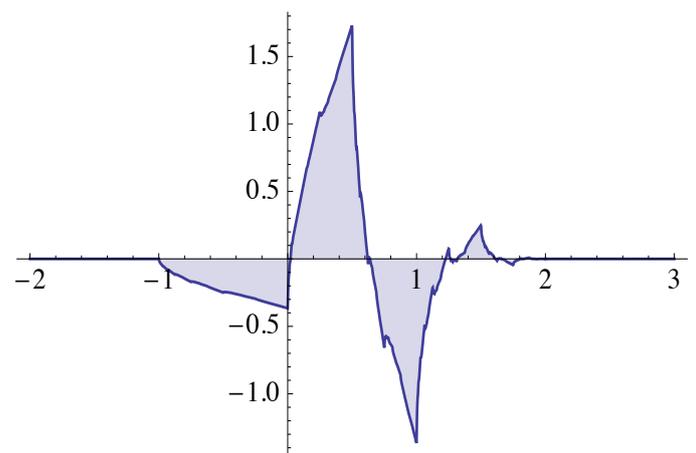
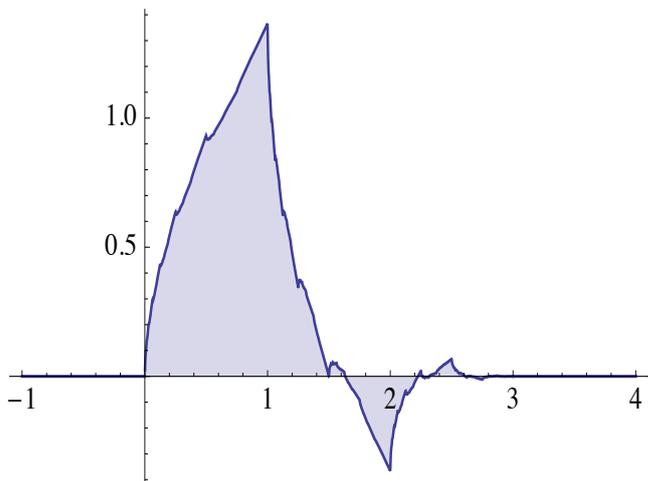


WTBV

Skript zur Vorlesung

Volker Strehl
Department Informatik der FAU



Copyright © 2017 Volker Strehl

PUBLISHED BY THE AUTHOR

BOOK-WEBSITE.COM

Licensed under the Creative Commons Attribution-NonCommercial 3.0 Unported License (the “License”). You may not use this file except in compliance with the License. You may obtain a copy of the License at <http://creativecommons.org/licenses/by-nc/3.0>. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Version vom 13. April 2017

Inhaltsverzeichnis

1	Einleitung	13
1.1	Approximation und Orthogonalität	13
1.1.1	Beispiel: Approximation durch Polynome (I)	15
1.1.2	Beispiel: Approximation durch trigonometrische Funktionen (I)	15
1.1.3	Beispiel: Approximation durch Treppenfunktionen (I)	15
1.1.4	Skalarprodukt und optimale Approximation	15
1.1.5	Beispiel: Approximation durch Polynome (II)	16
1.1.6	Beispiel: Approximation durch trigonometrische Funktionen (II)	17
1.1.7	Beispiel: Approximation durch Treppenfunktionen (II)	18
1.2	Beispiele für Approximationen	20
1.2.1	Approximation durch Polynome (III)	20
1.2.2	Approximation durch trigonometrische Funktionen (III)	23
1.2.3	Approximation durch Treppenfunktionen (III)	31
1.2.4	Approximation einer Testfunktion	32
1.3	Intermezzo: Approximation im \mathbb{R}^2	34
1.4	Approximation und Skalierung	38
1.5	Beispiele zur einstufigen Haar-Transformation	43
1.5.1	1. Beispiel	43
1.5.2	2. Beispiel	47
1.6	Haar-Transformation über mehrere Auflösungsstufen	47
1.6.1	1. Beispiel	49
1.6.2	2. Beispiel	53

1.7	Zweidimensionale Haar-Transformation	57
1.7.1	Das Prinzip	57
1.7.2	Ein Beispiel	59
1.8	Datenkompression mittels Haar-Transformation	73
1.9	Entrauschen mit Wavelets	85
1.9.1	Entrauschung und Rekonstruktion mit dem Haar-Wavelet	86
1.9.2	Entrauschung und Rekonstruktion mit dem Daubechies-12-Wavelet	87
1.10	(Naive) Kantenerkennung	91
1.10.1	Ein zweidimensionales Beispiel	93
1.11	Frequenztrennung	93
2	Mathematische Präliminarien	97
2.1	Normierte Vektorräume	98
2.2	Vektorräume mit innerem Produkt	99
2.3	Orthogonalität	102
2.4	Duale Basen (biorthogonale Basen)	105
2.5	Hilbert-Basen	108
2.6	Wichtige Beispiele von Hilbert-Basen	110
2.6.1	Der Hilbertraum $\ell^2(\mathbb{Z})$	110
2.6.2	Der Hilbertraum $\mathcal{L}^2([0, 1])$	110
2.6.3	Der Hilbertraum $\mathcal{L}^2(\mathbb{R})$	111
3	Fourierreihen	113
3.1	Motivation, Heuristik	113
3.2	Approximation, Gibbs-Wilbraham-Phänomen	115
3.3	Konvergenz von Fourierreihen	116
3.3.1	Die Sätze von Dirichlet und Fejér	118
3.3.2	Summationskerne und Konvergenzsätze	123
3.3.3	Eine Hilbert-Basis für $\mathcal{L}^2([0, 1])$	125
3.4	Kantenerkennung mittels Fourierreihen	127
3.5	Mathematica notebooks	146
4	Fouriertransformation	161
4.1	Heuristik	161
4.2	Definition der Fouriertransformation	162
4.3	Fouriertransformation und Faltung	165
4.4	Inversion: Fourier-Rücktransformation	167
4.5	Fouriertransformation und Differenzierbarkeit	169
4.6	\mathcal{L}^2-Fouriertransformation	169
4.7	Heisenbergs Unschärferelation	171

4.8	Diracs Funktionale $\delta_a(t)$	173
4.9	Eine kurze Exkursion in eine neue Welt	175
4.10	Diracs Kamm und die Formel von Poisson	177
4.11	Sampling und das Abtasttheorem	180
4.12	Mathematica notebooks	184
5	Das Haar-Wavelet	219
5.1	Zur Historie	219
5.2	Haar-Skalierungsfunktion und Haar-Waveletfunktion	221
5.3	Haar-Systeme auf $[0, 1]$	228
5.4	Haar-Systeme auf \mathbb{R}	228
5.5	Die Diskrete Haar-Transformation (DHT)	230
5.6	Analyse und Synthese: die Haar-Filterbank	235
5.7	Zweistufige Haar-Filterbank	239
5.8	Sichtweise der z -Transformation	241
5.9	2D-Haar-Wavelets	243
5.10	Mehrstufige 2D Haar-Transformation	248
5.11	Abschätzung der Größe von Haar-Waveletkoeffizienten	249
6	Diskrete Signale und Filter	251
6.1	2π -periodische Funktionen	251
6.2	Signale	252
6.3	Faltung	254
6.4	Filter	255
6.5	Downsampling und Upsampling	257
7	Orthogonale Filter und Rekonstruktion	259
7.1	Allgemeines	259
7.2	Orthogonalität	261
7.3	Signale endlicher Länge	266
8	Daubechies-Filter	269
8.1	Daubechies-Filter D_4	269
8.2	Daubechies-Filter D_6	273
8.3	Daubechies-Filter D_{2M}	273
8.4	Daubechies-Polynome	275
8.5	Nochmals D_4 und D_6	279
8.5.1	D_4 mittels Daubechies-Polynom	279
8.5.2	D_6 mittels Daubechies-Polynom	279

8.6	Nicht-kausale Filter	280
8.7	Coiflet-Filter	281
8.7.1	Idee und Konstruktion der Coiflet-Filter	281
8.7.2	C_6 mittels Daubechies-Polynom	283
8.7.3	C_6 direkt	284
8.7.4	C_{12} direkt	285
9	Multiresolution (MRA)	289
9.1	Schema der Multiresolution	289
9.2	Beispiele	292
9.2.1	Haar-MRA	292
9.2.2	Shannon-MRA	292
9.2.3	Stückweise lineare MRA	293
9.3	Approximation der Skalierungs- und Waveletfunktionen	296
9.3.1	Der Kaskadenalgorithmus	296
9.3.2	Dyadische Interpolation	297
9.4	Beweis des Theorems 9.1	297
9.5	Orthonormale Translationssysteme	301
9.6	Weitere Eigenschaften von MRAs (I)	304
9.7	Weitere Eigenschaften von MRAs (II)	306
9.7.1	Konvergenz der Fixpunktiteration	307
9.7.2	Endliche Filter und endlicher Träger	308
9.7.3	Momente, Glattheit, Rekonstruktionseigenschaften	308
9.8	Die Größe von Waveletkoeffizienten	311
9.9	Exakte Darstellung von Polynomen	312
9.10	Zusammenfassung	314
9.11	Mathematica notebooks	315
10	Biorthogonale Filterpaare	361
10.1	Biorthogonale Filterpaare	362
10.2	Spline-Filter	366
10.2.1	Symmetrische Tiefpassfilter	366
10.2.2	Spline-Funktionen	367
10.2.3	Daubechies biorthogonale Filter	367
10.3	Cohen-Daubechies-Feauveau-Filter	369
10.3.1	Nochmals Daubechies-Polynome	369
10.3.2	Symmetrische Filter ungerader Länge	373
10.3.3	Das Cohen-Daubechies-Feauveau-(7,9)-Filterpaar	373
11	Lifting	383
11.1	Faktorisierung der Haar-Transformation	383

11.2	Faktorisierung der Daubechies-4-Transformation	385
11.3	Die Polyphasenmatrix	386
11.4	Euklidische Division und Lifting	389
11.5	Die Wavelettransformationen in JPEG2000	392
11.5.1	Liftingschema für das Cohen-Daubechies-Feauveau-(7,9)-Wavelet	392
11.5.2	Liftingschema für das biorthogonale LeGall-(5,3)-Wavelet	393
12	Denoising using wavelets	397
13	DWT and the SPIHT algorithm	407
13.1	Introduction	407
13.1.1	What this chapter is about	407
13.1.2	References for this chapter	407
13.2	The scheme of wavelet transform image coding	409
13.3	Progressive transmission	410
13.3.1	Heuristics for progressive transmission	410
13.3.2	Progressive transmission, the procedures	411
13.3.3	An example for progressive transmission	412
13.4	Zig-zag ordering	413
13.5	Embedded trees in DWT-data	414
13.5.1	The 1D-case	414
13.5.2	The 2D-case (<i>spatial orientation trees</i>)	416
13.6	The SPIHT method	419
13.6.1	General remarks	419
13.6.2	Data for and verbal description of the SPIHT-algorithm	420
13.6.3	The procedures for SPIHT	424
13.7	Examples	426
13.7.1	Example 1	426
13.7.2	Example 2	428
13.7.3	Example 3	432
14	CWT and edge detection	455
15	Joseph Fourier und die Wärmeleitung	557
16	Anhang	567
16.1	Weitere Beispiele zur iterativen Konstruktion	567
16.2	Fourier's World	572
17	Literatur	579
	Lehrbücher und Monografien	579
	Zeitschriftenartikel	581

Vorwort

In diesem Text sind die wesentlichen theoretisch-mathematischen Inhalte der Vorlesung *Wavelettransformationen in der Bildverarbeitung* zusammengefasst, dazu Einblicke in drei Anwendungsbereiche: Rauschunterdrückung (*denoising*), Datenkompression und Kantenerkennung.

Wichtiger Hinweis: Zu den den theoretischen Ausführungen dieses Skripts werden im Rahmen der Vorlesung Beispiele in großem Umfang vorgeführt, meist in der Form von Mathematica notebooks. Diese finden sich, wie die Slides zur Vorlesung, auf der Webseite. Ausserdem sei ausdrücklich auf die Übungen hingewiesen, die neben den speziellen Anwendungen in der Bildverarbeitung auch Illustrationen des theoretischen Stoffs enthalten.

Übersicht: Dieses Skript beginnt mit einer sehr langen motivierenden Einführung, die ausgehend von dem Problem der *Approximation von Funktionen* (Signalen) auf die Besonderheiten des Wavelet-Ansatzes zur Transformation und Bearbeitung von Signalen hinführt. Dabei wird ausschliesslich das Haar-Wavelet benutzt, das sehr einfach aufgebaut ist, an dem man aber alle typischen Aspekte diskreter Wavelettransformationen kennenlernen kann. Typische Anwendungen, wie Datenkompression, Entrauschung, Kantenerkennung und Frequenztrennung werden an Beispielen demonstriert. Eine genauere Behandlung ist den Übungen vorbehalten.

Die Kapitel 2 bis 4 stellen die benötigten Kenntnisse aus den Bereichen *Fourierreihen* und *Fouriertransformation* zusammen. Das ist als Orientierung gedacht und ersetzt nicht ein genaueres Studium, wenn man wissen will, wie es wirklich funktioniert. Zwar kann man eine Diskrete Wavelettransformation implementieren und anwenden, ohne etwas von Fouriertheorie zu kennen, aber für die Fragen, wie man Wavelets mit guten Eigenschaften konstruiert und warum sie so gut funktionieren, sind Grundkenntnisse der Fouriertheorie unverzichtbar. Analytische Subtilitäten, wie z.B. die genaue Begründung von Konzepten und Resultaten, die ein Kenntnis des

Integralbegriffs von Lebesgue erfordern, sind aber nicht das Thema der Vorlesung und auch nicht des Skripts. Hingegen wird hier Wert gelegt auf einige Aspekte der Fouriertheorie, die gerade auch im Hinblick auf Anwendungen von Bedeutung sind: *Gibbs-Wilbraham-Phänomen*, *Faltungstheorem*, *Glattheit und Verschwinden im Unendlichen*, *Unschärferelation*, *Abtasttheorem*. Ein Exkurs, der den Umgang mit Fourierreihen illustriert, ist der Abschnitt 3.4, der das Problem der Kantenerkennung mit Hilfe von Fourierkoeffizienten behandelt.

Im Kapitel 5 werden das *Haar-Wavelet* und die *Haar-Wavelettransformation* ausführlich behandelt, bis hin zum Konzept der Filterbank, auf das später immer wieder Bezug genommen wird – dann mit anderen Wavelets als Bauelemente. Auch die Wavelettransformation von zweidimensionalen Signalen (*Bildern*) wird in diesem Kapitel am Beispiel der Haar-Transformation formalisiert.

Für den Umgang mit Filterbänken sind einige Grundbegriffe aus dem Bereich *Signale und Filter* erforderlich. Dafür ist das Kapitel 6 gedacht, das wie die Kapitel 2 bis 4 der Rekapitulation und Zusammenstellung dient.

Ab Kapitel 7 befindet man sich im Zentrum der Wavelettheorie. Das Ziel ist die Konstruktion von Waveletfiltern endlicher Länge, die bessere Eigenschaften aufweisen, als das Haar-Wavelet. An dieser Stelle ist die Bemerkung über das Verhältnis von Fouriertheorie und Wavelettheorie: in der Fouriertheorie geht es um *stationäre* (also lang andauernde) Aspekte von Signalen, die sich im Frequenzbereich (Spektrum) gut ausdrücken lassen. Bei den Wavelets geht es um das Erfassen von *transienten*, also kurzfristigen, im Orts/Zeitbereich gut lokalisierten Eigenschaften von Signalen. In diesem Kapitel 7 geht es um die präzise Beschreibung von Filtern und Filterpaaren, insbesondere von *orthogonalen Filtern* und deren Rekonstruktionseigenschaften.

Die Konstruktion von Paaren orthogonaler Hoch- und Tiefpassfilter endlicher Länge durch Ingrid Daubechies war ein Markstein in der Geschichte der Wavelettransformationen. Von der Komplexität dieser Aufgabe soll das Kapitel 8 einen handfesten Eindruck vermitteln. Solche “Wunschfilter” für die Wavelettransformation gibt es tatsächlich (was *à priori* nicht klar war!), aber man muss wirklich etwas tun, um sie zu konstruieren.

Das Kapitel 9 befasst sich mit dem Konzept der *Multiresolution* (MRA) von Stéphane Mallat, einem weiteren großen Schritt in der Geschichte der Wavelets. Das ist ein übergreifendes mathematisches Schema für den Umgang mit Wavelets für die Diskrete Wavelettransformation. In diesem Kontext, insbesondere wenn es um Qualitätsansprüche (Glattheitsforderungen) geht, wird von der Fouriertheorie erheblich Gebrauch gemacht.

Zu einer Multiresolution gehören (kontinuierliche) Skalierungs- und Waveletfunktionen, die etwas über die Eigenschaften der zugehörigen (diskreten) Wavelettransformationen aussagen. Diese Funktionen haben keine einfache analytische Beschreibung, sind teilweise sogar als “bizarr” anzusprechen. Wie man sie (approximiert) grafisch darstellt, wird in diesem Kapitel gezeigt. Weitere Beispiele dazu finden sich im Anhang (Kapitel 12).

Aus praktischen Gründen hat man die Idee der orthogonalen Filterpaare erweitert zu sogenannten *bi-orthogonalen Filterpaaren*. Insbesondere gehören die in dem Standard JPEG2000 festgelegten Waveletfilter zu dieser Familie. Das ist der Grund, sich deren Konstruktion im Kapitel 10 näher anzuschauen.

Die konkrete Implementierung bi-orthogonaler Filterpaare geschieht in der Praxis mittels der Technik des *Lifting*, die kurz im Kapitel 11 angesprochen wird. Sie hat eine interessante Verbindung zu dem altbekannten euklidischen Algorithmus für Polynome.

Die drei folgenden Kapitel befassen sich mit Anwendungen der Wavelettransformationen: Rauschunterdrückung (12), Datenkompression (13), und Kantenerkennung (14). Diese Kapitel sind der Vollständigkeit halber hinzugenommen worden, obwohl sie noch nicht voll ausgearbeitet sind. Immerhin enthält das Kapitel zur Datenkompression eine vollständige **Mathematica**-Implementierung der SPIHT-Methode. Im Kapitel zur Kantenerkennung wird ausserdem die kontinuierliche Wavelettransformation soweit eingeführt, wie das für diesen Zweck erforderlich ist. Für diese drei Anwendungskapitel gilt ganz besonders, dass sich auf den Webseiten zur Vorlesung reichhaltiges Material zur Illustration des behandelten Stoffes findet.

Das Kapitel 15 ist dem Andenken an den französischen Mathematiker, Physiker (und Politiker!) Joseph Fourier gewidmet, dem wir die Erfindung der nach ihm benannten Methoden (Fourierreihen, Fouriertransformation, Fourieranalyse,...) verdanken, die zu den epochalen Fortschritten der Wissenschaft zählen. Sein Ausgangspunkt war das Problem der Wärmeleitung in Festkörpern mathematisch zu fassen und so einer quantitativen Behandlung zugänglich zu machen. Davon gibt das Kapitel einen kurzen Einblick.

In einem Anhang (Kapitel 16) werden einige weitere Beispiele zur iterativen Konstruktion von Skalierungs- und Waveletfunktionen wiedergegeben, dazu einige tabellarische Zusammenstellungen von Fakten und Formeln zur Fourieranalyse und Multiresolutionsanalyse.

Eine Bitte: Dieses Skriptum wird zur Verfügung gestellt mit der dringenden Bitte, sich beim Verfasser zu melden, wenn bei der Lektüre Fehler gefunden werden, Unklarheiten auftreten, Verbesserungen oder/und Ergänzungen gewünscht werden. Auch die aktuelle Version hat davon schon profitiert.

Zum Schluss der Einleitung die Antwort auf eine Frage, die der Leser hat, aber vielleicht nicht zu stellen wagt:

“Woher kommt eigentlich das Wort *wavelet*?”.

Die Antwort: *wavelet* bedeutet soviel wie “kleine Welle”. Im Französischen heisst die Welle *onde* und die kleine Welle *ondelette*. Die Bezeichnung wurde inspiriert durch die ersten Funktionen, mit denen man (kontinuierliche) Wavelettransformationen ausgeführt hat. Hier sind einige typische Vertreter im Bild:

Analytisch betrachtet handelt es sich um die Funktionen

$$\begin{aligned} \psi_{mh}(x) &= -\frac{2e^{-x^2/2}(x^2-1)}{\sqrt{3}\sqrt[4]{\pi}} & \psi_{sh}(x) &= \frac{2(\sin(2\pi x) - \cos(\pi x))}{\pi - 2\pi x} \\ \psi_{mo}(x) &= \frac{e^{-x^2/2} \cos\left(\pi x \sqrt{\frac{2}{\log(2)}}\right)}{\sqrt[4]{\pi}} & \psi_{ga}(x) &= \frac{e^{-x^2/2+6ix}}{\sqrt[4]{\pi}} \end{aligned}$$

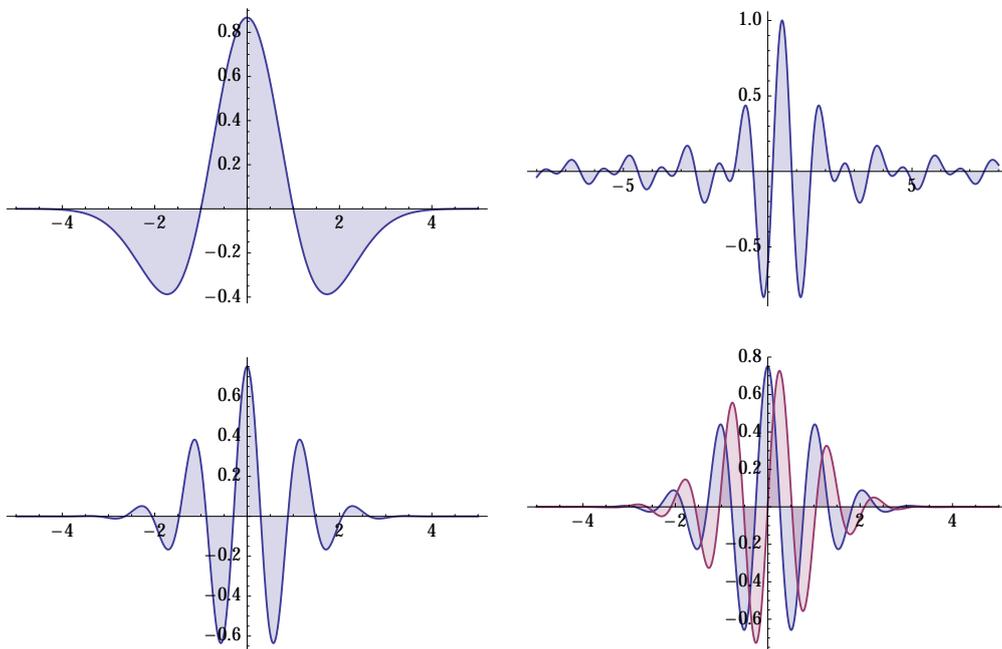


Abbildung 1: Grafische Darstellung des Mexikanischer-Hut-Wavelets (links oben), des Shannon-Wavelets (r.o.), des Morlet-Wavelets (l.u.) und des Gabor-Wavelets (Real-und Imaginärteil, r.u.)



1. Einleitung

Dieses einleitende Kapitel hat einen doppelten Zweck:

- Ausgehend von dem Problem der Approximation von Funktionen und verschiedenen Realisierungen (Approximation mit Polynomen, mit trigonometrischen Funktionen und mit Treppenfunktionen) mit ihren jeweiligen Eigenheiten soll anhand der Frage, wie sich Approximationen mit Treppenfunktionen unter verschiedenen Aufösungen verhalten, die Haar-Transformation als einfachste Wavelettransformation präsentiert werden. Hierbei hat man es, im deutlichen Gegensatz zur Approximation mit Polynomen oder mit trigonometrischen Funktionen, mit einer Approximation zu tun, sowohl im Orts/Zeitbereich, als auch im Frequenzbereich lokalisiert ist. Damit wird auch transiente (nicht stationäre) Information in der Approximation erfasst. Der formale Rahmen der *Multiresolution*, der im weiteren Verlauf der Vorlesung eine wesentliche Rolle spielen wird, kann schon an diesem einfachen Beispiel dargestellt und exemplarisch benutzt werden.
- Anhand der Haar-Transformation soll die Verwendung von Wavelettransformationen für Aufgaben der Bildverarbeitung (Datenkompression, Entrauschung, Kantenerkennung, Trennung von Bildinformationen nach Frequenzbereichen) beispielhaft erläutert werden. Detailliert werden diese Anwendungen dann in den Übungen zur Vorlesung behandelt.

1.1 Approximation und Orthogonalität

Benutzt man mathematische Konzepte, um Situationen und Objekte quantitativ zu modellieren, so tritt eigentlich immer und ganz besonders bei praktischen Anwendungen das Problem der numerischen Darstellung und damit der Approximation von Daten auf. Kontinuierliche Daten eines Modells (“Signale”) werden diskretisiert, um sie algorithmischer Transformation zugänglich zu machen.

In diesem einführenden Abschnitt sollen drei weit verbreitete Approximationstechniken nebeneinander gestellt werden.

Ausgangspunkt ist die Vorstellung, dass es bei den zu untersuchenden und zu transformierenden Signalen um reelle Funktionen $f : [a, b] \rightarrow \mathbb{R}$ handelt, die auf einem endlichen Intervall $[a, b] \subseteq \mathbb{R}$ definiert sind. In der Regel werden die betrachteten Funktionen nicht beliebig sein, sondern einer Familie von "gutartigen" Objekten angehören (z.B. stetige Funktionen, differenzierbare ("glatte") Funktionen, integrierbare Funktionen), was hier nicht im einzelnen ausgeführt werden soll. Solche Familien haben durchwegs die algebraische Struktur eines Vektorraumes, d.h. man kann mit solchen Funktionen nach den üblichen Regeln Linearkombinationen bilden, die dann wieder in der Familie liegen.

Es bezeichne nun \mathcal{V} einen reellen Vektorraum von Funktionen. Um von Approximation reden zu können, muss auf dem Raum \mathcal{V} eine Abstandsfunktion (*Metrik*) $d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$ definiert sein. Üblicherweise geschieht das dadurch, dass man die Metrik d aus einer *Norm* $\|\cdot\| : \mathcal{V} \rightarrow \mathbb{R}_{\geq 0}$ gewinnt:

$$d(f, g) = \|f - g\|,$$

man hat es also mit einem *normierten Vektorraum* $(\mathcal{U}, \|\cdot\|)$ zu tun, der in den meisten Fällen auf noch *vollständig* sein soll, d.h. ein Raum, in dem alle Cauchyfolgen konvergieren. Man spricht dann von einem *Banachraum*.

Beispiele für oft verwendete Normen auf einem passenden Vektorraum \mathcal{V} sind etwa

$$\begin{aligned} \|f\|_1 &= \int_a^b |f(t)| dt \\ \|f\|_2 &= \sqrt{\int_a^b |f(t)|^2 dt} \\ \|f\|_\infty &= \sup_{t \in [a, b]} f(t) \end{aligned}$$

"Passend" heisst dabei, dass die Integrale bzw. das Supremum existieren.

Das Approximationsproblem lässt sich dann folgendermaßen formulieren. Ist \mathcal{U} ein Unterraum von \mathcal{V} , der aus besonders "einfachen" Funktionen besteht (z.B. Polynome, Treppenfunktionen), so wird zu gegebenem $f \in \mathcal{V}$ ein Element $f^* \in \mathcal{U}$ gesucht, das so nahe wie möglich bei f liegt:

$$f^* = \operatorname{argmin}_{g \in \mathcal{U}} d(f, g)$$

Der Raum \mathcal{U} wird in der Regel endlich-dimensional sein. Meist wird man es auch mit einer parametrisierten Familie $(\mathcal{U}_N)_{N \in \mathbb{N}}$ von "Approximationsräumen" zu tun, die per Inklusion geordnet sind, d.h. Approximationen verschiedener Genauigkeitsstufen (*Auflösung*) beschreiben:

$$\dots \subset \mathcal{U}_{N-1} \subset \mathcal{U}_N \subset \mathcal{U}_{N+1} \subset \dots$$

Drei klassische Beispiele sollen hier genannt werden:

1.1.1 Beispiel: Approximation durch Polynome (I)

\mathcal{U}_N sei der Vektorraum der Polynome vom Grad $\leq N$, betrachtet auf dem Intervall $[a, b] = [-1, +1]$, also

$$g(t) = g_0 + g_1 t + g_2 t^2 + \cdots + g_N t^N.$$

Dies ist ein Vektorraum der Dimension $N + 1$.

1.1.2 Beispiel: Approximation durch trigonometrische Funktionen (I)

\mathcal{U}_N sei der Vektorraum der 2π -periodischen trigonometrischen Polynome

$$g(t) = \sum_{k=0}^N a_k \cos(kt) + \sum_{k=1}^N b_k \sin(kt),$$

betrachtet auf dem Intervall $[a, b] = [-\pi, \pi]$. Dies ist ein Vektorraum der Dimension $2N + 1$.

1.1.3 Beispiel: Approximation durch Treppenfunktionen (I)

\mathcal{U}_N sei der Vektorraum der Treppenfunktionen mit Stufenbreite 2^{-N} , betrachtet auf dem Intervall $[a, b] = [0, 1)$, also

$$g(t) = \sum_{0 \leq j < 2^N} \alpha_k \mathbf{1}_{I_{N,k}}(t),$$

wobei $I_{N,k} = [k 2^{-N}, (k+1) 2^{-N})$. Dies ist ein Vektorraum der Dimension 2^N .

1.1.4 Skalarprodukt und optimale Approximation

Der übliche Weg, um das Approximationsproblem handhabbar zu machen besteht darin, von der Norm (und damit von der Abstandsfunktion) vorauszusetzen, dass sie von einem *Skalarprodukt* oder *inneren Produkt* $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{C}$ herrührt. Typischerweise sehen in unserer Situation innere Produkte so aus:

$$\langle f | g \rangle = \int_a^b f(t) \overline{g(t)} w(t) dt,$$

wobei $w(t)$ eine geeignete Gewichtsfunktion ist. (Ausser im ersten Beispiel wird $w(t) \equiv 1$ sein, also gar nicht explizit in Erscheinung treten.)

Ein wichtiger Punkt ist: ein *inneres Produkt* erlaubt es, von *Winkeln* und von *Orthogonalität* zu sprechen. Für das Approximationsproblem, wie oben formuliert, gilt dann:

Das gesuchte Element f^* ist die *orthogonale Projektion* von f auf den Unterraum \mathcal{U} , d.h. $\langle f | g \rangle = \langle f^* | g \rangle$ gilt für alle $g \in \mathcal{U}$.

Ein Beweis dieser Aussage findet sich im folgenden Kapitel.

In dieser Situation kann die Berechnung der optimalen Approximation f^* auf die Berechnung von inneren Produkten zurückgeführt werden. Dazu sei nun $\mathcal{B} =$

$\{g^1, g^2, \dots, g^M\}$ eine *Orthonormalbasis* des Raumes \mathcal{U} , d.h. \mathcal{B} ist eine Basis von \mathcal{U} und es gilt $\langle g^i | g^j \rangle = \delta_{i,j}$ für $1 \leq i, j \leq M = \dim \mathcal{U}$.

Ist nun

$$f^* = \sum_{1 \leq i \leq M} \alpha_i g^i$$

die Darstellung der optimalen Approximation f^* von f in der Basis \mathcal{B} . Dann gilt wegen der Linearität des Skalarprodukts und der Orthonormalität von \mathcal{B} :

$$\langle f^* | g^j \rangle = \sum_{1 \leq i \leq M} \alpha_i \langle g^j | g^i \rangle = \sum_{1 \leq i \leq M} \alpha_i \delta_{j,i} = \alpha_j \quad (1 \leq j \leq M)$$

und das bedeutet

$$f^* = \sum_{1 \leq i \leq M} \langle f^* | g^i \rangle g^i = \sum_{1 \leq i \leq M} \langle f, g^i \rangle g^i.$$

Die Koeffizienten $\langle f | g^i \rangle$ der optimalen Approximation f^* bezeichnet man als die *verallgemeinerten Fourierkoeffizienten* von f bezüglich der Basis \mathcal{B} .

Dieses Vorgehen soll nun an den obigen drei Beispielen verdeutlicht werden.

1.1.5 Beispiel: Approximation durch Polynome (II)

Im Raum \mathcal{V} wird folgendes Skalarprodukt betrachtet:

$$\langle f | g \rangle = \int_{-1}^{+1} f(t) \overline{g(t)} \frac{1}{\sqrt{1-t^2}} dt.$$

Eine orthogonale Basis bezüglich $\langle \cdot, \cdot \rangle$ in \mathcal{U}_N bilden die ersten $N + 1$ *Chebyshev-T-Polynome* $\mathcal{B} = \{T_k(t); 0 \leq k \leq N\}$, rekursiv definiert durch

$$T_0(t) = 1, \quad T_1(t) = t, \quad T_{k+1}(t) = 2tT_k(t) - T_{k-1}(t) \quad (1 \leq k < N).$$

Die ersten Werte $T_n(t)$ ($0 \leq t \leq 5$) sind:

$$\begin{aligned} T_0(t) &= 1 \\ T_1(t) &= t \\ T_2(t) &= 2t^2 - 1 \\ T_3(t) &= 4t^3 - 3t \\ T_4(t) &= 8t^4 - 8t^2 + 1 \\ T_5(t) &= 16t^5 - 20t^3 + 5t \end{aligned}$$

Die grafische Darstellung des Werteverlaufs im Intervall $[-1, +1]$ ist in Abbildung 1.1 dargestellt.

Offensichtlich ist $T_k(t)$ ein Polynom vom Grad k , was die Basiseigenschaft von \mathcal{B} zeigt. Die Orthogonalität drückt sich so aus:

$$\langle T_k(t) | T_\ell(t) \rangle = \begin{cases} \pi & \text{falls } k = \ell = 0 \\ \pi/2 & \text{falls } k = \ell > 0 \\ 0 & \text{falls } k \neq \ell \end{cases}$$

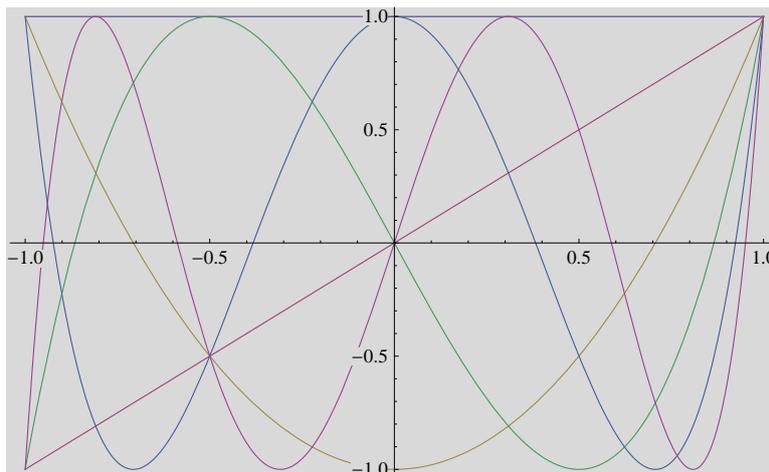


Abbildung 1.1: Grafische Darstellung der Polynome $T_k(t)$ für $0 \leq k \leq 5$ und $t \in [-1, +1]$. $T_k(t)$ ist das Polynom mit k Nullstellen in $[-1, +1]$.

Die optimale Approximation einer Funktion $f(t)$ im Raum \mathcal{U}_N schreibt sich also

$$f(t) \approx \frac{\langle f | 1 \rangle}{\pi} + \frac{2}{\pi} \sum_{k=1}^N \langle f | T_k \rangle T_k(t)$$

Diese *Chebyshev-Approximation* von Funktionen spielt in der numerischen Mathematik eine große Rolle!

Wichtig ist folgende Beobachtung: die Approximationskoeffizienten

$$\langle f | T_k \rangle = \int_{-1}^{+1} f(t) T_k(t) dt$$

der optimalen Approximation hängen vom Verlauf der zu approximierenden Funktion $f(t)$ im gesamten Intervall $[-1, +1]$ ab, geben also in erster Linie keine lokalen Eigenschaften wieder!

1.1.6 Beispiel: Approximation durch trigonometrische Funktionen (II)

Die oben gegebene Darstellung der Funktionen von \mathcal{U}_N suggeriert schon, dass man die trigonometrischen Funktionen

$$\mathcal{B} = \{\cos(kt); 0 \leq k \leq N\} \cup \{\sin(\ell t); 1 \leq \ell \leq N\}$$

als Basis \mathcal{U}_N nimmt.

In Abbildung 1.2 ist der Werteverlauf dieser Funktionen für $N = 5$ im Intervall $[-\pi, \pi)$ dargestellt.

Tatsächlich gilt Orthogonalität:

$$\begin{aligned} \int_{-\pi}^{\pi} \cos(kt) \cos(\ell t) dt &= \begin{cases} 2\pi & \text{falls } k = \ell = 0 \\ \pi \delta_{k,\ell} & \text{sonst} \end{cases} \\ \int_{-\pi}^{\pi} \sin(kt) \sin(\ell t) dt &= \pi \delta_{k,\ell} \\ \int_{-\pi}^{\pi} \cos(kt) \sin(\ell t) dt &= 0 \end{aligned}$$

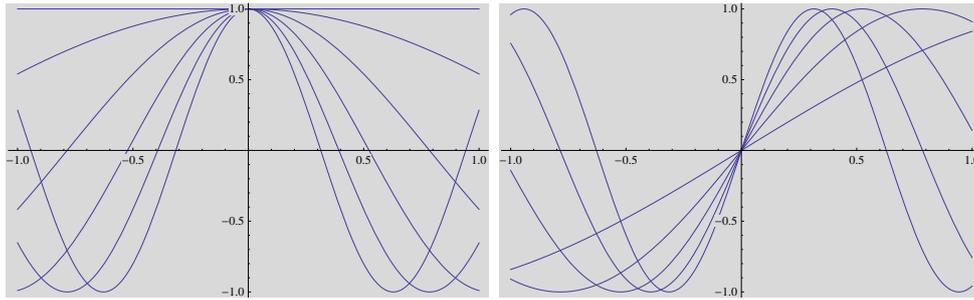


Abbildung 1.2: Grafische Darstellung der Funktionen $\cos(kt)$ ($0 \leq k \leq 5$) (links) und $\sin(kt)$ ($1 \leq k \leq 5$) (rechts) im Intervall $[-\pi, \pi]$.

Die optimale Approximation von Funktionen $f(t)$ in diesem Raum schreibt sich

$$f(t) \approx \frac{A_0}{2} + \sum_{k=1}^N A_k \cos(kt) + \sum_{\ell=1}^N B_\ell \sin(\ell t),$$

wobei die *Fourierkoeffizienten* so berechnet werden:

$$A_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(kt) dt \quad (0 \leq k \leq N)$$

$$B_\ell = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(\ell t) dt \quad (1 \leq \ell \leq N)$$

Auch hier gilt: diese Approximationskoeffizienten hängen vom Verlauf der Funktion $f(t)$ im ganzen Intervall $[-\pi, \pi]$ ab! Sie reflektieren also nicht (oder jedenfalls nicht unmittelbar) lokale Eigenschaften von $f(t)$.

Den Ausdruck für die approximierende Funktion bezeichnet man als *trigonometrisches Polynom*. Es handelt sich um das Anfangsstück einer *Fourierreihe*. Diese klassische Situation wird im weiteren Verlauf noch genauer besprochen und benötigt.

1.1.7 Beispiel: Approximation durch Treppenfunktionen (II)

Oben wurde bereits die Notation für die *dyadischen Intervalle*

$$I_{j,k} = [2^{-j}k, 2^{-j}(k+1)) \quad (j, k \in \mathbb{Z})$$

eingeführt. Die Indikatorfunktion für $I_{j,k}$ wird noch mit dem Skalierungsfaktor $2^{j/2}$ versehen. Man definiert:

$$\phi_{j,k}(t) = 2^{j/2} \mathbf{1}_{I_{j,k}}(t) = \begin{cases} 2^{j/2} & \text{für } 2^{-j}k \leq t < 2^{-j}(k+1) \\ 0 & \text{sonst} \end{cases} \quad (j, k \in \mathbb{Z}).$$

Hierbei ist j der Parameter für die *Skalierung*, *Dilation* (Auflösung) und k der Parameter für die *Lokalisierung* (Translation).

Die Funktion

$$\phi(t) = \mathbf{1}_{[0,1)}(t) = \phi_{0,0}(t)$$

wird als *Haar-Skalierungsfunktion* bezeichnet. Die Funktionen $\phi_{j,k}(t)$ gehen aus diesem $\phi(t)$ durch Skalierung und Translation hervor:

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k).$$

Dabei wurde der Skalierungsfaktor $2^{j/2}$ so gewählt, dass die Familie $\mathcal{B}_j = \{\phi_{j,k}(t); 0 \leq k < 2^j\}$ orthonormal bezüglich des Skalarproduktes

$$\langle f | g \rangle = \int_0^1 f(t) \overline{g(t)} dt$$

ist, d.h. es gilt

$$\langle \phi_{j,k} | \phi_{j,\ell} \rangle = \delta_{k,\ell} \quad (0 \leq k, \ell < 2^j).$$

\mathcal{B}_j ist also eine orthonormale Basis im Raum \mathcal{U}_j im Raum der Treppenfunktionen mit Stufenbreite 2^{-j} .

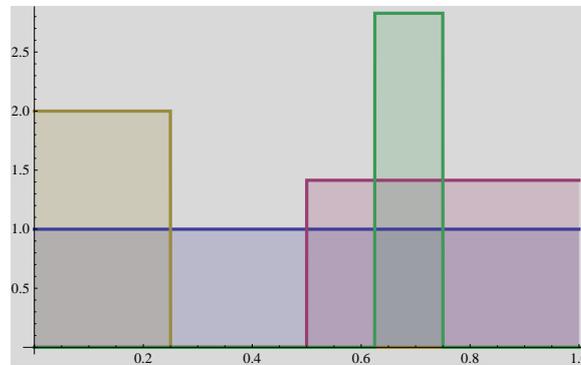


Abbildung 1.3: Grafische Darstellung der Funktionen $\phi_{0,0}, \phi_{1,1}, \phi_{2,0}, \phi_{3,5}$ im Intervall $[0, 1)$.

Die Koeffizienten der optimalen Approximation einer Funktion $f : [0, 1) \rightarrow \mathbb{R}$ sind also

$$a_{j,k} = \langle f | \phi_{j,k} \rangle = \int_0^1 f(t) \phi_{j,k}(t) dt = 2^{j/2} \int_{I_{j,k}} f(t) dt \quad (0 \leq k, \ell < 2^j)$$

und die optimale Approximation von $f(t)$ in \mathcal{U}_j schreibt sich als

$$f(t) \approx \sum_{k=0}^{2^j-1} a_{j,k} \phi_{j,k}(t).$$

Der Wert, den der rechts stehende Ausdruck für $t \in I_{j,k}$ annimmt, ist also

$$2^{j/2} a_{j,k} = 2^j \int_{I_{j,k}} f(t) dt = \frac{1}{2^{-j}} \int_{I_{j,k}} f(t) dt.$$

Das ist der *Mittelwert* von $f(t)$ im Intervall $I_{j,k}$. Im Gegensatz zu der Situation bei der Chebyshev-Approximation und der Approximation mittels trigonometrischer Polynome codiert hier der Approximationskoeffizient lokale Eigenschaften der zu approximierenden Funktion $f(t)$: er hängt ja nur vom Verhalten von $f(t)$ im Intervall $I_{j,k}$ ab!

1.2 Beispiele für Approximationen

1.2.1 Approximation durch Polynome (III)

In diesem Abschnitt sollen Funktionen $f(t)$, die im Intervall $[-1, +1]$ definiert sind, mittels Chebyshev-Polynomen approximiert werden.

1. Approximation eines Polynoms

Es sei $f(t) = -3t^4 + t^3 - 2t + 1$. Die ersten sechs Approximationskoeffizienten sind

$$(\langle f | T_k \rangle)_{0 \leq k \leq 5} = \left(-\frac{\pi}{8}, -\frac{5\pi}{8}, -\frac{3\pi}{4}, \frac{\pi}{8}, -\frac{3\pi}{16}, 0 \right)$$

und die optimalen Approximationen $f_k^*(t)$ von $f(t)$ in \mathcal{U}_k

$$\begin{aligned} f_0^*(t) &= -\frac{1}{8} \\ f_1^*(t) &= -\frac{5t}{4} - \frac{1}{8} \\ f_2^*(t) &= -3t^2 - \frac{5t}{4} + \frac{11}{8} \\ f_3^*(t) &= t^3 - 3t^2 - 2t + \frac{11}{8} \\ f_4^*(t) &= -3t^4 + t^3 - 2t + 1 \\ f_5^*(t) &= -3t^4 + t^3 - 2t + 1 \end{aligned}$$

Diese Funktionen sind grafisch im Vergleich zu $f(t)$ in Abbildung 1.4 dargestellt. Da $f(t)$ ein Polynom vierten Grades ist, gehört es selbst zum Raum \mathcal{U}_4 und damit gilt $f_k^*(t) = f_4^*(t) = f(t)$ für alle $k \geq 4$.

2. Approximation einer periodischen Funktion

Nun sei $g(t) = \frac{1}{\sqrt{\sin(10t)+2}}$ eine periodische Funktion. Die ersten sechs Approximationskoeffizienten dieser Funktion sind, numerisch berechnet,

$$(\langle g | T_k \rangle)_{0 \leq k \leq 5} = (2.32159, -0.0307786, -0.0204377, 0.0334497, -0.0167731, 0.144345)$$

Daraus berechnen sich die approximierenden Polynome $g_k^*(t)$ in \mathcal{U}_k zu

$$\begin{aligned} g_0^*(t) &= 0.738985 \\ g_1^*(t) &= 0.738985 - 0.0195943t \\ g_2^*(t) &= -0.0260221t^2 - 0.0195943t + 0.751996 \\ g_3^*(t) &= 0.0851791t^3 - 0.0260221t^2 - 0.0834786t + 0.751996 \\ g_4^*(t) &= -0.0854247t^4 + 0.0851791t^3 + 0.0594026t^2 - 0.0834786t + 0.741318 \\ g_5^*(t) &= 1.47029t^5 - 0.0854247t^4 - 1.75268t^3 + 0.0594026t^2 + 0.375987t + 0.741318 \end{aligned}$$

Die grafische Darstellung einiger Approximationen in Abbildung 1.6 zeigt, dass die Approximation einer periodischen Funktion durch Polynome auch in einem begrenzten Intervall durchaus mit Schwierigkeiten zu kämpfen hat. Global verhalten sich Polynome ganz anders als periodische Funktionen – das spürt man auch über kurze Distanzen, auch wenn es auf dem endlichen Intervall gelingt, $g(t)$ gut zu approximieren, siehe Abbildung 1.7.

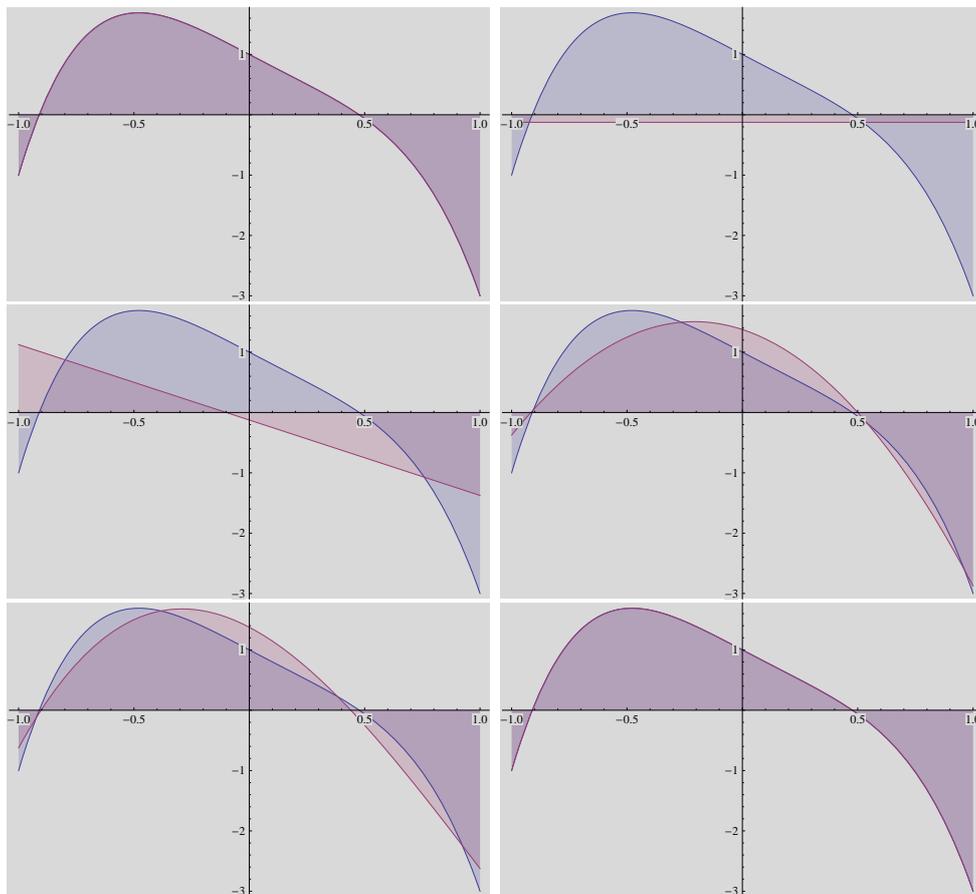


Abbildung 1.4: Grafische Darstellung der Funktion $f(t)$ und der Chebyshev-Approximationen $f_0^*(t), \dots, f_4^*(t)$ im Intervall $[-1, 1]$.

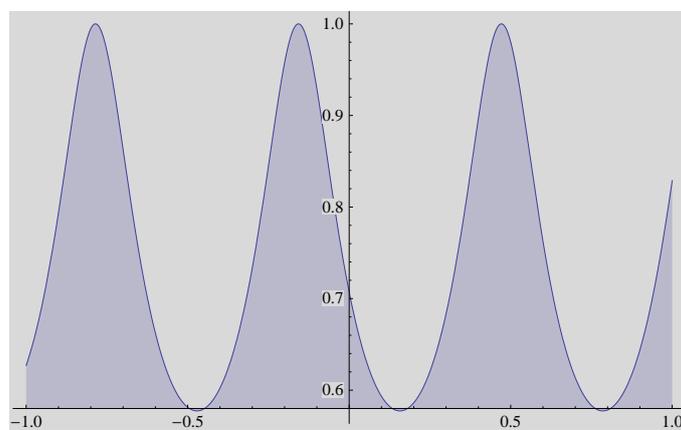


Abbildung 1.5: Grafische Darstellung der Funktion $g(t)$ im Intervall $[-1, 1]$.

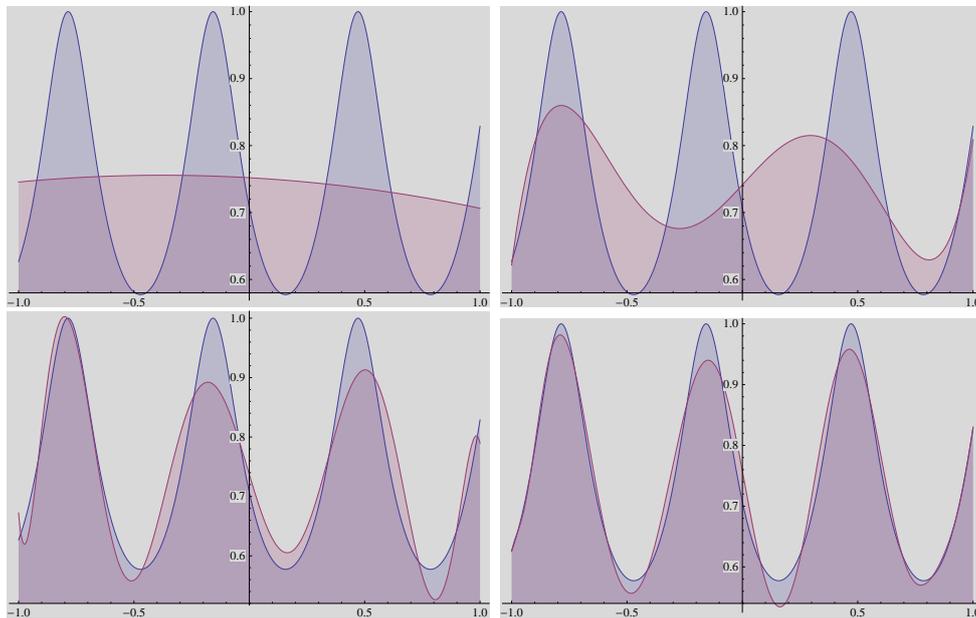


Abbildung 1.6: Grafische Darstellung der Funktion $g(t)$ und ihrer Chebyshev-Approximationen $g_2^*(t)$, $g_5^*(t)$, $g_{10}^*(t)$, $g_{15}^*(t)$ im Intervall $[-1, 1]$.

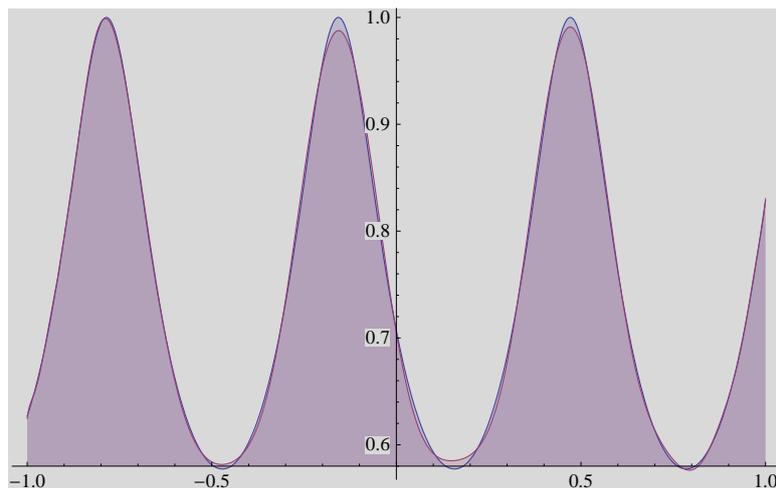


Abbildung 1.7: Grafische Darstellung der Funktion $g(t)$ und ihrer Chebyshev-Approximation $g_{25}^*(t)$ im Intervall $[-1, 1]$.

3. Approximation einer “glatten” Funktion

Polynome sind besonders einfache “glatte” Funktionen, wobei mit diesem Terminus mehrfache Differenzierbarkeit gemeint ist. Glatte Funktionen lassen sich in vielen Fällen mittels Polynomen niedrigen Grades im Intervall $[-1, 1]$ ganz brauchbar approximieren. Hierzu als Beispiel die Funktion

$$h(t) = \frac{e^{-t}}{\sqrt{t^2 + 1}}$$

mit der grafischen Darstellung in Abbildung 1.8. Auf die Angabe der Chebyshev-

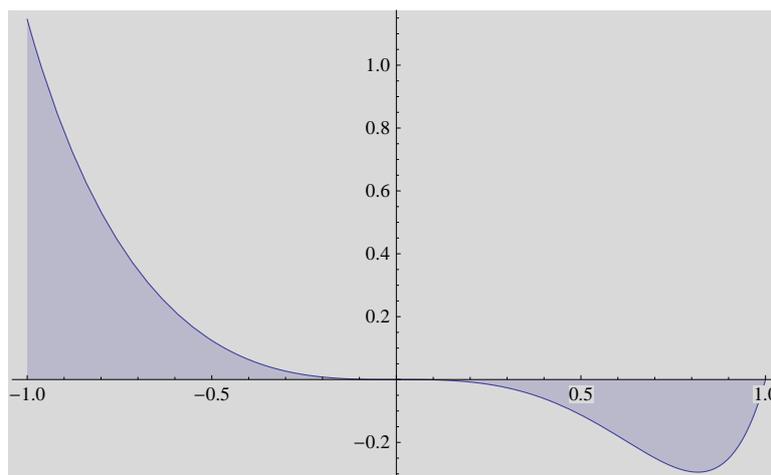


Abbildung 1.8: Grafische Darstellung der Funktion $h(t)$ im Intervall $[-1, 1]$.

Koeffizienten wird hier verzichtet. Der optische Eindruck einiger Approximationen soll genügen. Bei den ersten beiden Approximationen fällt auf, dass Abweichung am Rand des Intervalls weniger toleriert werden als Abweichungen im mittleren Bereich. Das liegt an der Gewichtsfunktion des Chebyshev-Skalarprodukts. Bei der Approximation mit einem Polynom 7. Grades fällt optisch kein Unterschied mehr zur Funktion $h(t)$ auf.

4. Approximation einer unstetigen Funktion

Zur Illustration soll hier die Box-Funktion

$$b(t) = \begin{cases} 1 & \text{falls } |t| \leq \frac{1}{2} \\ 0 & \text{sonst} \end{cases}$$

im Intervall $[-1, 1]$ approximiert werden, siehe Abbildung 1.10.

Es ist nicht überraschend, dass sehr glatte Funktionen wie Polynome große Mühe haben, Sprungstellen zu approximieren.

1.2.2 Approximation durch trigonometrische Funktionen (III)

1. Approximation einer trigonometrischen Funktion

Für eine Funktion wie

$$sa(t) = 2 \sin(2t) - 3 \cos(5t),$$

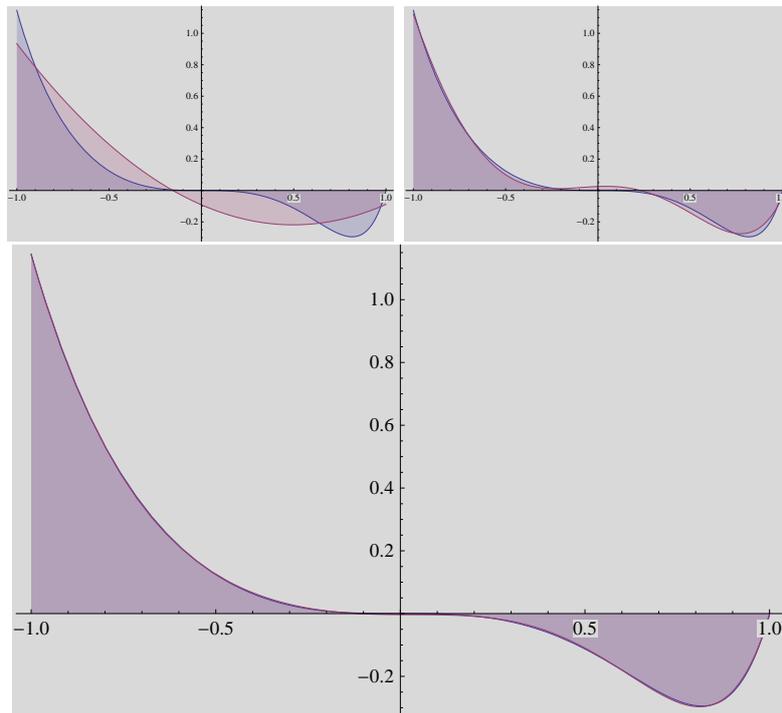


Abbildung 1.9: Grafische Darstellung der Funktion $h(t)$ und ihrer Approximationen $h_2^*(t)$, $h_5^*(t)$, $h_7^*(t)$ im Intervall $[-1, 1]$.

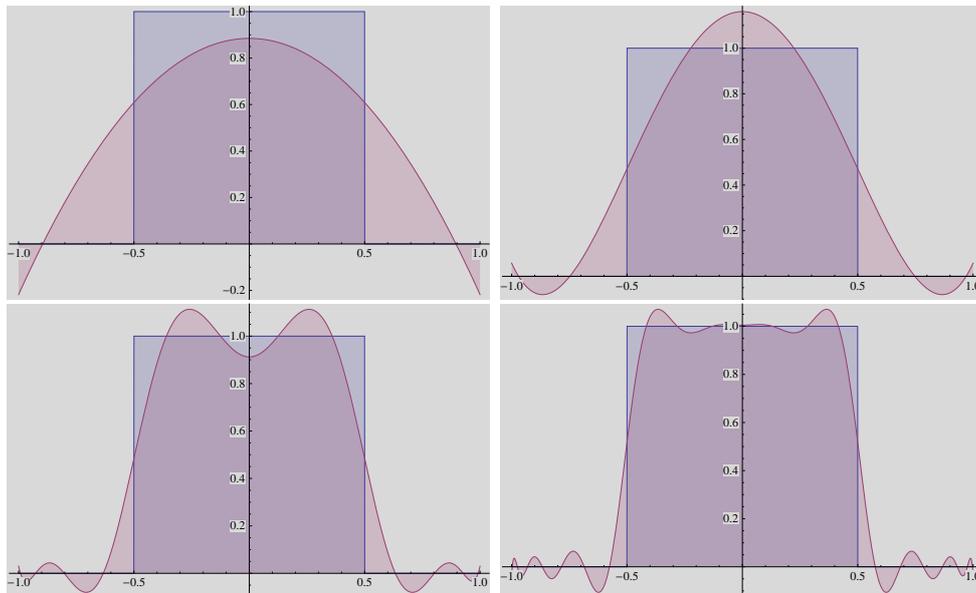


Abbildung 1.10: Grafische Darstellung der Funktion $p(t)$ und ihrer Approximationen $b_2^*(t)$, $b_5^*(t)$, $b_{10}^*(t)$, $b_{20}^*(t)$ im Intervall $[-1, 1]$.

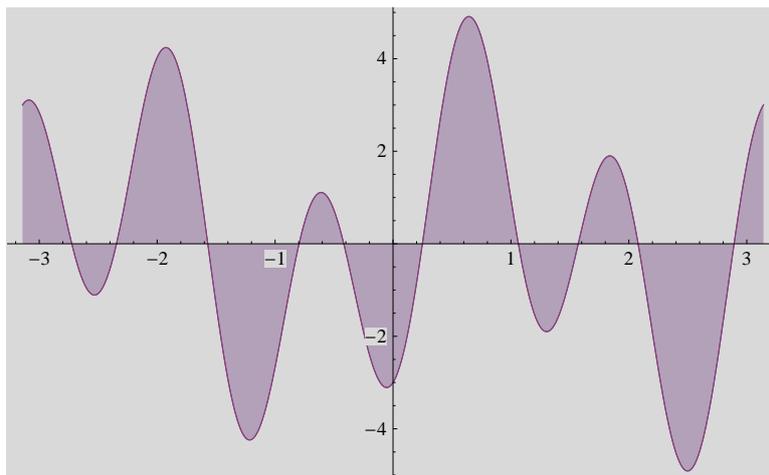


Abbildung 1.11: Grafische Darstellung der Funktion $sa(t)$ im Intervall $[-\pi, \pi]$.

siehe Abbildung 1.11, ist die Situation nicht überraschend.

Diese Funktion gehört selbst zum Approximationsraum \mathcal{U}_5 mit der Basis

$$\{\sin(kt)\}_{1 \leq k \leq 5} \cup \{\cos(kt)\}_{0 \leq k \leq 5}$$

und wird dort exakt dargestellt. Die ersten Approximationen sind

$$sa_k^*(t) = \begin{cases} 0 & \text{für } k \in \{0, 1\} \\ 2 \sin(2t) & \text{für } k \in \{2, 3, 4\} \\ 2 \sin(2t) - 3 \cos(5t) & \text{für } k \geq 5 \end{cases}$$

Der mittlere Fall $sa_2^*(t)$ wird in der Abbildung 1.12 dargestellt.

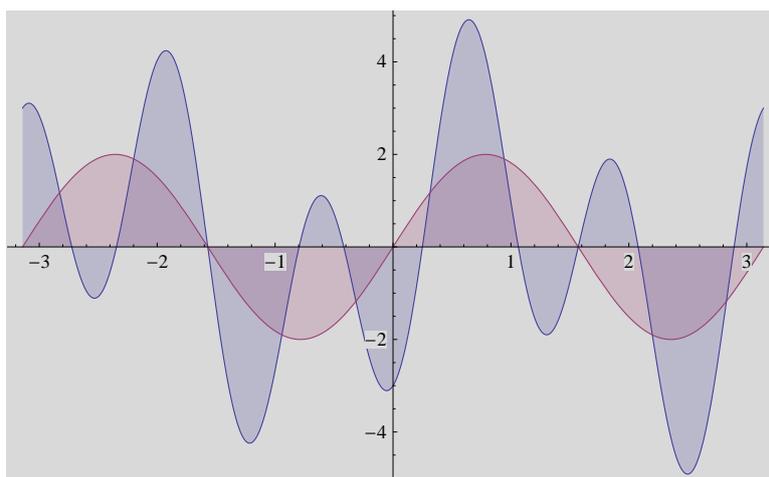


Abbildung 1.12: Grafische Darstellung der Funktion $s1(t)$ und ihrer Approximation $sa_2^*(t)$ im Intervall $[-\pi, \pi]$.

Interessanter ist der Fall einer periodischen Funktion wie

$$sb(t) = 2 \sin(t) - 3 \cos(\sqrt{2}t),$$

die zwar Linearkombination von periodischen Funktionen ist, deren Periodenlängen aber "inkommensurabel" sind, siehe Abbildung 1.13

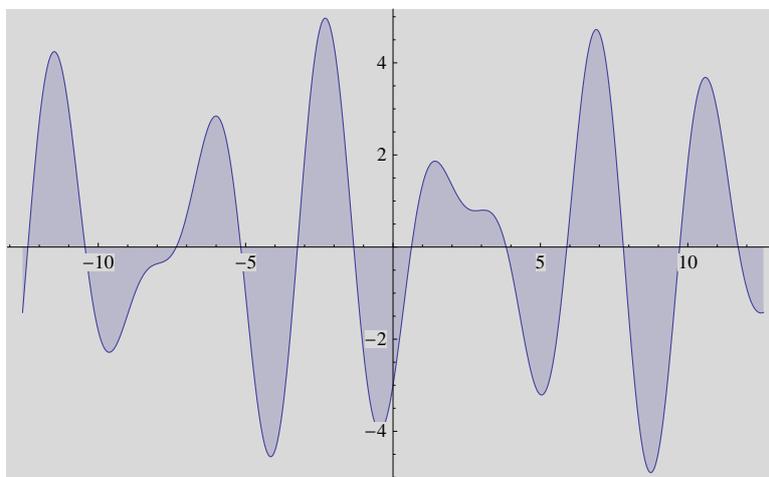


Abbildung 1.13: Grafische Darstellung der Funktion $sb(t)$ im Intervall $[-4\pi, 4\pi]$.

Hier sehen die approximierenden trigonometrischen Polynome viel abenteuerlicher aus:

$$sb_0^*(t) = -\frac{3 \sin(\sqrt{2}\pi)}{\sqrt{2}\pi}$$

$$sb_1^*(t) = \frac{3 \sin(\sqrt{2}\pi) (4 \cos(t) - 1)}{\sqrt{2}\pi}$$

$$sb_2^*(t) = \frac{4\pi \sin(2t) + 12\sqrt{2} \sin(\sqrt{2}\pi) \cos(t) + 6\sqrt{2} \sin(\sqrt{2}\pi) \cos(2t) - 3\sqrt{2} \sin(\sqrt{2}\pi)}{2\pi}$$

$$sb_3^*(t) = \frac{1}{14\pi} \left(28\pi \sin(2t) + 84\sqrt{2} \sin(\sqrt{2}\pi) \cos(t) + 42\sqrt{2} \sin(\sqrt{2}\pi) \cos(2t) \right. \\ \left. 12 - \sqrt{2} \sin(\sqrt{2}\pi) \cos(3t) - 21\sqrt{2} \sin(\sqrt{2}\pi) \right)$$

Hier gelingt die Approximation auf dem Intervall $[-\pi, \pi]$ schon mit wenigen Termen erstaunlich gut:

Man sollte sich aber nicht täuschen lassen. sieht man sich diese Approximation auf einem grösseren Intervall an, so sind die Abweichungen sofort drastisch, siehe Abbildung 1.15.

Warnung: Verwende nie "Approximationen" über einem Bereich, für den sie nicht gemacht wurden!

2. Approximation eines Polynoms

Für ein Polynom wie

$$p(t) = t^3 - t^2 + 1$$

beobachtet man ein ähnliches Verhalten der Approximationen $g_k^*(t)$ mit trigonometrischen Polynomen.

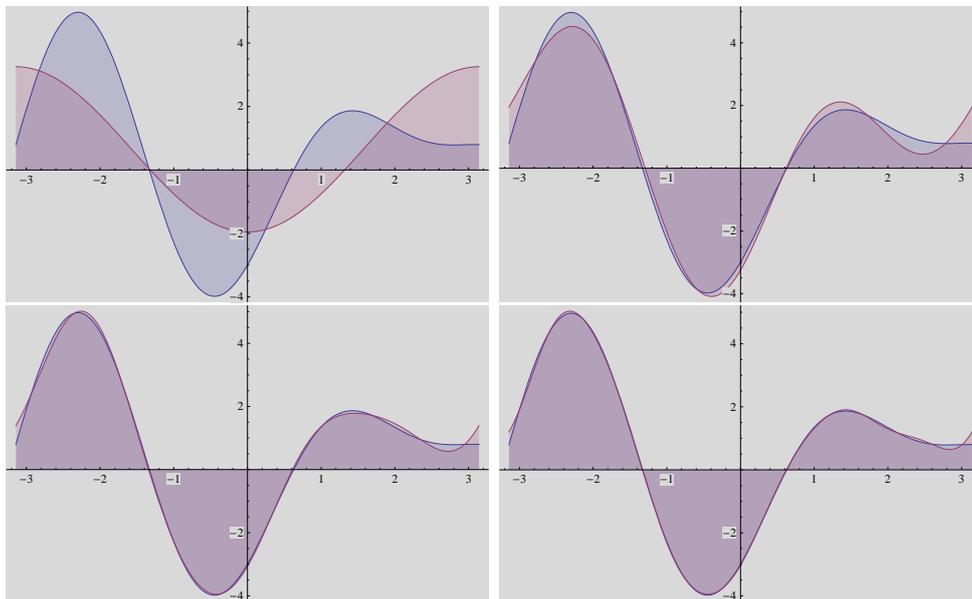


Abbildung 1.14: Grafische Darstellung der Funktion $sb(t)$ und ihrer Approximationen $sb_1^*(t)$, $sb_2^*(t)$, $sb_4^*(t)$, $sb_6^*(t)$ im Intervall $[-\pi, \pi]$.

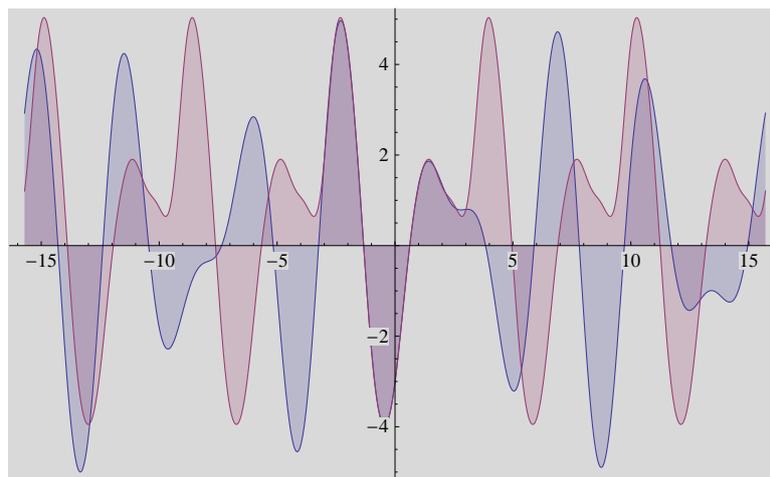


Abbildung 1.15: Grafische Darstellung der Funktionen $sb(t)$ und $sb_6^*(t)$ im Intervall $[-5\pi, 5\pi]$.

Die ersten Approximationen sind

$$\begin{aligned}
 g_0^*(t) &= 1 - \frac{\pi^2}{3} \\
 g_1^*(t) &= g_0^*(t) + 2(\pi^2 - 6)\sin(t) + 4\cos(t) \\
 g_2^*(t) &= g_1^*(t) + (3 - 2\pi^2)\sin(t)\cos(t) - \cos(2t) \\
 g_3^*(t) &= g_2^*(t) + \frac{2}{9}((3\pi^2 - 2)\sin(3t) + 2\cos(3t)) \\
 g_4^*(t) &= g_3^*(t) + \frac{1}{16}((3 - 8\pi^2)\sin(4t) - 4\cos(4t)) \\
 g_5^*(t) &= g_4^*(t) + \frac{2}{125}((25\pi^2 - 6)\sin(5t) + 10\cos(5t))
 \end{aligned}$$

In der grafischen Darstellung von Abbildung 1.16 erkennt man, dass man schnell eine gute Approximation von $g(t)$ im relevanten Intervall $[-\pi, \pi]$ erhält.

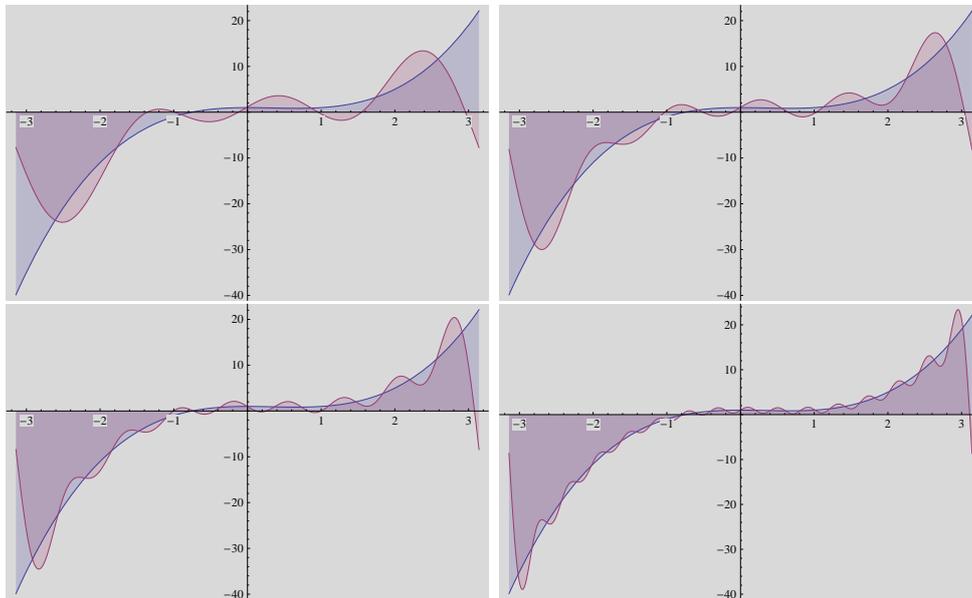


Abbildung 1.16: Grafische Darstellung der Funktion $g(t)$ und ihrer Approximationen $g_3^*(t)$, $g_4^*(t)$, $g_5^*(t)$, $g_8^*(t)$, $g_{15}^*(t)$ im Intervall $[-\pi, \pi]$.

Aber auch hier gilt, dass man diese Approximationen nur in dem für sie bestimmten Intervall betrachten soll, nicht aber darüber hinaus – siehe Abbildung 1.17. Polynome und periodische Funktionen verhalten sich global (auf der ganzen reellen Achse) grundsätzlich sehr verschieden!

3. Approximation einer “glatten” Funktion

Zur Illustration soll die weiter oben bereits verwendete Funktion

$$h(t) = \frac{e^{-t}}{\sqrt{1+t^2}}$$

auch mit trigonometrischen Polynomen approximiert werden. Die Fälle $h_3^*(t)$

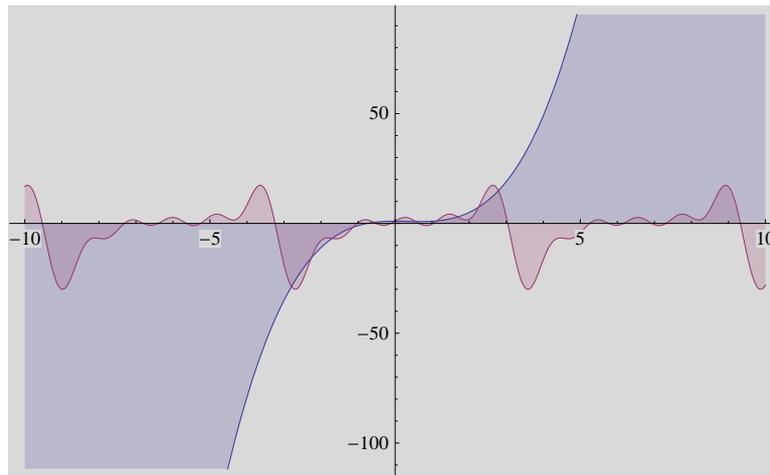


Abbildung 1.17: Grafische Darstellung der Funktion $g(t)$ und ihrer Approximationen $g_5^*(t)$ im Intervall $[-10, 10]$.

und $h_8^*(t)$ werden herausgegriffen.

$$\begin{aligned}
 h_3^*(t) &= 1.64123 - 1.69552 \sin(t) - 0.718812 \sin(3t) - 0.912758 \cos(t) \\
 &\quad + 0.363351 \cos(2t) - 0.154841 \cos(3t) + 1.82672 \sin(t) \cos(t) \\
 h_8^*(t) &= +1.64123 - 1.69552 \sin(t) - 0.718812 \sin(3t) + 0.531918 \sin(4t) \\
 &\quad - 0.438042 \sin(5t) + 0.365121 \sin(6t) - 0.315103 \sin(7t) \\
 &\quad + 0.27618 \sin(8t) - 0.912758 \cos(t) + 0.363351 \cos(2t) \\
 &\quad - 0.154841 \cos(3t) + 0.098006 \cos(4t) - 0.0605853 \cos(5t) \\
 &\quad + 0.0435327 \cos(6t) - 0.031737 \cos(7t) + 0.0245214 \cos(8t) \\
 &\quad + 1.82672 \sin(t) \cos(t)
 \end{aligned}$$

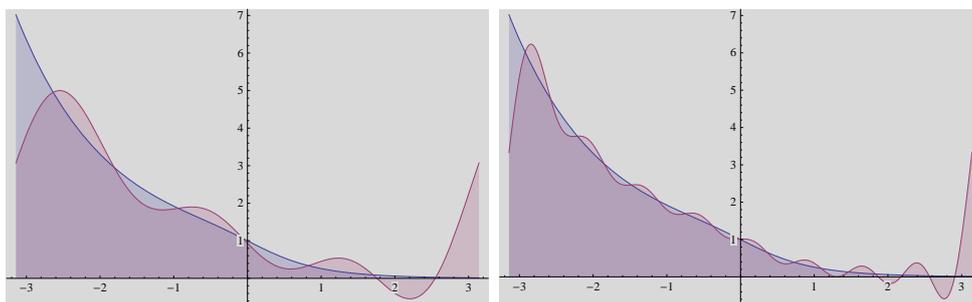


Abbildung 1.18: Grafische Darstellung der Funktion $h(t)$ und ihrer Approximationen $h_3^*(t)$, $h_8^*(t)$ im Intervall $[-10, 10]$.

In Abbildung 1.18 deutet sich schon an, dass das gute Approximationsverhalten ausserhalb des Intervalls $[-\pi, \pi]$ rasant verloren geht.

4. Approximation einer unstetigen Funktion

Jetzt soll die Funktion

$$b(t) = \begin{cases} 1 & \text{falls } |t| \leq \frac{1}{2} \\ 0 & \text{sonst} \end{cases}$$

im Intervall $[-1, 1]$ approximiert werden. Das ist besonders instruktiv, da es exemplarisch zeigt, dass die Fourier-Approximation mit den trigonometrischen Polynomen ihre Probleme mit Unstetigkeitsstellen hat. Man kann ohne Probleme die Approximationen explizit ausrechnen:

$$b_k^*(t) = \frac{1}{2\pi} + \frac{1}{\pi} \sum_{j=1}^k \frac{\sin(j/2)}{j/2} \cos(jt)$$

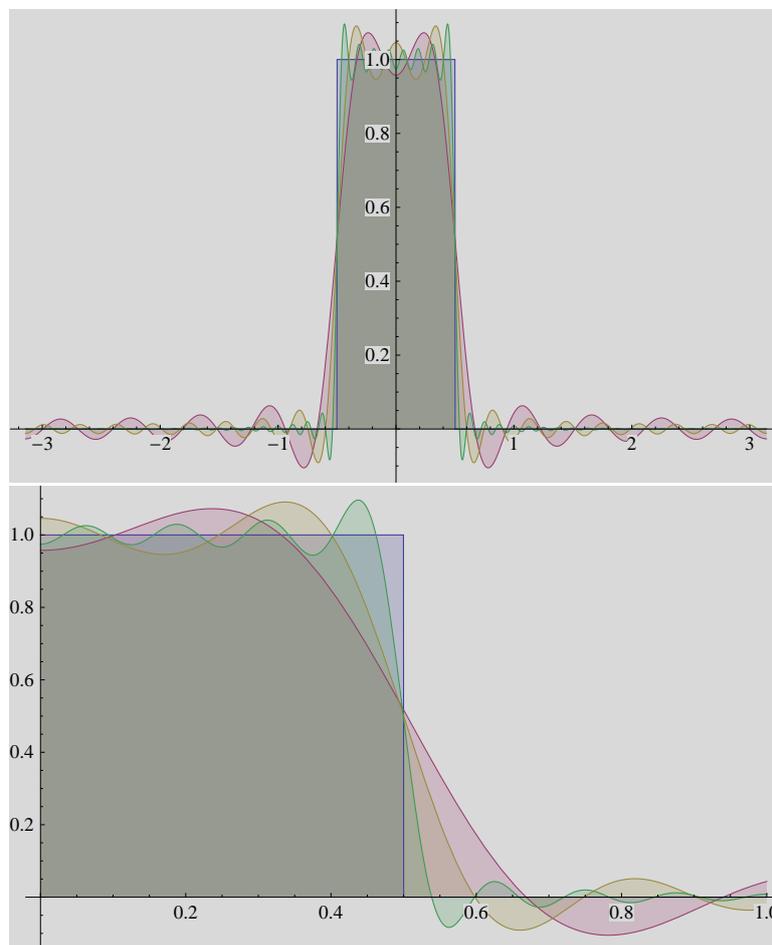


Abbildung 1.19: Grafische Darstellung der Funktion $b(t)$ und ihrer Approximationen $b_{10}^*(t)$, $b_{20}^*(t)$ und b_{50} im Intervall $[-\pi, \pi]$ und in der Vergrößerung in $[0, 1]$.

Abbildung 1.19 zeigt, dass bei steigender Approximation das “Überschwingen” an der Unstetigkeitsstelle keineswegs verschwindet!

Dieses Verhalten ist in der Literatur unter dem Namen *Gibbs-Wilbraham-Phänomen* bekannt. Es beschreibt die grundsätzlichen Probleme, die Fourier-Approximationen mit lokal begrenzten Ereignissen wie Unstetigkeiten haben.

1.2.3 Approximation durch Treppenfunktionen (III)

Approximationen mit Treppenfunktionen sind aus der Analysis hinlänglich und anschaulich bekannt, wenn es um den Integralbegriff geht. Es sollen einige wenige Beispiele genügen.

1. Approximation einer periodischen Funktion

Die Funktion

$$s(t) = \sin(10t)$$

soll im Intervall $[0, 1]$ approximiert werden. Mit den Basisfunktionen $\phi_{j,k}(t)$, wie sie eingeführt wurden, gilt beispielsweise:

$$\begin{aligned} s_0^*(t) &= \frac{\sin(5)^2}{5} \phi_{0,0}(t) = 0.183907 \\ s_1^*(t) &= \frac{\sqrt{2}}{5} \sin^2\left(\frac{5}{2}\right) \phi_{1,0}(t) + \frac{\cos(5) - \cos(10)}{5\sqrt{2}} \phi_{1,1}(t) \\ &= \begin{cases} 0.101305 & \text{für } 0 \leq t < 1/2 \\ 0.158779 & \text{für } 1/2 \leq t < 1 \end{cases} \\ s_2^*(t) &= \frac{2}{5} \sin^2\left(\frac{5}{4}\right) \phi_{2,0}(t) + \frac{1}{5} \left(\cos\left(\frac{5}{2}\right) - \cos(5) \right) \phi_{2,1}(t) \\ &\quad + \frac{1}{5} \left(\cos(5) - \cos\left(\frac{15}{2}\right) \right) \phi_{2,2}(t) + \frac{1}{5} \left(\cos\left(\frac{15}{2}\right) - \cos(10) \right) \phi_{2,3}(t) \\ &= \begin{cases} 0.360229 & \text{für } 0 \leq t < 1/4 \\ -0.216961 & \text{für } 1/4 \leq t < 1/2 \\ -0.0125946 & \text{für } 1/2 \leq t < 3/4 \\ 0.237141 & \text{für } 3/4 \leq t < 1 \end{cases} \end{aligned}$$

Die Approximationen durch Treppenfunktionen reagieren naturgemäß sehr sensibel auf lokale Variationen der zu approximierenden Funktion – die Approximationskoeffizienten sind strikt lokale Größen. Die Approximationsgüte wächst mit dem Auflösungsparameter j sehr rasch, aber man soll nicht übersehen, dass der Approximationsraum \mathcal{U}_j die Dimension 2^j hat, d.h., die Dimension (und damit auch der Rechenaufwand) wächst exponentiell in j !. Man muss sich also mit relativ niedrigen Werten von j bescheiden. Was für kleine Werte von j auffällt, sind die Unstetigkeiten der approximierenden (Treppen-)Funktion. Das wird (gerade auch bei Bildern) oft als nachteilig und optisch störend empfunden. Wavelets bieten eine interessante Alternative mit Approximationen, die lokal sensibel sind, aber “glattes” Verhalten aufweisen. wie man so etwas konstruiert, liegt aber keineswegs auf der Hand!

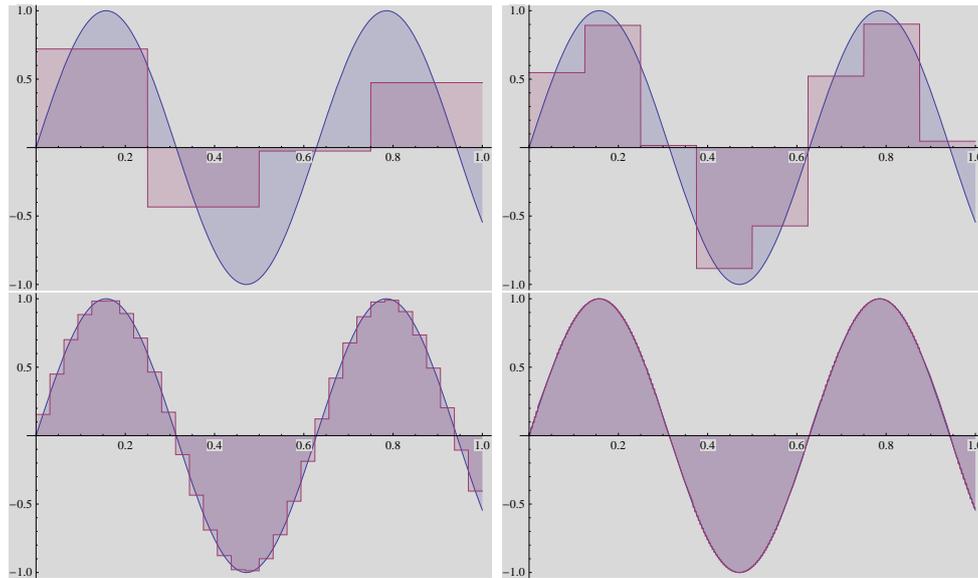


Abbildung 1.20: Grafische Darstellung der Funktion $s(t)$ und ihrer Approximationen $s_2^*(t)$, $s_3^*(t)$, $s_6^*(t)$, $s_8^*(t)$ im Intervall $[0, 1]$.

1.2.4 Approximation einer Testfunktion

Abschliessend soll aus verschiedenartigen Einzelteilen zusammengesetzte synthetisierte Testfunktion verschieden approximiert werden.

$$w(t) = \begin{cases} \frac{\sin(20t)}{t+2} & t < 0 \\ \pi(5t) & 0 \leq t < 0.2 \\ -\pi(2t) & 0.2 \leq t < 0.4 \\ \frac{\pi(t)}{2} & 0.4 \leq t < 0.5 \\ -e^{-8t^3} & 0.5 \leq t < 1 \end{cases}$$

Der Graph dieser Funktion im Intervall $[-1, 1]$ sieht so aus, wie in Abbildung 1.21 gezeigt.

1. Chebyshev-Approximation

Die optimale Approximation mittels der ersten 11 Chebyshev-Polynome ist

$$\sum_{k=0}^{10} \langle x | T_k \rangle T_k(t) = -34.1617t^{10} + 23.5804t^9 + 66.6662t^8 - 42.5675t^7 - 43.3937t^6 \\ + 24.0239t^5 + 11.8122t^4 - 4.67926t^3 - 1.66583t^2 + 0.229436t + 0.116685$$

Die grafische Darstellung von $w(t)$ und dieser Approximation in Abbildung 1.22 zeigt noch eine große Diskrepanz.

Das bessert sich, wenn man die Approximationen vom Grad 15 (Abbildung 1.23) und Grad 20 (Abbildung 1.24) betrachtet.

Man erkennt, dass die Approximationen in Bereichen, wo $w(t)$ ein glattes (und auch periodisches) Verhalten zeigt, zunehmend besser werden. Dort, wo $w(t)$

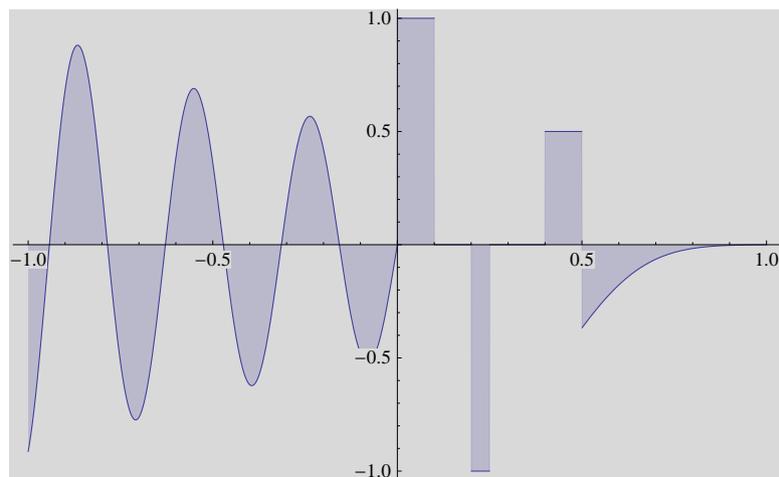


Abbildung 1.21: Grafische Darstellung der Funktion $w(t)$ im Intervall $[-1, 1]$.

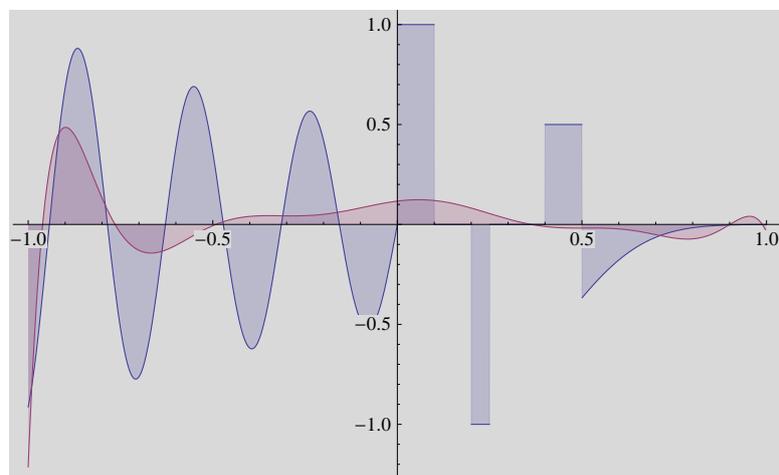


Abbildung 1.22: Grafische Darstellung der Funktion $w(t)$ und der Polynom-Approximation $w_{10}^*(t)$ im Intervall $[-1, 1]$.

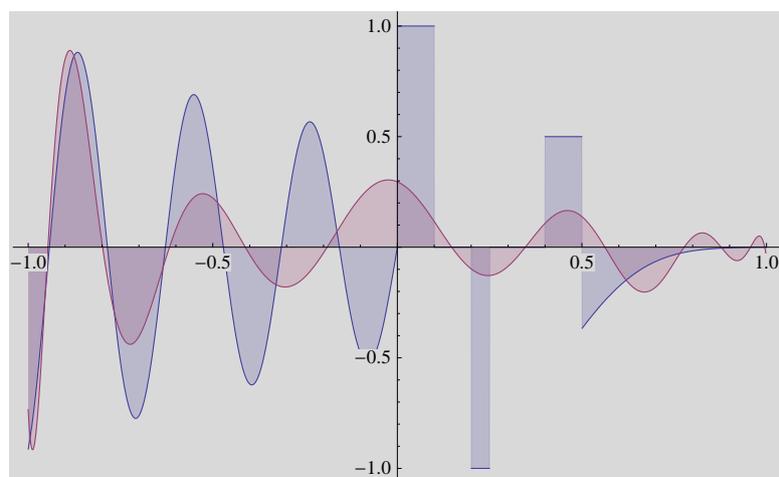


Abbildung 1.23: Grafische Darstellung der Funktion $w(t)$ und der Polynom-Approximation $w_{15}^*(t)$ im Intervall $[-1, 1]$.

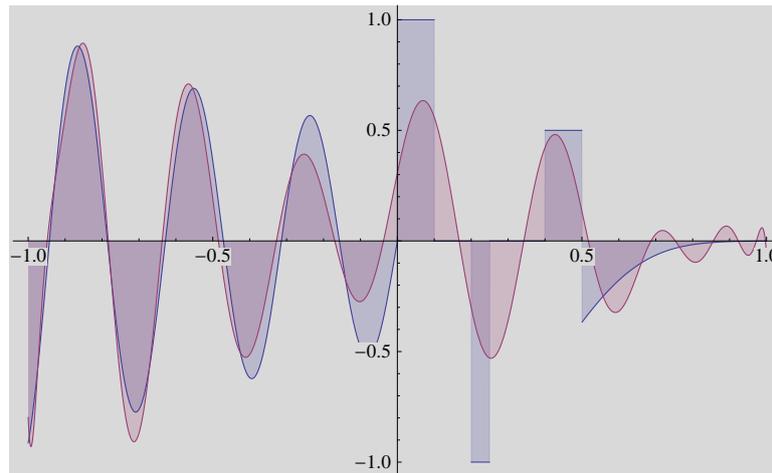


Abbildung 1.24: Grafische Darstellung der Funktion $w(t)$ und der Polynom-Approximation $w_{20}^*(t)$ im Intervall $[-1, 1]$.

auf kleinem Raum starke Änderungen aufweist, tut sich die globale Approximation aber schwer.

2. Approximation mit trigonometrischen Funktionen

Die Approximation von $w(t)$ im Raum \mathcal{U}_5 mit Basis $\{\sin(kt)\}_{1 \leq k \leq 5} \cup \{\cos(kt)\}_{0 \leq k \leq 5}$ ist

$$\begin{aligned} w_5^*(t) = & 0.0251826 - 0.0151537 \sin(t) - 0.0477764 \sin(3t) + 0.0355833 \sin(4t) \\ & + 0.00329177 \sin(5t) + 0.0360625 \cos(t) + 0.0628503 \cos(2t) + 0.0126241 \cos(3t) \\ & + 0.128708 \cos(4t) + 0.0685737 \cos(5t) + 0.0651213 \sin(t) \cos(t) \end{aligned}$$

Die grafische Darstellung in Abbildung 1.25 ist keineswegs überzeugend. Für die ‘‘doppelte Auflösung’’ (Abbildung 1.26) ergibt sie partiell ein deutlich günstigeres Bild. Allerdings machen sich die Unstetigkeiten von $w(t)$ erwartungsgemäß deutlich bemerkbar.

3. Approximation mit Treppenfunktionen

Der Vollständigkeit halber sind die Approximationen von $w(t)$ durch Treppenfunktionen in \mathcal{U}_5 und in \mathcal{U}_8 sind in den Abbildungen 1.27 und 1.28 wiedergegeben. Sie bieten keine Überraschungen. Die Approximationsgüte von $w_8^{(t)}$ ist natürlich der hohen Auflösung durch die Stufenbreite 2^{-8} geschuldet.

1.3 Intermezzo: Approximation im \mathbb{R}^2

Als Einstieg in den folgenden Abschnitt, der in das Gebiet der Wavelets und Wavelettransformationen überleitet, soll hier ein ganz simples Approximationsproblem betrachtet werden. Dabei sei $\mathcal{V} = \mathbb{R}^2$ (Vektoren als Spaltenvektoren geschrieben), ausgestattet mit der Standardbasis, bestehend aus

$$\varepsilon_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{und} \quad \varepsilon_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

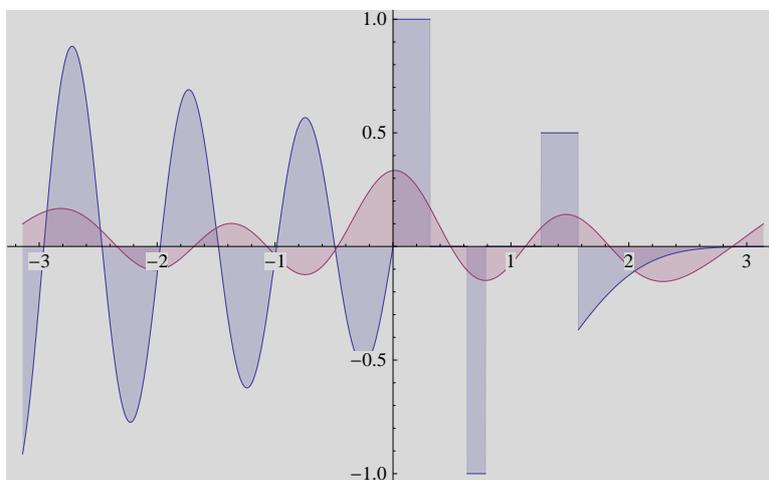


Abbildung 1.25: Grafische Darstellung der Funktion $w(t/\pi)$ und der trigonometrischen Approximation $w_5^*(t)$ im Intervall $[-\pi, \pi]$.

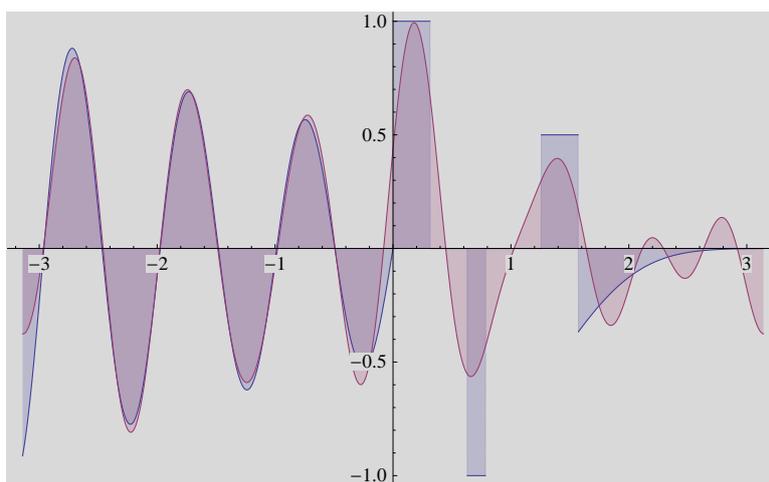


Abbildung 1.26: Grafische Darstellung der Funktion $w(t/\pi)$ und der trigonometrischen Approximation $w_{10}^*(t)$ im Intervall $[-\pi, \pi]$.

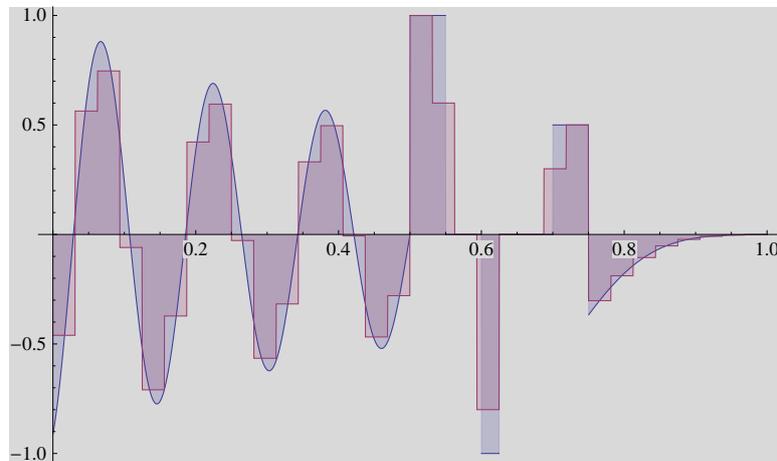


Abbildung 1.27: Grafische Darstellung der Funktion $w(2t-1)$ und der Approximation mit Treppenfunktionen $w_5^*(t)$ im Intervall $[0, 1]$.

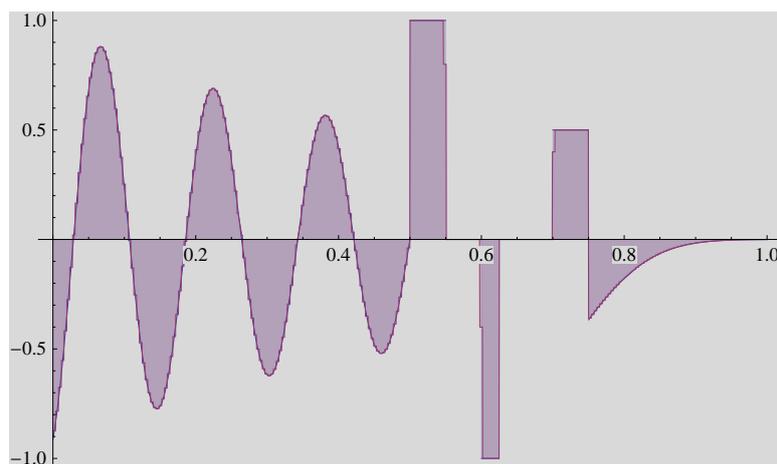


Abbildung 1.28: Grafische Darstellung der Funktion $w(2t-1)$ und der Approximation mit Treppenfunktionen $w_8^*(t)$ im Intervall $[0, 1]$.

Als Skalarprodukt wird das übliche Skalarprodukt genommen, bezüglich dem $\{\varepsilon_1, \varepsilon_2\}$ eine orthonormale Basis von \mathcal{V} ist.

Der Unterraum der “einfachen” Vektoren sei die Diagonale

$$\mathcal{U} = \left\{ \begin{bmatrix} x \\ x \end{bmatrix}; x \in \mathbb{R} \right\}.$$

Eine Basis des eindimensionalen Raumes \mathcal{U} ist gegeben durch den normierten Vektor

$$\alpha = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

Man kann diese Basis zu einer orthonormalen Basis von \mathcal{V} ergänzen, indem man den zu α orthogonalen Vektor

$$\delta = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

hinzunimmt.

Die optimale Approximation eines beliebigen Vektors $\begin{bmatrix} a \\ b \end{bmatrix} \in \mathcal{V}$ im Raum \mathcal{U} ist gegeben durch die orthogonale Projektion

$$\begin{bmatrix} a \\ b \end{bmatrix} \mapsto \langle \begin{bmatrix} a \\ b \end{bmatrix} | \alpha \rangle \alpha = \frac{a+b}{\sqrt{2}} \alpha = \frac{a+b}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

beinhaltet also den *Mittelwert* der Komponenten a und b . Die Darstellung in der Basis $\{\alpha, \delta\}$ ist

$$\begin{bmatrix} a \\ b \end{bmatrix} = \frac{a+b}{\sqrt{2}} \alpha + \frac{a-b}{\sqrt{2}} \delta = \frac{a+b}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{a-b}{2} \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$

was keine Überraschung darstellen sollte. Neben dem Mittelwert tritt nun auch die *Abweichung vom Mittelwert* als zweiter Koeffizient in Erscheinung.

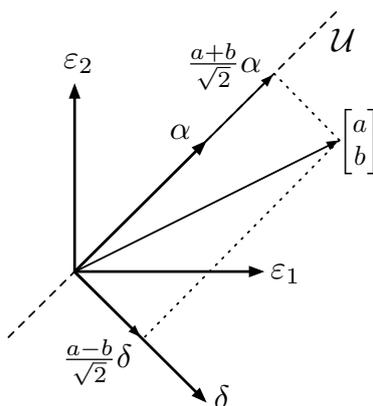


Abbildung 1.29: Geometrie der optimalen Approximation im \mathbb{R}^2

Man bezeichnet den Mittelwert $\frac{a+b}{2}$ bzw. $\frac{a+b}{\sqrt{2}}$ (als Koeffizienten von α) als *Approximationskoeffizienten* und $\frac{a-b}{2}$ bzw. $\frac{a-b}{\sqrt{2}}$ (als Koeffizienten von δ) als *Detailkoeffizienten*.

Diese Aufspaltung in Grobstruktur (=Approximation) und Feinstruktur (=Detail) ist (neben der Lokalisation) ein charakteristisches Merkmal von Wavelettransformationen, von denen der hier geschilderte Vorgang das allereinfachste Beispiel ist.

Der Vorgang, der hier beschrieben wurde, ist der einer Basistransformation zwischen den beiden Basen $\{\varepsilon_1, \varepsilon_2\}$ und $\{\alpha, \delta\}$ des Raumes $\mathcal{V} = \mathbb{R}^2$. Die Matrix dieser Basistransformation ist

$$H = [\alpha \ \delta] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

genannt *Hadamard*-Matrix¹ der Ordnung 2. Offensichtlich gilt

$$H.H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \text{also } H = H^{-1},$$

und damit erweist sich die symmetrische Matrix als orthogonal (was ja klar ist, da die Vektoren $\{\alpha, \delta\}$ eine orthonormale Basis der \mathbb{R}^2 bilden). Für die Koeffizienten von Vektoren gilt also

$$\begin{bmatrix} a \\ b \end{bmatrix} \xleftrightarrow{H} \frac{1}{\sqrt{2}} \begin{bmatrix} a+b \\ a-b \end{bmatrix}.$$

Die Tatsache, dass es sich dabei um eine orthogonale Transformation handelt, besagt nichts anderes, als dass die beiden links und rechts auftretenden Vektoren gleich Länge haben. Das sieht man ganz direkt:

$$\left\| \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = a^2 + b^2 = \left(\frac{a+b}{\sqrt{2}} \right)^2 + \left(\frac{a-b}{\sqrt{2}} \right)^2 = \left\| \frac{1}{\sqrt{2}} \begin{bmatrix} a+b \\ a-b \end{bmatrix} \right\|^2.$$

Zur Terminologie: Man bezeichnet das Quadrat der euklidischen Länge $\|v\|^2 = \langle v | v \rangle = \sum_{j=1}^n |v_n|^2$ eines Vektors (diskreten Signals) $v = (v_1, v_2, \dots, v_n)$ komplexer Zahlen als die *Energie* des Signals v . Analog spricht man bei Funktionen (kontinuierlichen Signalen) $f(t)$ von der Energie $\|f\|^2 = \int |f(t)|^2 dt$. Orthogonale Transformationen sind also energieerhaltende Transformationen.

1.4 Approximation und Skalierung

In diesem Abschnitt wird das Beispiel der Approximation mit Treppenfunktionen weitergeführt. Von Interesse sind nun die Approximationen benachbarter Auflösungsstufen. Anders als bei den Beispielen der Chebyshev-Approximation und den Fourierreihen sind hier die Basisvektoren von \mathcal{U}_j nicht zugleich Basisvektoren von \mathcal{U}_{j+1} , obwohl natürlich $\mathcal{U}_j \subset \mathcal{U}_{j+1}$ gilt.

Es bezeichne nun $\mathcal{B}_j = \{\phi_{j,k}; 0 \leq k < 2^j\}$ die Basis des Raums der Treppenfunktionen \mathcal{U}_j und entsprechend $\mathcal{B}_{j+1} = \{\phi_{j+1,k}; 0 \leq k < 2^{j+1}\}$ die Basis des Raums der Treppenfunktionen \mathcal{U}_{j+1} .

Folgende Beziehung ist ebenso einfach, wie fundamental:

$$\begin{aligned} \phi_{0,0}(t) &= \mathbf{1}_{[0,1)}(t) \\ &= \mathbf{1}_{[0,1/2)}(t) + \mathbf{1}_{[1/2,1)}(t) \\ &= \frac{1}{\sqrt{2}}(\phi_{1,0}(t) + \phi_{1,1}(t)) \end{aligned}$$

¹ Benannt nach dem französischen Mathematiker Jacques HADAMARD (1865–1963)

und diese Beziehung erweitert sich mittels Dilation und Translation zu

$$\phi_{j,k}(t) = \frac{1}{\sqrt{2}} (\phi_{j+1,2k}(t) + \phi_{j+1,2k+1}(t)) \quad (j, k \in \mathbb{Z}).$$

Diese Beziehung, die die Basisfunktionen aus \mathcal{B}_j und aus \mathcal{B}_{j+1} miteinander verknüpft, nennt man *Skalierungsgleichung* für die Haar-Basisfunktionen, oder kurz: *Haar-Skalierungsgleichung*.

Betrachtet man nun die Approximationskoeffizienten einer Funktion $f(t)$

$$a_{j,k} = \langle f | \phi_{j,k} \rangle$$

für zwei benachbarte Auflösungsstufen j und $j+1$, so ergibt sich

$$\begin{aligned} a_{j,k} &= \langle f | \phi_{j,k} \rangle \\ &= \frac{1}{\sqrt{2}} (\langle f | \phi_{j+1,2k} \rangle + \langle f | \phi_{j+1,2k+1} \rangle) \\ &= \frac{1}{\sqrt{2}} (a_{j+1,2k} + a_{j+1,2k+1}). \end{aligned}$$

Dies ist nun aber genau die Mittelwertbildung aus dem vorigen Abschnitt! Es liegt also auf der Hand, zu den *Approximationskoeffizienten* $a_{j,k}$ auch *Detailkoeffizienten* $d_{j,k}$ zu betrachten, für die

$$d_{j,k} = \frac{1}{\sqrt{2}} (a_{j+1,2k} - a_{j+1,2k+1}) \quad (j, k \in \mathbb{Z})$$

gilt – oder eleganter mit der Hadamard-Matrix H ausgedrückt:

$$\begin{bmatrix} a_{j,k} \\ d_{j,k} \end{bmatrix} = H \begin{bmatrix} a_{j+1,2k} \\ a_{j+1,2k+1} \end{bmatrix}.$$

Da H zu sich selbst invers ist, gilt ebenso

$$\begin{bmatrix} a_{j+1,2k} \\ a_{j+1,2k+1} \end{bmatrix} = H \begin{bmatrix} a_{j,k} \\ d_{j,k} \end{bmatrix}.$$

Man sieht sofort, dass sich die Detailkoeffizienten ebenfalls als innere Produkte schreiben lassen. Dazu definiert man die *Haar-Waveletfunktion*

$$\begin{aligned} \psi(t) &= \mathbf{1}_{[0,1/2)}(t) - \mathbf{1}_{[1/2,1)}(t) \\ &= \frac{1}{\sqrt{2}} (\phi_{1,0}(t) - \phi_{1,1}(t)) \end{aligned}$$

und die Funktionen, die daraus mittels Skalierung und Translation hervorgehen:

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad (j, k \in \mathbb{Z}).$$

Man sieht sofort:

$$\psi_{j,k}(t) = \frac{1}{\sqrt{2}} (\phi_{j+1,2k}(t) - \phi_{j+1,2k+1}(t)) \quad (j, k \in \mathbb{Z}).$$

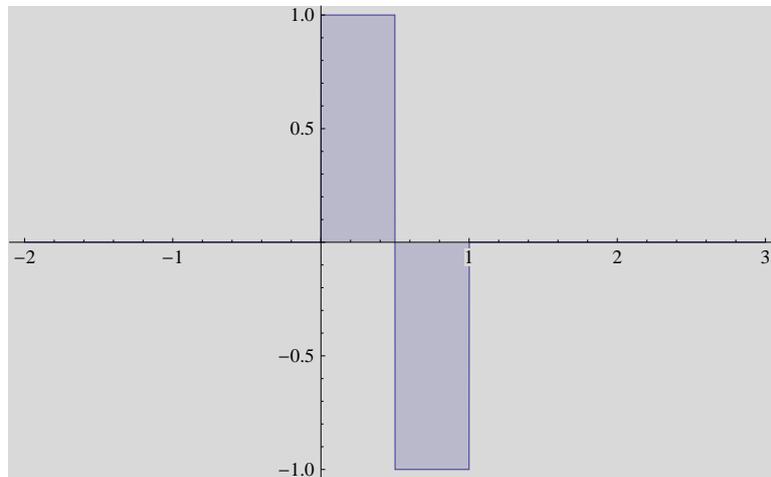


Abbildung 1.30: Grafische Darstellung der Haar-Waveletfunktion $\psi(t)$.

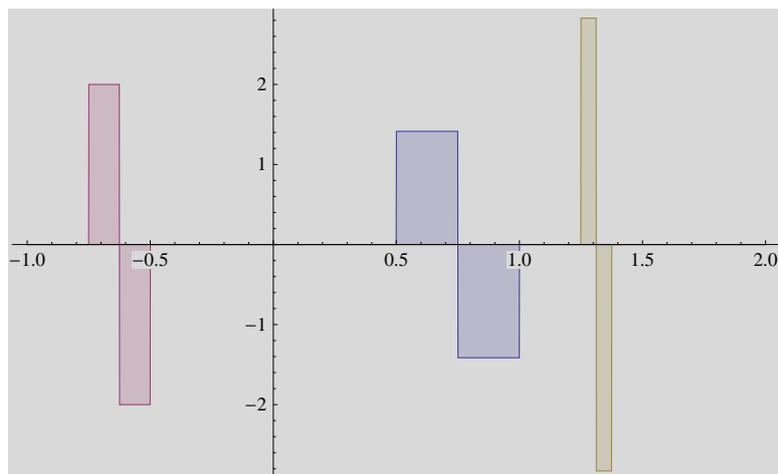


Abbildung 1.31: Grafische Darstellung der Haar-Waveletfunktionen $\psi_{1,1}(t)$, $\psi_{2,-3}(t)$ und $\psi_{3,10}(t)$.

Diese Beziehung bezeichnet man als *Haar-Waveletgleichung*, und für die Detailkoeffizienten einer Funktion $f(t)$ gilt

$$d_{j,k} = \frac{1}{\sqrt{2}} (a_{j+1,2k} - a_{j+1,2k+1}) = \langle f | \psi_{j,k} \rangle \quad (j, k \in \mathbb{Z}).$$

Auch die Waveletfunktion $\psi_{j,k}$ ist nur auf dem Intervall $I_{j,k}$ von Null verschieden – aber im Gegensatz zur Skalierungsfunktion $\phi_{j,k}(t)$ registrieren die Detailkoeffizienten die “Variation” von $f(t)$ im Intervall $I_{j,k}$:

$$2^{j/2} d_{j,k} = 2^j \left(\int_{I_{j+1,2k}} f(t) dt - \int_{I_{j+1,2k+1}} f(t) dt \right).$$

Skalierungs- und Waveletgleichung kann man zusammenfassen zu

$$\begin{bmatrix} \phi_{j,k}(t) \\ \psi_{j,k}(t) \end{bmatrix} = H \begin{bmatrix} \phi_{j+1,2k}(t) \\ \phi_{j+1,2k+1}(t) \end{bmatrix}$$

und äquivalent dazu ist

$$\begin{bmatrix} \phi_{j+1,2k}(t) \\ \phi_{j+1,2k+1}(t) \end{bmatrix} = H \begin{bmatrix} \phi_{j,k}(t) \\ \psi_{j,k}(t) \end{bmatrix}.$$

Die Waveletfunktionen $\psi_{j,k}(t)$ sind normiert und Orthogonalität gilt sogar über alle Skalierungsstufen:

$$\langle \psi_{i,k} | \psi_{j,\ell} \rangle = \int \psi_{i,k}(t) \psi_{j,\ell}(t) dt = \delta_{i,j} \delta_{k,\ell} \quad (i, j, k, \ell \in \mathbb{Z}).$$

Schliesslich sind Skalierungs- und Waveletfunktionen orthogonal zueinander:

$$\langle \phi_{i,k} | \psi_{j,\ell} \rangle = \int \phi_{i,k}(t) \psi_{j,\ell}(t) dt = 0 \quad (i, j, k, \ell \in \mathbb{Z}).$$

Mit

$$\mathcal{B}_j = \{ \phi_{j,k}(t); 0 \leq k < 2^j \}$$

wird hier² die (orthonormale) Basis des *Approximationsraumes* \mathcal{U}_j der Stufe j bezeichnet. Die Waveletfunktionen

$$\mathcal{C}_j = \{ \psi_{j,\ell}(t); 0 \leq \ell < 2^j \}$$

sind (orthonormale) Basen des *Detailraumes* \mathcal{W}_j der Stufe j .

Aus der Orthogonalität von Skalierungsfunktionen $\phi_{i,k}$ und Waveletfunktionen $\psi_{j,\ell}$ und den Skalierungs- und Waveletgleichungen folgt, dass

$$\text{sowohl } \mathcal{B}_{j+1}, \text{ als auch } \mathcal{B}_j \cup \mathcal{C}_j,$$

orthonormale Basen des Raumes \mathcal{U}_{j+1} sind. Die von der H -Matrix vermittelte Transformation ist gerade der Basiswechsel zwischen diesen beiden Basen. Jedes Element

²Später wird eine etwas andere Bezeichnung verwendet

$f \in \mathcal{U}_{j+1}$ ist somit in eindeutiger Weise als Summe $g + h$ mit $g \in \mathcal{U}_j$ und $h \in \mathcal{W}_j$ darstellbar. Mathematisch formuliert geht es um die direkte Summe von Vektorräumen:

$$\mathcal{U}_{j+1} = \mathcal{U}_j \oplus \mathcal{W}_j$$

Ist also $f(t) : [0, 1] \rightarrow \mathbb{R}$ eine Funktion und ist

$$\mathbf{a}^{(j+1)} = [a_{j+1,0}, a_{j+1,1}, a_{j+1,2}, \dots, a_{j+1,2^{j+1}-1}]$$

der Vektor der Approximationskoeffizienten auf Stufe $j + 1$ (= Koeffizienten der Approximation in der Basis \mathcal{B}_{j+1}), entsprechend

$$\mathbf{a}^{(j)} = [a_{j,0}, a_{j,1}, a_{j,2}, \dots, a_{j,2^j-1}]$$

der Vektor der Approximationskoeffizienten auf Stufe j und

$$\mathbf{d}^{(j)} = [d_{j,0}, d_{j,1}, d_{j,2}, \dots, d_{j,2^j-1}]$$

der Vektor der Detailkoeffizienten auf Stufe j , so sind $\mathbf{a}^{(j)}, \mathbf{d}^{(j)}$ die Koeffizienten bezüglich der Basis \mathcal{C}_{j+1} . Die Transformation

$$\mathbf{a}^{(j+1)} \mapsto (\mathbf{a}^{(j)}, \mathbf{d}^{(j)}), \text{ \textit{ausgeschrieben} : } \begin{bmatrix} a_{j,k} \\ d_{j,k} \end{bmatrix} = H \begin{bmatrix} a_{j+1,2k} \\ a_{j+1,2k+1} \end{bmatrix} \quad (0 \leq k < 2^j - 1)$$

bezeichnet man als (einstufige) *Haar-Analysetransformation* und die Umkehrung

$$(\mathbf{a}^{(j)}, \mathbf{d}^{(j)}) \mapsto \mathbf{a}^{(j+1)}, \text{ \textit{ausgeschrieben} : } \begin{bmatrix} a_{j+1,2k} \\ a_{j+1,2k+1} \end{bmatrix} = H \begin{bmatrix} a_{j,k} \\ d_{j,k} \end{bmatrix} \quad (0 \leq k < 2^j - 1)$$

als (einstufige) *Haar-Synthesetransformation*. Explizit:

$$\begin{bmatrix} a_{j,0} & a_{j,1} & \dots & a_{j,2^j-1} \\ d_{j,0} & d_{j,1} & \dots & d_{j,2^j-1} \end{bmatrix} = H \begin{bmatrix} a_{j+1,0} & a_{j+1,2} & \dots & a_{j+1,2^{j+1}-2} \\ a_{j+1,1} & a_{j+1,3} & \dots & a_{j+1,2^{j+1}-1} \end{bmatrix}$$

bzw.

$$\begin{bmatrix} a_{j+1,0} & a_{j+1,2} & \dots & a_{j+1,2^{j+1}-2} \\ a_{j+1,1} & a_{j+1,3} & \dots & a_{j+1,2^{j+1}-1} \end{bmatrix} = H \begin{bmatrix} a_{j,0} & a_{j,1} & \dots & a_{j,2^j-1} \\ d_{j,0} & d_{j,1} & \dots & d_{j,2^j-1} \end{bmatrix}.$$

Schematisch:

$$\begin{array}{ccccc} & & \mathbf{a}^{(j+1)} & & \\ \text{Analyse:} & \swarrow & & \searrow & \\ \mathbf{a}^{(j)} & & & & \mathbf{d}^{(j)} \\ \text{Synthese:} & \swarrow & & \searrow & \\ \mathbf{a}^{(j)} & & \mathbf{a}^{(j+1)} & & \mathbf{d}^{(j)} \end{array}$$

Bei der Analysetransformation

$$\mathbf{a}^{(j+1)} \mapsto (\mathbf{a}^{(j)}, \mathbf{d}^{(j)})$$

entsteht der Vektor $\mathbf{a}^{(j)}$ durch Mittelwertbildung, der Vektor $\mathbf{d}^{(j)}$ durch Differenzbildung. Die Operation

$$\mathbf{a}^{(j+1)} \mapsto \mathbf{a}^{(j)}$$

ist also eine ‘‘Glättungsoperation’’, bei der Unterschiede benachbarter Koeffizienten des Vektors $\mathbf{a}^{(j+1)}$ nivelliert werden. In der nachrichtentechnischen Systemtheorie würde man von einem *Tiefpassfilter (mit downsampling)* sprechen. $\mathbf{a}^{(j)}$ ist der *niederfrequente* Anteil des Signals $\mathbf{a}^{(j+1)}$.

Die Operation

$$\mathbf{a}^{(j+1)} \mapsto \mathbf{d}^{(j)}$$

verstärkt dagegen die Differenzen benachbarter Koeffizienten des Vektors $\mathbf{a}^{(j+1)}$. In der nachrichtentechnischen Systemtheorie würde man von einem *Hochpassfilter (mit downsampling)* sprechen. $\mathbf{d}^{(j)}$ ist der *hochfrequente* Anteil des Signals $\mathbf{a}^{(j+1)}$.

Die Orthogonalität der Transformation besagt

$$\|\mathbf{a}^{(j+1)}\|^2 = \|\mathbf{a}^{(j)}\|^2 + \|\mathbf{d}^{(j)}\|^2.$$

1.5 Beispiele zur einstufigen Haar-Transformation

1.5.1 1. Beispiel

Betrachten wir die Funktion

$$f(t) = \begin{cases} 0 & 0 \leq t < 0.2 \\ t - 0.2 & 0.2 \leq t < 0.6 \\ 1.5 - 2.5t & 0.6 \leq t < 0.8 \\ 0 & 0.8 \leq t < 1 \end{cases}$$

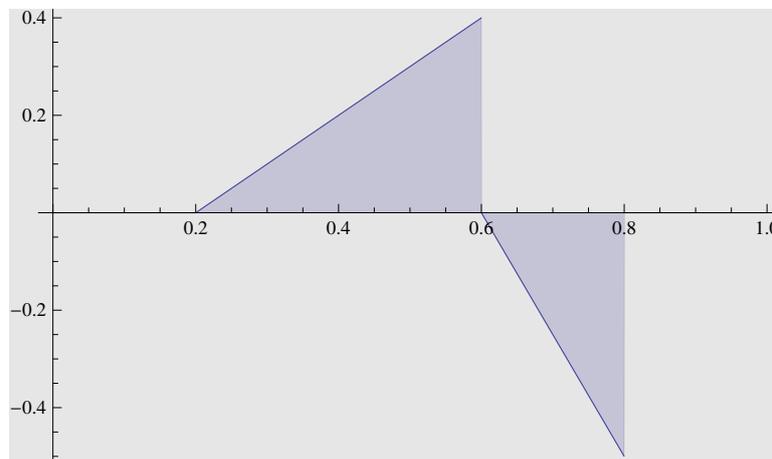


Abbildung 1.32: Plot der Funktion $f(t)$

Haar-Approximationskoeffizienten der Stufe $j = 3$

$$\mathbf{a}^{(3)} = (0., 0.00353553, 0.0397748, 0.0839689, 0.0967852, -0.0773398, -0.0618718, 0.)$$

Haar-Transformation von $\mathbf{a}^{(3)}$

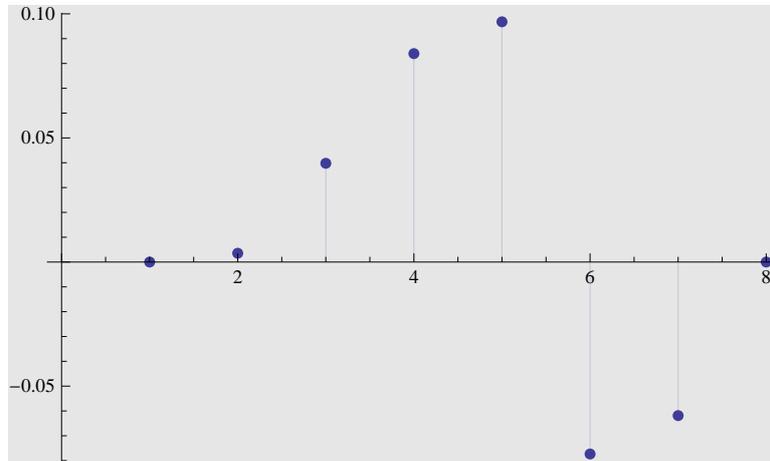


Abbildung 1.33: Approximationskoeffizienten $\mathbf{a}^{(3)}$ von $f(t)$ in \mathcal{U}_3

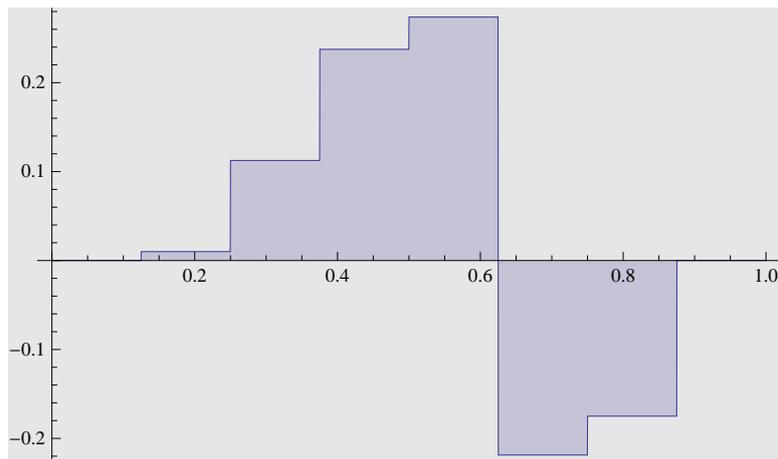


Abbildung 1.34: Approximation von $f(t)$ in \mathcal{U}_3

$$\underbrace{\{0.0025, 0.0875, 0.01375, -0.04375\}}_{\mathbf{a}^{(2)}} \underbrace{\{-0.0025, -0.03125, 0.123125, -0.04375\}}_{\mathbf{d}^{(2)}}$$

Energieerhaltung:

$$\begin{array}{lll} \|\mathbf{a}^{(3)}\| = 0.1668 & \|\mathbf{a}^{(2)}\| = 0.0988212 & \|\mathbf{d}^{(2)}\| = 0.134375 \\ \|\mathbf{a}^{(3)}\|^2 = 0.0278223 & \|\mathbf{a}^{(2)}\|^2 = 0.00976563 & \|\mathbf{d}^{(2)}\|^2 = 0.0180566 \end{array}$$

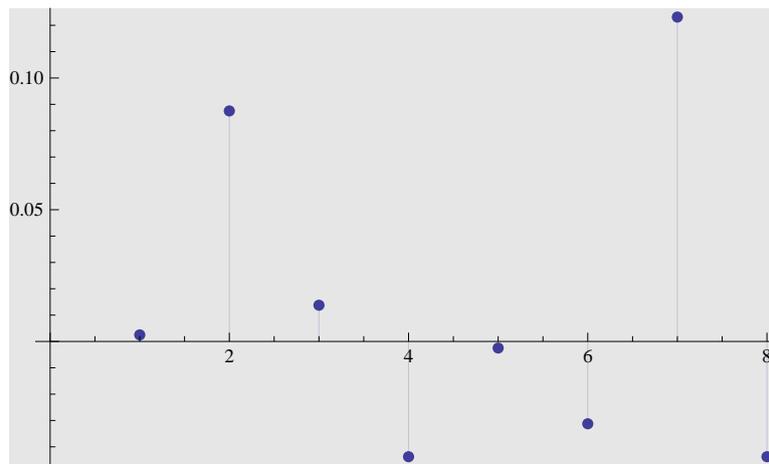


Abbildung 1.35: Haar-Transformierte ($\mathbf{a}^{(2)}, \mathbf{d}^{(2)}$) der Approximationskoeffizienten von $f(t)$ in \mathcal{U}_3

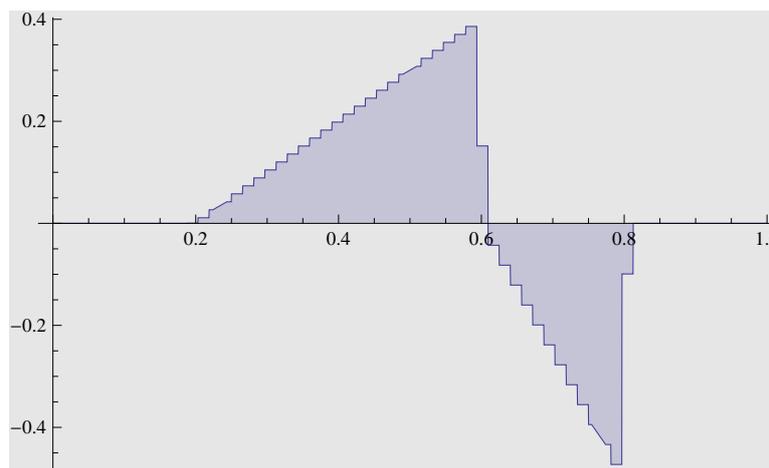


Abbildung 1.36: Approximation von $f(t)$ in \mathcal{U}_6

Energieerhaltung:

$$\begin{array}{lll} \|\mathbf{a}^{(6)}\| = 0.191641 & \|\mathbf{a}^{(5)}\| = 0.187777 & \|\mathbf{d}^{(5)}\| = 0.0383274 \\ \|\mathbf{a}^{(6)}\|^2 = 0.0367264 & \|\mathbf{a}^{(5)}\|^2 = 0.0352574 & \|\mathbf{d}^{(5)}\|^2 = 0.00146899 \end{array}$$

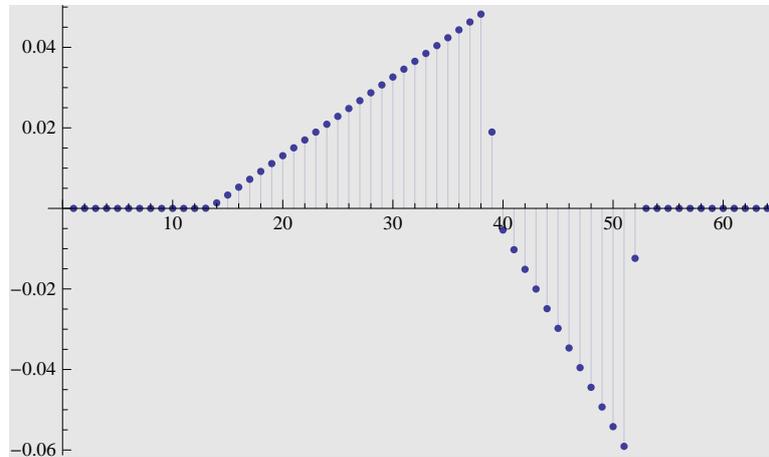


Abbildung 1.37: Approximationskoeffizienten $\mathbf{a}^{(6)}$ von $f(t)$ in \mathcal{U}_6

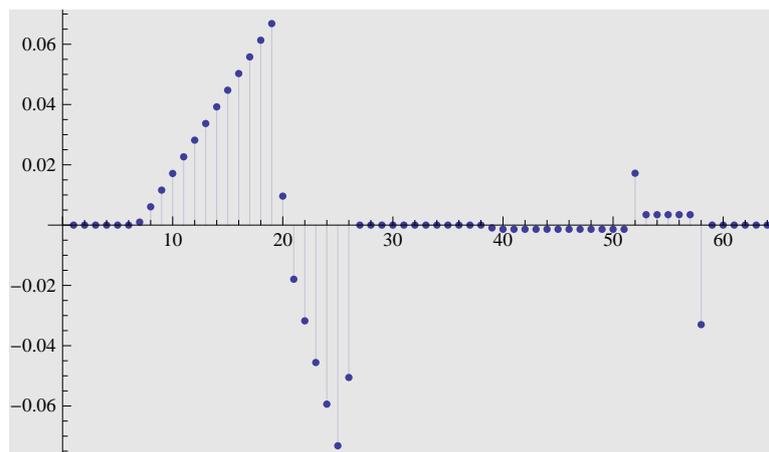


Abbildung 1.38: Haar-Transformierte $(\mathbf{a}^{(5)}, \mathbf{d}^{(5)})$ der Approximationskoeffizienten von $f(t)$ in \mathcal{U}_6

1.5.2 2. Beispiel

Betrachten wir die Funktion

$$g(t) = \frac{t \sin(10\pi t)}{0.2 + (t - 0.5)^2}$$

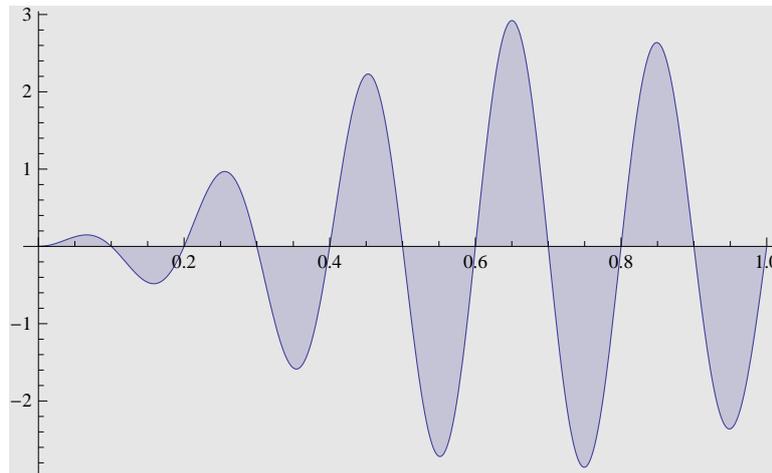


Abbildung 1.39: Plot der Funktion $g(t)$

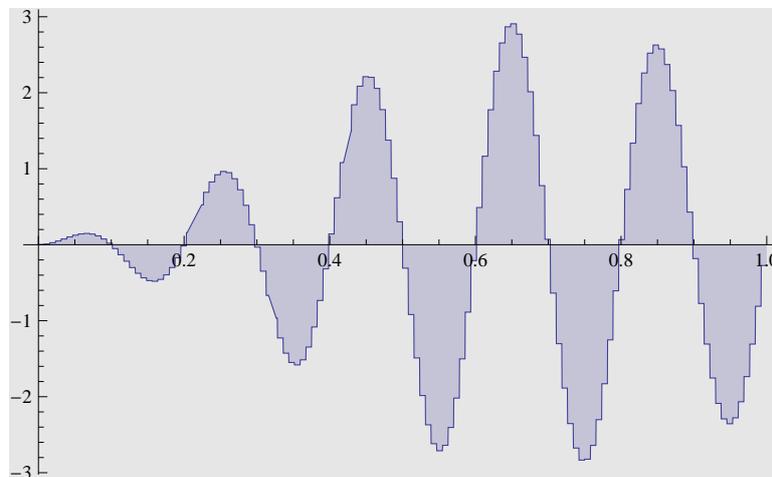
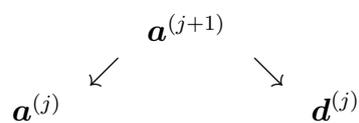


Abbildung 1.40: Approximation von $g(t)$ in \mathcal{U}_7

1.6 Haar-Transformation über mehrere Auflösungsstufen

Das Prinzip

Das Schema



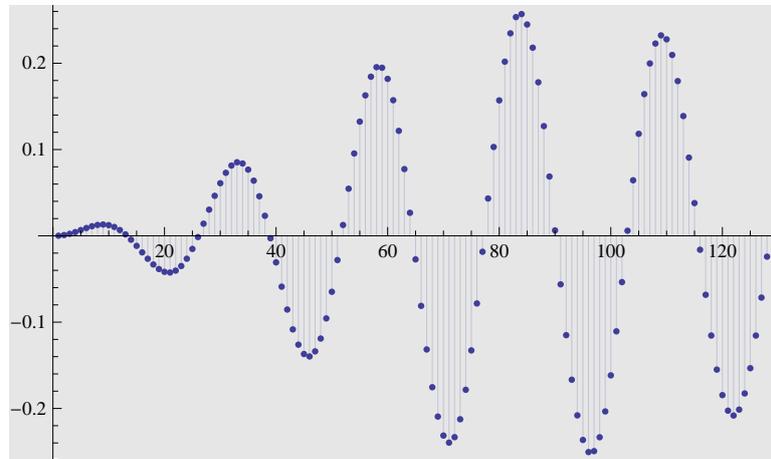


Abbildung 1.41: Approximationskoeffizienten $\mathbf{a}^{(7)}$ von $g(t)$ in \mathcal{U}_7

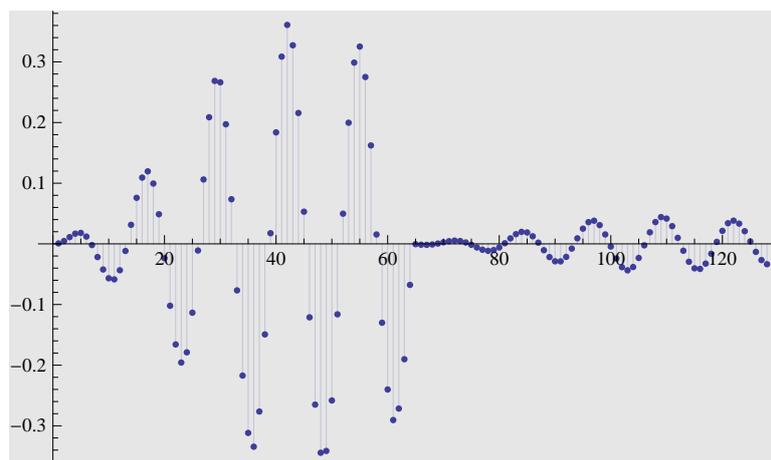
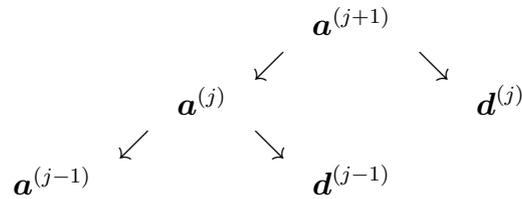


Abbildung 1.42: Haar-Transformierte $(\mathbf{a}^{(6)}, \mathbf{d}^{(6)})$ der Approximationskoeffizienten von $g(t)$ in \mathcal{U}_7

der Haar-Analysetransformation lässt sich iteriert anwenden, indem man auf die Approximationskomponente $\mathbf{a}^{(j)}$ des transformierten Signals die gleiche Transformation anwendet, wie zuvor auf $\mathbf{a}^{(j+1)}$. Schematisch:



In linearer Darstellung schreibt sich diese *zweistufige Haar-Analysetransformation* als

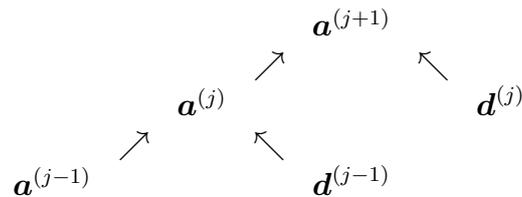
$$\mathbf{a}^{(j+1)} \mapsto (\mathbf{a}^{(j-1)}, \mathbf{d}^{(j-1)}, \mathbf{d}^{(j)}).$$

Beachte: Ist $\mathbf{a}^{(j+1)}$ ein Vektor der Dimension N , so hat $\mathbf{d}^{(j)}$ die Länge $N/2$ und $\mathbf{a}^{(j-1)}$ und $\mathbf{d}^{(j-1)}$ haben beide die Länge $N/4$.

Diese zweistufige Transformation lässt sich umkehren, linear geschrieben:

$$(\mathbf{a}^{(j-1)}, \mathbf{d}^{(j-1)}, \mathbf{d}^{(j)}) \mapsto \mathbf{a}^{(j+1)},$$

nach dem Schema



Dies ist die *zweistufige Haar-Synthesetransformation*. Auch diese Transformation ist eine orthogonale Basistransformation: in der oben eingeführten Notation ist auch

$$\mathcal{B}_{j-1} \cup \mathcal{C}_{j-1} \cup \mathcal{C}_j = \{\phi_{j-1,k}; 0 \leq k < 2^{j-1}\} \cup \{\psi_{j-1,\ell}; 0 \leq \ell < 2^{j-1}\} \cup \{\psi_{j,m}; 0 \leq m < 2^j\}$$

eine Basis von \mathcal{U}_{j+1} und es ist

$$\mathcal{U}_{j+1} = \mathcal{U}_{j-1} \oplus \mathcal{W}_{j-1} \oplus \mathcal{W}_j.$$

Es sollte klar sein, dass man dieses Vorgehen fortsetzen kann zu n -stufigen Transformationen. Das wird in einem späteren Kapitel im Detail dargestellt.

1.6.1 1. Beispiel

Die Funktion

$$h(t) = t \sin(3\pi t) \quad (0 \leq t \leq 1)$$

wird approximiert im Raum \mathcal{U}_8 .

1. Signal
2. Approximationskoeffizienten des Signals

Das Signal aus Abbildung 1.43 wird im Raum \mathcal{U}_8 approximiert. Die Approximationskoeffizienten sind in Abbildung 1.44 dargestellt. Der grafische Verlauf entspricht natürlich dem des eigentlichen Signals aus Abbildung 1.43, aber die Höhen sind um den Faktor $\sqrt{2^{-8}} = 2^{-4}$ geschrumpft (das liegt an der Normierung der Haar-Skalierungsfunktionen). Die Energie des Koeffizientenvektors ist $\|\mathbf{a}^{(8)}\|^2 = 0.163833$.

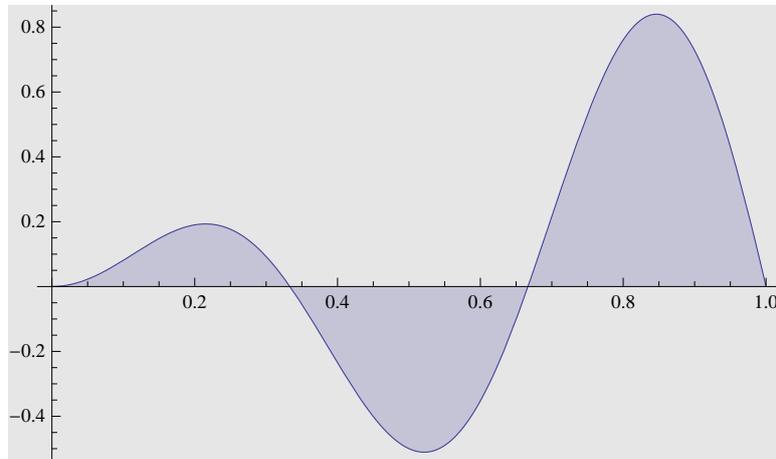


Abbildung 1.43: Grafische Darstellung der Funktion $h(t)$

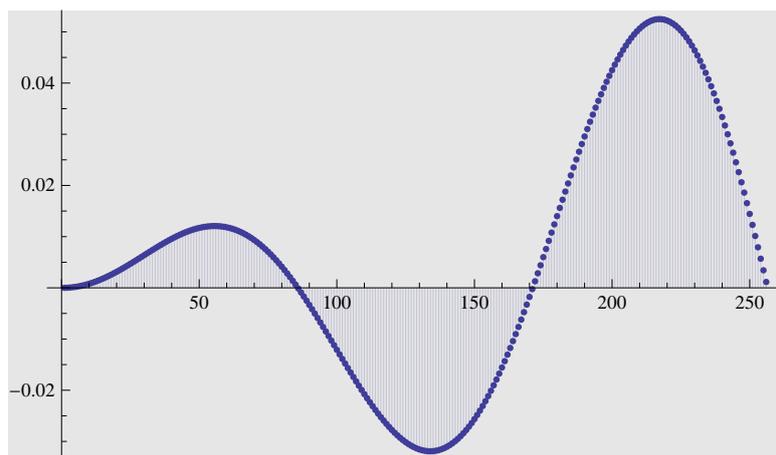


Abbildung 1.44: Approximationskoeffizienten $\mathbf{a}^{(8)}$ von $h(t)$

3. Einstufige Haar-Transformation der Approximationskoeffizienten

Eine einstufige Haartransformation liefert den Koeffizientenvektor $(\mathbf{a}^{(7)}, \mathbf{d}^{(7)})$, der in Abbildung 1.45 dargestellt ist. Für diesen Vektor gilt

$$\|\mathbf{a}^{(7)}\|^2 = 0.163776, \quad \|\mathbf{d}^{(7)}\|^2 = 0.0000574112$$

und man erkennt, dass (bis auf numerische Ungenauigkeiten) Energieerhaltung gilt, d.h. $\|\mathbf{a}^{(7)}\|^2 + \|\mathbf{d}^{(7)}\|^2 = \|\mathbf{a}^{(8)}\|^2$. Die relative Verteilung der Energie auf die beiden Teilvektoren ist

$$\mathbf{a}^{(7)} \mapsto 0.9996, \quad \mathbf{d}^{(7)} \mapsto 0.000350425.$$

Fast die gesamte Energie geht also in den niederfrequenten Anteil. Das setzt sich auch so fort.

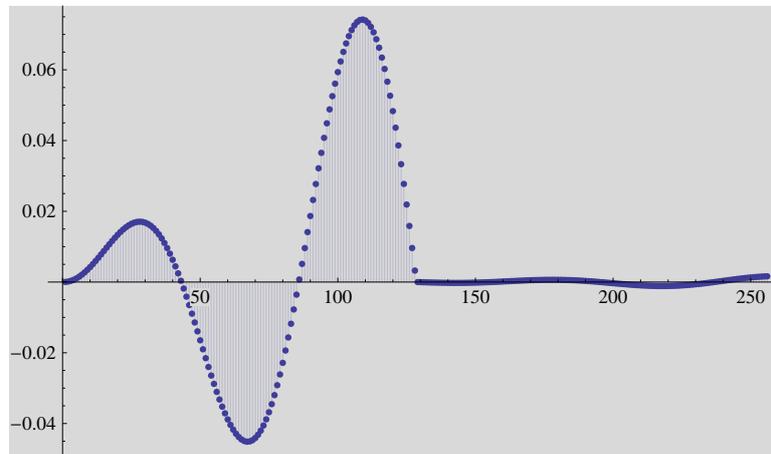


Abbildung 1.45: Koeffizienten $(\mathbf{a}^{(7)}, \mathbf{d}^{(7)})$ der einstufigen HT von $\mathbf{a}^{(8)}$

4. Zweistufige Haar-Transformation der Approximationskoeffizienten

In einer zweiten Stufe wird der Vektor der Approximationskoeffizienten $\mathbf{a}^{(7)}$ einer einstufigen Haar-Transformation unterworfen, d.h. $\mathbf{a}^{(7)} \rightarrow (\mathbf{a}^{(6)}, \mathbf{d}^{(6)})$. Der Vektor der Detailkoeffizienten $\mathbf{d}^{(7)}$ wird nicht mehr verändert. Man erhält insgesamt den Vektor

$$(\mathbf{a}^{(6)}, \mathbf{d}^{(6)}, \mathbf{d}^{(7)}).$$

In Abbildung 1.46 ist er grafisch dargestellt. Beachte die unterschiedlichen Längen: $\mathbf{a}^{(6)}$ und $\mathbf{d}^{(6)}$ haben nur noch die halbe Länge von $\mathbf{d}^{(7)}$. Die Energie der drei Teilvektoren ist

$$\|\mathbf{a}^{(6)}\|^2 = 0.163546, \quad \|\mathbf{d}^{(6)}\|^2 = 0.000229442, \quad \|\mathbf{d}^{(7)}\|^2 = 0.0000574112,$$

und man kann wieder die Energieerhaltung überprüfen. Die relative Verteilung der Energie ist

$$\mathbf{a}^{(6)} \mapsto 0.998249, \quad \mathbf{d}^{(6)} \mapsto 0.00140046, \quad \mathbf{d}^{(7)} \mapsto 0.000350425.$$

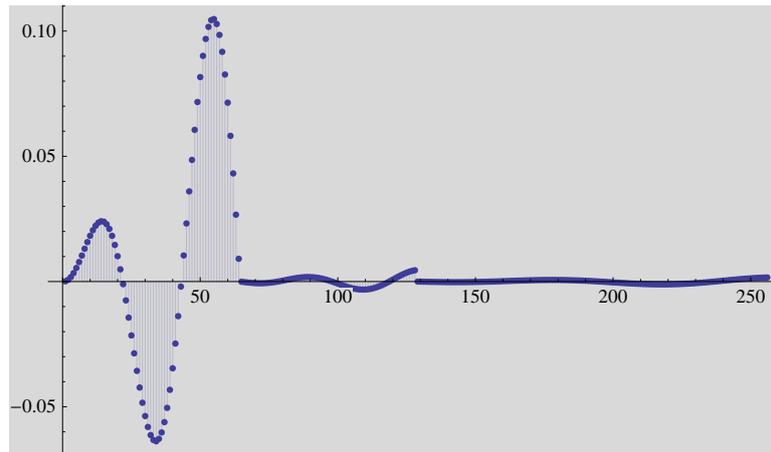


Abbildung 1.46: Koeffizienten $(\mathbf{a}^{(6)}, \mathbf{d}^{(6)}, \mathbf{d}^{(7)})$ der zweistufigen HT von $\mathbf{a}^{(8)}$

5. Dreistufige Haar-Transformation der Approximationskoeffizienten

Analog wird nun die nächste Stufe der Haar-Transformation dadurch realisiert, dass man den Teilvektor $\mathbf{a}^{(6)}$ einstufig transformiert. Das Resultat ist der Vektor

$$(\mathbf{a}^{(5)}, \mathbf{d}^{(5)}, \mathbf{d}^{(6)}, \mathbf{d}^{(7)}),$$

der in Abbildung 1.47 grafisch wiedergegeben ist. Die Energie der vier Teilvektoren ist

$$\|\mathbf{a}^{(5)}\|^2 = 0.162632, \quad \|\mathbf{d}^{(5)}\|^2 = 0.00091453, \quad \|\mathbf{d}^{(6)}\|^2 = 0.000229442, \quad \|\mathbf{d}^{(7)}\|^2 = 0.0000574112$$

und die relativen Anteile sind

$$\mathbf{a}^{(5)} \mapsto 0.992667, \quad \mathbf{d}^{(5)} \mapsto 0.00558211, \quad \mathbf{d}^{(6)} \mapsto 0.00140046, \quad \mathbf{d}^{(7)} \mapsto 0.000350425.$$

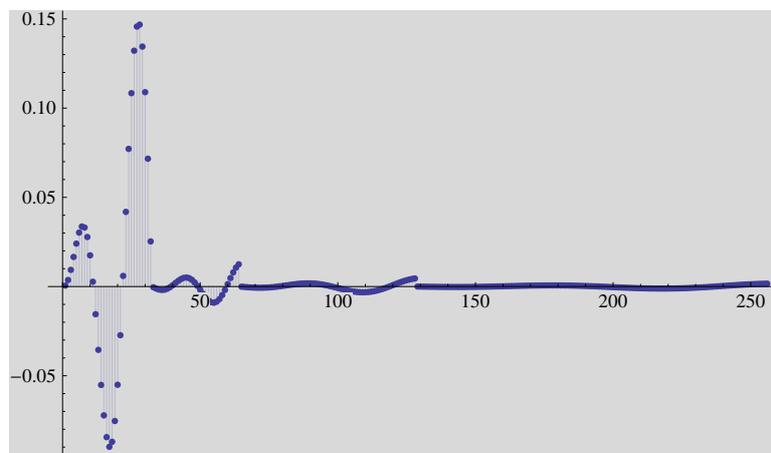


Abbildung 1.47: Koeffizienten $(\mathbf{a}^{(5)}, \mathbf{d}^{(5)}, \mathbf{d}^{(6)}, \mathbf{d}^{(7)})$ der dreistufigen HT von $\mathbf{a}^{(8)}$

6. Rekonstruktion der originalen Approximationskoeffizienten

Eine inverse Haar-Transformation über drei Stufen liefert ein Signal $\tilde{\mathbf{a}}$, das optisch nicht von $\mathbf{a}^{(8)}$ zu unterscheiden ist. Die Abweichung ist in der Tat äusserst gering und nur als “numerisches Rauschen” zu verstehen. In Abbildung 1.48 ist die Differenz geplottet. Es gilt $\|\mathbf{a}^{(8)} - \tilde{\mathbf{a}}\| \approx 2.3889 \cdot 10^{-16}$.

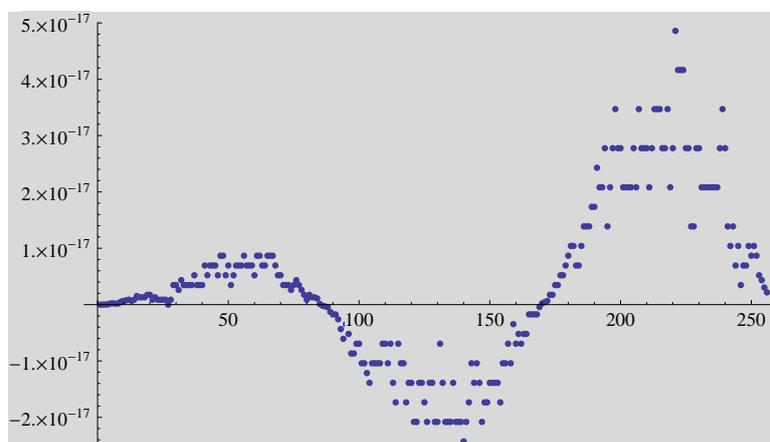


Abbildung 1.48: Koeffizienten von $\mathbf{a}^{(8)} - \tilde{\mathbf{a}}$, wobei $\tilde{\mathbf{a}}$ die iHT-Rekonstruktion über drei Stufen aus $(\mathbf{a}^{(5)}, \mathbf{d}^{(5)}, \mathbf{d}^{(6)}, \mathbf{d}^{(7)})$ ist.

1.6.2 2. Beispiel

Die Funktion des vorigen Beispiels wird nun durch zwei “Störungen” unterschiedlicher Art modifiziert: einen deutlich ausgeprägte lokale Spitze und eine kurzes hochfrequente Rauschen. Diese Änderungen machen sich auch in den Haar-Transformierten des Signals bemerkbar. Im Übrigen ist das Vorgehen genauso wie im vorigen Beispiel. Auf die Angaben zu den Energien wird verzichtet.

1. Modifiziertes Signal

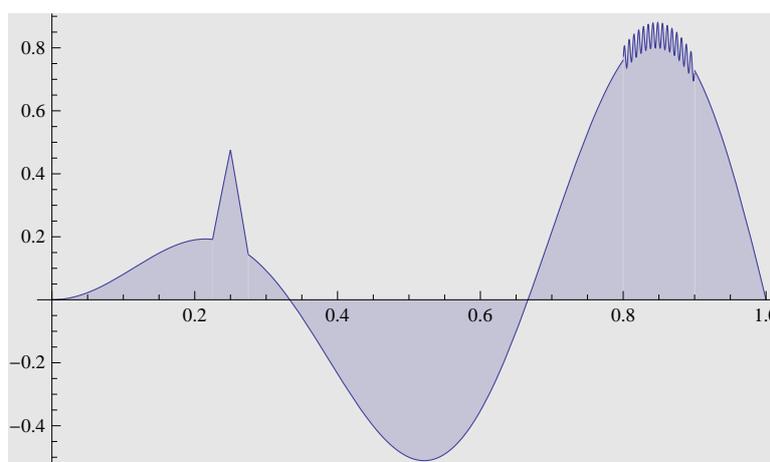


Abbildung 1.49: Modifiziertes Signal

2. Approximationskoeffizienten des modifiziertes Signals

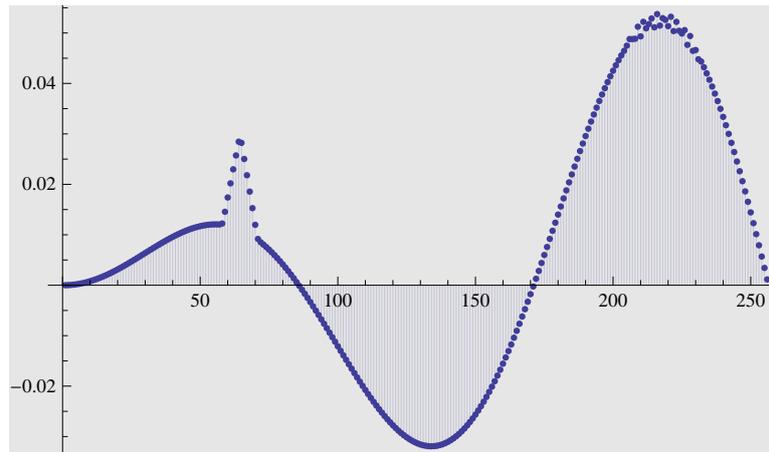


Abbildung 1.50: Approximationskoeffizienten $\mathbf{a}^{(8)}$ des modifiziertes Signals bei Approximation in \mathcal{U}_8 .

3. Einstufige Haar-Transformation

Das Resultat ist in Abbildung 1.51 wiedergegeben. Zu beachten ist, dass sich die “Störungen” sowohl im Approximationsbereich $\mathbf{a}^{(7)}$, als auch im Detailbereich $\mathbf{d}^{(7)}$ bemerkbar machen, und zwar relativ an den gleichen Positionen, wie in $\mathbf{a}^{(8)}$.

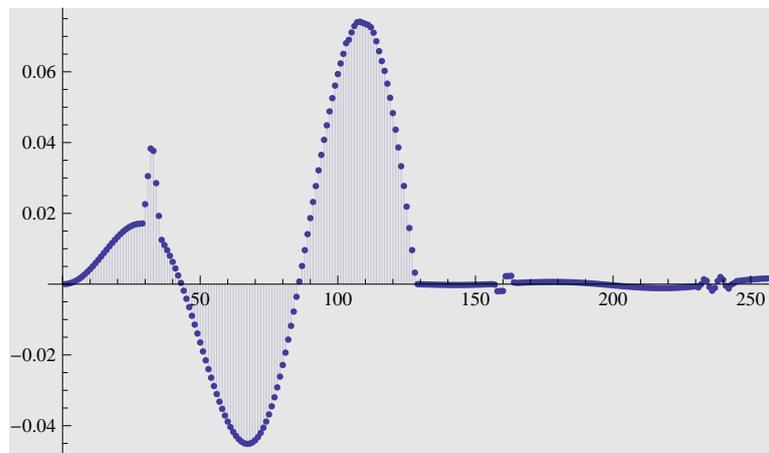


Abbildung 1.51: Einstufige Haar-Transformation der Koeffizienten aus Abbildung 1.50

4. Zweistufige Haar-Transformation

Das Resultat ist in Abbildung 1.52 wiedergegeben. Zu beachten ist, dass sich die “Störungen” sowohl im Approximationsbereich $\mathbf{a}^{(6)}$, als auch den Detailbereichen $\mathbf{d}^{(6)}$ und $\mathbf{d}^{(7)}$ bemerkbar machen, und zwar relativ an den gleichen Positionen, wie in $\mathbf{a}^{(8)}$.

5. Dreistufige Haar-Transformation

Das Resultat ist in Abbildung 1.53 wiedergegeben. Zu beachten ist, dass sich die “Störungen” sowohl im Approximationsbereich $\mathbf{a}^{(5)}$, als auch den Detailbereichen $\mathbf{d}^{(5)}$, $\mathbf{d}^{(6)}$ und $\mathbf{d}^{(7)}$ bemerkbar machen, und zwar relativ an den

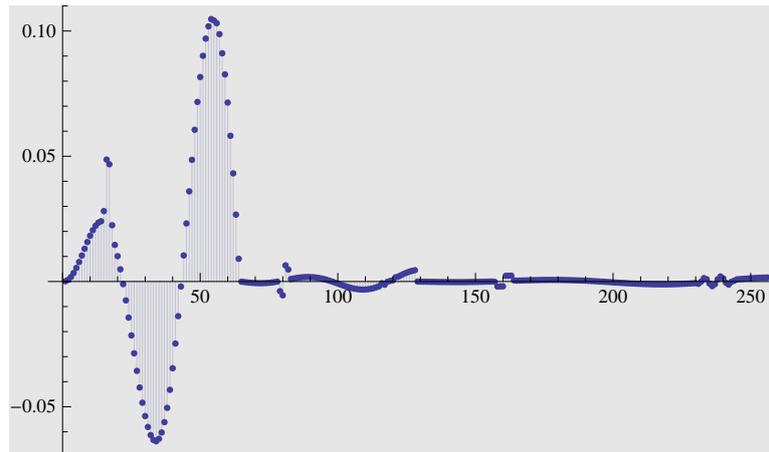


Abbildung 1.52: Zweistufige Haar-Transformation der Koeffizienten aus Abbildung 1.50

gleichen Positionen, wie in $\mathbf{a}^{(8)}$.

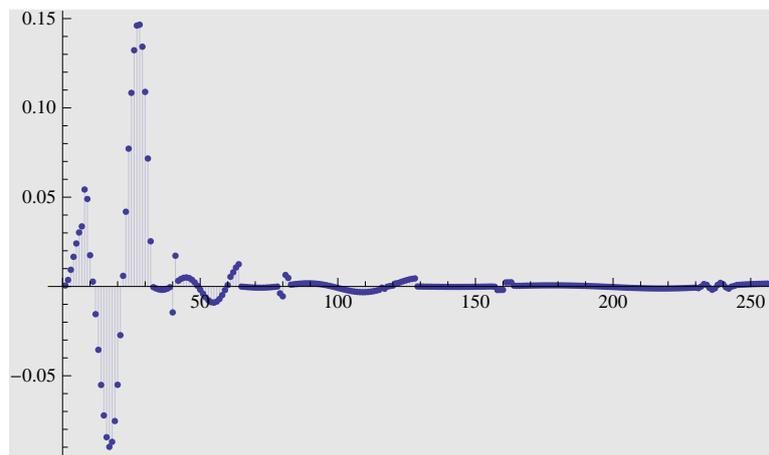


Abbildung 1.53: Dreistufige Haar-Transformation der Koeffizienten aus Abbildung 1.50

6. Rekonstruktion nach dreistufiger Haar-Transformation

Wie im vorigen Beispiel können mittels Rücktransformation die originalen Approximationskoeffizienten (bis auf Rundungsfehler) reproduziert werden. Nur die Differenz ist in Abbildung 1.54 wiedergegeben.

7. Modifikation der Daten nach zweistufiger Transformation

Die Daten nach zweistufiger Transformation, also die aus Abbildung 1.52, werden nun modifiziert, indem in den Detailbereichen $\mathbf{d}^{(6)}$ und $\mathbf{d}^{(6)}$ alle Intensitätswerte, die absolut kleiner als 0.002 sind, auf Null gesetzt werden (Abbildung 1.55). Auf diese Weise werden hochfrequente Signalanteile schwacher Energie (z.B. Rauschen) eliminiert.

Sodann wird eine zweistufige Rücktransformation vorgenommen. Man erkennt in Abbildung 1.56, dass die im rechten Bildteil befindliche hochfrequente Störung deutlich reduziert wurde, während die markante Spitze im linken Bildteil kaum beeinträchtigt wurde – siehe auch die Darstellung als Treppenfunktion

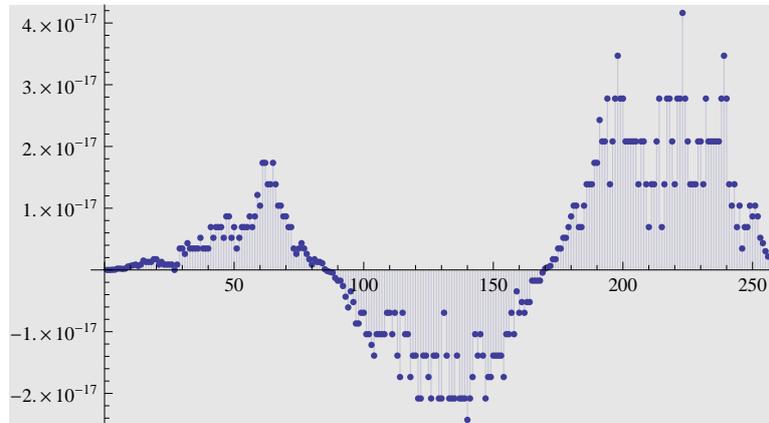


Abbildung 1.54: Rekonstruktion nach dreistufiger Haar-Transformation des Signals aus Abbildung 1.50. Gezeigt wird die Differenz zwischen Original und Rekonstruktion.

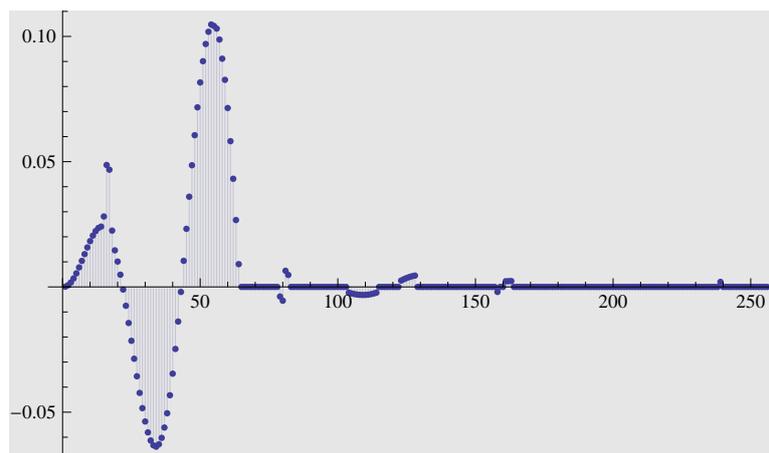


Abbildung 1.55: Modifizierte Daten

in Abbildung 1.58. Allerdings beinhaltet diese Operation auch unerwünschte Nebenwirkungen, die in der sehr einfachen Natur des Haar-Wavelets begründet sind! Zum Vergleich in Abbildung das Resultat bei Verwendung eines anderen Wavelets.

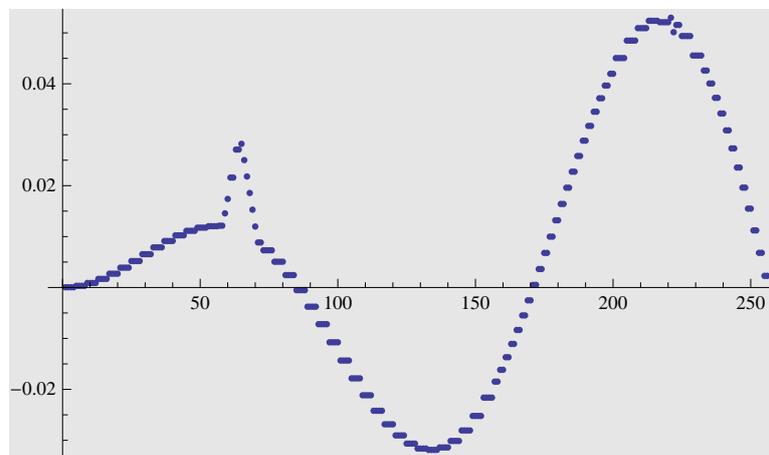


Abbildung 1.56: Rekonstruktion ausgehend von den modifizierten Daten aus Abbildung 1.55

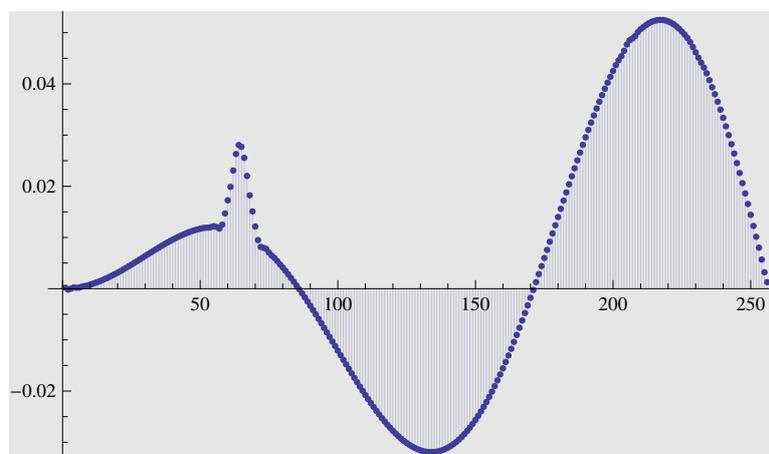


Abbildung 1.57: Rekonstruktion ausgehend von den modifizierten Daten aus Abbildung 1.55. Transformation und Rekonstruktion mittels Daubechies-D8-Wavelet.

1.7 Zweidimensionale Haar-Transformation

1.7.1 Das Prinzip

An dieser Stelle soll nur das Prinzip erläutert werden, das dann anhand eines Beispiels der Haar-Transformation eines Bildes illustriert wird. Der Formalismus wird im Detail in einem späteren Kapitel behandelt.

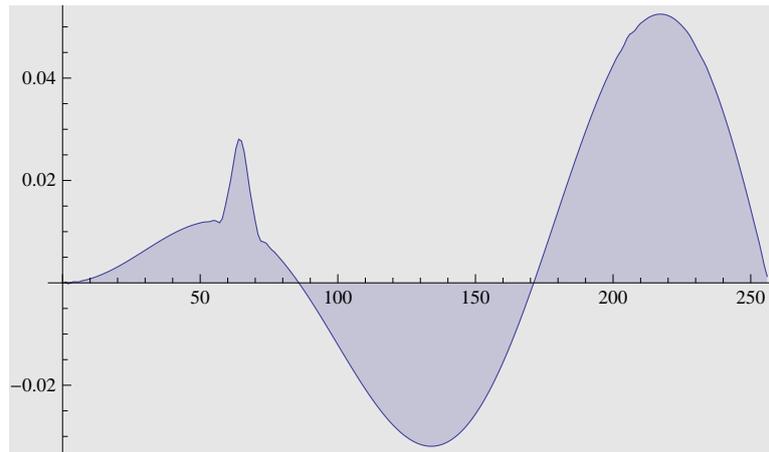


Abbildung 1.58: Darstellung der Rekonstruktion als Treppenfunktion.

Das Prinzip

Dazu betrachten wir eine 4×4 -Matrix

$$\begin{pmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{pmatrix}$$

In einem ersten Schritt wird eine Haar-Transformation separat auf jeder *Zeile* dieser Matrix durchgeföhrt. Man erhalt

$$(A|D) = \frac{1}{\sqrt{2}} \begin{pmatrix} a+b & c+d & a-b & c-d \\ e+f & g+h & e-f & g-h \\ i+j & k+l & i-j & k-l \\ m+n & o+p & m-n & o-p \end{pmatrix},$$

wobei die linke Teilmatrix

$$A = \frac{1}{\sqrt{2}} \begin{pmatrix} a+b & c+d \\ e+f & g+h \\ i+j & k+l \\ m+n & o+p \end{pmatrix}$$

die jeweiligen Approximationskoeffizienten und die rechte Teilmatrix

$$D = \frac{1}{\sqrt{2}} \begin{pmatrix} a-b & c-d \\ e-f & g-h \\ i-j & k-l \\ m-n & o-p \end{pmatrix}$$

die Detailkoeffizienten enthalt.

In einem zweiten Schritt wird eine Haar-Transformation separat auf jeder *Spalte* der eben produzierten Matrix $(A|D)$ durchgeföhrt. Dies ergibt die Matrix

$$\begin{pmatrix} AA & AD \\ DA & DD \end{pmatrix} = \frac{1}{2} \begin{pmatrix} a+b+e+f & c+d+g+h & a-b+e-f & c-d+g-h \\ i+j+m+n & k+l+o+p & i-j+m-n & k-l+o-p \\ a+b-e-f & c+d-g-h & a-b-e+f & c-d-g+h \\ i+j-m-n & k+l-o-p & i-j-m+n & k-l-o+p \end{pmatrix}.$$

In jeder Spalte sind Approximations- bzw. Detailinformationen getrennt (oben bzw. unten) getrennt enthalten, so dass sich jetzt folgendes Schema bietet:

$$\begin{pmatrix} AA & AD \\ DA & DD \end{pmatrix}$$

wobei die Teilmatrix

$$AA = \frac{1}{2} \begin{pmatrix} a + b + e + f & c + d + g + h \\ i + j + m + n & k + l + o + p \end{pmatrix}$$

die horizontale und vertikale Approximationsinformation enthält, die Matrix

$$AD = \frac{1}{2} \begin{pmatrix} a - b + e - f & c - d + g - h \\ i - j + m - n & k - l + o - p \end{pmatrix}$$

die horizontale Detail- und vertikale Approximationsinformation,

$$DA = \frac{1}{2} \begin{pmatrix} a + b - e - f & c + d - g - h \\ i + j - m - n & k + l - o - p \end{pmatrix}.$$

und schliesslich

$$DD = \frac{1}{2} \begin{pmatrix} a - b - e + f & c - d - g + h \\ i - j - m + n & k - l - o + p \end{pmatrix}.$$

die horizontalen und vertikalen Detailinformationen.

Anders formuliert: AA liefert eine durch Mittelung entstandene Approximation, AD bzw. DA registrieren horizontale bzw. vertikale Unterschiede, während DD diagonale Unterschiede wiedergibt.

Die geschilderte Transformation, die man entsprechend auf jeder Matrix (mit gerader Zeilen- und Spaltenanzahl) durch führen kann, ist eine *einstufige Haar-Transformation*. Auch dies ist eine orthogonale Transformation, also insbesondere umkehrbar. Die technischen Details werden später erläutert.

Eine zweistufige Haar-Transformation gewinnt man dadurch, dass man das gleiche (einstufige) Verfahren auf die Teilmatrix AA anwendet, die anderen Teilmatrizen AD , DA , DD aber nicht verändert. diesen Prozess kann man natürlich noch weiter analog fortführen.

1.7.2 Ein Beispiel

Aus der Sicht der Bildverarbeitung ist ein *Bild* eine Matrix von Intensitätswerten³.

Als Bildbeispiel (Abbildung 1.59) wurde hier die *Melencolia* von Albrecht Dürer (1471–1528) gewählt⁴. In der Wikipedia findet man Informationen und viele Links zur kunsthistorischen Bedeutung dieses Kupferstichs. Die mathematischen Aspekte werden beispielsweise in Eric W. Weissteins Lexikon *MathWorld*⁵ erläutert.

³ Der Einfachheit halber geht es hier nur um Schwarz-Weiss-Bilder.

⁴ Die Idee, diese fränkische Ikone der Kunstgeschichte als Demonstrationsbeispiel zu verwenden, habe ich von dem empfehlenswerten Buch von David F. Walnut [Wal02] übernommen

⁵ <http://mathworld.wolfram.com/DuerersSolid.html>



Abbildung 1.59: Albrecht Dürers Kupferstich *Melencolia* aus dem Jahr 1514 mit magischem Quadrat rechts oben.

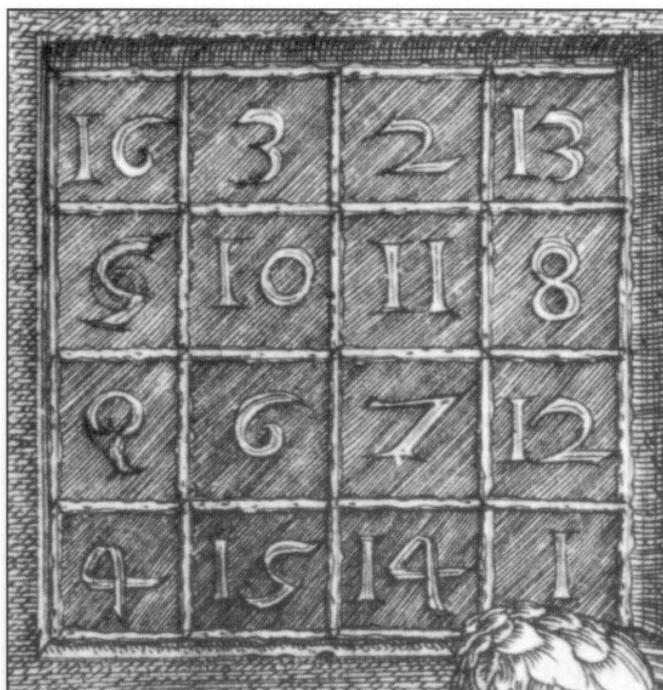


Abbildung 1.60: Das magische Quadrat aus Albrecht Dürers *Melancolia*.

Ein zufällig gewählter Ausschnitt von 6×6 Pixeln etwa besteht aus der Matrix

$$\begin{pmatrix} 0.439216 & 0.337255 & 0.345098 & 0.431373 & 0.478431 & 0.619608 \\ 0.388235 & 0.403922 & 0.505882 & 0.541176 & 0.705882 & 0.694118 \\ 0.431373 & 0.584314 & 0.666667 & 0.698039 & 0.721569 & 0.733333 \\ 0.635294 & 0.666667 & 0.745098 & 0.705882 & 0.658824 & 0.556863 \\ 0.741176 & 0.745098 & 0.713725 & 0.607843 & 0.556863 & 0.494118 \\ 0.733333 & 0.678431 & 0.619608 & 0.564706 & 0.6 & 0.666667 \end{pmatrix}$$

Das Bild insgesamt besteht aus 256×256 Pixeln von Grauwerten. In den folgenden Beispielen geht es um den linken oberen Quadranten vom Format 128×128 .

Das Ergebnis einer *zeilenweisen* Haar-Transformation auf dem Bildausschnitt zeigt die Abbildung 1.62. In der linken (Approximations-)Hälfte erkennt man eine horizontal gestauchte Version des Originals. In der rechten (Detail-)Hälfte erkennt man, ebenfalls gestaucht, vage vertikale Strukturen, wie sie in dem Original vorhanden sind. Man kann das Ergebnis für die rechte Hälfte durch Adjustieren der Intensität (das ist nicht Bestandteil der Transformation, sondern dient nur der Veranschaulichung!) besser sichtbar machen, siehe Abbildung 1.63.

Das Ergebnis einer *spaltenweisen* Haar-Transformation auf dem Bildausschnitt zeigt die Abbildung 1.64. In der oberen (Approximations-)Hälfte erkennt man eine vertikal gestauchte Version des Originals. In der unteren (Detail-)Hälfte erkennt man, ebenfalls gestaucht, vage horizontale Strukturen, wie sie in dem Original vorhanden sind. Man kann das Ergebnis für die untere Hälfte durch Adjustieren der Intensität (das ist nicht Bestandteil der Transformation, sondern dient nur der Veranschaulichung!) besser sichtbar machen, siehe Abbildung 1.65.

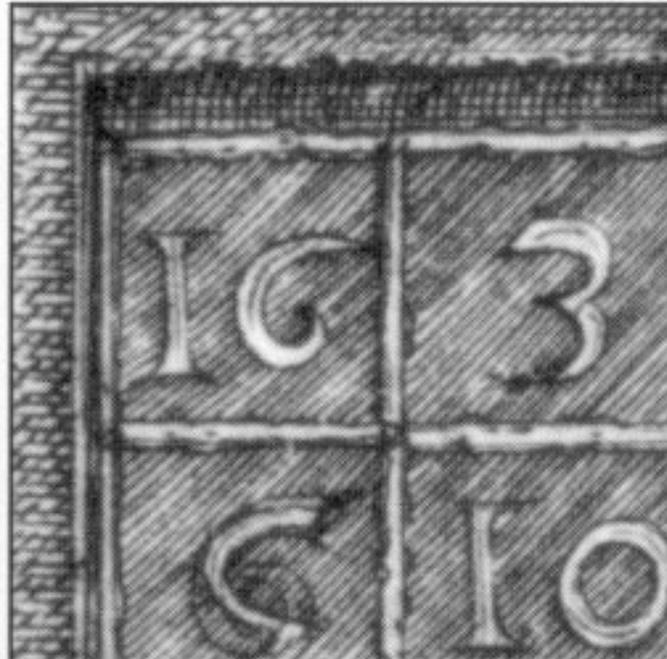


Abbildung 1.61: Ein Ausschnitt aus dem magischen Quadrat.

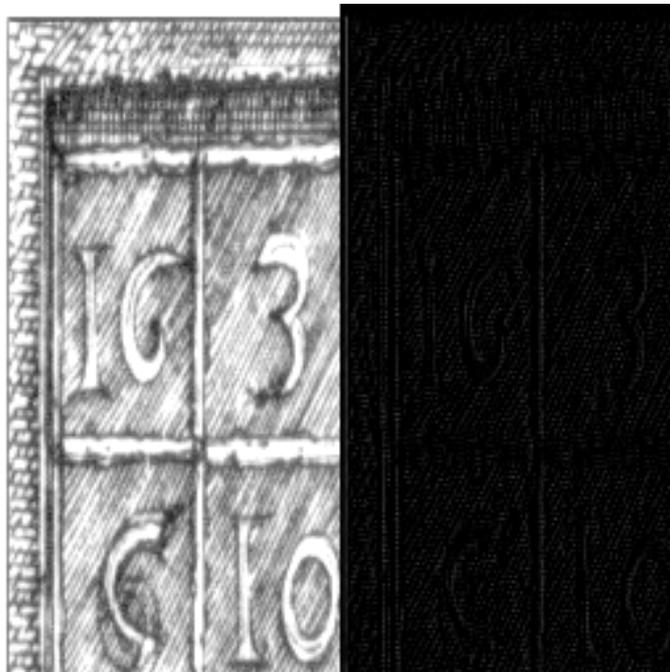


Abbildung 1.62: Haar-Transformation zeilenweise.

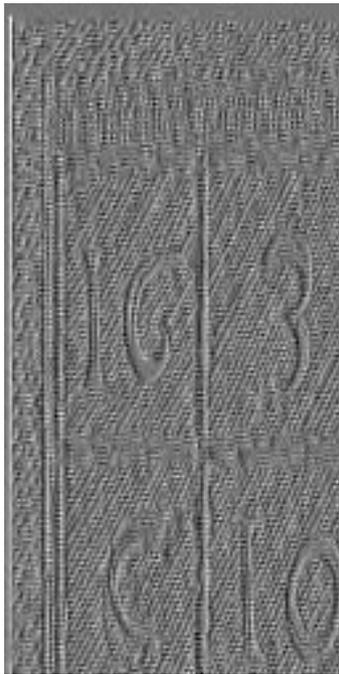


Abbildung 1.63: Haar-Transformation zeilenweise, rechte Hälfte, mit angepasster Intensität.

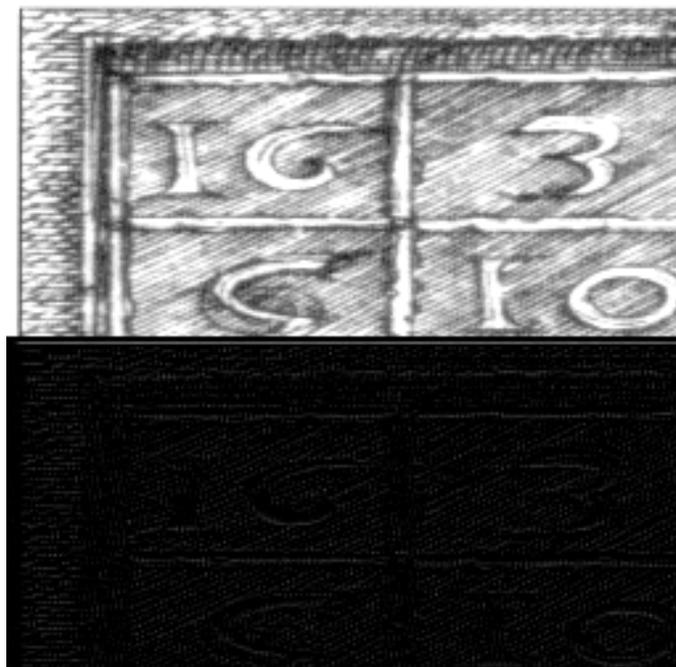


Abbildung 1.64: Haar-Transformation spaltenweise

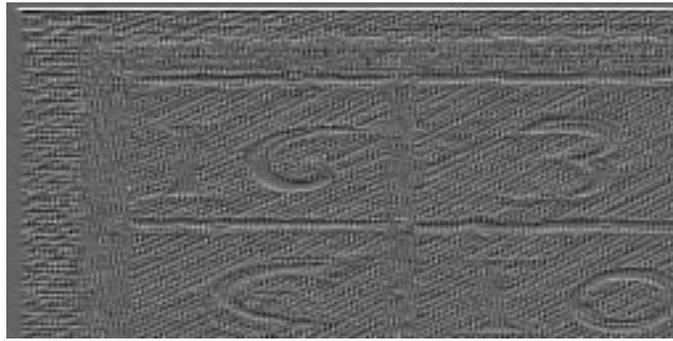


Abbildung 1.65: Haar-Transformation spaltenweise, untere Hälfte, mit angepasster Intensität

Ein einstufige zweidimensionale Haar-Transformation besteht aus einer zeilenweisen Haar-Transformation, gefolgt von einer spaltenweisen Haar-Transformation auf dem Resultat des ersten Transformationsschrittes. Man kann es auch in anderer Reihenfolge machen: erst spaltenweise, dann zeilenweise eindimensionale Transformation – das Ergebnis ist das gleiche.

Das Resultat ist in Abbildung 1.66 wiedergegeben. Eine Darstellung mit angepassten Intensitäten gibt Abbildung 1.67. Man erkennt nun vier Bereiche:

- links oben den AA-Bereich der horizontalen und vertikalen Approximation, der i.w. verkleinert und vergrößert das ursprüngliche Bild darstellt;
- rechts oben den Bereich der horizontalen Details und der vertikalen Approximation – hier sind i.w. vertikale Strukturen des Ausgangsbildes zu sehen;
- links unten den Bereich der horizontalen Approximation und der vertikalen Details – hier sind i.w. horizontale Strukturen des Ausgangsbildes zu sehen;
- rechts unten den Bereich der horizontalen und der vertikalen Details – hier sind, wenn man überhaupt noch etwas erkennt, diagonale Strukturen des Ausgangsbildes zu sehen;

Aufschlussreich ist die Verteilung der Energie auf die vier Teilbilder.

$$\begin{pmatrix} 17821.7 & 20.9915 \\ 53.1318 & 9.11389 \end{pmatrix}$$

Eine Berechnung der Energie des ursprünglichen Bildes aus Abbildung 1.61 ergibt den Wert 17890.3. Man sieht die Energieerhaltung einigermaßen gut bestätigt.

Aus dem in Abbildung 1.66 dargestellten Bild, der einstufigen zweidimensionalen Haartransformation des Bildes aus Abbildung 1.61, kann man durch Rücktransformation das Original wiederherstellen. Die Differenz zwischen Original und Rekonstruktion zeigt die Abbildung 1.68. Die Intensitäten der Differenz liegen im Bereich von 2^{-14} , sind also vernachlässigbar.

An zwei Modifikationen soll nun noch der Nutzen der Haar-Transformation erläutert werden.

Zunächst wird in der Haartransformierten von Abbildung bzw. Abbildung mit den adjustierten Intensitäten, das rechte untere Teilbild durch ein Nullbild ersetzt. Sodann wird die Rekonstruktionsabbildung durchgeführt. Ausgangspunkt der Rekonstruktion sind also das (adjustierten) Daten wie in Abbildung 1.69.

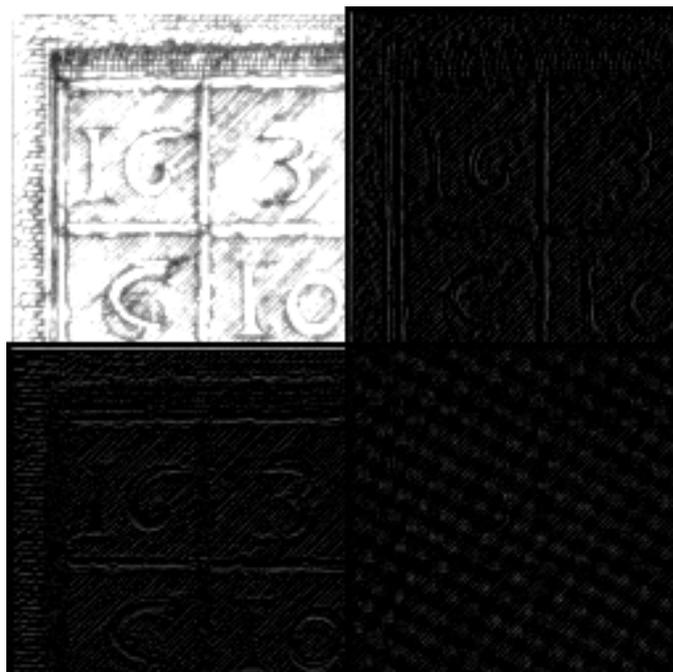


Abbildung 1.66: Einstufige Haar-Transformation des Bildes aus Abbildung 1.61.

Vergleicht man nun die beiden Rekonstruktionen aus Abb. 1.66 und Abb. 1.69, so erhält man in Abbildung 1.70 in dem rechten Teil ein Bild, das im Vergleich zur perfekten Rekonstruktion des Originals (links) einen ganz leicht “verwaschenen” Eindruck macht, was mit dem Verlust (Null-Setzen) hochfrequenter Bildinformation zu tun hat. Die Differenz ist sehr gering, in Abbildung 1.71 ist sie in adjustierter Intensität dargestellt. Die Energie der Differenz ist 9.11389.

In einer weiteren Rekonstruktion wird nun der AA-Tiefpassanteil des transformierten Bildes durch ein Nullbild ersetzt – siehe Abbildung 1.72. Die Rekonstruktion beruht nun nur auf den hochfrequenten Bildanteilen. Das Ergebnis wird nun von dem Original deutlich verschieden sein (die allermeisten Energieanteile wurden ja “unterschlagen”). Die Rekonstruktion in Abbildung 1.73 muss man sich mit adjustierter Intensität anschauen. Man erkennt, dass hier die (hochfrequenten) Feinstrukturen des Originals deutlich hervortreten. Dies ist ein Hinweis darauf, dass und wie man Wavelettransformationen zur Erkennung von Bildstrukturen verwenden kann.

Man kann die zweidimensionale Haar-Transformation ebenfalls mehrstufig durchführen, und zwar analog dazu, wie das im eindimensionalen Fall gemacht wurde: nur der Approximationsanteil, also das AA-Teilbild nach der ersten Transformationsphase, wird einer weiteren Haar-Transformation unterzogen. Die AD-, DA- und DD-Teilbilder bleiben unverändert. Das Ergebnis (mit globaler Adjustierung der Intensitäten) ist in Abbildung 1.74 dargestellt. Die vier Teilbilder, die aus dem AA-Teilbild der ersten Transformationsphase in der zweiten Transformationsphase generiert werden, sind (mit separat adjustierten Intensitäten) noch einmal separat in Abbildung 1.75 dargestellt. Der Vergleich mit Abbildung 1.67 ist aufschlussreich: man erkennt eine Vergrößerung des Resultats der ersten Transformationsphase. Das weist auf ein nützliches Prinzip hin:

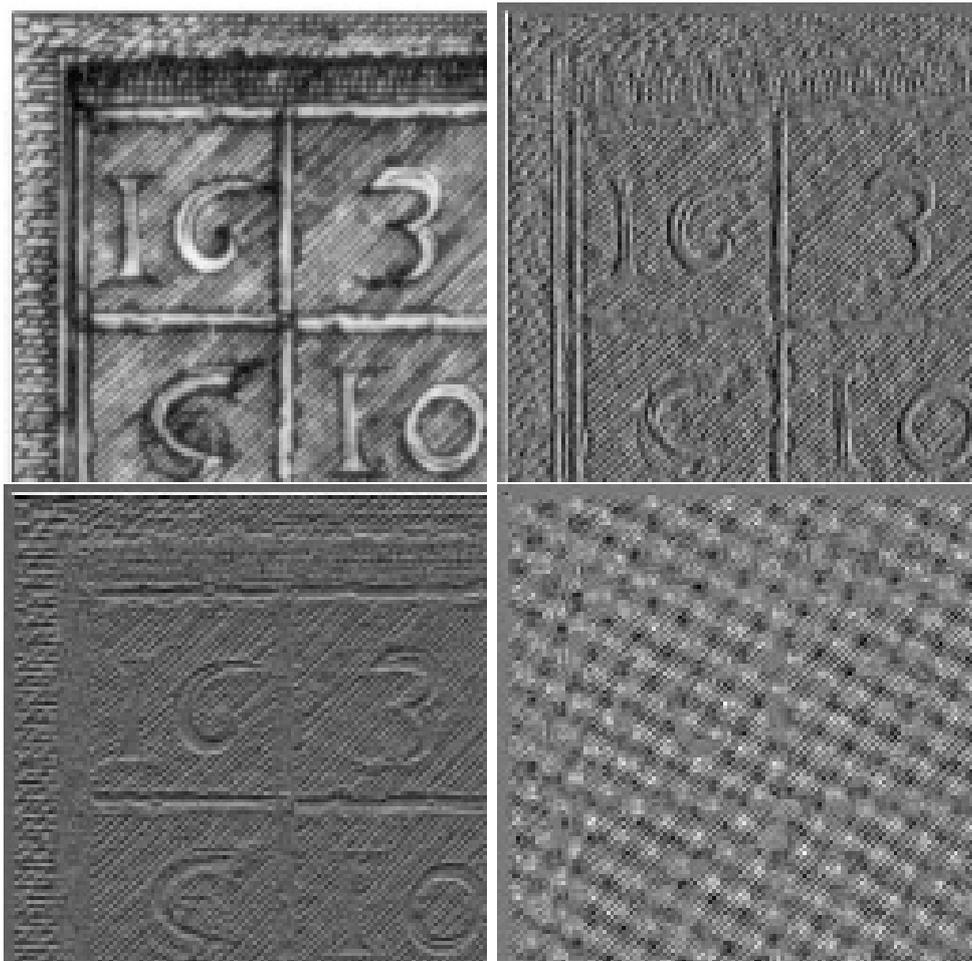


Abbildung 1.67: Einstufige Haar-Transformation des Bildes aus Abbildung 1.61 mit separat adjustierten Intensitäten für die vier Teilbilder.

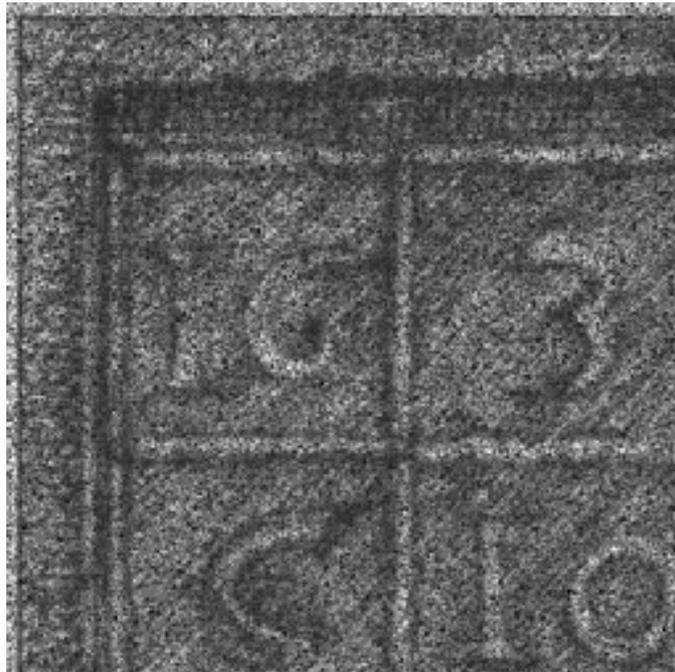


Abbildung 1.68: Einstufige Haar-Transformation des Bildes aus Abbildung 1.61, Differenz zwischen Original und Rekonstruktion mit angepasster Intensität.

Strukturen, die bei Wavelettransformationen über mehrere Auflösungsstufen in entsprechender Position erkennbar sind, können nicht durch zufällige Schwankungen entstanden sein, sondern müssen von markanten Strukturen des Originalbildes herrühren.

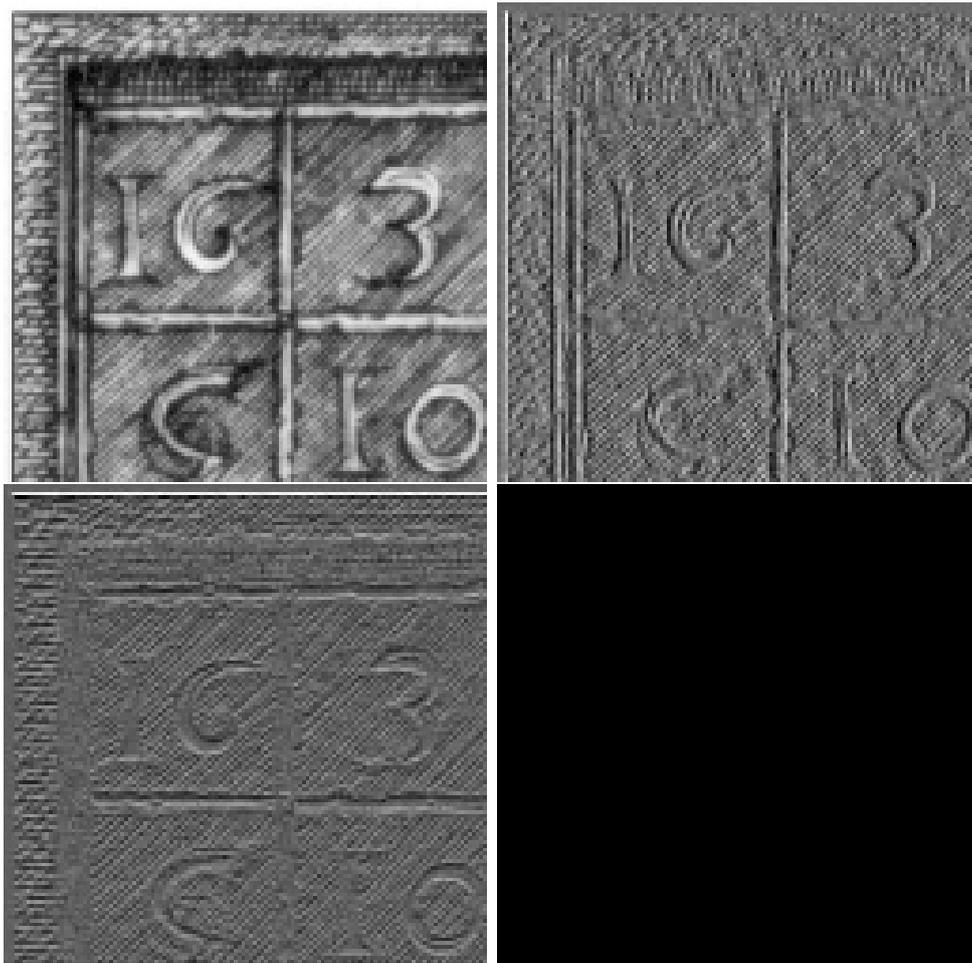


Abbildung 1.69: Einstufige Haar-Transformation des Bildes aus Abbildung 1.61 mit separat adjustierten Intensitäten für die vier Teilbilder, DD-Anteil durch Nullbild ersetzt.

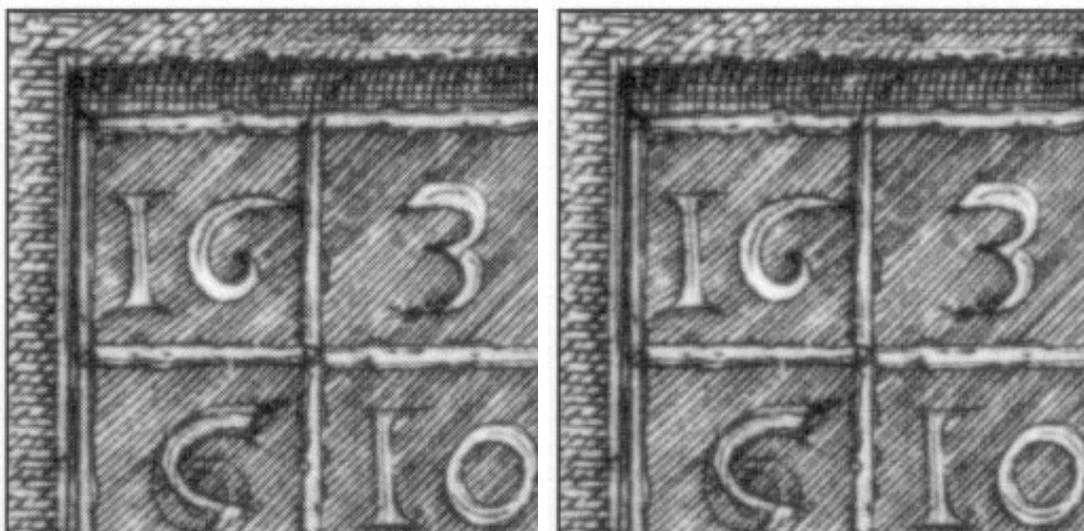


Abbildung 1.70: Einstufige Haar-Transformation des Bildes aus Abbildung 1.61, Rekonstruktion ohne und mit Modifikation des DD-Teilbildes.

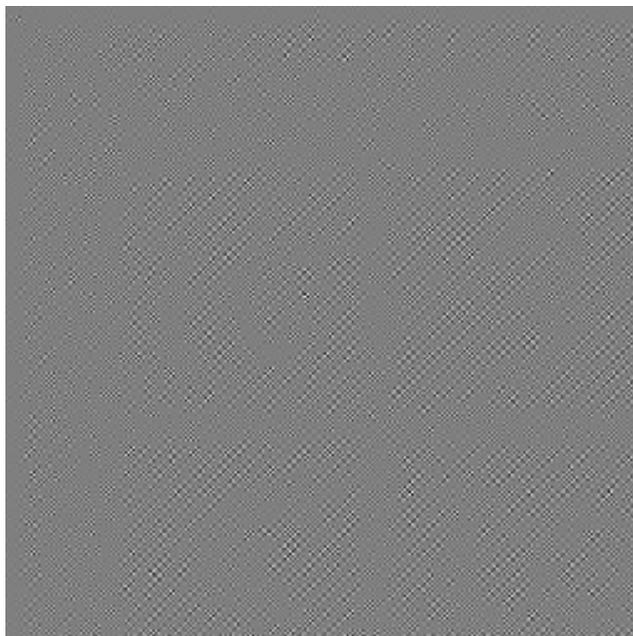


Abbildung 1.71: Differenz (mit adjustierter Intensität) der Rekonstruktionen ohne und mit Modifikation des DD-Teilbildes.

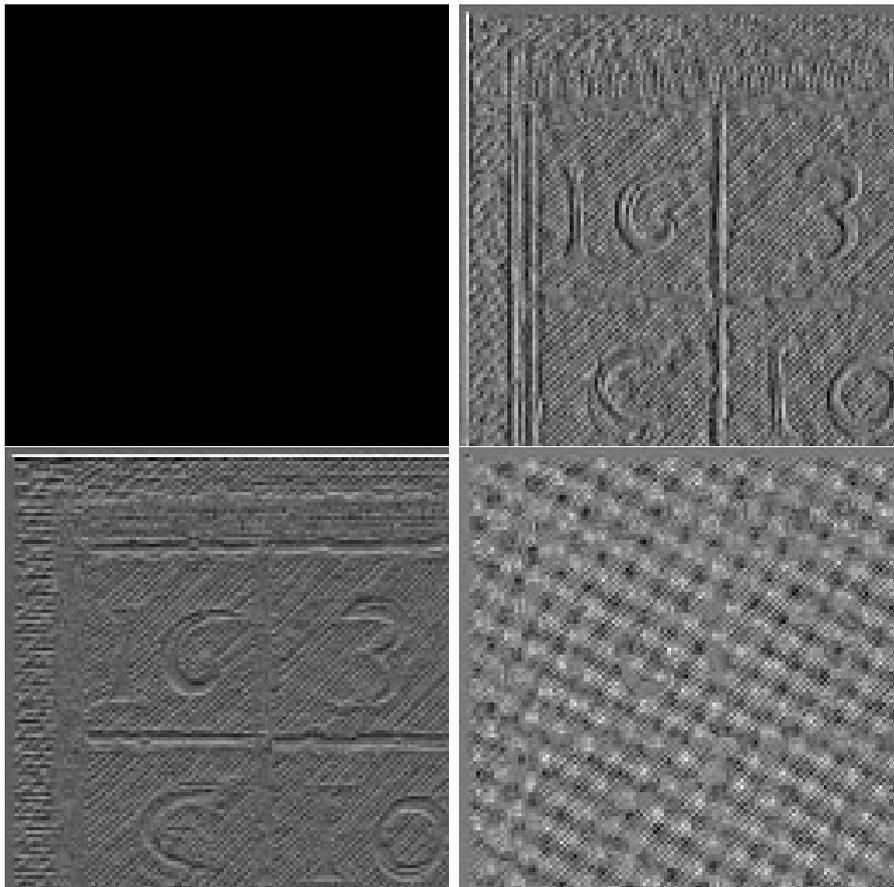


Abbildung 1.72: Einstufige Haar-Transformation des Bildes aus Abbildung 1.61 mit separat adjustierten Intensitäten für die vier Teilbilder, AA-Anteil durch Nullbild ersetzt.

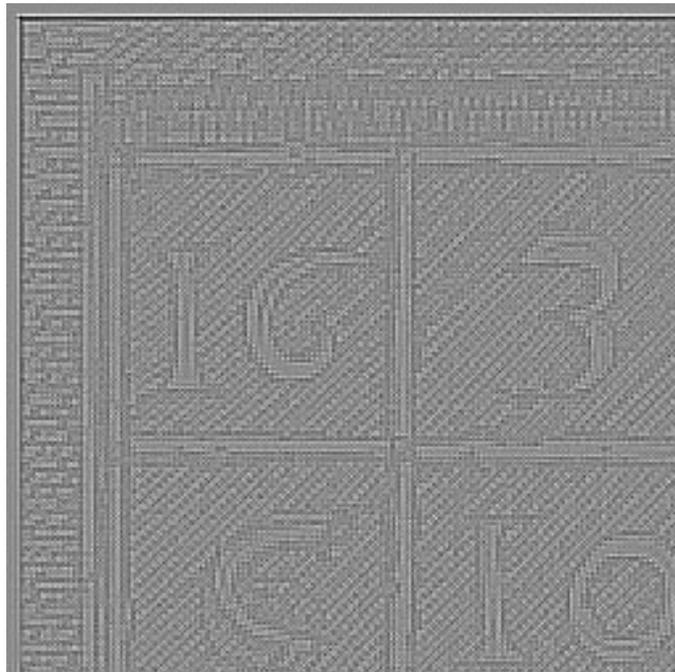


Abbildung 1.73: Rekonstruktion (mit adjustierter Intensität) nach Modifikation des AA-Teilbildes, siehe Abbildung 1.72.



Abbildung 1.74: Zweistufige Haar-Transformation (mit global adjustierten Intensitäten des Bildes aus Abbildung 1.61).

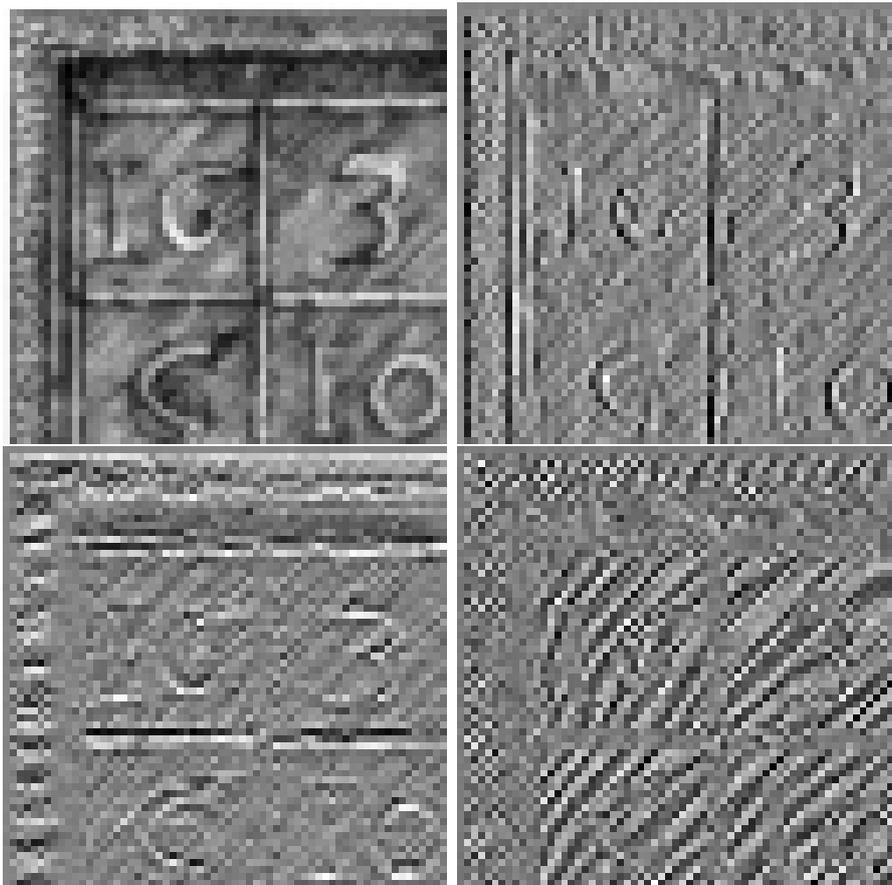


Abbildung 1.75: Einstufige Haar-Transformation des AA-Teilbildes aus Abbildung 1.66 mit separat adjustierten Intensitäten.

1.8 Datenkompression mittels Haar-Transformation

Wavelettransformationen eignen sich sehr gut zu Zwecken der Datenkompression. Das soll an einem Beispiel demonstriert werden. Dazu betrachten wir das magische Quadrat im Format 552×576 . Die Intensitätswerte dieses Bildes liegen zwischen 0.00784314 und 1.0.

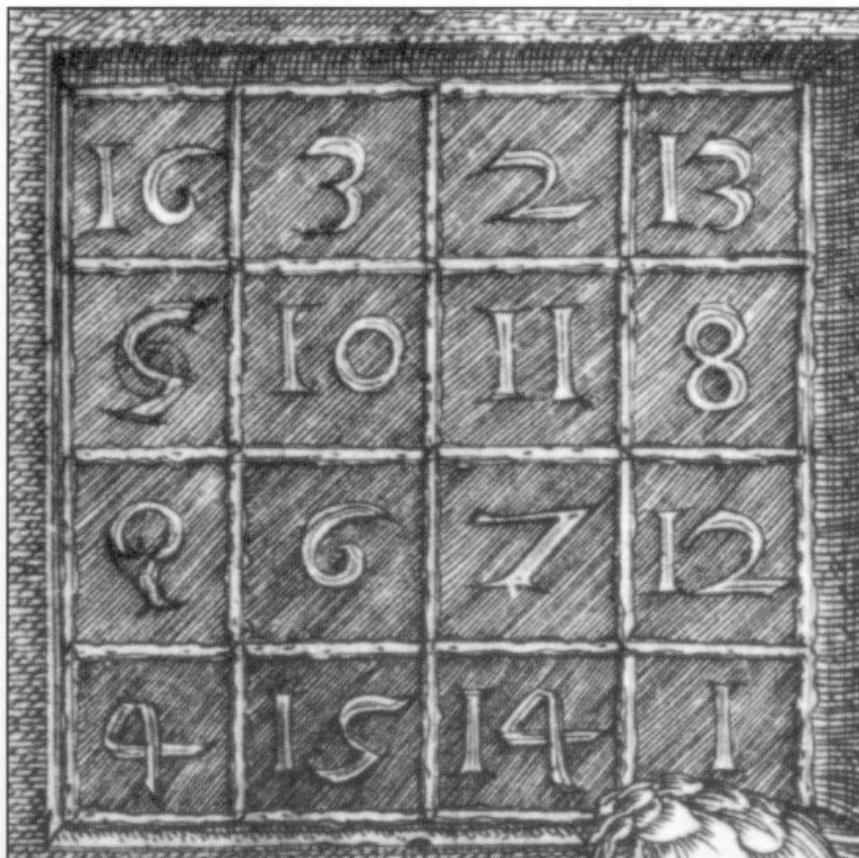


Abbildung 1.76: Magisches Quadrat (552×552)

Auf diesem Bild wird eine dreistufige Haar-Transformation durchgeführt. Das Resultat ist ein 552×552 -Bild, das aus 10 Teilbildern besteht, die gemäß ihrer "Abstammung" und nummeriert sind (Abbildung 1.77). Eine Darstellung als Abstammungsbaum gibt Abbildung 1.78.

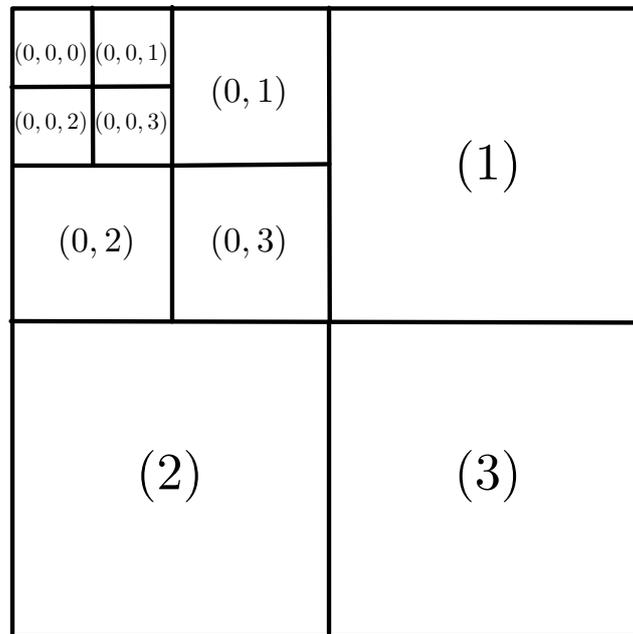


Abbildung 1.77: Zerlegungsschema nach dreistufiger Transformation

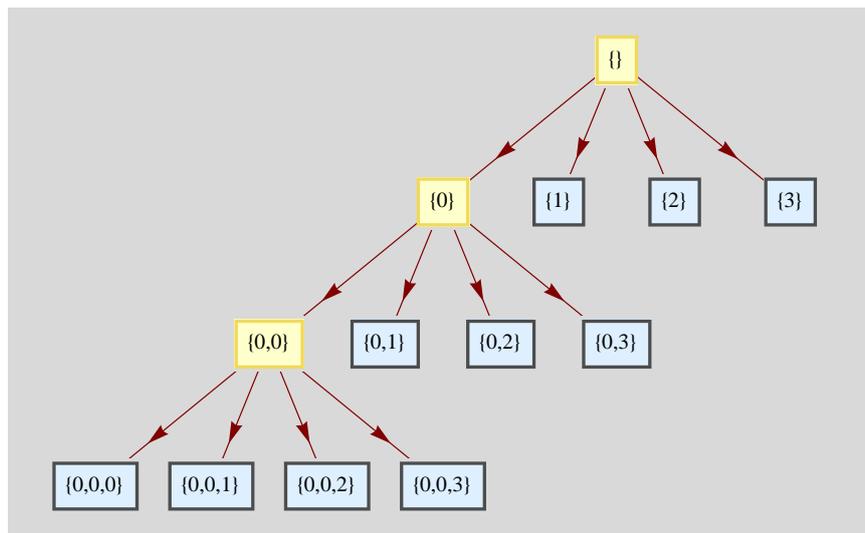


Abbildung 1.78: Baumdarstellung des Zerlegungsschemas nach dreistufiger Transformation

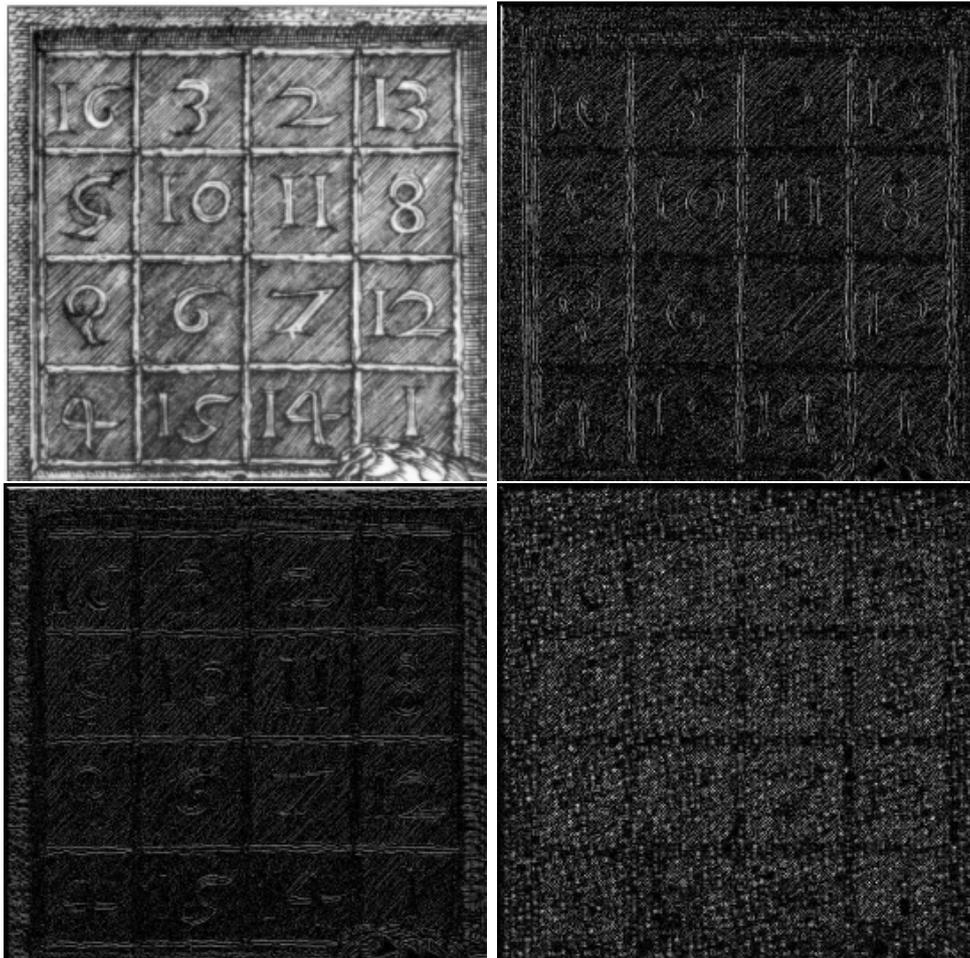


Abbildung 1.79: Zerlegung nach der ersten Stufe der dreistufigen Transformation, Teilbilder (0), (1), (2), (3) mit jeweils 276×276 Pixeln.

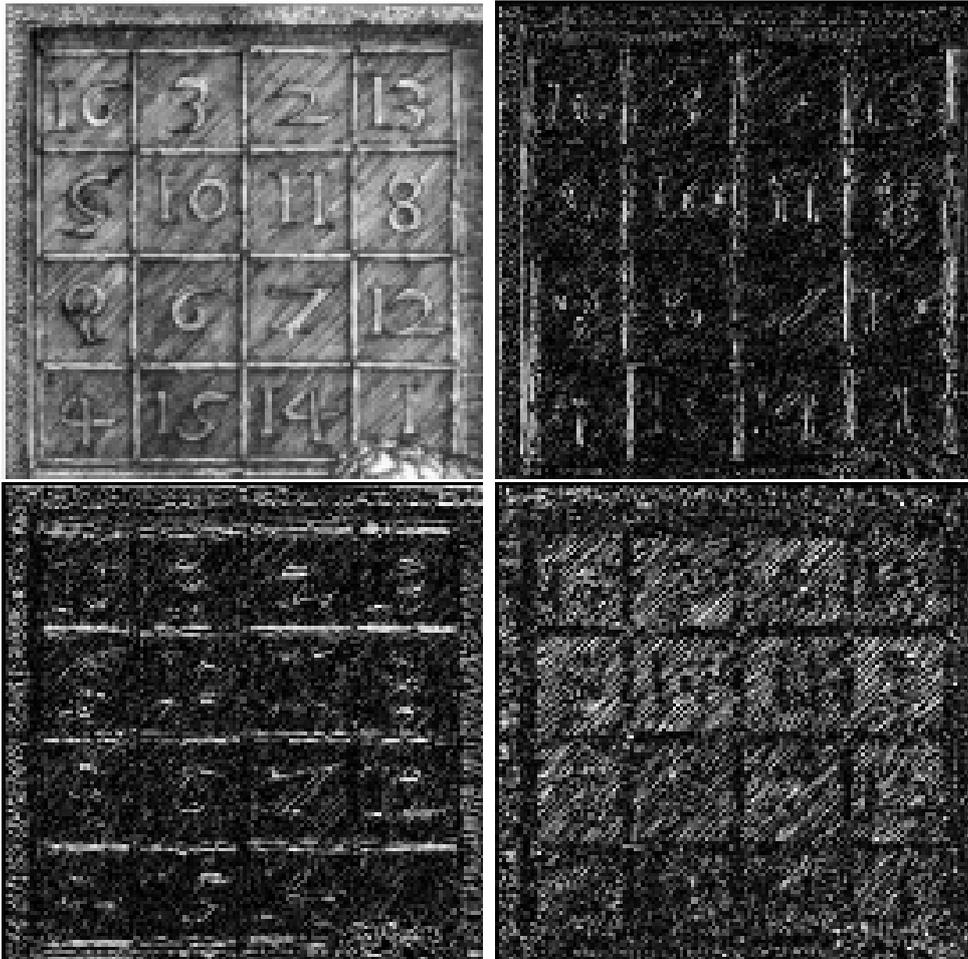


Abbildung 1.80: Zerlegung von Teilbild (0) nach der ersten Stufe der dreistufigen Transformation, Teilbilder $(0, 0)$, $(0, 1)$, $(0, 2)$, $(0, 3)$ mit jeweils 138×138 Pixeln.

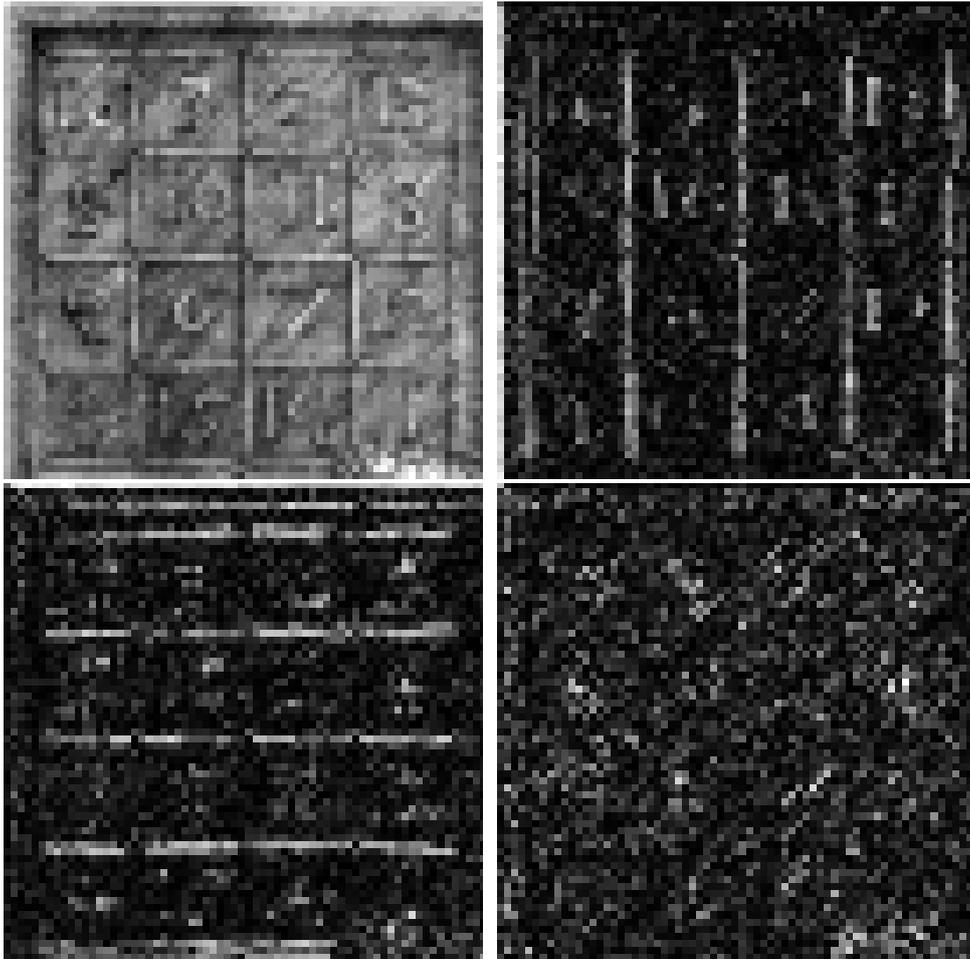


Abbildung 1.81: Zerlegung von Teilbild $(0,0)$ nach der dritten Stufe der dreistufigen Transformation, Teilbilder $(0,0,0)$, $(0,0,1)$, $(0,0,2)$, $(0,0,3)$ mit jeweils 69×69 Pixeln.

Die Energieverteilung auf diese Teilbilder ist aufschlussreich:

$$\begin{aligned}
 \{1\} &\rightarrow 0.00422301 \\
 \{2\} &\rightarrow 0.00540121 \\
 \{3\} &\rightarrow 0.00245289 \\
 \{0, 1\} &\rightarrow 0.00771271 \\
 \{0, 2\} &\rightarrow 0.00730358 \\
 \{0, 3\} &\rightarrow 0.00196534 \\
 \{0, 0, 1\} &\rightarrow 0.0124388 \\
 \{0, 0, 2\} &\rightarrow 0.0127398 \\
 \{0, 0, 3\} &\rightarrow 0.00206025 \\
 \{0, 0, 0\} &\rightarrow 0.943702
 \end{aligned}$$

Die Verteilung der Anzahl von Koeffizienten $\neq 0$ auf die Teilbilder ist

$$\begin{aligned}
 \{1\} &\rightarrow 75380 \\
 \{2\} &\rightarrow 75673 \\
 \{3\} &\rightarrow 75468 \\
 \{0, 1\} &\rightarrow 19000 \\
 \{0, 2\} &\rightarrow 19020 \\
 \{0, 3\} &\rightarrow 19004 \\
 \{0, 0, 1\} &\rightarrow 4754 \\
 \{0, 0, 2\} &\rightarrow 4761 \\
 \{0, 0, 3\} &\rightarrow 4761 \\
 \{0, 0, 0\} &\rightarrow 4761
 \end{aligned}$$

Gesamtzahl von Koeffizienten $\neq 0$: 302582 bei einer Gesamtzahl von $552^2 = 304704$ Pixeln.

1. Rekonstruktion nach Schwellwertbildung mit 0.15 in den Detailbildern

Jetzt werden die durch dreistufige Haar-Transformation entstandenen Teilbilder modifiziert: in den Detailbildern (also allen Teilbildern, ausgenommen das Approximationsbild $(0, 0, 0)$), werden alle Intensitätswerte von Pixeln, die absolut kleiner als $\lambda = 0.15$ sind, werden durch Wert 0 ersetzt. Dies bezeichnet man als *hard thresholding* mit dem Schwellwert λ .

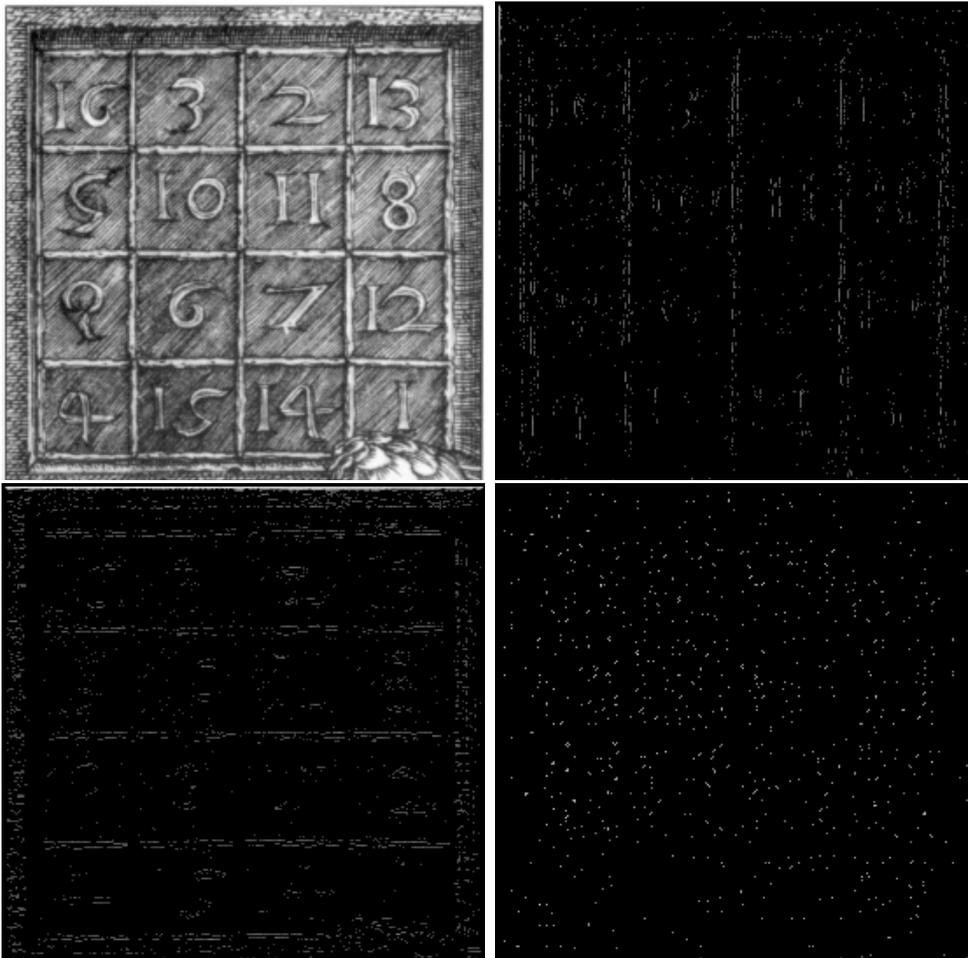


Abbildung 1.82: Schwellwert-modifizierte Teilbilder der ersten Stufe der dreistufigen Transformation, Teilbilder (0), (1), (2), (3) mit jeweils 276×276 Pixeln.

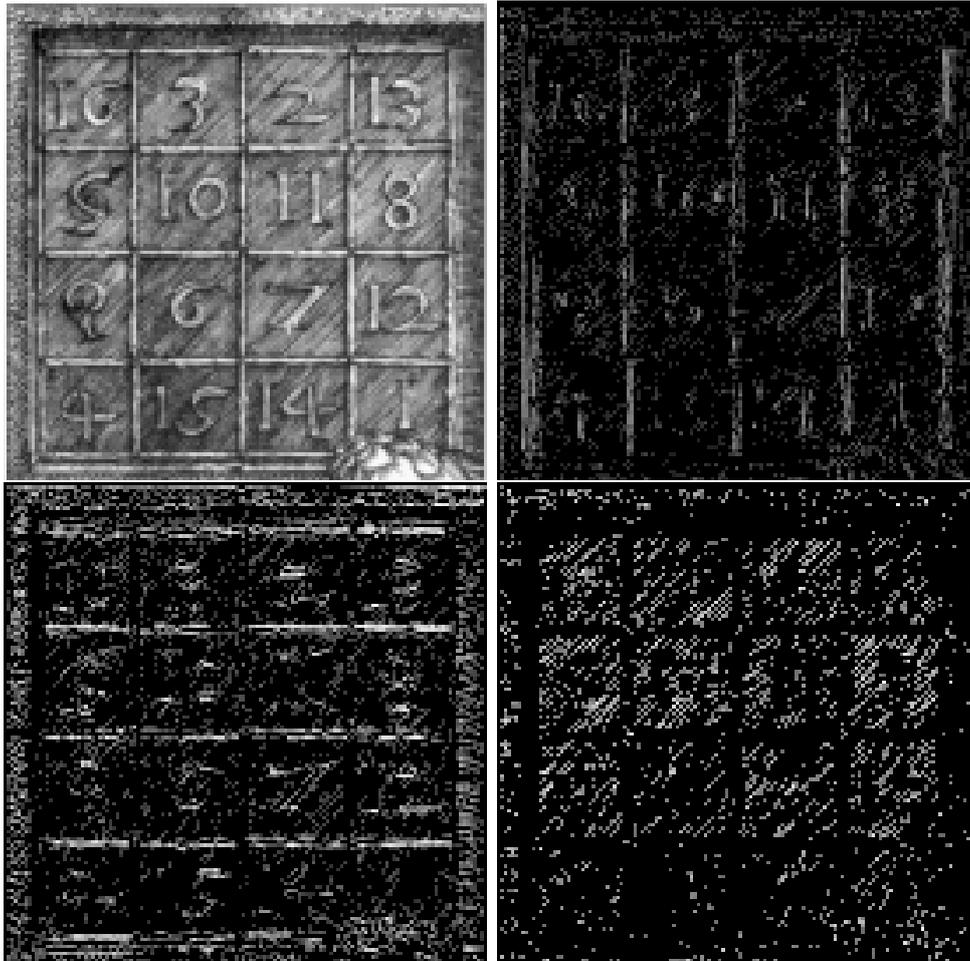


Abbildung 1.83: Schwellwert-modifizierte Teilbilder von Teilbild (0) nach der ersten Stufe der dreistufigen Transformation, Teilbilder $(0, 0)$, $(0, 1)$, $(0, 2)$, $(0, 3)$ mit jeweils 138×138 Pixeln.

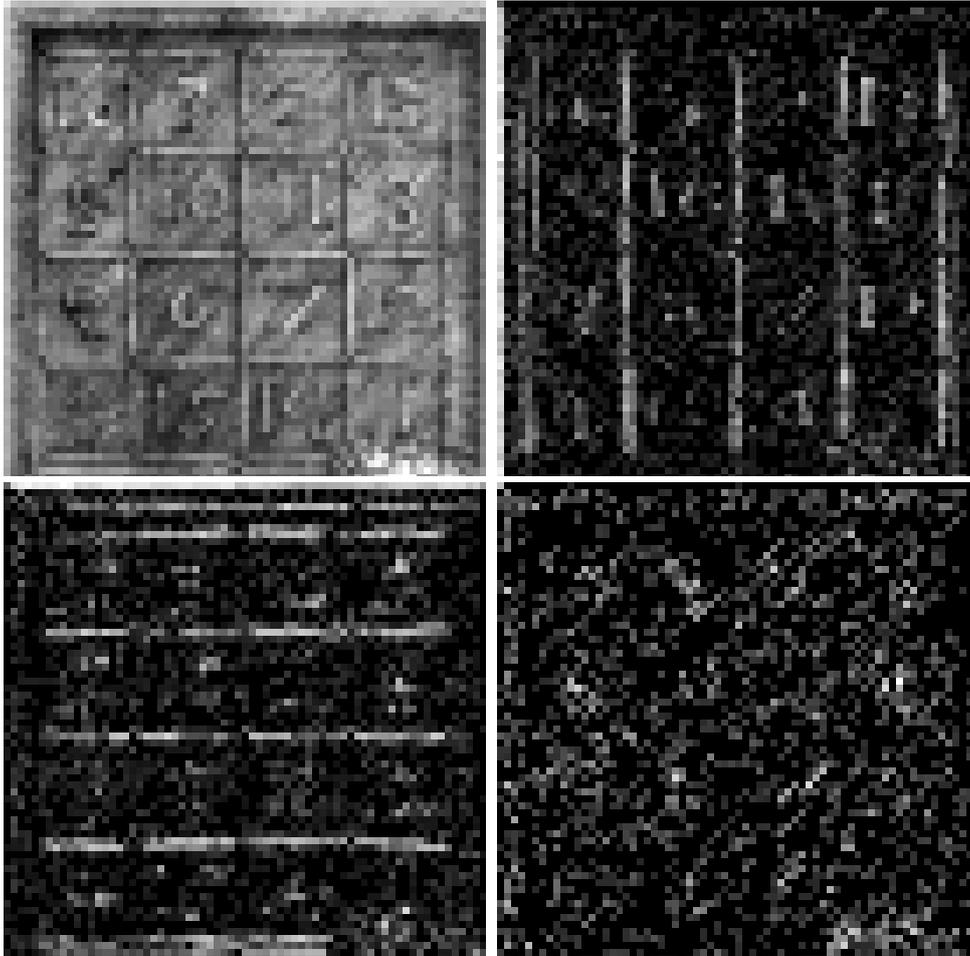


Abbildung 1.84: Schwellwert-modifizierte Teilbilder von Teilbild (0,0) nach der dritten Stufe der dreistufigen Transformation, Teilbilder (0,0,0), (0,0,1), (0,0,2), (0,0,3) mit jeweils 69×69 Pixeln.

Die Energieverteilung auf die Schwellwert- modifizierten Teilbilder ist nun:

$$\begin{aligned}
 \{1\} &\rightarrow 0.00130732 \\
 \{2\} &\rightarrow 0.00245536 \\
 \{3\} &\rightarrow 0.000268546 \\
 \{0, 1\} &\rightarrow 0.00693957 \\
 \{0, 2\} &\rightarrow 0.00655401 \\
 \{0, 3\} &\rightarrow 0.00108347 \\
 \{0, 0, 1\} &\rightarrow 0.0124334 \\
 \{0, 0, 2\} &\rightarrow 0.0127357 \\
 \{0, 0, 3\} &\rightarrow 0.00187427 \\
 \{0, 0, 0\} &\rightarrow 0.954348
 \end{aligned}$$

Die Verteilung der Anzahl von Koeffizienten $\neq 0$ auf die Teilbilder ist:

$$\begin{aligned}
 \{1\} &\rightarrow 2734 \\
 \{2\} &\rightarrow 4200 \\
 \{3\} &\rightarrow 809 \\
 \{0, 1\} &\rightarrow 5601 \\
 \{0, 2\} &\rightarrow 6027 \\
 \{0, 3\} &\rightarrow 2510 \\
 \{0, 0, 1\} &\rightarrow 2866 \\
 \{0, 0, 2\} &\rightarrow 2827 \\
 \{0, 0, 3\} &\rightarrow 1811 \\
 \{0, 0, 0\} &\rightarrow 4761
 \end{aligned}$$

Die Anzahl der Koeffizienten $\neq 0$ ist 34146. Bei einer Pixelanzahl von 304704 sind das nur noch 11.2849% der Pixel!

Weitere Rekonstruktion auf der Basis von Schwellwert-modifizierten Transformationsbildern folgen.

2. **Rekonstruktion nach Schwellwertbildung mit $\lambda = 0.1$**

Hier ist die Anzahl der Koeffizienten $\neq 0$ in den Schwellwert-modifizierten Bildern = 65992. Das sind 21.7865% der ursprünglichen Anzahl.

3. **Rekonstruktion nach Schwellwertbildung mit $\lambda = 0.2$**

Hier ist die Anzahl der Koeffizienten $\neq 0$ in den Schwellwert-modifizierten Bildern = 21315. Das sind 7.04437% der ursprünglichen Anzahl.

4. **Rekonstruktion nach Schwellwertbildung mit $\lambda = 0.3$**

Hier ist die Anzahl der Koeffizienten $\neq 0$ in den Schwellwert-modifizierten Bildern = 12943 Das sind 4.27752% der ursprünglichen Anzahl.

5. **Rekonstruktion nach Schwellwertbildung mit $\lambda = 0.40$**

Hier ist die Anzahl der Koeffizienten $\neq 0$ in den Schwellwert-modifizierten Bildern = 9777. Das sind 3.23119% der ursprünglichen Anzahl.

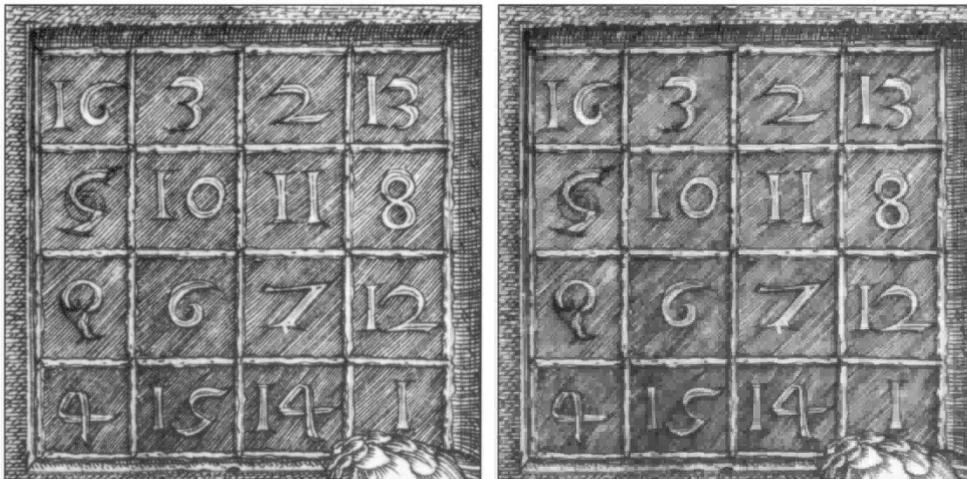


Abbildung 1.85: Vergleich der Originals (links) mit der Rekonstruktion auf der Basis von 11.3% der originalen Pixelanzahl.

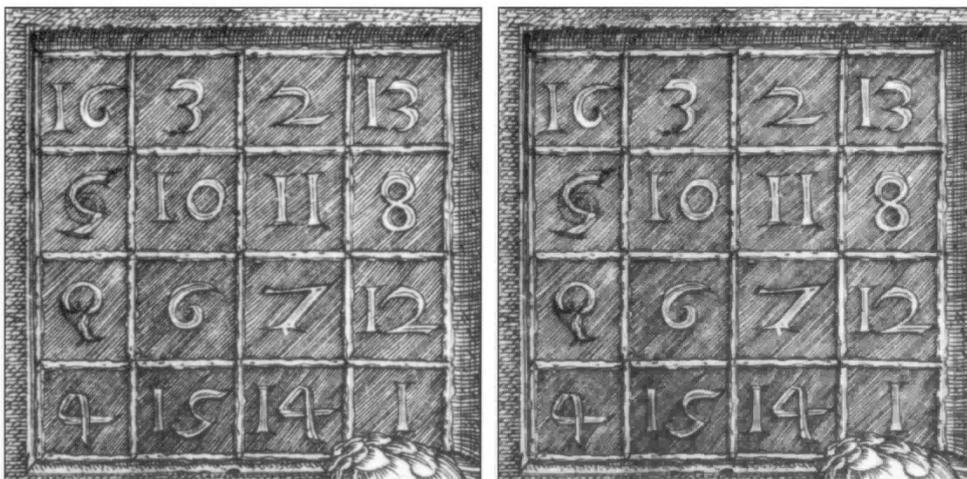


Abbildung 1.86: Vergleich der Originals (links) mit der Rekonstruktion auf der Basis von 21.8% der originalen Pixelanzahl.

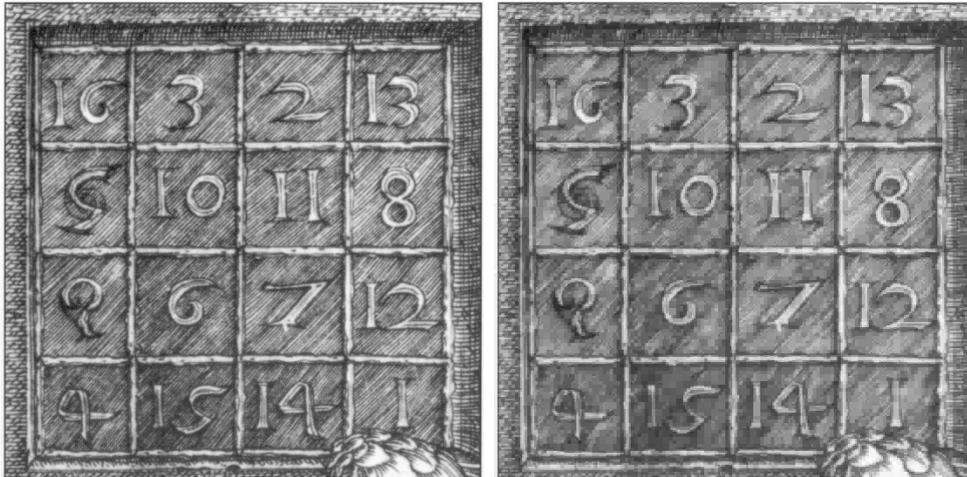


Abbildung 1.87: Vergleich der Originals (links) mit der Rekonstruktion auf der Basis von 7% der originalen Pixelanzahl.

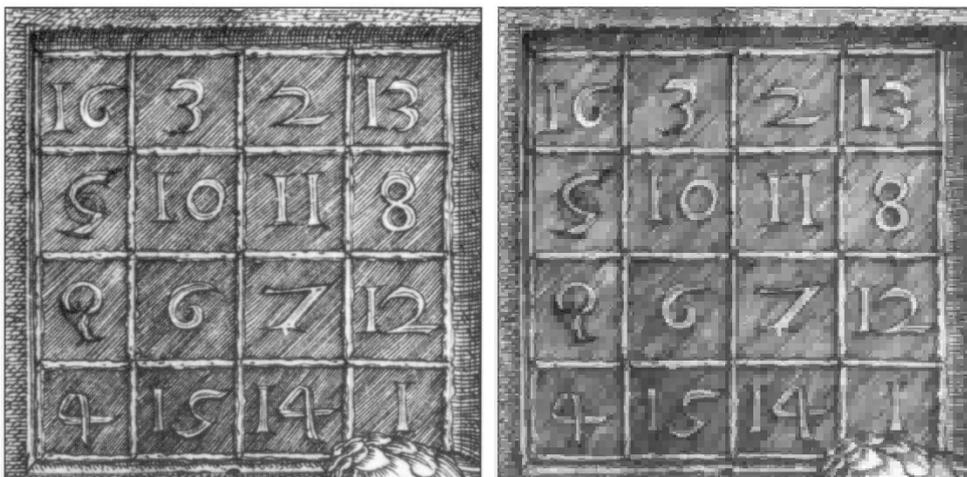


Abbildung 1.88: Vergleich der Originals (links) mit der Rekonstruktion auf der Basis von 4.3% der originalen Pixelanzahl.

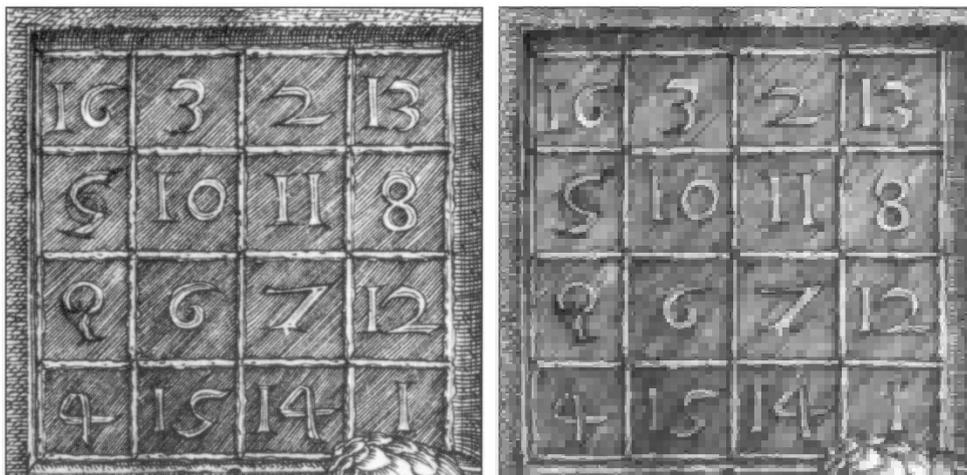


Abbildung 1.89: Vergleich der Originals (links) mit der Rekonstruktion auf der Basis von 3.2% der originalen Pixelanzahl.

1.9 Entrauschen mit Wavelets

Die Eigenschaft von Wavelets, Signale nach Frequenzen zu analysieren (und dies auch im Ortsbereich lokalisiert, anders als die Fouriertransformation) kann man dazu verwenden, verrauschte Signale von den unerwünschten Frequenzanteilen zu "säubern". Rauschen als Störung bringt hochfrequente Signalanteile, die zudem in der Regel im Vergleich zum eigentlichen Signal weniger energiereich sind. Natürlich darf man nicht einfach hochfrequente Anteile blind eliminieren, denn wesentliche Bildinformation wie Kanten sind auch hochfrequent und sollen ja tunlichst erhalten bleiben. Man muss also zu einer Hypothese kommen, wie hoch (energetisch gesehen) die Rauschanteile an einem Signal sind, d.h. man muss aus dem Signal (sofern man keine anderweitige Information über das Signal und den Kanal, in dem es übertragen wurde, hat) den Rauschlevel schätzen. Das wird in der Vorlesung an späterer Stelle behandelt. Hier soll nur das Prinzip des Einsatzes von Wavelets für die Entrauschung illustriert werden.

Abbildung 1.90 zeigt ein verrauschtes Signal, wobei die Rauschanteile in ihrer Intensität offenbar zeitlich variieren.

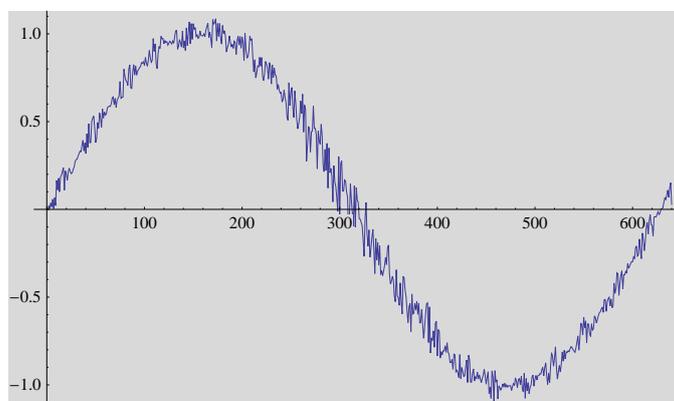


Abbildung 1.90: Verrauschtes Signal

1.9.1 Entrauschung und Rekonstruktion mit dem Haar-Wavelet

Dieses Signal wird nun einer vierstufigen eindimensionalen Haartransformation unterzogen. Abbildung 1.91 zeigt die Struktur der entstehenden Daten. Die Binärcodierung deutet mit jeder "0" eine Tiefpassfilterung und mit jeder "1" eine Hochpassfilterung an. Dieses Indizierungsschema wird in der Folge beibehalten.

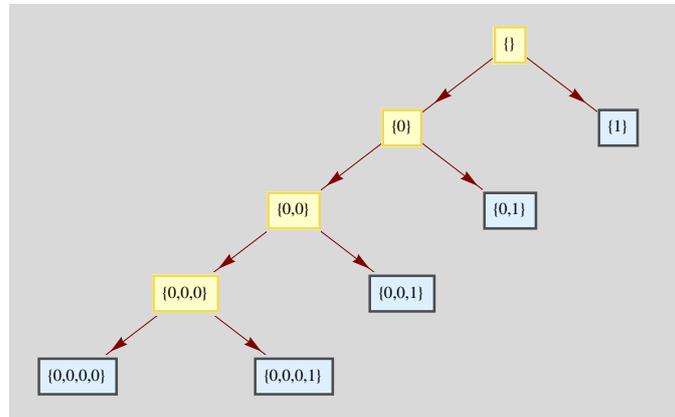


Abbildung 1.91: Schema einer 4-stufigen Wavelettransformation

Die bei der Wavelettransformation entstehenden Daten sehen im Plot so aus:

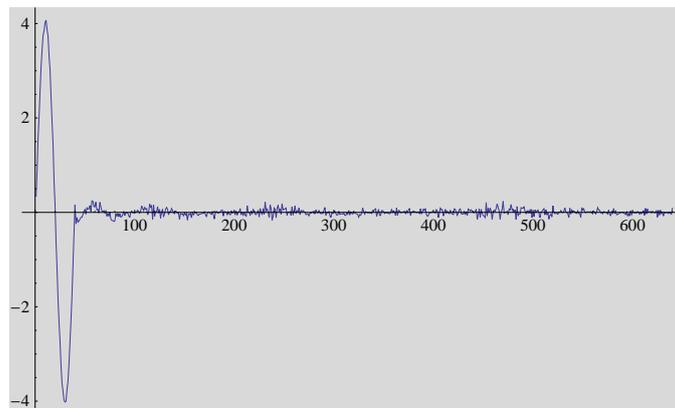


Abbildung 1.92:

Die Datenvektoren für die Bereiche $(0, 0, 0, 0)$, $(0, 0, 0, 1)$, $(0, 0, 1)$, $(0, 1)$ und (1) sind in der grafischen Darstellung von Abbildung 1.92 in dieser Reihenfolge aneinandergelagert.

Aufschlussreich ist die Energieverteilung auf die einzelnen Teile:

$$\begin{aligned} \{1\} &\rightarrow 0.00312411 \\ \{0, 1\} &\rightarrow 0.00173473 \\ \{0, 0, 1\} &\rightarrow 0.00124577 \\ \{0, 0, 0, 1\} &\rightarrow 0.00206121 \\ \{0, 0, 0, 0\} &\rightarrow 0.991834 \end{aligned}$$

Die Idee ist nun, alle Bereiche aus der Rekonstruktion mittels inverser Haar-Transformation auszuschliessen, deren Energieanteil unterhalb eines bestimmten Werte λ liegt.

1. Rekonstruktion nach vierstufiger Wavelettransformation mit dem Haar-Wavelet und Rauschunterdrückung mit $\lambda = 0.002$

Wählt man etwas $\lambda = 0.002$, so werden die Bereiche $(0, 1)$ und $(0, 0, 1)$ aus der Rekonstruktion ausgeschlossen, d.h. man hat einen Vektor mit Energieverteilung

$$\begin{aligned}\{1\} &\rightarrow 0.00313345 \\ \{0, 1\} &\rightarrow 0. \\ \{0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 1\} &\rightarrow 0.00206738 \\ \{0, 0, 0, 0\} &\rightarrow 0.994799\end{aligned}$$

Eine Rekonstruktion mittels inverser Haartransformation liefert

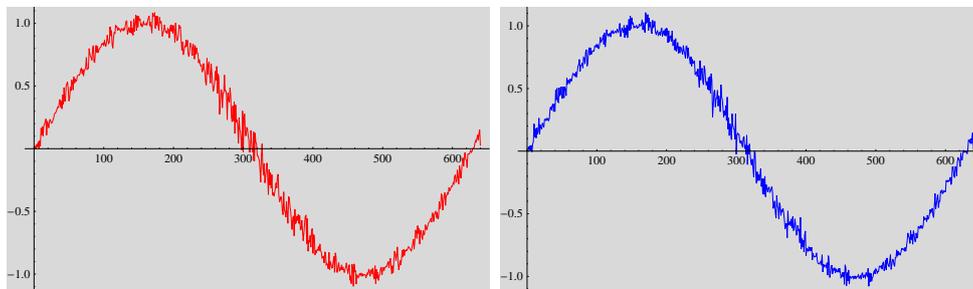


Abbildung 1.93: Vergleich von Ergebnissen der Rücktransformation der H4-Transformationsdaten ohne (links) und mit (rechts) der Elimination niedrigerenergetischer Bereiche ($\lambda = 0.002$)

2. Rekonstruktion nach vierstufiger Wavelettransformation mit dem Haar-Wavelet und Rauschunterdrückung mit $\lambda = 0.004$

Man erkennt kaum eine Verbesserung. Also wird man λ erhöhen, etwa $\lambda = 0.004$ setzen, was offensichtlich zu der Energieverteilung

$$\begin{aligned}\{1\} &\rightarrow 0. \\ \{0, 1\} &\rightarrow 0. \\ \{0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 0\} &\rightarrow 1.\end{aligned}$$

führt. Die Rekonstruktion ist ernüchternd: die Elimination sämtlicher hochfrequenten Information führt zu einer blockigen Stufenfunktion – das ist der Preis für die Verwendung des Haar-Wavelets.

1.9.2 Entrauschung und Rekonstruktion mit dem Daubechies-12-Wavelet

Die enttäuschenden Beobachtungen bei der Anwendung der Haar-Wavelettransformation im vorigen Abschnitt sollten Motivation genug sein, nach Alternativen Ausschau

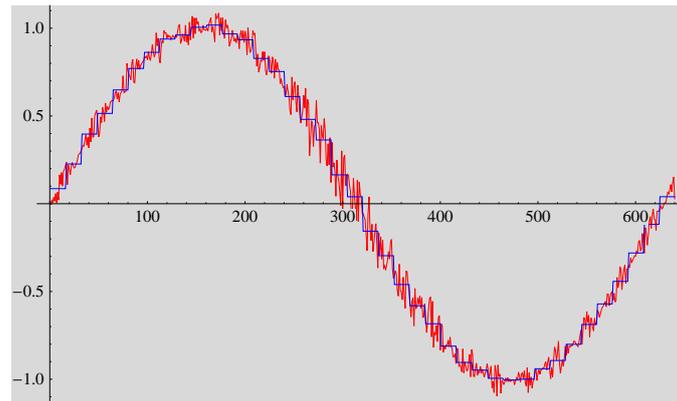


Abbildung 1.94: Vergleich von “wahrer” Rekonstruktion (rot) und Rekonstruktion nach Unterdrückung hochfrequenter Anteile mit $\lambda = 0.004$ (blau).

zu halten, die nach ähnlichen Prinzipien vorgehen, aber bessere Resultate liefern. Tatsächlich ist das Haar-Wavelet nur das einfachste einer inzwischen sehr umfangreichen Familie von Wavelets. Allerdings: wie man solche besseren Wavelets konstruiert, ist eine keineswegs offensichtliche Angelegenheit. Als Vertreter einer interessanten und viel verwendeten Klasse von Wavelets wird hier – ohne nähere Erläuterungen, die kommen in einem späteren Kapitel – die Entrauschung mit Hilfe des sogenannten Daubechies-12-Wavelets gezeigt. Dabei sollte ins Auge springen, dass man viel bessere Chancen als bei Verwendung des Haar-Wavelets hat, bei Entrauschung von ursprünglich “glatten” Signalen wieder glatte Signale und nicht mit Unstetigkeiten gespickte Treppenfunktionen zu erhalten.

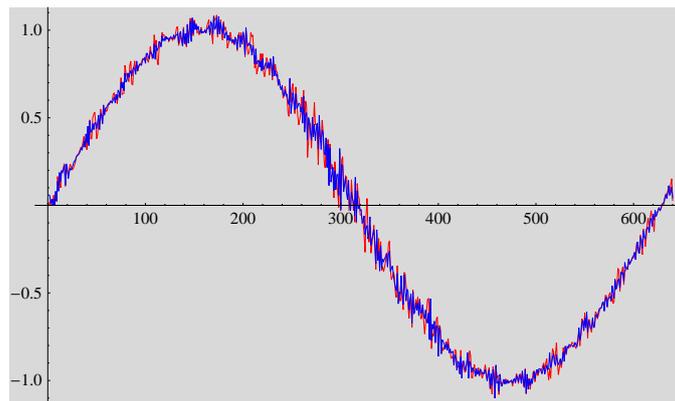
1. Energieverteilung nach vierstufiger Wavelettransformation mit dem Daubechies-12-Wavelet

$$\begin{aligned} \{1\} &\rightarrow 0.00245876 \\ \{0, 1\} &\rightarrow 0.00135729 \\ \{0, 0, 1\} &\rightarrow 0.00112498 \\ \{0, 0, 0, 1\} &\rightarrow 0.00275573 \\ \{0, 0, 0, 0\} &\rightarrow 0.992303 \end{aligned}$$

2. Rekonstruktion nach vierstufiger Wavelettransformation mit dem Daubechies-12-Wavelet und Rauschunterdrückung mit $\lambda = 0.003$.

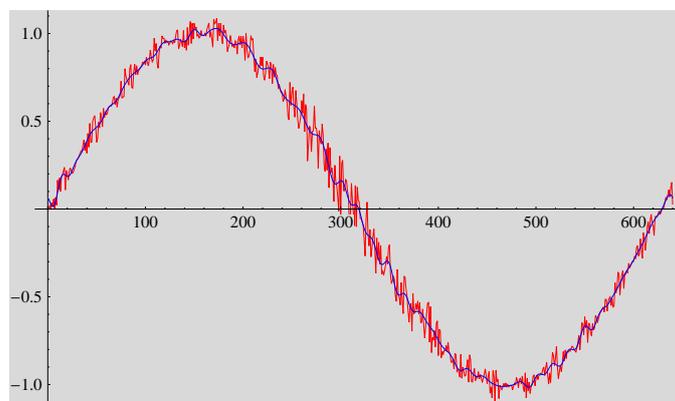
$$\begin{aligned} \{1\} &\rightarrow 0.00246488 \\ \{0, 1\} &\rightarrow 0. \\ \{0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 1\} &\rightarrow 0.00276259 \\ \{0, 0, 0, 0\} &\rightarrow 0.994773 \end{aligned}$$

3. Rekonstruktion nach vierstufiger Wavelettransformation mit dem Daubechies-

Abbildung 1.95: Vierstufige D12-Transformation und $\lambda = 0.002$

12-Wavelet und Rauschunterdrückung mit $\lambda = 0.0025$.

$$\begin{aligned} \{1\} &\rightarrow 0. \\ \{0, 1\} &\rightarrow 0. \\ \{0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 1\} &\rightarrow 0.00276941 \\ \{0, 0, 0, 0\} &\rightarrow 0.997231 \end{aligned}$$

Abbildung 1.96: Vierstufige D12-Transformation und $\lambda = 0.0025$

4. Rekonstruktion nach vierstufiger Wavelettransformation mit dem Daubechies-12-Wavelet und Rauschunterdrückung mit $\lambda = 0.003$

$$\begin{aligned} \{1\} &\rightarrow 0. \\ \{0, 1\} &\rightarrow 0. \\ \{0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 1\} &\rightarrow 0. \\ \{0, 0, 0, 0\} &\rightarrow 1. \end{aligned}$$

5. Rekonstruktion nach sechsstufiger Wavelettransformation mit dem Daubechies-12-Wavelet und Rauschunterdrückung mit $\lambda = 0.003$

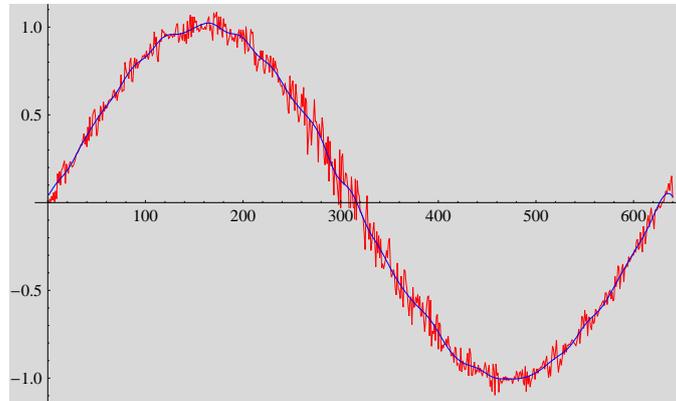


Abbildung 1.97: Vierstufige D12-Transformation und $\lambda = 0.003$

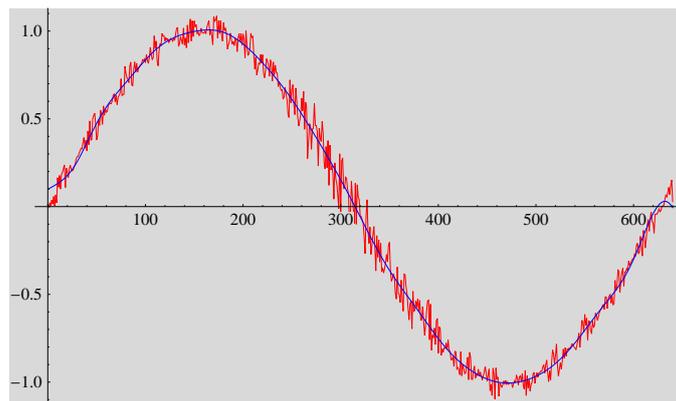


Abbildung 1.98: Sechsstufige D12-Transformation und $\lambda = 0.01$

1.10 (Naive) Kantenerkennung

In den diese Einführung abschliessenden Beispielen soll noch einmal auf die schon weiter vorne kurz angesprochene Verwendung von Wavelets zum Erkennen von Bildstrukturen, wie zB. Kanten, eingegangen werden. Das Vorgehen ist reichlich naiv und soll nur eine Idee vermitteln. Später wird Kantenerkennung mit raffinierteren Wavelet-basierten Algorithmen noch behandelt.

Ein eindimensionales Beispiel

Abbildung 1.99 zeigt den Graphen einer einfachen Funktion, die an den Stellen $x_1 = -1$, $x_2 = -0.2$, $x_3 = 0$, $x_4 = 0.6$ senkrechte Kanten (Unstetigkeiten) aufweist.

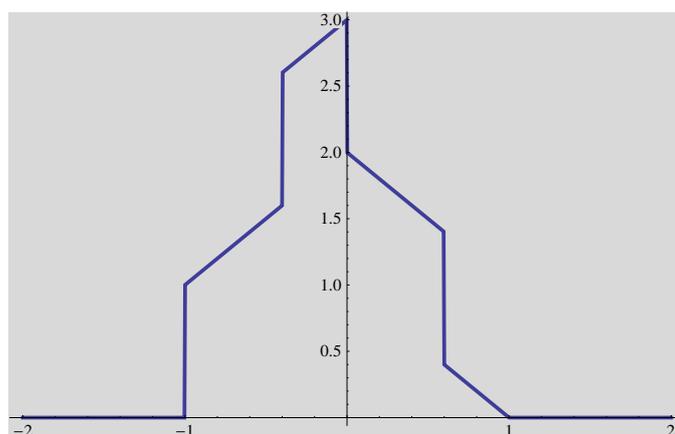


Abbildung 1.99: Eine Funktion mit Unstetigkeiten.

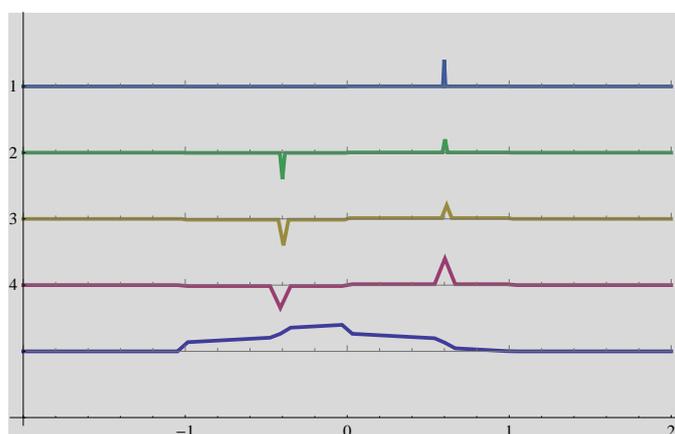


Abbildung 1.100: Vierstufige Haartransformation der Funktion aus Abbildung 1.99 mit auf gemeinsame x -Achse skalierten Detaildaten.

In Abbildung wurden die Detaildaten der Bereiche $1 = (1)$, $2 = (0, 1)$, $3 = (0, 0, 1)$, $4 = (0, 0, 0, 1)$ und schliesslich die Approximationsdaten (Bereich $(0, 0, 0, 0)$) so skaliert aufgetragen, dass sie eine gemeinsame x -Achse haben, die der in Abbildung 1.99 entspricht. Es fällt auf, dass sich die Unstetigkeiten bei $x_2 = -0.2$ und bei $x_4 = 0.6$ über alle (fast) Auflösungsstufen deutlich bemerkbar machen, dass aber die Unstetigkeiten bei $x_1 = -1$ und $x_3 = 0$ nicht registriert werden! Das ist ein

typischer Defekt des Haar-Wavelets (und nur des Haar-Wavelets), da sich die eigentlichen Transformationsintervalle der Länge 2 nicht überlappen. Sprünge über diese Intervallgrenzen hinweg kann man nicht erfassen.

Hier ist noch die Darstellung der Detailinformation als sogenanntes Skalogramm:

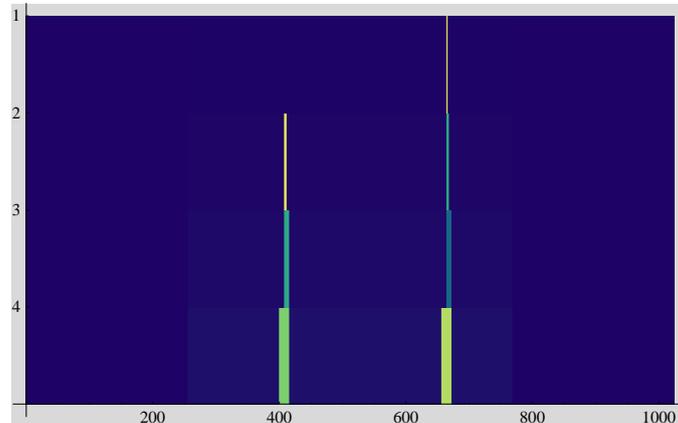


Abbildung 1.101: Skalogramm-Darstellung der Wavelettransformierten (nur Detailinformation) aus Abbildung 1.100.

Man erkennt an den lokalen Entsprechungen der Intensitäten über die Skalierungsstufen hinweg, dass man es mit zwei Kanten zu tun hat. Die beiden anderen Kanten werden jedoch überhaupt nicht erkannt.

Jetzt wird die Funktion aus Abbildung 1.99 einer vierstufigen Wavelettransformation unterzogen, jetzt aber unter Verwendung des Daubechies-10-Wavelets.

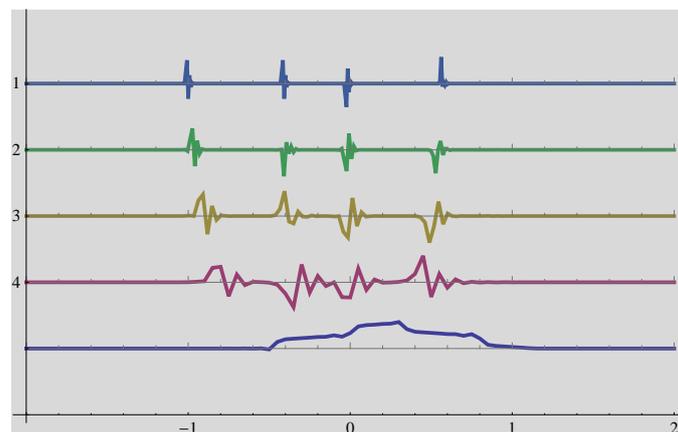


Abbildung 1.102: Vierstufige Daubechies-10-Transformation der Funktion aus Abbildung 1.99 mit auf gemeinsame x -Achse skalierten (leicht verzerrten) Detaildaten.

Hier werden alle vier senkrechten Kanten klar erkannt.

Das Prinzip sollte klar geworden sein: markante Informationen, die sich an lokal entsprechenden Positionen über mehrere Auflösungsstufen hinweg feststellen lassen, deuten auf markante Strukturen im Ausgangsbild hin.

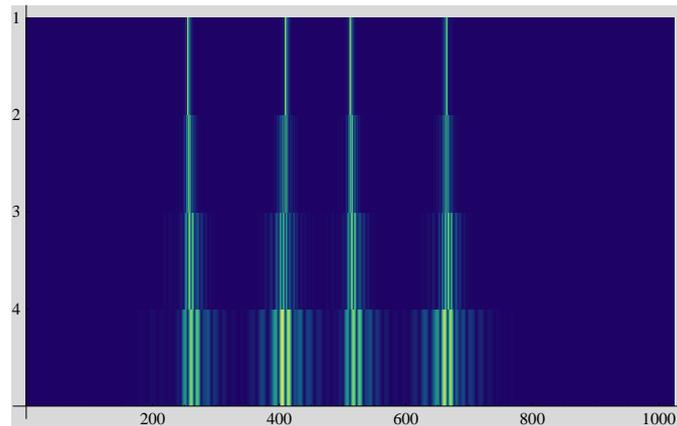


Abbildung 1.103: Skalogramm-Darstellung der Wavelettransformierten (nur Detailinformation) aus Abbildung 1.102.

1.10.1 Ein zweidimensionales Beispiel

Hier wird das Portrait eines bekannten Künstlers einer vierstufigen Wavelettransformation mit dem Symlet-4-Wavelet unterworfen, die Approximationsinformation wird gelöscht und dann zurücktransformiert.



Abbildung 1.104: Salavador Dali

1.11 Frequenztrennung

Als abschliessendes Beispiel soll gezeigt werden, wie man überlagerte Signale mit verschiedenen Frequenzen mittels Wavelettransformation trennen kann. Das Prinzip sollte sich aus den vorangehenden Beispielen ohne weiteres erschliessen.

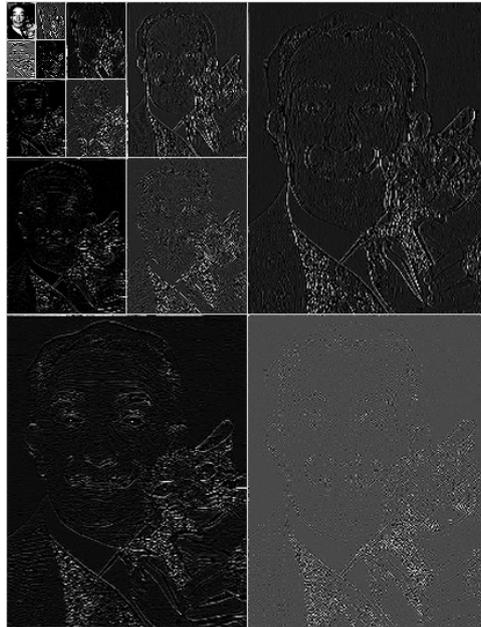


Abbildung 1.105: Vierstufige Wavelettransformation (mit Symlet-4) von Abbildung 1.104.



Abbildung 1.106: Vierstufige Wavelet-Rücktransformation (mit Symlet-4) ohne Approximationsinformation von Abbildung 1.104.

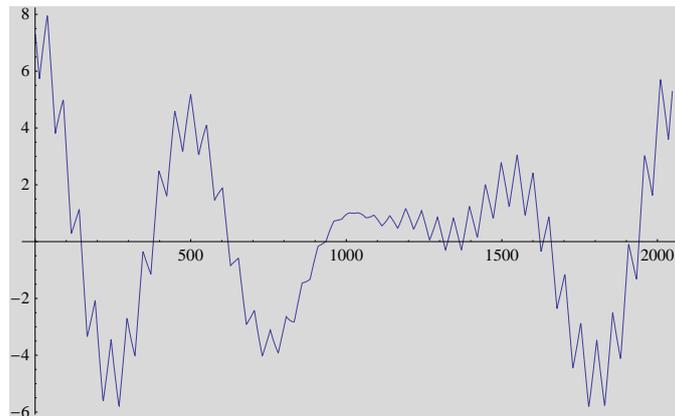


Abbildung 1.107: Überlagertes Signal

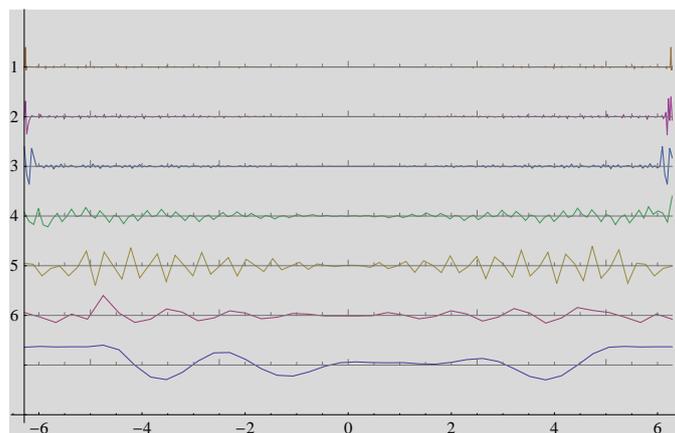


Abbildung 1.108: Detail- und Approximationsinformationen nach sechstufiger Transformation mit Symlet-6

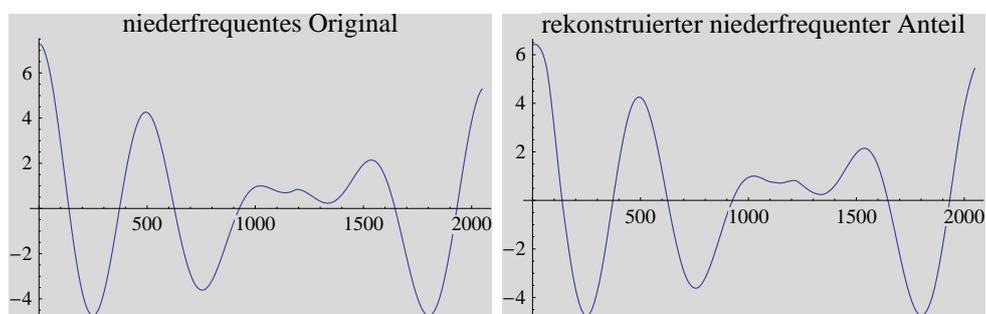


Abbildung 1.109: Anteile niedriger Frequenz am Signal von Abbildung 1.107

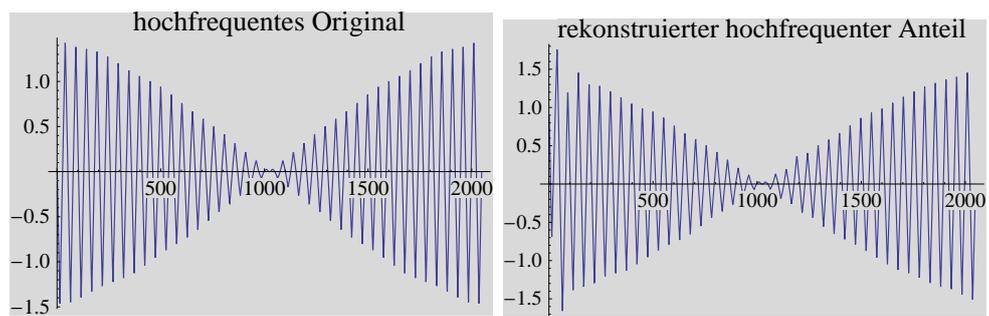


Abbildung 1.110: Anteile hoher Frequenz am Signal von Abbildung 1.107

2. Mathematische Präliminarien

Transformationen, wie die Fourier-Transformation und Wavelet-Transformationen, sind Aktionen, die man auf *Signalen* im weiten Sinne ausführt. Also ist erst einmal zu klären, wie der Begriff des Signals mathematisch gefasst werden soll.

- Zunächst einmal sind Signale (im umgangssprachlichen Sinn), wie Bilder, Audiodaten, Zeitreihen (Börsendaten, seismische Daten u.v.a.m.) Funktionen, die von einer oder mehreren diskreten oder kontinuierlich variierenden Variablen (Zeit, Ort) abhängen. Diese Funktionen nehmen reelle (oder auch komplexe) Werte an, die über einen diskreten oder kontinuierlichen Bereich variieren. Auch wenn im Computer repräsentierte Signale immer diskrete Variable und diskrete Werte haben, ist es sinnvoll, wegen der mathematischen Ökonomie (der Effizienz des Apparats der klassischen Analysis) auch Signale mit kontinuierlichen Variablen und/oder Werten zu betrachten. Typischerweise werden eindimensionale (1D) Signale also Funktionen von folgender Art¹ sein:
 - $f : \mathbb{R} \rightarrow \mathbb{R}$ oder \mathbb{C} ;
 - $f : [a, b) \rightarrow \mathbb{R}$ (oder \mathbb{C}), wobei $a, b \in \mathbb{R}$ mit $a < b$;
 - $f : \mathbb{Z} \rightarrow \mathbb{R}$ (oder \mathbb{C} , \mathbb{Z} , \mathbb{Z}_N, \dots);
 - $f : \mathbb{Z}_N \rightarrow \mathbb{R}$ (oder \mathbb{C} , \mathbb{Z} , \mathbb{Z}_N , \mathbb{C}, \dots).
- Signale werden transformiert, um spezifische Eigenschaften zu erkennen oder zu modifizieren. Typische Aufgaben im Bereich der Bildverarbeitung sind etwa: Filterung, Kompression, Entrauschung, Erkennung von speziellen Merkmalen

¹ Zur Nomenklatur: \mathbb{R} = reelle Zahlen, $\mathbb{R}_{\geq 0}$ = nicht-negative reelle Zahlen, $\mathbb{R}_{>0}$ = positive reelle Zahlen, \mathbb{C} = komplexe Zahlen, \mathbb{Z} = ganze Zahlen, \mathbb{N} = natürliche Zahlen, $\mathbb{N}_{>0}$ = positive natürliche Zahlen, $\mathbb{Z}_N = \{0, 1, 2, \dots, N-1\}$. Für $a, b \in \mathbb{R}$ mit $a < b$ ist $[a, b] = \{r \in \mathbb{R}; a \leq r \leq b\}$ das abgeschlossene endliche Intervall, $[a, b) = \{r \in \mathbb{R}; a \leq r < b\}$ das halboffene endliche Intervall usw.

(feature detection), Fusion, ...

- Signale werden der mathematische Ökonomie wegen grundsätzlich als Elemente eines geeigneten (reellen oder komplexen) Vektorraums aufgefasst. Dann kann man Signale Addieren, Subtrahieren, Multiplizieren, ... und die angesprochenen Transformationen können sehr oft (aber nicht immer!) als *lineare Transformationen* solcher Vektorräume verstanden werden. Beispielsweise wird das Filtern von Signalen meist mittels einer *Faltungsoperation* auf dem Vektorraum der Signale realisiert, der zu diesem Zweck noch die zusätzliche algebraische Eigenschaft haben muss, dass der Definitionsbereich der Signale $(\mathbb{R}, [a, b], \mathbb{Z}, \mathbb{Z}_N)$ eine additive Gruppenstruktur trägt. Für \mathbb{R} und \mathbb{Z} ist das die gewöhnliche Addition. Für \mathbb{Z}_N ist das die "Addition modulo N ". Für endliche Intervalle $[a, b)$ stellt man sich Funktionen $f : [a, b) \rightarrow \mathbb{R}$ periodisch fortgesetzt als Funktionen $f^* : \mathbb{R} \rightarrow \mathbb{R}$ vor², für die man die Faltung wie in \mathbb{R} berechnet. Das Resultat ist eine $(b - a)$ -periodische Funktion, die auf ganz \mathbb{R} definiert ist und die man wieder als Funktion auf $[a, b)$ auffassen kann. Das Kapitel über Fourier-Reihen zeigt, wie das gemeint ist und benutzt wird.

2.1 Normierte Vektorräume

Alle hier betrachteten Vektorräume sind Vektorräume über den komplexen Zahlen \mathbb{C} . In vielen Fällen, und die Wavelet-Transformationen gehören dazu, genügt es, für die Transformationen selbst reelle Vektorräume zu betrachten. Analysiert man aber die Eigenschaften solcher Transformationen, so kommt Fourier-Analyse hinzu, und damit befindet man sich wieder im Bereich komplexer Vektorräume, da die Basisfunktionen der Fourier-Analyse nun einmal komplexe Exponentialfunktionen sind. So ist es sinnvoll, von vorne herein komplexe Vektorräume zu betrachten.

Definition 2.1 (Normierter Vektorraum).

Ein normierter Vektorraum ist ein Vektorraum V mit einer Normabbildung (kurz: Norm)

$\|\cdot\| : V \rightarrow \mathbb{R}$, d.h., es gilt für alle $u, v \in V$ und alle $a \in \mathbb{C}$:

1. $\|v\| \geq 0$ und $\|v\| = 0 \Leftrightarrow v = 0$,
2. $\|a \cdot v\| = |a| \cdot \|v\|$,
3. $\|u + v\| \leq \|u\| + \|v\|$.

Es gilt dann übrigens auch immer

$$\|u - v\| \geq \left| \|u\| - \|v\| \right| \quad (u, v \in V).$$

■ Beispiel 2.1

1. Der endlich-dimensionale Vektorraum $\mathcal{V} = \mathbb{C}^n$ kann auf verschiedene Weisen zu einem normierten Vektorraum gemacht werden. Für $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{V}$:
 - (a) 1-Norm: $\|\mathbf{x}\|_1 = \sum_j |x_j|$

² Genau: $f^*(r) = f(r - \lfloor \frac{r-a}{b-a} \rfloor \cdot (b-a))$ für alle $r \in \mathbb{R}$.

$$(b) \text{ 2-Norm: } \|\mathbf{x}\|_2 = \sqrt{\sum_j x_j \bar{x}_j}$$

$$(c) \text{ } \infty\text{-Norm: } \|\mathbf{x}\|_\infty = \max_j |x_j|$$

$$(d) \text{ allgemein } p\text{-Norm (für } p \geq 1\text{): } \|\mathbf{x}\|_p = \left(\sum_j |x_j|^p\right)^{1/p}$$

2. Der Vektorraum $\mathcal{V} = \mathcal{C}^0[a, b]$ der stetigen Funktionen $f : [a, b] \rightarrow \mathbb{C}$ mit $a, b \in \mathbb{R}$ kann auf verschiedene Weisen zu einem normierten Vektorraum gemacht werden.

$$(a) \text{ 1-Norm: } \|f\|_1 = \int_a^b |f(t)| dt$$

$$(b) \text{ 2-Norm: } \|f\|_2 = \sqrt{\int_a^b f(t) \cdot \bar{f}(t) dt}$$

$$(c) \text{ } \infty\text{-Norm: } \|f\|_\infty = \max_{t \in [a, b]} |f(t)|$$

$$(d) \text{ allgemein } p\text{-Norm (für } p \geq 1\text{): } \|f\|_p = \left(\int_a^b |f(t)|^p dt\right)^{1/p}$$

■

Jede Norm $\|\cdot\|$ auf einem Vektorraum \mathcal{V} definiert eine *Metrik* auf \mathcal{V} :

$$d : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}_{\geq 0} : (u, v) \mapsto d(u, v) = \|u - v\|,$$

wobei für alle $u, v, w \in \mathcal{V}$ folgende Eigenschaften gelten:

1. $d(u, v) \geq 0$ und $d(u, v) = 0 \Leftrightarrow u = v$,
2. $d(u, v) = d(v, u)$,
3. $d(u, w) \leq d(u, v) + d(v, w)$.

Damit einher geht ein *Konvergenzbegriff*: für eine Folge $(v^n)_{n \geq 0}$ in \mathcal{V} und ein Element $w \in \mathcal{V}$ definiert man:

$$\lim_{n \rightarrow \infty} v^n = w \Leftrightarrow \lim_{n \rightarrow \infty} d(v^n, w) = 0.$$

Der Begriff der *Cauchyfolge*³ wird wie üblich in Räumen mit einer Metrik definiert.

Definition 2.2 (Vollständigkeit, Banachraum^a).

Ein normierter Vektorraum, in dem jede Cauchyfolge konvergiert, heisst vollständig, oder auch Banachraum.

^a Benannt nach dem polnischen Mathematiker Stefan BANACH (1892–1945)

2.2 Vektorräume mit innerem Produkt

Ein besonders interessanter Fall von normierten Vektorräumen sind solche, bei denen die Norm durch ein sogenanntes *inneres Produkt* (häufig auch *Skalarprodukt* genannt) definiert wird. Innere Produkte erlauben es – und das ist im Kontext dieser Veranstaltung von grundlegender Bedeutung – von Winkeln und von *Orthogonalität* zu sprechen.

³ Benannt nach dem französischen Mathematiker Augustin-Louis Cauchy (1789–1857)

Definition 2.3 (Inneres Produkt).

Ein Vektorraum mit innerem Produkt (oft auch als hermitesches^a oder komplexes Skalarprodukt bezeichnet) ist ein komplexer Vektorraum \mathcal{V} mit einer Abbildung

$$\langle \cdot | \cdot \rangle : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{C} : (u, v) \mapsto \langle u | v \rangle,$$

so dass für alle $u, v, w \in \mathcal{V}$ und für alle $a, b \in \mathbb{C}$ folgendes gilt:

1. $\langle v | v \rangle \geq 0$ und $\langle v | v \rangle = 0 \Leftrightarrow v = 0$,
2. $\langle a \cdot u + b \cdot v | w \rangle = a \langle u | w \rangle + b \langle v | w \rangle$,
3. $\langle u | v \rangle = \overline{\langle v | u \rangle}$.

Ein solcher Vektorraum wird gelegentlich als Prä-Hilbertraum bezeichnet.

^a Benannt nach dem französischen Mathematiker Charles HERMITE (1822–1901)

Aus einem inneren Produkt $\langle \cdot | \cdot \rangle$ erhält man eine Norm $\| \cdot \|$ vermöge

$$\|v\| = \sqrt{\langle v | v \rangle}.$$

Einige wichtige Beziehungen in dieser Situation sind:

Folgerung 2.1.

Für $u, v \in \mathcal{V}$ gilt:

1. $|\langle u, v \rangle| \leq \|u\| \cdot \|v\|$ (die Cauchy-Schwarz-Ungleichung)^a,
2. $\|u + v\|^2 = \|u\|^2 + \|v\|^2 + 2 \Re \langle u, v \rangle$,
3. $\|u\|^2 + \|v\|^2 = (\|u + v\|^2 + \|u - v\|^2)/2$
4. $\langle u, v \rangle = 4(\|u + v\|^2 - \|u - v\|^2 + i\|u + iv\|^2 - i\|u - iv\|^2)$.

^asiehe den folgenden Abschnitt

Die *Polarisationsgleichung* 4. zeigt, dass Norm und inneres Produkt sich gegenseitig eindeutig bestimmen.

■ **Beispiel 2.2**

1. \mathbb{C}^n mit dem üblichen Skalarprodukt für $\mathbf{x} = (x_1, \dots, x_n)$, $\mathbf{y} = (y_1, \dots, y_n)$:

$$\langle \mathbf{x} | \mathbf{y} \rangle = \sum_{j=1}^n x_j \cdot \bar{y}_j.$$

Die zugehörige Norm ist die übliche *euklidische Norm* $\|\mathbf{x}\| = \sqrt{\sum_{i=1}^n |x_i|^2}$.

2. Der Raum $\ell^2(\mathbb{Z})$ aller zweiseitig unendlichen Folgen $\mathbf{x} = (x_j)_{j \in \mathbb{Z}}$ mit “endlicher Energie” $\|\mathbf{x}\|_2^2 = \sum_{j \in \mathbb{Z}} |x_j|^2 < \infty$. Das innere Produkt für $\mathbf{x} = (x_j)$, $\mathbf{y} = (y_j)$ ist das Skalarprodukt

$$\langle \mathbf{x} | \mathbf{y} \rangle = \sum_{j \in \mathbb{Z}} x_j \cdot \bar{y}_j.$$

3. Der Raum $\mathcal{C}^0[a, b]$ der stetigen Funktionen auf einem Intervall $a, b \in \mathbb{R}$ mit $-\infty < a < b < \infty$ und

$$\langle f | g \rangle = \int_a^b f(t) \cdot \overline{g(t)} dt.$$

4. Der Raum $\mathcal{L}^2[a, b]$ (für $a, b \in \mathbb{R} \cup \{\pm\infty\}$) aller *quadrat-integrierbaren Funktionen* $f : [a, b] \rightarrow \mathbb{C}$, d.h. Funktionen mit “endlicher Energie” $\int_a^b |f(t)|^2 dt < \infty$ und innerem Produkt

$$\langle f | g \rangle = \int_a^b f(t) \cdot \overline{g(t)} dt.$$

■

Definition 2.4 (Hilbertraum^a).

Einen Vektorraum mit innerem Produkt $\langle \cdot | \cdot \rangle$, der bezüglich der von der zugehörigen Norm $\|v\| = \sqrt{\langle v | v \rangle}$ entsprechenden Metrik vollständig ist, nennt man einen Hilbertraum.

^a Benannt nach dem deutschen Mathematiker David HILBERT (1862–1943)

Die endlich-dimensionalen Räume $\mathcal{V} = \mathbb{C}^n$ sind (auf offensichtliche Weise) Hilberträume. $\ell^2(\mathbb{Z})$ und $\mathcal{L}^2[a, b]$ sind Hilberträume, $\mathcal{C}^0[a, b]$ dagegen nicht.

- R** Ein Hilbertraum \mathcal{H} heisst *separabel*, wenn es eine abzählbar-unendliche Teilmenge $\mathcal{E} \subset \mathcal{H}$ gibt, die in \mathcal{H} *dicht* ist, d.h., zu jedem $h \in \mathcal{H}$ und jedem $\varepsilon > 0$ gibt es ein Element $e \in \mathcal{E}$ mit $\|h - e\| < \varepsilon$.

Im Kontext dieser Vorlesung werden nur separable Hilberträume auftreten.

- R** Die Funktionenräume $\mathcal{L}^p(I)$ für $1 \leq p \leq \infty$ und ein endliches oder unendliches Intervall $I \subseteq \mathbb{R}$ werden in Analogie zu \mathcal{L}^2 definiert als die Menge der Funktionen $f : I \rightarrow \mathbb{C}$ mit $\|f\|_p = (\int_I |f(t)|^p dt)^{1/p} < \infty$. Dabei ist der Fall $p = \infty$ als $\|f\|_\infty = \sup_{t \in I} |f(t)|$ zu lesen. All diese Räume sind normiert, jeweils mittels $f \mapsto \|f\|_p$. Allerdings sind diese Räume *nur im Fall* $p = 2$ Hilberträume! Das ist ein Grund für die Ausnahmestellung der \mathcal{L}^2 -Räume.

Caveat: Streng genommen sind die Elemente von $\mathcal{L}^2[a, b]$ nicht “Funktionen” im üblichen Sinn, sondern Äquivalenzklassen von Funktionen. Dabei werden zwei Funktionen als äquivalent betrachtet, wenn sie sich nur auf einer Menge vom (Lebesgueschen) Maß 0 unterscheiden. Das erfordert eine Kenntnis des Lebesgueschen Maß- und Integrationsbegriffes – insofern sind alle späteren Ausführungen zu \mathcal{L}^2 -Räumen entsprechend zu lesen. Für praktische Anwendungen hat dieser Aspekt meist keine grosse Bedeutung.

2.3 Orthogonalität

In diesem Abschnitt soll \mathcal{V} zunächst einen Vektorraum mit innerem Produkt bezeichnen, also nicht notwendigerweise einen Hilbertraum. Erst am Ende des Abschnitts wird auch Vollständigkeit vorausgesetzt.

Definition 2.5 (Orthogonalsystem).

1. Vektoren $u, v \in \mathcal{V}$ sind orthogonal, geschrieben $u \perp v$, falls $\langle u | v \rangle = 0$.
2. Eine Menge $\{v^1, v^2, \dots\} \subseteq \mathcal{V}$ ist ein Orthogonalsystem, falls $\forall i, j (i \neq j) : v^i \perp v^j$ und $\forall j : \|v^j\| > 0$.
3. Eine Menge $\{v^1, v^2, \dots\} \subseteq \mathcal{V}$ ist ein Orthonormalsystem (ONS), falls $\forall i, j (i \neq j) : v^i \perp v^j$ und $\forall j : \|v^j\| = 1$.

Für orthogonale Vektoren $u, v \in \mathcal{V}$ gilt die *Pythagoras-Gleichung*

$$\|u + v\|^2 = \|u\|^2 + \|v\|^2$$

und entsprechend gilt für ein endliches Orthogonalsystem $\{v^1, v^2, \dots, v^n\} \subset \mathcal{V}$

$$\|v^1 + \dots + v^n\|^2 = \|v^1\|^2 + \dots + \|v^n\|^2.$$

Der Beweis der *Cauchy-Schwarz-Ungleichung* in Folgerung (2.1) ergibt sich unmittelbar aus der orthogonalen Zerlegung. Sind nämlich $u, v \in \mathcal{V}$ Vektoren mit $\|v\| = 1$, so ist der Vektor $\langle u | v \rangle v$ die *Projektion* von u in Richtung v . Offensichtlich gilt

$$u - \langle u | v \rangle v \perp v \quad \text{und} \quad u = (u - \langle u | v \rangle v) + \langle u | v \rangle v$$

ist die eindeutige Zerlegung von u in einen zu v orthogonalen und einen zu v parallelen Vektor. Daher gilt

$$\|u\|^2 = \|u - \langle u | v \rangle v\|^2 + |\langle u | v \rangle|^2 \cdot \|v\|^2$$

und als Konsequenz erhält man $|\langle u | v \rangle|^2 \leq \|u\|^2$ im speziellen Fall $\|v\| = 1$. Der allgemeine Fall ergibt sich daraus unmittelbar.

Die Cauchy-Schwarz-Ungleichung ist ein wichtiges technisches Hilfsmittel in vielen Situationen. Sie zeigt beispielsweise, dass inneres Produkt und Norm in einem (prä-)Hilbertraum *stetige* Funktionen ihrer Argumente sind:

$$\begin{aligned} |\langle u + h | v + k \rangle - \langle u | v \rangle| &= |\langle u | k \rangle + \langle h | v \rangle + \langle h | k \rangle| \\ &\leq \|u\| \|k\| + \|h\| \|v\| + \|h\| \|k\| \end{aligned}$$

und daher gilt

$$u \rightarrow u', v \rightarrow v' \quad \Rightarrow \quad \langle u | v \rangle \rightarrow \langle u' | v' \rangle,$$

wobei mit $u \rightarrow u'$ und $v \rightarrow v'$ die Norm-Konvergenz, gemeint ist, mit $\langle u | v \rangle \rightarrow \langle u' | v' \rangle$ dagegen die Konvergenz in \mathbb{R} .

Das folgende Theorem verallgemeinert die spezielle Version der Cauchy-Schwarz-Ungleichung:

Theorem 2.1 — Bessels Ungleichung^a.

Ist $\{v^1, v^2, \dots\}$ ein Orthonormalsystem in \mathcal{V} , so gilt für jedes $u \in \mathcal{V}$

$$\sum_{j \geq 1} |\langle u | v^j \rangle|^2 \leq \|u\|^2.$$

^a Benannt nach dem deutschen Astronomen, Mathematiker und Physiker Friedrich Wilhelm BESSEL (1784–1846)

Der *Beweis* ergibt sich für $k \rightarrow \infty$ aus

$$0 \leq \|u - \sum_{1 \leq j \leq k} \langle u | v^j \rangle \cdot v^j\|^2 = \dots = \|u\|^2 - \sum_{1 \leq j \leq k} |\langle u | v^j \rangle|^2.$$

Fourier- und Wavelet-Transformationen, wie viele andere orthogonale Transformationen, stützen sich auf grundlegende Eigenschaften orthogonaler Projektionen. Eine sehr allgemeine Aussage ist in der folgenden Formulierung enthalten.

Theorem 2.2 — Orthogonale Projektion (1).

Ist $\mathcal{U} \subseteq \mathcal{V}$ ein Unterraum des Vektorraums \mathcal{V} und $v \in \mathcal{V}$ ein Vektor, so sind für einen Vektor $u' \in \mathcal{U}$ die folgenden Aussagen äquivalent:

1. $\|v - u'\| = \inf_{u \in \mathcal{U}} \|v - u\| =: d(v, \mathcal{U})$,
2. $v - u' \perp \mathcal{U}$, d.h. $v - u' \perp u$ für alle $u \in \mathcal{U}$.

Man bezeichnet, wenn es existiert, u' als *orthogonale Projektion* von v auf \mathcal{U} .

Wenn eine solche orthogonale Projektion u' existiert, ist sie eindeutig bestimmt.

Über Existenz wird hierbei (noch) nichts gesagt. Zum Beweis betrachtet man für beliebiges $u \in \mathcal{U}$ die quadratische Funktion

$$f_u : t \mapsto \|v - u' - t \cdot u\|^2 = \|v - u'\|^2 + 2\Re\langle v - u' | u \rangle \cdot t + \|u\|^2 \cdot t^2.$$

Gilt die Aussage 1., so ist $\partial_t f_u(0) = 0$ und das bedeutet $\Re\langle v - u' | u \rangle = 0$. Ganz analog kann man aus $\partial_t f_{iu}(0) = 0$ folgern, dass $\Im\langle v - u' | u \rangle = 0$ ist. Insgesamt ist also $\langle v - u' | u \rangle = 0$ für beliebiges $u \in \mathcal{U}$, d.h. Aussage 2. gilt.

Diese Argumentation lässt sich problemlos umkehren, sodass aus 2. auch 1. folgt.

Die Eindeutigkeit der orthogonalen Projektion ist eine Folge der Parallelogrammgleichung. Sind nämlich u' und u'' zwei Elemente von \mathcal{U} mit

$$d(v, \mathcal{U}) = \|v - u'\| = \|v - u''\|,$$

so ist wegen $(u' + u'')/2 \in \mathcal{U}$

$$\begin{aligned} \|u' - u''\|^2 &= 2\|v - u'\|^2 + 2\|v - u''\|^2 - 4\|v - (u' + u'')/2\|^2 \\ &\leq 4d(v, \mathcal{U})^2 - 4d(v, \mathcal{U})^2 = 0. \end{aligned}$$

Also ist $u' = u''$. □

Eine unmittelbare Anwendung betrifft den Fall eines endlich-dimensionalen Unterraumes \mathcal{U} . In diesem Fall kann man das optimal approximierende Element u' explizit angeben, die Existenz macht also keine Schwierigkeiten.

Theorem 2.3 — Orthogonale Projektion (2).

Ist $\{u^1, \dots, u^n\}$ eine orthonormale Basis des endlich-dimensionalen Unterraums \mathcal{U} von \mathcal{V} und ist $v \in \mathcal{V}$, so ist die optimale Approximation von v in \mathcal{U} gegeben durch

$$u' = \sum_{1 \leq j \leq n} \langle v | u^j \rangle u^j.$$

Ist $\{u^1, \dots, u^n\}$ nur ein Orthogonalsystem, so gilt

$$u' = \sum_{1 \leq j \leq n} \frac{\langle v | u^j \rangle}{\langle u^j | u^j \rangle} u^j.$$

Zum Beweis muss man nur beachten, dass für jedes k ($1 \leq k \leq n$)

$$\langle u' | u^k \rangle = \sum_{j=1}^n \langle v | u^j \rangle \langle u^j | u^k \rangle = \langle v | u^k \rangle, \quad \text{also } \langle v - u' | u^k \rangle = 0,$$

gilt. Das bedeutet $v - u' \perp \mathcal{U}$. □

Definition 2.6.

1. Ein Unterraum \mathcal{U} eines Hilbertraumes \mathcal{H} heisst abgeschlossen (oder Hilbert-Unterraum), wenn er selbst vollständig ist, d.h., wenn alle Cauchyfolgen von Elementen aus \mathcal{U} gegen ein Element aus \mathcal{U} konvergieren.
2. Ist \mathcal{H} ein Hilbertraum und $\mathcal{U} \subseteq \mathcal{H}$ ein Unterraum, so wird der Unterraum

$$\mathcal{U}^\perp = \{v \in \mathcal{H}; \langle v | u \rangle = 0 \forall u \in \mathcal{U}\}$$

als orthogonales Komplement von \mathcal{U} in \mathcal{H} bezeichnet.

Aus der Stetigkeit des inneren Produkts folgt:

Lemma 2.1 Ist \mathcal{U} ein Unterraum von \mathcal{H} , so ist das orthogonale Komplement \mathcal{U}^\perp ein abgeschlossener Unterraum von \mathcal{H} (selbst dann, wenn \mathcal{U} selbst nicht abgeschlossen ist).

In diesem allgemeineren Rahmen kann man nun das

Prinzip der optimalen Approximation durch orthogonale Projektion

formulieren:

Theorem 2.4 — Orthogonale Projektion (3).

Ist \mathcal{H} ein Hilbertraum und \mathcal{U} ein abgeschlossener Unterraum, so gilt

$$\mathcal{H} = \mathcal{U} \oplus \mathcal{U}^\perp,$$

d.h., zu jedem $v \in \mathcal{H}$ gibt es eindeutig bestimmte $u' \in \mathcal{U}$, $u'' \in \mathcal{U}^\perp$ mit $v = u' + u''$.
Zudem gilt

$$\|v - u'\| = d(v, \mathcal{U}) = \min_{u \in \mathcal{U}} \|v - u\|.$$

Der Beweis verläuft im wesentlichen wie vorher, nur dass jetzt die Vollständigkeit die Existenz der orthogonalen Projektionen u' (auf \mathcal{U}) und u'' (auf \mathcal{U}^\perp) sicherstellt.

Definition 2.7. *Ist \mathcal{H} ein Hilbertraum und \mathcal{U} ein abgeschlossener Unterraum, so ist (mit den Bezeichnungen wie im Theorem)*

$$P_{\mathcal{U}} : \mathcal{H} \rightarrow \mathcal{U} : v \mapsto u'$$

die orthogonale Projektion von \mathcal{H} auf \mathcal{U} .

Die folgenden Aussagen sind unmittelbare Folgerungen aus dem schon Hergeleiteten:

Satz 2.1

1. Die Projektionen $P_{\mathcal{U}} : \mathcal{H} \rightarrow \mathcal{U}$ sind lineare Transformationen.
2. Sind $\mathcal{U} \subseteq \mathcal{V} \subseteq \mathcal{H}$ abgeschlossene Unterräume, so gilt $P_{\mathcal{U}} \circ P_{\mathcal{V}} = P_{\mathcal{U}}$, insbesondere ist $P_{\mathcal{U}} \circ P_{\mathcal{U}} = P_{\mathcal{U}}$, d.h., $P_{\mathcal{U}}$ ist idempotent.
3. $P_{\mathcal{U}}$ ist stetig, es gilt sogar $\|P_{\mathcal{U}}(v)\| \leq \|v\|$ für jedes $v \in \mathcal{H}$.

2.4 Duale Basen (biorthogonale Basen)

Im Rahmen eines kurzen Exkurses soll die Idee der *dualen Basis* zu einer gegebenen Basis eines Vektorraums erwähnt werden, ohne tiefer (d.h. im Bezug auf beliebige Hilberträume) darauf einzugehen. Der Grund dafür ist, dass später im Rahmen der Konstruktion von Filtern, die für Wavelettransformationen geeignet sind, sogenannte *biorthogonale* Filterpaare vorkommen werden (Kapitel 10 und Kapitel 11). Hier geht es nur darum, die Bezeichnung *biorthogonal* von Ihrer Bedeutung in Vektorräumen mit innerem Produkt her zu motivieren. Der Einfachheit halber soll es in diesem Abschnitt nur um *endlich-dimensionale* Vektorräume gehen.

Es sei also V ein endlich-dimensionaler (reeller oder komplexer) Vektorraum mit innerem Produkt $\langle \cdot | \cdot \rangle$ und einer orthonormalen Basis $\mathcal{E} = (e^1, \dots, e^n)$, i.e., es gilt

$$\langle e^i | e^j \rangle = \delta_{i,j} \quad (1 \leq i, j \leq n).$$

Aus der Orthonormalität ergibt sich, wie bereits betont, die folgende fundamentale Aussage, die dem Prinzip der optimalen Approximation zugrunde liegt:

Für jeden Vektor $\mathbf{v} \in V$ gilt

$$\mathbf{v} = \sum_{i=1}^n \varepsilon_i \mathbf{e}^i \quad \text{with} \quad \varepsilon_i = \langle \mathbf{v} | \mathbf{e}^i \rangle,$$

d.h., die Koeffizienten ε_i von \mathbf{v} bezüglich der orthonormalen Basis \mathcal{E} sind die inneren Produkte $\langle \mathbf{v} | \mathbf{e}^i \rangle$.

Nun sei $\mathcal{B} = (\mathbf{b}^1, \dots, \mathbf{b}^n)$ eine beliebige Basis von V , also nicht notwendigerweise eine orthogonale. Dann kann jeder Vektor $\mathbf{v} \in V$ als Linearkombination von \mathcal{E} geschrieben werden:

$$\mathbf{v} = \sum_{i=1}^n \beta_i \mathbf{b}^i,$$

aber was sind dann die Koeffizienten β_i im Bezug auf das innere Produkt $\langle \cdot | \cdot \rangle$?

Mit \mathcal{E} und \mathcal{B} gilt

$$\mathbf{b}^i = \sum_{j=1}^n b_{i,j} \mathbf{e}^j, \quad \text{wobei} \quad b_{i,j} = \langle \mathbf{b}^i | \mathbf{e}^j \rangle.$$

Das definiert die Matrix

$$B = [b_{i,j}]_{1 \leq i,j \leq n},$$

und diese Matrix muss invertierbar sein, da ja \mathcal{B} eine Basis ist.

Sei nun $\mathcal{C} = (\mathbf{c}^1, \dots, \mathbf{c}^n)$ eine weitere Basis von V mit der entsprechenden Matrix

$$C = [c_{i,j}]_{1 \leq i,j \leq n}, \quad \text{wobei} \quad c_{i,j} = \langle \mathbf{c}^i | \mathbf{e}^j \rangle.$$

Definition 2.8. Die beiden Basen \mathcal{B} und \mathcal{C} sind zueinander duale Basen (oder auch: bilden ein Paar von biorthogonalen Basen), wenn

$$\langle \mathbf{b}^i | \mathbf{c}^j \rangle = \delta_{i,j} \quad (1 \leq i, j \leq n).$$

Damit stellen sich zwei Fragen

1. Existiert zu einer gegebenen Basis \mathcal{B} immer eine duale Basis \mathcal{C} ?
Und wenn ja, ist diese eindeutig bestimmt?
2. Wozu dienen duale Basen?

- Zunächst die Antwort auf die zweite Frage:

Angenommen, \mathcal{B} und \mathcal{C} sind zueinander duale Basen, da gilt für $\mathbf{v} \in V$

$$\mathbf{v} = \sum_{i=1}^n \beta_i \mathbf{b}^i = \sum_{j=1}^n \gamma_j \mathbf{c}^j$$

und somit einerseits

$$\langle \mathbf{v} | \mathbf{c}^j \rangle = \sum_{i=1}^n \beta_i \langle \mathbf{b}^i | \mathbf{c}^j \rangle = \beta_j,$$

als auch andererseits

$$\langle \mathbf{v} | \mathbf{b}^i \rangle = \sum_{j=1}^n \gamma_j \langle \mathbf{c}^j | \mathbf{b}^i \rangle = \gamma_i.$$

Damit ist

$$\mathbf{v} = \sum_{i=1}^n \langle \mathbf{v} | \mathbf{c}^i \rangle \mathbf{b}^i = \sum_{j=1}^n \langle \mathbf{v} | \mathbf{b}^j \rangle \mathbf{c}^j$$

und das klärt die oben gestellte Frage nach der Interpretation der Koeffizienten eines Vektors v bezüglich einer beliebigen Basis von V : das sind die inneren Produkte von v mit den Elementen der dualen Basis! Die Situation einer orthonormalen Basis \mathcal{E} ist also sehr spezielle: die duale Basis zu einer orthonormalen Basis ist diese Basis selbst!

- Die Antwort auf die erste Frage ist sehr einfach:

Sind \mathcal{B} und \mathcal{C} zwei beliebige Basen von V , wie oben, dann gilt auf Grund der Orthonormalität von \mathcal{E} für beliebige $1 \leq i, k \leq n$:

$$\begin{aligned} \langle \mathbf{b}^i | \mathbf{c}^k \rangle &= \left\langle \sum_{j=1}^n b_{i,j} \mathbf{e}^j \mid \sum_{\ell=1}^n c_{k,\ell} \mathbf{e}^\ell \right\rangle \\ &= \sum_{j=1}^n \sum_{\ell=1}^n b_{i,j} \overline{c_{k,\ell}} \langle \mathbf{e}^j | \mathbf{e}^\ell \rangle = \sum_{j=1}^n b_{i,j} \overline{c_{k,j}}. \end{aligned}$$

In Matrizenform geschrieben:

$$[\langle \mathbf{b}^i | \mathbf{c}^k \rangle]_{1 \leq i, k \leq n} = B \cdot C^\dagger,$$

wobei C^\dagger die konjugierte und transponierte Matrix von C bezeichnet, die sogenannte *adjungierte Matrix* von C . Also gilt

$$\mathcal{B} \text{ und } \mathcal{C} \text{ sind zueinander duale Basen} \iff B \cdot C^\dagger = I_n \iff B^{-1} = C^\dagger$$

und damit ist die Frage der Existenz und Eindeutigkeit geklärt.

2.5 Hilbert-Basen

Definition 2.9.

1. Ist \mathcal{H} ein Hilbertraum und $M \subset \mathcal{H}$ eine Menge von Vektoren, so bezeichnet $\mathcal{F}(M)$ (oder auch $\text{span}(M)$) die Menge der (endlichen!) Linearkombinationen von Elementen aus M .
 $\mathcal{F}(M)$ ist der kleinste Unterraum von \mathcal{H} , der M enthält und wird als der von M erzeugte Unterraum von \mathcal{H} bezeichnet.
 $(\mathcal{F}(M)$ ist aber i.a. nicht vollständig, also i.a. kein abgeschlossener Unterraum von \mathcal{H}).
2. Ist \mathcal{H} ein Hilbertraum und $M \subset \mathcal{H}$ eine Menge von Vektoren, so bezeichnet \overline{M} , die Menge M , erweitert um alle Elemente von \mathcal{H} , die sich mittels Elementen aus M in der Norm-Metrik approximieren lassen (Grenzwerte von Cauchyfolgen). Dies ist der Abschluss von M in \mathcal{H} ,
3. Ist \mathcal{H} ein Hilbertraum und $M \subset \mathcal{H}$ eine Menge von Vektoren, so ist $\overline{\mathcal{F}(M)} = \overline{\text{span}(M)}$ der von M erzeugte Hilbert-Unterraum von \mathcal{H} , i.e., der kleinste M enthaltende abgeschlossene Unterraum von \mathcal{H} .

Aus einem System unabhängiger Vektoren in einem Vektorraum mit innerem Produkt kann man iterativ ein Orthonormalsystem konstruieren.

Theorem 2.5 — Gram-Schmidt-Orthogonalisierung.

Ist $U = \{u^1, u^2, \dots\}$ eine linear-unabhängige Familie von Vektoren in \mathcal{V} , so kann man ein Orthonormalsystem $V = \{v^1, v^2, \dots\}$ in \mathcal{V} konstruieren, so dass gilt:

- $\langle v^j, u^j \rangle > 0$ für alle $j \in \mathbb{N}$,
- $\text{span}\{u^1, \dots, u^k\} = \text{span}\{v^1, \dots, v^k\}$ für alle $k \in \mathbb{N}$.

Das Gram-Schmidt-Verfahren⁴ erlaubt es auch, aus einer beliebigen Folge $U = \{u^1, u^2, \dots\}$ von Vektoren iterativ ein ONS zu gewinnen. Man muss nur bei der Konstruktion die Vektoren u^k mit $u^k \in \text{span}(u^1, u^2, \dots, u^{k-1})$ einfach übergehen. Natürlich ist dann nicht garantiert, dass tatsächlich ein unendliches ONS erzeugt wird – wenn nämlich ganz U bereits in einem endlich-dimensionalen Teilraum enthalten ist.

Theorem 2.6 — Orthonormale Systeme.

Ist \mathcal{H} ein Hilbertraum und $V = (v^n)_{n \geq 1}$ ein ONS in \mathcal{H} , $\mathcal{V} = \overline{\text{span}}(V)$ der von V erzeugte Hilbert-Unterraum von \mathcal{H} , so gilt:

1. Ist $(\alpha_n)_{n \geq 1}$ eine komplexe Folge, so ist

$$\sum_{n \geq 1} \alpha_n v^n \text{ konvergiert in } \mathcal{H} \quad \Leftrightarrow \quad \sum_{n \geq 1} |\alpha_n|^2 < \infty$$

⁴ Benannt nach dem dänischen Mathematiker Jørgen Pedersen GRAM (1850–1916) und dem deutschen Mathematiker Erhard SCHMIDT (1876–1959)

und in diesem Fall gilt $\|\sum_{n \geq 1} \alpha_n v^n\|^2 = \sum_n |\alpha|^2$.

2. Bessels Ungleichung: Für alle $u \in \mathcal{H}$ gilt

$$\sum_{n \geq 1} |\langle u | v^n \rangle|^2 \leq \|u\|^2.$$

3. Für alle $u \in \mathcal{H}$ konvergiert

$$\sum_{n \geq 1} \langle u | v^n \rangle v^n = P_{\mathcal{V}}(u).$$

4. Für alle $u, w \in \mathcal{H}$ gilt (mit absoluter Konvergenz)

$$\sum_{n \geq 1} \langle u | v^n \rangle \overline{\langle w | v^n \rangle} = \langle P_{\mathcal{V}}(u) | P_{\mathcal{V}}(w) \rangle.$$

Zum Beweis einige Anmerkungen:

1. Aus der Pythagoras-Identität folgt, dass $(\sum_{n=1}^N \alpha_n v^n)_{N \geq 1}$ genau dann eine Cauchyfolge in \mathcal{H} ist, wenn $(\sum_{n=1}^N |\alpha_n|^2)_{N \geq 1}$ eine Cauchyfolge in \mathbb{R} ist. Ausserdem verwendet man die Stetigkeit der Norm.
2. Folgt aus der Bessel-Ungleichung für $\mathcal{V}_n = \text{span}(v^1, \dots, v^n)$ für $n \rightarrow \infty$.
3. Aus der Bessel-Ungleichung und 1. folgt, dass $\sum_{n \geq 1} \langle u | v^n \rangle v^n$ in \mathcal{V} konvergiert. Andererseits ist $u - \sum_{n \geq 1} \langle u | v^n \rangle v^n$ in \mathcal{V}^\perp , so dass die Behauptung aus der Prinzip der orthogonalen Projektion folgt.
4. Für $N \geq 1$ folgt aus der Cauchy-Schwarz-Ungleichung

$$\left(\sum_{n=1}^N |\langle u | v^n \rangle \overline{\langle w | v^n \rangle}| \right)^2 \leq \sum_{n=1}^N |\langle u | v^n \rangle|^2 \cdot \sum_{n=1}^N |\langle w | v^n \rangle|^2 \leq \|u\|^2 \|w\|^2,$$

so dass $\sum_{n=1}^N \langle u | v^n \rangle \overline{\langle w | v^n \rangle}$ absolut konvergiert. Ausserdem ist

$$\left\langle \sum_{n=1}^N \langle u | v^n \rangle v^n, \sum_{n=1}^N \langle w | v^n \rangle v^n \right\rangle = \sum_{n=1}^N \langle u | v^n \rangle v^n \overline{\langle w | v^n \rangle v^n},$$

so dass die Behauptung für $N \rightarrow \infty$ aus der Stetigkeit des inneren Produktes folgt. \square

Das folgende Theorem fasst die Situation zusammen. Es ergibt sich aus den vorhergehenden Aussagen ohne weitere technische Probleme.

Theorem 2.7 — Hilbert-Basen.

Für einen Hilbertraum \mathcal{H} mit ONS $V = (v^n)_{n \geq 1}$ sind folgende Aussagen äquivalent:

1. V erzeugt \mathcal{H} : $\overline{\text{span}(V)} = \mathcal{H}$.
2. $V^\perp = \{0\}$.

3. Parseval-Plancherel Identität: Für alle $u \in \mathcal{H}$ gilt

$$\|u\|^2 = \sum_{n \geq 1} |\langle u | v^n \rangle|^2.$$

4. Für alle $u \in \mathcal{H}$ gilt

$$u = \sum_{n \geq 1} \langle u | v^n \rangle v^n.$$

Sind diese Aussagen erfüllt, so bezeichnet man V als eine *Hilbert-Basis* oder auch *vollständige Basis* von \mathcal{H} . Die konvergierende Reihe in 4. wird oft als *verallgemeinerte Fourierreihe* bezeichnet.

Eine Frage ist noch zu klären: Gibt es Hilbert-Basen überhaupt? Klar ist: wenn es eine solche Basis gibt, muss der Raum \mathcal{H} separabel sein! Ist nun $W = (w^n)_{n \geq 1} \subset \mathcal{H}$ eine abzählbare, dicht liegende Teilmenge, so kann man mittels Gram-Schmidt Orthogonalisierung daraus ein ONS $V = (v^n)_{n \geq 1}$ gewinnen. Dieses ist eine Hilbert-Basis von \mathcal{H} , denn es ist $V^\perp = \{0\}$:

Ist nämlich $u \in \mathcal{H}$ und gilt $\langle u | v^n \rangle = 0$ für alle $n \geq 1$, so gilt (wegen Gram-Schmidt) auch $\langle u | w^n \rangle = 0$ für alle $n \geq 1$. Da W dicht in \mathcal{H} liegt, gilt wegen der Stetigkeit des inneren Produktes auch $\langle u | w \rangle = 0$ für alle $w \in \mathcal{H}$, insbesondere trifft das auch auf $w = u$ zu. Somit ist $\langle u | u \rangle = 0$ und $u = 0$. \square

Theorem 2.8 Jeder separable Hilbertraum hat eine Hilbert-Basis.

2.6 Wichtige Beispiele von Hilbert-Basen

2.6.1 Der Hilbertraum $\ell^2(\mathbb{Z})$

Zur Erinnerung: Der Raum $\ell^2(\mathbb{Z})$ besteht aus allen zweiseitig unendlichen Folgen $\mathbf{x} = (x_j)_{j \in \mathbb{Z}}$ mit "endlicher Energie" $\|\mathbf{x}\|_2^2 = \sum_{j \in \mathbb{Z}} |x_j|^2 < \infty$. Das innere Produkt für $\mathbf{x} = (x_j)$, $\mathbf{y} = (y_j)$ ist das Skalarprodukt

$$\langle \mathbf{x} | \mathbf{y} \rangle = \sum_{j \in \mathbb{Z}} x_j \cdot \bar{y}_j.$$

Die "Einheitsvektoren" $E = (\mathbf{e}^n)_{n \in \mathbb{Z}}$ mit $(\mathbf{e}^n)_j = \delta_{n,j}$ ($n, j \in \mathbb{Z}$) bilden offensichtlich ein ONS in $\ell^2(\mathbb{Z})$.

Ist $\mathbf{x} = (x_j)_{j \in \mathbb{Z}}$ ein beliebiges Element von $\ell^2(\mathbb{Z})$, so gilt ganz einfach

$$\langle \mathbf{x} | \mathbf{e}^n \rangle = x_n \quad (n \in \mathbb{Z}),$$

und das zeigt $E^\perp = \{0\}$. Also ist E eine Hilbert-Basis von $\ell^2(\mathbb{Z})$.

2.6.2 Der Hilbertraum $\mathcal{L}^2([0, 1])$

Zur Erinnerung: Der Raum $\mathcal{L}^2[a, b]$ (für $a, b \in \mathbb{R} \cup \{\pm\infty\}$) besteht aus allen quadratintegrierbaren Funktionen $f : [a, b] \rightarrow \mathbb{C}$, d.h. Funktionen mit "endlicher Energie"

$\int_a^b |f(t)|^2 dt < \infty$, und innerem Produkt

$$\langle f | g \rangle = \int_a^b f(t) \cdot \overline{g(t)} dt.$$

Falls $[a, b]$ ein *endliches* Intervall ist, z.B. $[a, b] = [0, 1]$, so findet man leicht ein ONS E in $\mathcal{L}^2([0, 1])$, bestehend aus komplexen Exponentialfunktionen:

$$E = \{w^n(t); n \in \mathbb{Z}\} \quad \text{wobei} \quad w^n(t) = e^{2\pi i n t}.$$

(Man könnte genausogut auch die trigonometrische Funktionen $\sin(2\pi n t)$ ($n \geq 1$) und $\cos(2\pi n t)$ ($n \geq 0$) nehmen).

Der Nachweis der Orthogonalität ist eine leichte Aufgabe:

$$\langle w^m | w^n \rangle = \int_0^1 e^{2\pi i(m-n)t} dt = \begin{cases} 0 & \text{falls } m \neq n \\ 1 & \text{falls } m = n. \end{cases}$$

Bleibt noch nachzuweisen, dass tatsächlich $E^\perp = \{0\}$ und somit E eine Hilbert-Basis von $\mathcal{L}^2([0, 1])$ ist. Das wird im folgenden Kapitel über *Fourierreihen* im Abschnitt 3.3.3 ausgeführt. Alles was über das Intervall $[0, 1]$ ausgesagt wird, lässt sich problemlos auf beliebige *endliche* Intervalle $I \subset \mathbb{R}$ übertragen.

Im Zusammenhang mit Wavelets treten ganz andere Hilbert-Basen von $\mathcal{L}^2(I)$ für endliche Intervalle in Erscheinung.

2.6.3 Der Hilbertraum $\mathcal{L}^2(\mathbb{R})$

Zunächst eine Ergänzung zum Theorem 2.7. Wie im Abschnitt 3.3.2, siehe Folgerung 3.1, im Zusammenhang mit Summationskernen erwähnt, kann man beliebige Funktionen aus $\mathcal{L}^2(\mathbb{R})$ mittels stetiger Funktionen mit kompaktem Träger (das sind stetige Funktionen auf \mathbb{R} , die ausserhalb eines *endlichen* Intervalles identisch verschwinden; die Menge dieser Funktionen wird mit \mathcal{C}_c^0 bezeichnet) in der \mathcal{L}^2 -Norm approximieren. Das hat zur Folge, dass es genügt, für die Charakterisierung der Vollständigkeit eine Basis die \mathcal{C}_c^0 -Funktionen heranzuziehen, was technisch ererblich einfacher ist.

Ergänzung zu Theorem 2.7

- Für ein ONS $V = \{v^n\}_{n \in \mathbb{N}}$ sind folgende Aussagen äquivalent:
 1. V ist eine Hilbert-Basis von $\mathcal{L}^2(\mathbb{R})$.
 2. $\mathcal{C}_c^0 \subseteq \overline{\text{span}}(V)$.
 3. Die Identität von Parseval-Plancherel gilt für alle Elemente von \mathcal{C}_c^0 .

Es ist nun nicht klar, wie man überhaupt ein ONS in $\mathcal{L}^2(\mathbb{R})$ angeben soll. Die komplexen Exponentialfunktionen aus dem vorigen Beispiel kommen nicht in Frage, da sie nicht einmal zu $\mathcal{L}^2(\mathbb{R})$ gehören!

Das einfachste (aber immer noch keineswegs offensichtliche) Beispiel einer Hilbertbasis für $\mathcal{L}^2(\mathbb{R})$ führt bereits in das Reich der Wavelet-Transformationen. Es ist die Haar-Basis, bestehend aus der Familie H aller Funktionen

$$\psi_{j,k}(t) = 2^{j/2} \psi_H(t) \quad (j, k \in \mathbb{Z}),$$

wobei $\psi_H(t)$ die Wavelet-Funktion des sog. Haar-Wavelets ist:

$$\psi_H(t) = \begin{cases} 0 & \text{falls } t < 0 \text{ oder } t \geq 1, \\ 1 & \text{falls } 0 \leq t < 1/2, \\ -1 & \text{falls } 1/2 \leq t < 1, \end{cases}$$

Die Orthonormiertheit der Familie H ist leicht nachzuprüfen, aber für die Vollständigkeit $H^\perp = \{0\}$ muss man etwas tun — siehe Kapitel 5.

3. Fourierreihen

Für die seriöse Beschäftigung mit Wavelets sind solide Kenntnisse der Fourieranalysis (Fourierreihen, Fouriertransformation) unerlässlich, auch wenn sich der Anwender nicht mit den (bisweilen äusserst subtilen) Feinheiten der mathematischen Aspekte nicht im Detail beschäftigen will. In diesem und dem folgenden Kapitel werden die wesentlichen Tatsachen präsentiert, meist ohne Herleitungen, oder wenn doch, dann mit Beweisen, die den Gedankengang aufzeigen, ohne jedoch den Anspruch zu erheben, alles genau auszuleuchten. Dafür gibt es einen reichhaltigen, geradezu unübersehbaren Fundus and Literatur. An dieser Stelle nur drei Hinweise: [GW99], [Kam07], [PW12].

3.1 Motivation, Heuristik

Definition 3.1 (Periodische Funktionen).

1. Es sei $a > 0$ eine reelle Zahl. Eine Funktion $f : \mathbb{R} \rightarrow \mathbb{C}$ heisst a -periodisch, wenn $f(t + a) = f(t)$ für alle $t \in \mathbb{R}$ gilt.
2. Die a -periodischen Basisfunktionen sind die komplexen Exponentialfunktionen

$$w_{a,n}(t) = e^{2\pi i n(t/a)} \quad (n \in \mathbb{Z}).$$

Alternativ kann man als Basisfunktionen auch die Funktionen

$$\cos(2\pi n(t/a)) \quad (n \in \mathbb{N}), \quad \sin(2\pi n(t/a)) \quad (n \geq 1)$$

wählen. Wegen des Bezugs zur Fouriertransformation ist aber naheliegend, mit den komplexen Exponentialfunktionen als Basis umzugehen.

Eine a -periodische Funktion ist durch ihre Werte auf dem Intervall $[0, a)$ oder auf jedem anderen Intervall $[b, b + a)$ vollständig bestimmt.

Theorem 3.1 — Orthonormale Basis in $\mathcal{L}^2[0, a]$.

Die Familie $\{w_{a,n}(t)\}_{n \in \mathbb{Z}}$ ist ein ONS in $\mathcal{L}^2[0, a]$ bezüglich des Skalarprodukts

$$\langle f | g \rangle = \frac{1}{a} \int_0^a f(t) \overline{g(t)} dt.$$

Zum Beweis hat man nur nachzuprüfen, dass

$$\langle w_{a,n} | w_{a,m} \rangle = \frac{1}{a} \int_0^a e^{2\pi i n(t/a)} e^{-2\pi i m(t/a)} dt = \begin{cases} 0 & \text{falls } n \neq m, \\ 1 & \text{falls } n = m. \end{cases}$$

Die Idee der Fourier-Theorie ist es, a -periodische Funktionen $f : \mathbb{R} \rightarrow \mathbb{C}$ möglichst allgemeiner Natur als Linearkombinationen der Basisfunktionen darzustellen, also in der Form

$$f(t) \simeq \sum_{n \in \mathbb{Z}} c[n] w_{a,n}(t)$$

oder durch endlich Abschnitte, sog. *trigonometrische Polynome*

$$f(t) \approx \sum_{n=-N}^N c[n] w_{a,n}(t)$$

zu approximieren.

Die Fragen, die sich stellen sind diese:

- Wenn eine solche Darstellung möglich ist bzw. eine Approximation so “gut” wie möglich ist, was sind dann die Koeffizienten $c[n]$?
- In welchem Sinn, wenn überhaupt, konvergiert eine solche Reihenentwicklung?
- Und selbst wenn die Reihenentwicklung konvergiert, gegen welche Funktion konvergiert sie dann?

Diese Fragen haben die Mathematiker in der Nachfolge von Joseph FOURIER¹ seit mehr als 200 Jahren (Vortrag vor der *Académie des Sciences* in Paris am 21. Dezember 1807, Hauptwerk *Théorie analytique de la chaleur*, [Fou22]) intensiv beschäftigt. Die Antworten und die Methoden, sie zu erhalten, sind alles andere als einfach! Was sich daraus entwickelt hat, gehört als *Harmonische Analyse* zum Kernbestand der Reinen und der Angewandten Mathematik. Einen Eindruck davon, welche Art von physikalischen Problemen (Wärmeleitung) Joseph Fourier mit welchen Methoden behandelt hat, gibt der letzte Abschnitt dieses Kapitels.

Die ONS-Eigenschaft besagt, dass es für die “optimalen” Koeffizienten $c[n]$ nur eine vernünftige Wahl gibt:

$$c_f[n] = \langle f | w_{a,n} \rangle = \frac{1}{a} \int_0^a f(t) e^{-2\pi i n(t/a)} dt,$$

sofern denn das Integral überhaupt existiert. Dies sind die *Fourierkoeffizienten* der Funktion der Funktion $f(t)$. Häufig verwendet man auch die entsprechenden Darstellungen, die sich aus der Aufspaltung der komplexen Exponent in Real- und Imaginärteil ergeben, also mit den mit den $\cos(2\pi n(t/a))$ und $\sin(2\pi n(t/a))$ als Basisfunktionen.

¹ Jean Baptiste Joseph FOURIER (1768–1830), französischer Mathematiker und Physiker

Definition 3.2 (Formale Fourierreihe).

Für $f \in \mathcal{L}^1[0, a]$ heisst die Reihe

$$\sum_{n \in \mathbb{Z}} c_f[n] w_{a,n}(t)$$

mit den Koeffizienten

$$c_f[n] = \langle f | w_{a,n} \rangle = \frac{1}{a} \int_0^a f(t) e^{-2\pi i n(t/a)} dt$$

die formale Fourierreihe von f .

Das große Problem ist also: wenn man eine a -periodische Funktion $f \in \mathcal{L}^1[0, a]$ hat, in welchem Sinne konvergiert deren formale Fourierreihe, oder genauer gesagt: in welchem Sinne konvergiert die Folge der Partialsummen

$$S_N^f(t) = \sum_{n=-N}^{+N} \langle f | w_{a,n} \rangle w_{a,n}(t) \quad (N \geq 0)$$

bei denen es sich um sehr “glatte” (i.e., unendlich oft differenzierbare Funktionen) handelt, für $N \rightarrow \infty$? Der folgende Abschnitt gibt einen visuellen Eindruck von den Problemen, mit denen man es zu tun hat.

3.2 Approximation, Gibbs-Wilbraham-Phänomen

Hier vorerst nur der Hinweis auf dieses Phänomen: approximiert man eine Funktion $f \in \mathcal{L}^1[0, a]$ durch die Partialsummen ihrer Fourierreihe

$$S_N^f(t) = \sum_{n=-N}^{+N} \langle f | w_{a,n} \rangle w_{a,n}(t),$$

so stellt man fest, dass bei wachsendem N die Funktion $f(t)$ überall dort, wo $f(t)$ “glatt”, also stetig differenzierbar ist, immer besser approximiert wird, nicht jedoch an Unstetigkeitsstellen. Grafisch zeigt sich dort ein “Überschwingen”, das mit wachsendem N immer hochfrequenter wird, aber dessen Amplitude nicht kleiner wird! Dies zeigt drastisch, dass die Fourier-Approximation an Unstetigkeitsstellen der darzustellenden Funktion ein grundsätzliches Problem hat! Das hat etwas damit zu tun, dass die Basisfunktionen $w_n(t)$ zwar im Frequenzbereich scharf lokalisiert sind, aber im Orts-/Zeitbereich überhaupt nicht.

■ **Beispiel 3.1** Sei $\mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]}(t)$ die Indikatorfunktion des Intervalles $[-\frac{1}{2}, \frac{1}{2}]$, die man sich im Rahmen der Fourierreihen zu einer 2-periodischen Funktion fortgesetzt denkt. Die Fourierkoeffizienten lassen sich ganz einfach berechnen:

$$\int_{-1}^1 \mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]}(t) e^{-i\pi(2n-1)t} dt = \int_{-1/2}^{1/2} e^{-i\pi(2n-1)t} dt = \frac{2}{\pi} \frac{(-1)^{1+n}}{(2n-1)}$$

und

$$\int_{-1}^1 \mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]}(t) e^{-i\pi(2n)t} dt = \int_{-1/2}^{1/2} e^{-i\pi(2n)t} dt = 0.$$

Man erwartet also

$$f(t) = \frac{1}{2} + \frac{2}{\pi} \cos(\pi t) - \frac{2}{\pi} \frac{\cos(3\pi t)}{3} + \frac{2}{\pi} \frac{\cos(5\pi t)}{5} - \frac{2}{\pi} \frac{\cos(7\pi t)}{7} + \frac{2}{\pi} \frac{\cos(9\pi t)}{9} \mp \dots$$

Setzt man nun

$$S_N(t) = \frac{1}{2} + \sum_{n=1}^N (-1)^{n-1} \frac{2 \cos((2n-1)\pi t)}{2n-1}$$

so erhält man für $N = 5$ folgendes Bild:

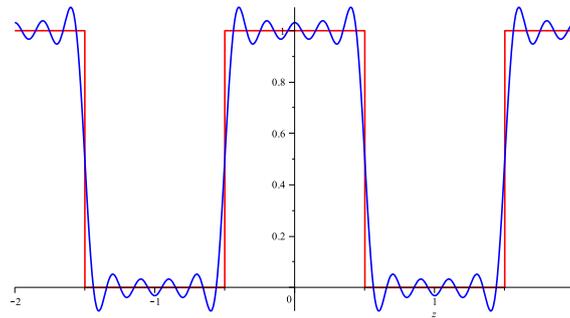


Abbildung 3.1: Gibbs-Wilbraham-Phänomen: $\mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]}(t)$ (rot), $S_5(t)$ (blau)

Das “Überschwingen” in der Nähe der Unstetigkeitsstellen von $\mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]}(t)$, dessen Amplitude durch Erhöhen der Approximationsparameters N keineswegs zum Verschwinden gebracht werden kann – siehe Abbildung 3.2 – bezeichnet man als Gibbs-Phänomen, oder auch zutreffender als Gibbs-Wilbraham-Phänomen². Man kann übrigens genau angeben, um wieviel, relativ zur Sprunghöhe einer Unstetigkeit die Approximation immer überschwingt, ganz unabhängig, mit wievielen Glieder man approximiert. Es ist dies

$$\frac{1}{\pi} \int_0^\pi \frac{\sin t}{t} dt - \frac{1}{2} \approx 0,08949$$

also etwa 9% – immerhin! ■

3.3 Konvergenz von Fourierreihen

Das Gibbs-Wilbraham-Phänomen deutet schon an, dass es mit der Konvergenz eine schwierige Angelegenheit sein könnte. Und das ist es in der Tat! Eine gründliche

² Josiah Willard GIBBS (1839–1903), amerikanischer Physiker, insbesondere für seine Arbeiten zur Thermodynamik bekannt; Henry WILBRAHAM (1825–1883), englischer Mathematiker. Wilbraham beschrieb das Phänomen bereits im Jahr 1848, was weitgehend unbeachtet blieb. Die Arbeiten von Gibbs, nach dem dieses Phänomen benannt wurde, stammen aus dem Jahr 1898, also 50 Jahre später!

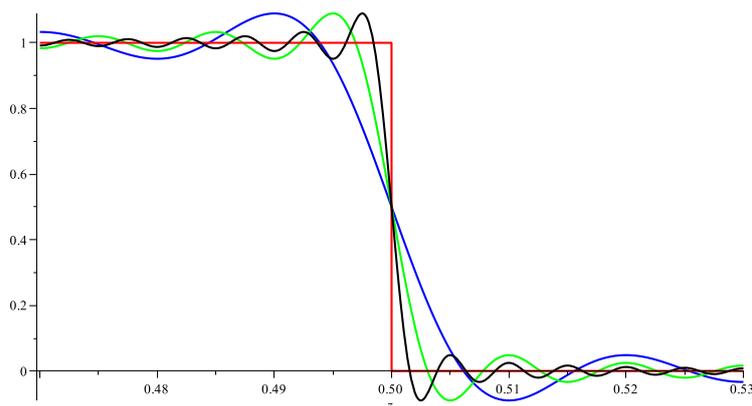


Abbildung 3.2: Approximation der Unstetigkeit von $\mathbf{1}_{[-\frac{1}{2}, \frac{1}{2}]}(t)$ mit S_{50} (blau), S_{100} (grün) und S_{200} (schwarz).

Behandlung dieser Problematik kann nicht Gegenstand dieser Veranstaltung sein, dafür gibt es ausserdem umfangreiche Literatur. Hier werden nur einige wichtige Resultate (ohne Beweise) zitiert, um einen kleinen Einblick zu geben.

Die erste Aussage ist fundamental, denn sie beinhaltet, dass $\mathcal{L}^2([0, a])$ für jedes endliche Intervall ein Hilbertraum ist und die komplexen Exponentialfunktionen $\{w_{a,n}(t)\}_{n \in \mathbb{Z}}$ eine Hilbertbasis dieses Raums bilden. Ein Beweis dieser Aussage wird im Abschnitt 3.3.3 skizziert.

\mathcal{L}^2 -Konvergenz (die gute Nachricht)

- Für jedes $f \in \mathcal{L}^2([0, a])$ gilt $S_N^f \rightarrow f$ im Sinne der \mathcal{L}^2 -Norm:

$$\lim_{N \rightarrow \infty} \int_0^a |S_N^f(t) - f(t)|^2 dt = 0.$$

Ein klassisches, positives Resultat ist das folgende Theorem (das man auch konkret an dem Beispiel zum Gibbs-Phänomen nachvollziehen kann):

DIRICHLETS Theorem (1829, eine gute Nachricht)

- Für *stückweise stetig differenzierbare* Funktionen $f \in \mathcal{L}^1([0, a])$ gilt für alle $t \in [0, a)$ punktweise Konvergenz:

$$\lim_{N \rightarrow \infty} S_N^f(t) \rightarrow \frac{\lim_{\tau \downarrow t} f(\tau) + \lim_{\tau \uparrow t} f(\tau)}{2}.$$

Für alle $t \in [0, a)$, wo f stetig ist, gilt also

$$\lim_{N \rightarrow \infty} S_N^f(t) = f(t).$$

Mehr zu Dirichlets Theorem (aber kein Beweis) im Abschnitt 3.3.1.

Nun ein erstes negatives Resultat, das Hoffnungen auf eine leicht überschaubare Regelung der Konvergenzproblematik dämpft.

DU BOIS-REYMONDSs Beispiel (1873, eine schlechte Nachricht)

- Es gibt eine *stetige* 1-periodische Funktion $f : \mathbb{R} \rightarrow \mathbb{C}$, bei der die Folge $\left(S_N^f(t)\right)_{N \rightarrow \infty}$ für $t = 0$ nicht konvergiert.

Man kann das bedauerliche Problem der Nicht-Konvergenz bei stetigen Funktionen mit einem “Summationstrick” (Stichwort: Césaro-Mittelwerte) umgehen, wie es im Theorem 3.4 von Fejér (1900) gemacht wird.

Weitere klassische positive Konvergenzresultate für Funktionen, die etwas bessere Eigenschaften als nur die Stetigkeit haben, wurden bspw. von JORDAN (1881) für *Funktionen mit beschränkter Variation* und DINI (1878) für Funktionen, die einer *Lipschitz-Bedingung* genügen, erzielt. Dabei blieb unklar, was für \mathcal{L}^2 -Funktionen generell gilt, die ja i.a. unstetig sind.

Dann gab es ein Beispiel, das für den Raum $\mathcal{L}^1([0, 2\pi))$ alle Hoffnungen auf ein “schönes” Konvergenzverhalten noch weit schlimmer zunichte macht, als es das Beispiel von Du Bois-Reymond an einem Punkt zeigt.

KOLMOGOROVs Beispiel (1926, eine sehr schlechte Nachricht)

- Es gibt eine Funktion $f \in \mathcal{L}^1([0, 2\pi))$, bei der für jedes $t \in [0, 2\pi)$ (!!) die Folge $\left(S_N^f(t)\right)_{N \rightarrow \infty}$ divergiert (mit $\limsup_{n \rightarrow \infty} |S_N^f(t)| = \infty$).

Aber die Geschichte endet 160 Jahre nach Fourier dann doch versöhnlich.

CARLESONs Theorem (1966, die überraschende gute Nachricht)

- Für jede Funktion $f \in \mathcal{L}^2([0, 2\pi))$, gilt für (Lebesgue) fast alle $t \in [0, 2\pi)$

$$\lim_{N \rightarrow \infty} S_N^f(t) \rightarrow f(t).$$

Diese Aussage gilt sogar für alle $f \in \mathcal{L}^p([0, 2\pi))$, wenn $1 < p \leq \infty$ ist (Hunt 1968).

Der Fall $p = 1$ (Kolmogorov) ist also eher als “pathologische Ausnahme” zu sehen. Typisch ist das positive Resultat von Carleson, eines der herausragenden Resultate der Analysis im 20. Jahrhundert (und der Beweis ist sehr schwierig!).

3.3.1 Die Sätze von Dirichlet und Fejér

Die wichtigsten positiven Aussagen über die punktweise Konvergenz von Fourierreihen für hinreichend gutartige periodische Funktionen $f(t)$ verdankt man dem deutschen Mathematiker Peter Gustav Lejeune DIRICHLET (1805–1859) und dem Ungarn Leopold FEJÉR (1870–1959).

Unter “hinreichend gutartig” versteht man genauer *stückweise Differenzierbarkeit*, was mit Hilfe des Begriffs der *stückweisen Stetigkeit* definiert wird.

Definition 3.3 (Stückweise Stetigkeit). *Eine auf einem endlichen Intervall I definierte Funktion $f : I \rightarrow \mathbb{C}$ heißt stückweise stetig, wenn sie nur endlich viele Unstetigkeitsstellen hat und an jeder solchen Unstetigkeitsstelle t_0 die Grenzwerte $\lim_{\tau \uparrow t_0} f(\tau)$ (von links) und $\lim_{\tau \downarrow t_0} f(\tau)$ (von rechts) existieren.*

An einer Unstetigkeitsstelle einer stückweise stetigen Funktion $f(t)$ sind also mindestens zwei der drei Werte $f(t_0)$, $\lim_{\tau \uparrow t_0} f(\tau)$ und $\lim_{\tau \downarrow t_0} f(\tau)$ verschieden. Es handelt sich also um “Sprung-Unstetigkeiten”, die Differenz $\lim_{\tau \downarrow t_0} f(\tau) - \lim_{\tau \uparrow t_0} f(\tau)$ gibt die “Sprunghöhe” an der Stelle t_0 an.

R Ein kurzer Hinweis: eine stückweise stetige Funktionen $f(t)$ auf einem endlichen Intervall I ist dort beschränkt, die Integrale über die Teilintervalle, in denen $f(t)$ stetig ist, existieren und sind endlich, das gilt also auch für die Funktion $f(t)$ insgesamt, d.h. also gilt auch $\int_I |f(t)| dt < \infty$ und $\int_I |f(t)|^2 dt < \infty$. Für stückweise differenzierbare a -periodische Funktionen, die im Zusammenhang mit Fourierreihen interessieren, bedeutet das, das es nur auf das Verhalten der Funktion im Intervall $[0, a]$ ankommt. Daraus ergibt sich: stückweise differenzierbare a -periodische Funktionen gehören zu $\mathcal{L}^\infty([0, a])$ (Beschränktheit), $\mathcal{L}^1([0, a])$ (Integrierbarkeit) und $\mathcal{L}^2([0, a])$ (Quadratintegrierbarkeit).

Definition 3.4 (Stückweise Differenzierbarkeit).

1. *Eine auf einem endlichen Intervall I definierte Funktion $f : I \rightarrow \mathbb{C}$ heißt stückweise differenzierbar, wenn sie und ihre Ableitung auf I stückweise stetig sind.^a*
2. *$f(t)$ ist auf einem unendlichen Intervall stückweise differenzierbar, wenn $f(t)$ auf jedem endlichen Teilintervall stückweise differenzierbar ist*

^a Die Ableitung $f'(t)$ ist an endlich vielen Stellen möglicherweise nicht definiert, aber rechts- und linksseitige Ableitungen existieren.

Für eine periodische, auf \mathbb{R} stückweise differenzierbare Funktion $f(t)$ konvergieren die Partialsummen $S_N^f(t)$ für $N \rightarrow \infty$ tatsächlich für jedes $t \in \mathbb{R}$, an den Stetigkeitsstellen gegen den “richtigen” Wert $f(t)$, an den Unstetigkeitsstellen gegen den Mittelwert aus links- und rechtsseitigem Grenzwert $(\lim_{\tau \uparrow t} f(\tau) + \lim_{\tau \downarrow t} f(\tau)) / 2$.

Theorem 3.2 — Dirichlet.

Ist $f : \mathbb{R} \rightarrow \mathbb{C}$ eine a -periodische Funktion und ist f stückweise differenzierbar auf \mathbb{R} , so gilt für jedes $t \in \mathbb{R}$ (punktweise Konvergenz!)

$$S_N^f(t) = \sum_{n=-N}^N \langle f | w_{a,n} \rangle w_{a,n}(t) \rightarrow_{N \rightarrow \infty} \tilde{f}(t) \quad \text{für alle } t \in \mathbb{R},$$

wobei $\tilde{f}(t) = (\lim_{\tau \uparrow t} f(\tau) + \lim_{\tau \downarrow t} f(\tau)) / 2$.

Insbesondere konvergieren für eine *stetige* a -periodische und stückweise differenzierbare Funktion die Partialsummen $S_N^f(t)$ gegen eine *stetige* Funktion (nämlich $f(t)$ selbst), während für eine *unstetige* a -periodische und stückweise differenzierbare Funktion $f(t)$ die Partialsummen $S_N^f(t)$ gegen eine *unstetige* Funktion konvergieren. Für den stetigen Fall gilt sogar:

Theorem 3.3 — Gleichmässige punktweise Konvergenz.

Ist $f : \mathbb{R} \rightarrow \mathbb{C}$ eine a -periodische, stückweise differenzierbare und stetige Funktion \mathbb{R} , so gilt gleichmässige Konvergenz:

$$\|S_N^f(t) - f(t)\|_\infty \rightarrow_{N \rightarrow \infty} 0.$$

Das folgt aus der Tatsache, dass wegen $f \in \mathcal{L}^1([0, a])$

$$|S_N^f(t)| = \left| \sum_{n=-N}^N c_f[n] e^{2\pi i n t} \right| \leq \sum_{n=-N}^N |c_f[n]| \leq \sum_{n=-\infty}^{\infty} |c_f[n]| < \infty$$

ist. Daraus ergibt sich, dass die stetigen (und unendlich oft differenzierbaren) Partialsummen $S_N^f(t)$ *gleichmässig* konvergieren.

Als weitere Konsequenz erkennt man, dass zumindest in diesem Fall die Folge der Fourierkoeffizienten $(c_f[n])_{n \in \mathbb{Z}}$ die Funktion $f(t)$ eindeutig bestimmt.

Theorem 3.4 — Fejér.

Ist $f : \mathbb{R} \rightarrow \mathbb{C}$ eine a -periodische Funktion und ist f stetig auf \mathbb{R} , so gilt gleichmässige Konvergenz:

$$\|\sigma_N^f(t) - f(t)\|_\infty \rightarrow_{N \rightarrow \infty} 0,$$

wobei $\sigma_N^f(t) = (1/N) \sum_{k=0}^{N-1} S_k^f(t)$ der Mittelwert der ersten N Partialsummen ist.

Technische Hilfsmittel zum Beweis der Theorem von Dirichlet und Féjer sind sogenannte *Kerne*, das sind durch einen Parameter k indizierte Folgen von Funktionen, die aus einer Basisfunktion durch Skalierung hervorgehen, wobei sich bei wachsendem k die Funktionen immer mehr in der Umgebung von $t = 0$ konzentrieren – sie die Abbildungen 3.3 und 3.4. Übrigens: wie man auf solche Funktionen kommt erkennt man, wenn man sich deren Fouriertransformierte anschaut!

Definition 3.5 (Dirichlet- und Féjer-Kern).

Für $a > 0$ bezeichnet man

– als Dirichlet-Kern die Folge der Funktionen

$$D_{a,k}(t) = \sum_{n=-k}^k w_{a,n}(t) = \sum_{n=-k}^k e^{2\pi i n t/a} \quad (k \in \mathbb{N});$$

– als Fejér-Kern die Folge der Funktionen

$$F_k(t) = \frac{1}{k} \sum_{m=0}^{k-1} D_{a,m}(t) \quad (k \in \mathbb{N}).$$

Eine einfache Übung in der Summation endlicher geometrischer Reihen ergibt eine handliche Darstellung dieser Funktionen.

Theorem 3.5 Für $a > 0, k \in \mathbb{N}$ gilt

- für den Dirichlet-Kern:

$$D_{a,k}(t) = \frac{\sin((2k+1)\pi t/a)}{\sin(\pi t/a)};$$

- für den Fejér-Kern:

$$F_{k,a}(t) = \frac{1}{k} \left(\frac{\sin(k\pi t/a)}{\sin(\pi t/a)} \right)^2;$$

- für “jede” a -periodische Funktion f :

$$S_k^f(t) = \frac{1}{a} \int_0^a f(s) D_{k,a}(t-s) ds = (D_{k,a} \star_a f)(t),$$

$$\sigma_k^f(t) = \frac{1}{a} \int_0^a f(s) F_{k,a}(t-s) ds = (F_{k,a} \star_a f)(t),$$

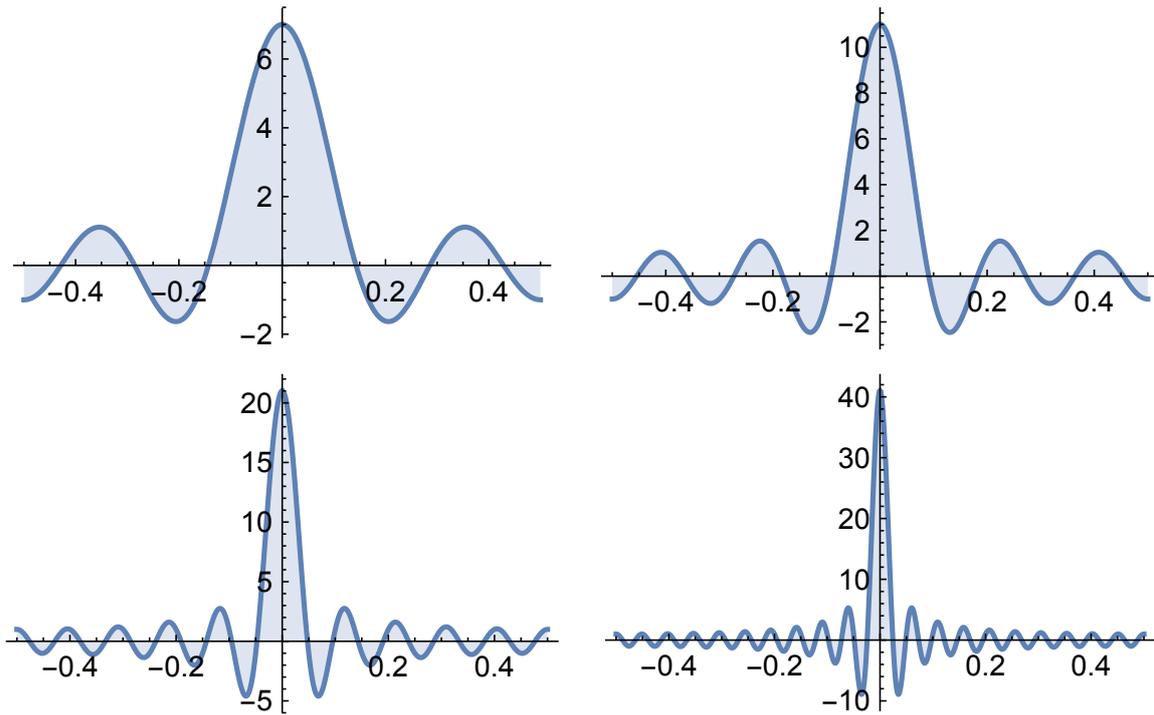
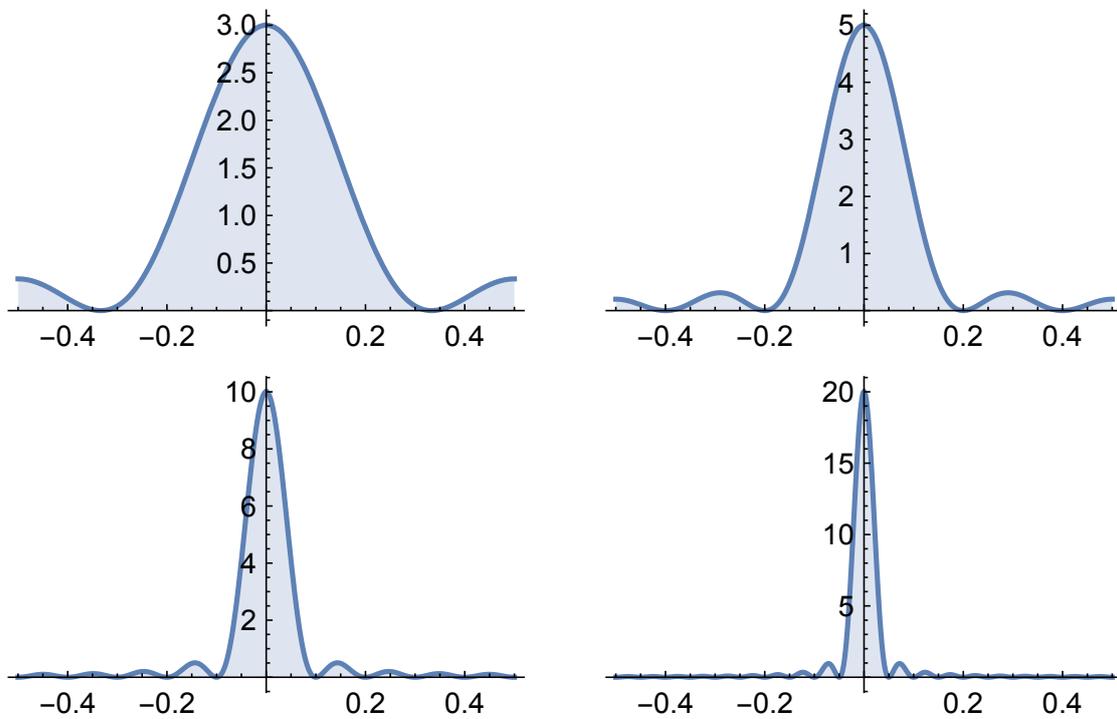
wobei die Integrale suggestiv als Faltungsoperationen für a -periodische Funktionen geschrieben werden.

Die Abbildung 3.3 bzw. 3.4 stellen die Dirichlet-Kerne $D_{k,1}(t)$ bzw. die Fejér-Kerne $F_{k,1}(t)$ für $k = 3, 5, 10, 20$ dar. Mit steigendem k “konzentrieren” sich diese 1-periodischen und unendlich-oft differenzierbaren Funktionen immer näher bei $t = 0$. Daraus ergibt sich zumindest qualitativ, dass für festes $t \in \mathbb{R}$ die Faltungen $(D_{k,1} \star f)(t)$ bzw. $(F_{k,1} \star f)(t)$ nur vom Verhalten der Funktion $f(\tau)$ in einer immer enger werdenden Umgebung von $\tau = t$ abhängen, im Limes für $k \rightarrow \infty$ also nur noch von dem Grenzwertverhalten von $\lim_{\tau \uparrow t} f(\tau)$ und $\lim_{\tau \downarrow t} f(\tau)$.

Die Sätze von Dirichlet und Fejér lassen sich also auch so formulieren:

Dirichlet	$\lim_{k \rightarrow \infty} \frac{1}{a} \int_0^a f(s) D_k(t-s) ds = \lim_{k \rightarrow \infty} (D_{k,a} \star_a f)(t) = \tilde{f}(t)$	punktweise
Fejér	$\lim_{k \rightarrow \infty} \frac{1}{a} \int_0^a f(s) F_k(t-s) ds = \lim_{k \rightarrow \infty} (F_{k,a} \star_a f)(t) = f(t)$	gleichmässig

Die grafische Darstellungen lassen (zu Recht) vermuten, dass die Folgen $(D_{k,a}(t))_{k \geq 0}$ bzw. $(F_{k,a}(t))_{k \geq 0}$ nicht in irgendeinem vernünftigen Sinn gegen eine Funktion konvergieren. Aber man kann spekulieren, dass man den Objekten

Abbildung 3.3: Dirichlet-Kerne für $a = 1$ und $k = 3, 5, 10, 20$.Abbildung 3.4: Fejér-Kerne für $a = 1$ und $k = 3, 5, 10, 20$.

$\Delta_a(t) = \lim_{k \rightarrow \infty} (D_{k,a}(t))_{k \geq 0}$ bzw. $\Phi_a(t) = \lim_{k \rightarrow \infty} (F_{k,a}(t))_{k \geq 0}$ doch einen Sinn geben kann, der die Aussagen $\Delta_a \star_a f(t) = \tilde{f}(t)$ bzw. $\Phi_a \star_a f(t) = f(t)$ rechtfertigt. Insbesondere sollten also hinreichend gutartige (z.B. stetig-differenzierbare) a -periodische Funktionen $f(t)$ durch Faltung mit $\Delta_a(t)$ bzw. $\Phi_a(t)$ in sich selbst transformiert werden, d.h., $\Delta_a(t)$ bzw. $\Phi_a(t)$ wären neutrale Elemente (oder *Eins-Elemente*) bezüglich der Faltungsoperation \star_a . Die Folgen $(D_{k,a}(t))_{k \geq 0}$ ($F_{k,a}(t))_{k \geq 0}$ approximieren also das Verhalten von solchen "virtuelle" einselementen und werden deshalb auch als *Approximations der Eins* bezeichnet – siehe den folgenden Abschnitt.

3.3.2 Summationskerne und Konvergenzsätze

Definition 3.6 (Summationskern).

$I = (-a, a) \subseteq \mathbb{R}$ sei ein endliches oder unendliches Intervall.

Eine Familie von auf I definierten Funktionen $\{K_r(t)\}_{r>0}$ heisst Summationskern auf I , wenn gilt:

1. $\forall r : \int_I K_r(t) dt = 1$
2. $\exists M > 0 \forall r > 0 : \int_I |K_r(t)| dt < M$
3. $\forall 0 < \delta < a : \lim_{r \downarrow 0} \int_{\delta < |t| < a} |K_r(t)| dt = 0$

Beispiele für Summationskerne sind:

1. Box-Funktionen ($0 < r < 1, a = 1$)

$$K_r(t) = \frac{1}{2r} \mathbf{1}_{[-r,r]}(t)$$

2. Hat-Funktionen ($0 < r < 1, a = 1$)

$$K_r(t) = \frac{1}{r} \left(1 - \frac{|t|}{r}\right) \mathbf{1}_{[-r,r]}(t)$$

3. Gauss-Funktionen ($0 < r < 1, a = \infty$)

$$K_r(t) = \frac{1}{\tau} \cdot e^{-\pi(t/\tau)^2}$$

4. Fejér ($0 < r, a = 1$)

$$K_r(t) = r \cdot \left(\frac{\sin(\pi t/r)}{\sin(\pi t)}\right)^2 \cdot \chi_{[-1/2, 1/2]}(t)$$

Diese Kerne sind exemplarisch in Abbildung 3.5 für vier Werte von r dargestellt. Man erkennt, was man auch aus der Definition ablesen kann: für $r \downarrow 0$ konzentrieren sich die Funktionsgraphen um den Nullpunkt. Das hat zur Folge, dass die Faltungen $(K_r \star f)(t)$ für $r \downarrow 0$ (pauschal gesagt) gegen $f(t)$ konvergieren, wie es das Theorem 3.6 zum Ausdruck bringt, also $\lim_{r \downarrow 0} (K_r \star f)(t) = f(t)$, sodass wiederum, wie im vorigen Abschnitt, die Folge der Funktionen $(K_r(t))_{r>0}$ im Limes $r \downarrow 0$ die

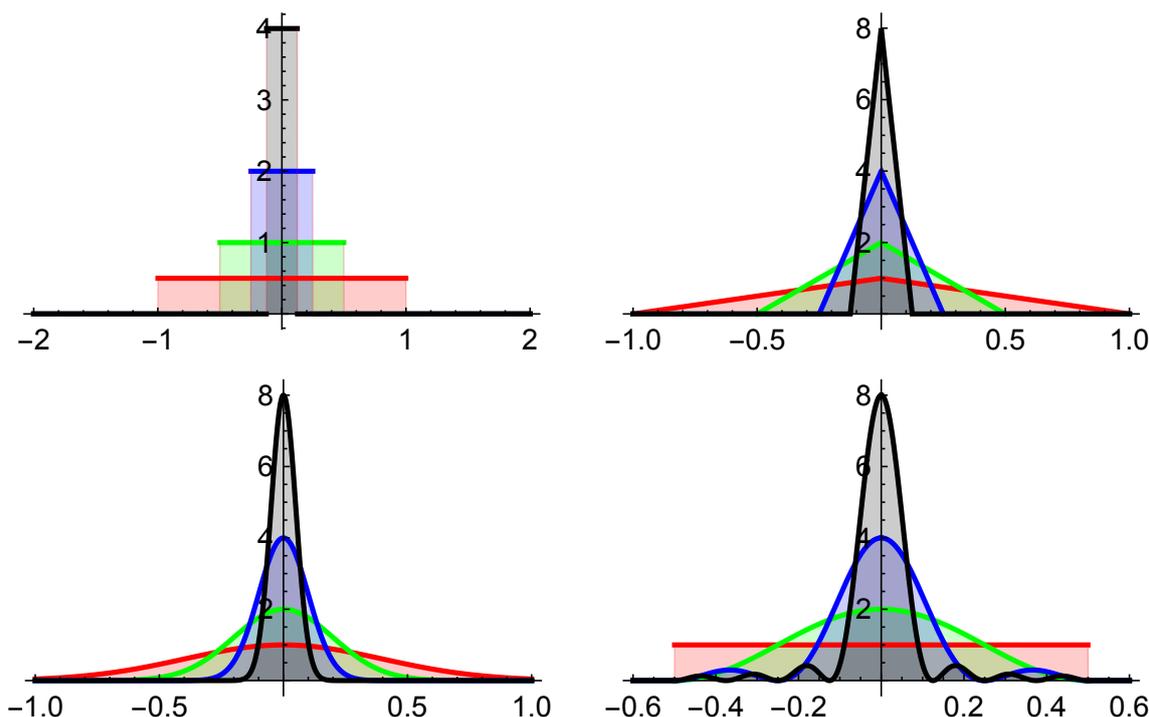


Abbildung 3.5: Summationskerne für Box, Hat, Gauss, Fejér mit $r = 1$ (rot), $r = 1/2$ (grün), $r = 1/4$ (blau), $r = 1/8$ (schwarz).

Funktion $f(t)$ reproduziert, also als Einselement bezüglich der Faltung operiert. Summationskerne werden daher auch als *Approximationen der Eins* (approximation of unity) bezeichnet. Sie sind ein halbwegs elementares Handwerkszeug, um die Einführung von verallgemeinerten Funktionen (*Distributionen*) mit ihrer tieferen Theorie zu umgehen. Man kann intuitiv sagen, dass der “Limes für $r \downarrow 0$ ” eines Summationskerns das Diracsche δ -Funktional ist, vgl. Abschnitt 4.8, insbesondere Satz 4.7 mit der Aussage $\delta \star f = f$.

Theorem 3.6 — Summationskerne und Konvergenz.

Ist $\{K_r(t)\}_{r>0}$ ein Summationskern auf \mathbb{R} , so gilt die Konvergenzaussage

$$\lim_{r \downarrow 0} (K_r \star f)(x) = \lim_{r \downarrow 0} \int_{\mathbb{R}} f(t) K_r(x-t) dt = f(x)$$

in den folgenden Fällen:

- für punktweise Konvergenz: $f \in L^\infty(\mathbb{R})$ und stetig in $t = x$;
- für gleichmässige (L^∞) Konvergenz : $f \in L^\infty(\mathbb{R})$ und gleichm. stetig in \mathbb{R} ;
- für $\|\cdot\|_1$ -Konvergenz : $f \in \mathcal{L}^1(\mathbb{R})$;
- für $\|\cdot\|_2$ -Konvergenz : $f \in \mathcal{L}^2(\mathbb{R})$.

Summationskerne sind also ein technisches Hilfsmittel um Konvergenzaussagen zu beweisen, siehe hierzu auch den Abschnitt 4.4, bei dem es um die Definition der

Fourier-Rücktransformation geht. Auch strukturelle Aussagen lassen sich mittels Summationskernen gewinnen:

Folgerung 3.1. *Die Menge C_c^0 der stetigen Funktionen mit kompakten Träger liegt dicht, und zwar sowohl in $\mathcal{L}^1(\mathbb{R})$, als auch in $\mathcal{L}^2(\mathbb{R})$.*

3.3.3 Eine Hilbert-Basis für $\mathcal{L}^2([0, 1])$

Die Aussage, dass für jedes *endliche* Intervall $I \subset \mathbb{R}$ der Raum $\mathcal{L}^2(I)$ ein Hilbertraum ist, soll nun damit abgeschlossen werden, dass gezeigt wird: das ONS der komplexen Exponentialfunktionen ist vollständig. Es genügt, dies für das Intervall $I = [0, 1)$ zu zeigen. Dann ist

$$\mathcal{E} = \{e^n(t); n \in \mathbb{Z}\} \quad \text{mit} \quad e^n(t) = e^{2\pi i n t}$$

ein ONS in $\mathcal{L}^2([0, 1])$. Für $f \in \mathcal{L}^2([0, 1])$ sind die Fourierkoeffizienten definiert durch

$$c_f[n] = \langle f | e^n \rangle = \int_0^1 f(t) e^{-2\pi i n t} dt \quad (n \in \mathbb{Z}),$$

und

$$f(t) \approx \sum_{n \in \mathbb{Z}} c_f[n] e^n(t)$$

ist die formale Fourierreihe von $f(t)$. Das Ziel ist der Nachweis, dass für $N \rightarrow \infty$ die Partialsummen

$$S_N^f(t) = \sum_{-N \leq n \leq N} c_f[n] e^n(t)$$

in der \mathcal{L}^2 -Norm gegen $f(t)$ konvergieren. Wegen der orthogonalen Zerlegung

$$f(t) = S_N^f(t) + (f(t) - S_N^f(t))$$

und Bessels Formel gilt

$$\|f\|^2 = \int_0^1 |f(t)|^2 dt = \sum_{-N \leq n \leq N} |c_f[n]|^2 + \int_0^1 |f(t) - S_N^f(t)|^2 dt$$

und daher genügt es zu zeigen (vgl. Theorem 2.7), dass

$$\|f\|^2 = \sum_{n \in \mathbb{Z}} |c_f[n]|^2.$$

Sei zunächst $f(t)$ eine stetig differenzierbare 1-periodische Funktion, also $f \in \mathcal{C}^1$. Nun definiert man eine weitere stetig differenzierbare 1-periodische Funktion

$$g(x) = \int_0^1 f(x+t) \overline{f(t)} dt.$$

Diese Funktion ist ebenfalls stetig differenzierbar und 1-periodisch stetig und es ist

$$g(0) = \int_0^1 f(t) \overline{f(t)} dt = \|f\|^2.$$

Die Fourierkoeffizienten von $g(x)$ lassen sich einfach berechnen:

$$\begin{aligned} c_g[n] &= \langle g | e^n \rangle = \int_0^1 \int_0^1 f(x+t) \overline{f(t)} e^{-2\pi i n x} dt dx \\ &= \int_0^1 \int_0^1 f(x+t) e^{-2\pi i n(x+t)} \overline{f(x)} e^{2\pi i n t} dt dx \\ &= \int_0^1 f(s) e^{2\pi i n s} ds \int_0^1 \overline{f(t)} e^{-2\pi i n t} dt \\ &= |c_f[n]|^2. \end{aligned}$$

Daraus ergibt sich

$$\left\| \sum_n c_g[n] e^n(t) \right\|_\infty \leq \sum_n |c_g[n]| \leq \sum_n |c_f[n]|^2 < \infty,$$

und das impliziert, dass die formale Fourierreihe von $g(t)$ gleichmässig konvergiert und somit als Folge von 1-periodischen stetigen Funktionen gegen eine 1-periodische stetige Funktion $h(t)$ konvergiert. Indem man auch die Ableitung von $g(t)$ und ihrer Fourierreihe betrachtet, findet man heraus, dass $h(t)$ sogar eine stetig differenzierbare 1-periodische Funktion ist.

Nun soll gezeigt werden, dass die beiden Funktionen 1-periodischen und stetig differenzierbaren Funktionen $g(t)$ und $h(t)$ identisch sind. Dies folgt aus der Tatsache, dass ihre Fourierkoeffizienten identisch sind - siehe die Bemerkung nach Theorem 3.3 ! Tatsächlich ist

$$\begin{aligned} c_h[n] &= \int_0^1 h(t) \overline{e^n(t)} dt \\ &= \int_0^1 \left(\sum_{k=-\infty}^{\infty} c_g[k] e^k(t) \right) \overline{e^n(t)} dt \\ &= \sum_{k=-\infty}^{\infty} c_g[k] \int_0^1 e^k(t) \overline{e^n(t)} dt = c_g[n], \end{aligned}$$

wobei die Vertauschung von Integral und unendlicher Summe durch die gleichmässige Konvergenz gerechtfertigt wird.

Es gilt also $g(x) = h(x)$ für alle (!) $x \in \mathbb{R}$, d.h. die formale Fourierreihe von $g(x)$ konvergiert punktweise (und gleichmässig) gegen $g(x)$, man darf also

$$\int_0^1 f(x+t) \overline{f(t)} dt = g(x) = \sum_{n=-\infty}^{\infty} c_g[n] e^n(t)$$

schreiben. Speziell für $x = 0$ ergibt sich, wie erwünscht,

$$\|f\|^2 = \int_0^1 |f(t)|^2 dt = g(0) = \sum_{n=-\infty}^{\infty} c_g[n] = \sum_{n=-\infty}^{\infty} |c_f[n]|^2.$$

Der allgemeine Fall ergibt sich routinemässig aus der Tatsache, dass die 1-periodischen stetig differenzierbaren Funktionen in $\mathcal{L}^2([0, 1])$ dicht liegen, mittels eines \mathcal{L}^2 -Approximationsarguments wie folgt:

Ist $\ell(t) \in \mathcal{L}^2([0, 1])$ und $S_N^\ell(t) = \sum_{n=-N}^N c_\ell[n] e^{in} t$ die bandbegrenzte Fourierreihe von $\ell(t)$, sowie $f(t) \in \mathcal{C}^1([0, 1])$ eine stetig differenzierbare 1-periodische Funktion, wie oben dargestellt, mit bandbegrenzter Fourierreihe $S_N^f(t)$. Dann ist für jedes $N \in \mathbb{N}$

$$\ell(t) - S_N^\ell(t) = \ell(t) - f(t) + f(t) - S_N^f(t) + S_N^f(t) - S_N^\ell(t)$$

und somit

$$\begin{aligned} \|\ell - S_N^\ell\| &\leq \|\ell - f\| + \|f - S_N^f\| + \|S_N^f - S_N^\ell\| \\ &= \underbrace{\|\ell - f\|}_{(1)} + \underbrace{\|f - S_N^f\|}_{(2)} + \underbrace{\|S_N^{f-\ell}\|}_{(3)} \end{aligned}$$

Diese drei Summanden kann man simultan beliebig klein machen, denn:

- (1) Wegen der Dichtheit von \mathcal{C}^1 in $\mathcal{L}^2([0, 1])$ gibt es zu jedem $\ell \in \mathcal{L}^2([0, 1])$ und $\varepsilon > 0$ ein $f \in \mathcal{C}^1$ mit $\|\ell - f\| < \varepsilon$.
- (2) Wegen der Gültigkeit von $\|f - S_N^f\| \rightarrow_{N \rightarrow \infty} 0$ (das zeigt der erste Teil des Beweises) gilt dann $\|f - S_N^f\| < \varepsilon$ für hinreichend große N .
- (3) Wegen der Ungleichung von Bessel gilt $\|S_N^{f-\ell}\| \leq \|f - \ell\| < \varepsilon$.

Damit ist $\|\ell - S_N^\ell\| \rightarrow_{N \rightarrow \infty} 0$ gezeigt. \square

Als wichtige Folgerung ergibt sich nun die Eindeutigkeit der Fourierkoeffizienten.

Theorem 3.7 — Eindeutigkeit der Fourierkoeffizienten.

Sind $f, g \in \mathcal{L}^2([0, 1])$, so gilt

$$f(t) = g(t) \text{ fast überall}^a \Leftrightarrow c_f[k] = c_g[k] \text{ für alle } k \in \mathbb{Z}.$$

^aim Sinne von Lebesgue

Es genügt zu zeigen, dass aus dem Verschwinden aller Fourierkoeffizienten für eine Funktion $f \in \mathcal{L}^2([0, 1])$, also $c_f[k] = 0$ für alle $k \in \mathbb{Z}$, die Aussage $\|f\|^2 = 0$ folgt. Das ist aber eine unmittelbare Folgerung aus der Parseval-Plancherel Beziehung

$$\int_0^1 |f(t)|^2 dt = \lim_{N \rightarrow \infty} \sum_{k=-N}^N |c_f[k]|^2$$

\square

3.4 Kantenerkennung mittels Fourierreihen

Hinweis: Die Anregung zu diesem Abschnitt hat der Artikel *Edge Detection Using Fourier Coefficients* von Shlomo Engelberg [Eng08] gegeben.

Ihrer Natur nach sind Fourierreihen und Fouriertransformation erst einmal keine geeigneten Hilfsmittel, um scharfe transiente Ereignisse, wie z.B. Kanten, in

einem Signal zu detektieren. Im Frequenzbereich geht jegliche Lokalisierung verloren, da die Basisfunktionen (komplexe Exponentialfunktion oder trigonometrische Funktionen) scharf lokalisiert im Frequenzbereich sind, jedoch überhaupt nicht im Orts/Zeitbereich.

Ganz aussichtslos ist die Angelegenheit aber trotzdem nicht! Dazu betrachte man die Situation bei 1-periodischen stückweise differenzierbaren Signalen. Zunächst ein Beispiel.

■ **Beispiel 3.2** Betrachten wir die stückweise differenzierbare Funktion

$$f(t) = \begin{cases} 1 - t & 0 \leq t < 1/6 \\ |t - 0.3| & 1/6 \leq t < 1/2 \\ \sin(20t) & 1/2 \leq t < 3/4 \\ -t^4 & 1/4 \leq t < 1 \end{cases}$$

die bis auf die Sprungstellen bei $t = 1/6, 1/2, 3/4$ und $t = 1$ stetig und, abgesehen von den Sprungstellen, bis auf den “Knick” bei $t = 0.3$ sogar differenzierbar ist.

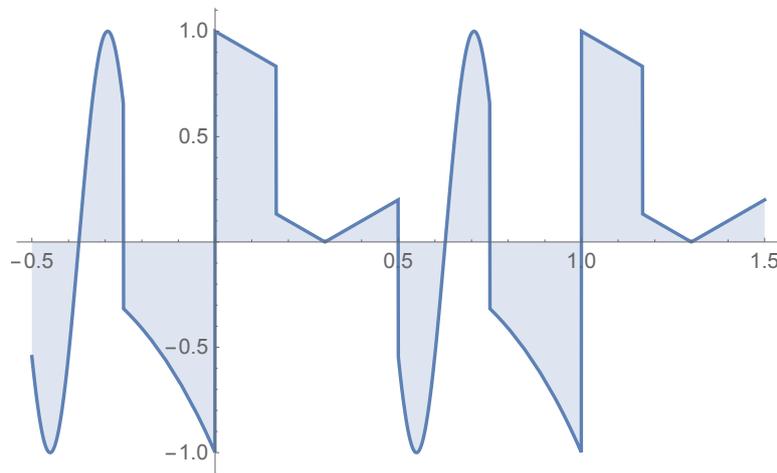


Abbildung 3.6: Die stückweise differenzierbare Funktion $f(t)$ von Beispiel 3.2

Diese Funktion sollte man sich zu einer 1-periodischen Funktion fortgesetzt denken. Die Frage ist: kann man die Sprungstellen von $f(t)$ mit Hilfe der Fourierreihe identifizieren?

Hier sind die ersten 21 Terme der Fourierreihe:

$$\begin{aligned} S_{20}^f(t) = & 0.0251584 + 0.235017 \sin(2\pi t) + 0.377806 \sin(4\pi t) + 0.540989 \sin(6\pi t) \\ & - 0.119918 \sin(8\pi t) + 0.223626 \sin(10\pi t) + 0.0849666 \sin(12\pi t) + 0.095746 \sin(14\pi t) \\ & + 0.015485 \sin(16\pi t) + 0.135425 \sin(18\pi t) + 0.082121 \sin(20\pi t) + 0.117 \cos(2\pi t) \\ & - 0.190523 \cos(4\pi t) + 0.202766 \cos(6\pi t) - 0.0374503 \cos(8\pi t) - 0.189097 \cos(10\pi t) \\ & + 0.056604 \cos(12\pi t) + 0.0514062 \cos(14\pi t) + 0.0266491 \cos(16\pi t) - 0.0515104 \cos(18\pi t) \\ & - 0.00414729 \cos(20\pi t) \end{aligned}$$

Das ist also eine unendlich-oft differenzierbare Funktion, die, wie erwartet, den Funktionsverlauf zwar ganz gut wiedergibt, in der grafischen Darstellung von den Sprungstellen aber nichts ahnen lässt.

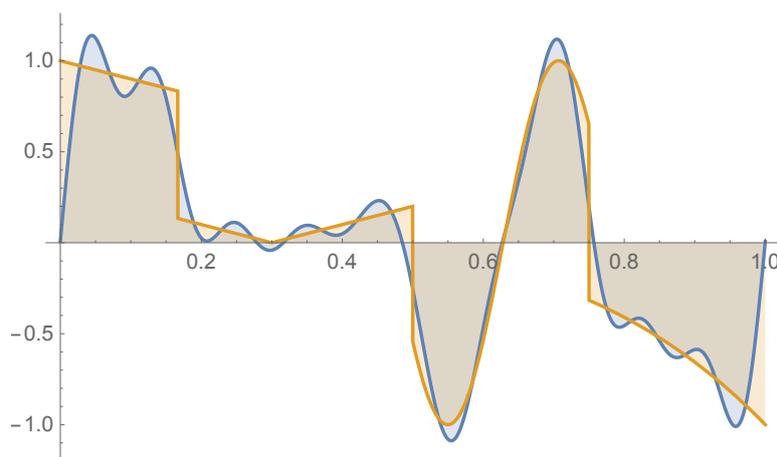


Abbildung 3.7: Die stückweise differenzierbare Funktion $f(t)$ von Beispiel 3.2 und ihre Approximation durch $S_{10}^f(t)$.

Die Information über die Funktion $f(t)$, die hier interessiert, sind die Sprungstellen $0 < t_1 < t_2 < \dots < t_m \leq 1$ und die Sprunghöhen h_1, h_2, \dots, h_m mit $h_j = \lim_{\tau \downarrow t_j} f(\tau) - \lim_{\tau \uparrow t_j} f(\tau)$, im konkreten Fall also

j	1	2	3	4
t_j	$1/6$	$1/2$	$3/4$	1
h_j	-0.7	-0.744	-0.967	2

Man kann nun diese Information aus der Funktion $f(t)$ extrahieren. Dazu definiert man eine “Einheits-Sprungfunktion”

$$k(t) = -1/2 + t \quad \text{für } 0 \leq t < 1,$$

die man sich (wie alle Funktionen in diesem Abschnitt) zu einer 1-periodischen Funktion fortgesetzt denkt. Die Funktion $k(t)$, siehe Abbildung 3.8, ist dann eine Art “Sägezahnfunktion”. Nun definiert man die stückweise differenzierbare Funktion

$$h(t) = \sum_{j=1}^m h_j \cdot k(t - t_j),$$

die Lage und Höhe der Sprungstellen von $f(t)$ codiert. Im konkreten Fall ist dies die Funktion

$$\begin{aligned} h(t) = & -0.967 \left(\left[t - \frac{3}{4} \right] - t + \frac{5}{4} \right) + 0.744 \left(\left[t - \frac{1}{2} \right] - t + 1 \right) \\ & + 0.7 \left(\left[t - \frac{1}{6} \right] - t + \frac{2}{3} \right) - 2 \left([t] - t + \frac{1}{2} \right), \end{aligned}$$

deren Graph in Abbildung 3.9 dargestellt ist.

Wenn man nun diese Sprungstellen-Funktion $h(t)$ von $f(t)$ abzieht, erhält man eine Funktion $g(t) = f(t) - h(t)$, die stückweise differenzierbar ist, aber zudem auch noch *stetig*, da ja alle Sprungstellen aus $f(t)$ extrahiert wurden, siehe Abbildung 3.10. Die Abbildung 3.11 zeigt die Funktionen $f(t), g(t), h(t)$ im Zusammenhang. ■

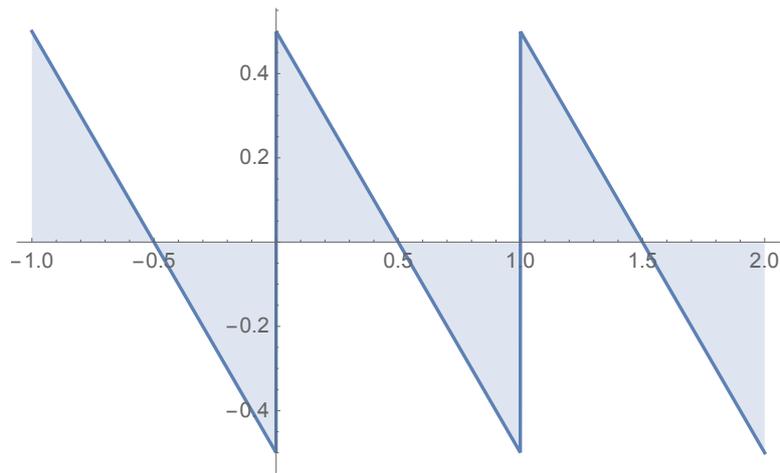


Abbildung 3.8: Die stückweise differenzierbare Sägezahn-Funktion $k(t)$.

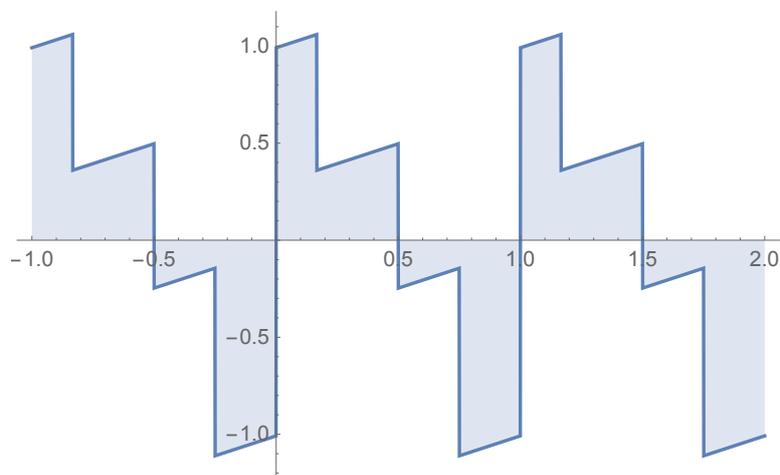


Abbildung 3.9: Die stückweise differenzierbare Funktion $h(t)$ der Sprungstellen von $f(t)$.

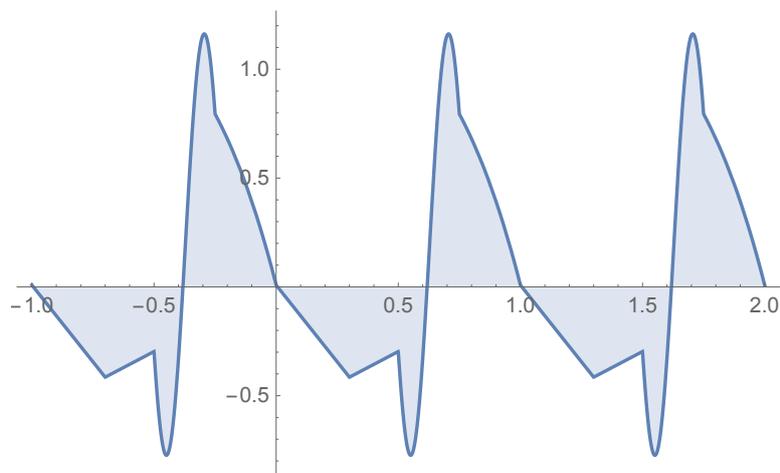


Abbildung 3.10: Die stückweise differenzierbare und stetige Funktion der um die Sprungstellen von $f(t)$ verminderten Funktion $g(t) = f(t) - h(t)$.

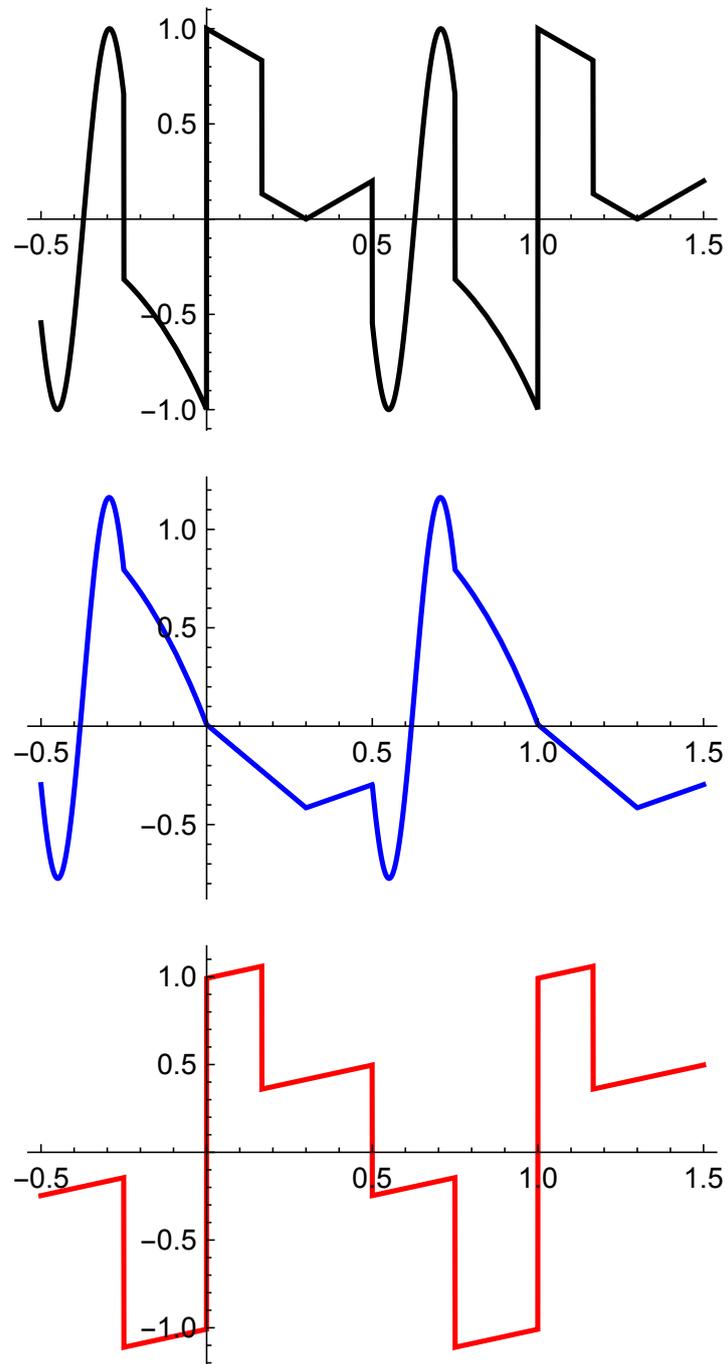


Abbildung 3.11: Die stückweise differenzierbaren Funktionen $f(t)$ (schwarz), $g(t)$ (blau) und $h(t)$ (rot).

Zusammenfassend die allgemeine Aussage: Jede stückweise differenzierbare 1-periodische Funktion $f(t)$ lässt sich zerlegen in

$$f(t) = g(t) + h(t), \quad \text{wobei} \quad \begin{cases} g(t) \text{ stückweise differenzierbar und stetig,} \\ h(t) \text{ stückweise differenzierbar, aber nicht stetig (oder } h \equiv 0), \end{cases}$$

und wo die komplette Information über die Sprungstellen von $f(t)$ (Lagen und Sprunghöhen) in

$$h(t) = \sum_{j=1}^m h_j \cdot k(t - t_j)$$

enthalten ist.

Wozu kann das gut sein, wenn man doch die Sprungstellen einer Funktion garnicht kennt, sondern erst bestimmen möchte? Die Antwort auf diese (sehr berechtigte!) Frage liegt in dem unterschiedlichen Verhalten der Fourier-Koeffizienten von stetigen und nicht-stetigen stückweise differenzierbaren Funktionen. Es gilt nämlich:

Theorem 3.8 — Unstetigkeiten und Fourierkoeffizienten.

Ist $f(t)$ eine 1-periodische stückweise differenzierbare Funktion, sind $c_f[n], n \in \mathbb{Z}$ deren Fourier-Koeffizienten, so gilt

$$\sum_{n \in \mathbb{Z}} |c_f[n]| \quad \text{ist} \quad \begin{cases} < \infty & \text{wenn } f(t) \text{ stetig ist,} \\ = \infty & \text{wenn } f(t) \text{ nicht stetig ist.} \end{cases}$$

So schön dieses Kriterium auf den ersten Blick aussieht, so problematisch ist es auf den zweiten: die Fourier-Koeffizienten einer Funktion lassen sich nur in ganz seltenen Fällen in geschlossener Form angeben, ansonsten kann man sie zwar numerisch einzeln berechnen, aber daraus eine Konvergenzaussage für $\sum_{n \in \mathbb{Z}} |c_f[n]|$ abzuleiten, dürfte äusserst gewagt sein.

Vor dem Beweis zunächst ein Blick auf die Einheitssprungfunktion $k(t)$. Deren Fourierkoeffizienten berechnet man leicht:

$$c_k[n] = \begin{cases} 0 & \text{für } n = 0, \\ \frac{1}{2\pi i n} & \text{für } n \neq 0. \end{cases}$$

Man sieht, dass $\sum_{n \in \mathbb{Z}} |c_k[n]| = \infty$ ist (Stichwort: Divergenz der hamonischen Reihe), wie nach der Aussage der Theorems zu erwarten.

Es ist interessant, sich auch die Summe der Quadrate $\sum_{n \in \mathbb{Z}} |c_k[n]|^2$ anzuschauen. Diese Summe ist endlich (wegen der Zugehörigkeit zu $\mathcal{L}^2([0, 1])$) und wegen der Parseval-Plancherel Beziehung gilt

$$\sum_{n \in \mathbb{Z}} |c_k[n]|^2 = \int_0^1 |k(t)|^2 dt.$$

Das Integral berechnet sich zu $1/12$ und daher hat man ein klassisches Resultat gefunden:

$$\sum_{n \geq 1} \frac{1}{n^2} = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \dots = \frac{\pi^2}{6}.$$

Nun zum Beweis von Theorem 3.8.

- Der Beweis, dass die absolute Summierbarkeit der Fourierkoeffizienten, i.e., $\sum_{n \in \mathbb{Z}} |c_f[n]| < \infty$ die (sogar gleichmässige) Konvergenz gegen eine stetige Funktion (nämlich $f(t)$) zur Folge hat, wurde schon im Zusammenhang von 3.3 skizziert.
- Sei nun $f(t)$ eine *stetige* stückweise differenzierbare 1-periodische Funktion. Dann ist die Ableitung $f'(t)$ eine stückweise stetige 1-periodische Funktion, hat also im Intervall $[0, 1]$ höchstens endlich-viele Unstetigkeitsstellen $0 < t_1 < t_2 < \dots < t_m \leq 1$. Natürlich gehört $f'(t)$ als stückweise stetige Funktion zu $\mathcal{L}^2([0, 1])$ – das wird weiter unten verwendet!

Mit $d[n]$ sollen nun die Fourierkoeffizienten von $f'(t)$ bezeichnet werden. Die kann man auf Grund der Voraussetzungen leicht berechnen. Mittels partieller Integration erhält man

$$\begin{aligned}
 d[n] &= \int_0^1 f'(t) e^{-2\pi i n t} dt \\
 &= \int_0^{t_1} (\dots) + \int_{t_1}^{t_2} (\dots) + \dots + \int_{t_m}^1 (\dots) \\
 &= f(t) e^{-2\pi i n t} dt \Big|_0^{t_1} + \dots + f(t) e^{-2\pi i n t} dt \Big|_{t_m}^1 \\
 &\quad + (2\pi i n) \left(\int_0^{t_1} f(t) e^{-2\pi i n t} dt + \dots + \int_{t_m}^1 f(t) e^{-2\pi i n t} dt \right) \\
 &= (2\pi i n) \left(\int_0^{t_1} f(t) e^{-2\pi i n t} dt + \dots + \int_{t_m}^1 f(t) e^{-2\pi i n t} dt \right) \\
 &= 2\pi i n \cdot c_f[n].
 \end{aligned}$$

Bitte beachten: bei der vorletzten Gleichheit wurde die Stetigkeit von $f(t)$ entscheidend verwendet!

Jetzt kann man mit Hilfe der Ungleichung von Cauchy-Schwarz die Summe $\sum_{n \in \mathbb{Z}} |c_f[n]|$ abschätzen, wobei auch noch die Parseval-Plancherel Beziehung für die Ableitung $f'(t)$ verwendet wird

$$\sum_{n \in \mathbb{Z}} |d_n|^2 = \int_0^1 |f'(t)|^2 dt < \infty.$$

Cauchy-Schwarz besagt

$$\begin{aligned}
 |\langle (1/n)_{n < 0}, (|d[n]|)_{n < 0} \rangle|^2 &\leq \|(1/n)_{n < 0}\|^2 \cdot \|(|d[n]|)_{n < 0}\|^2 \\
 &= \left(\sum_{n \geq 1} \frac{1}{n^2} \right)^2 \cdot \sum_{n < 0} |d[n]|^2 < \infty \\
 |\langle (1/n)_{n > 0}, (|d[n]|)_{n > 0} \rangle|^2 &\leq \|(1/n)_{n > 0}\|^2 \cdot \|(|d[n]|)_{n > 0}^{-1}\|^2 \\
 &= \left(\sum_{n \geq 1} \frac{1}{n^2} \right)^2 \cdot \sum_{n > 0} |d[n]|^2 < \infty.
 \end{aligned}$$

Damit wird

$$\begin{aligned} \sum_{n \in \mathbb{Z}} |c_f[n]| &= \frac{1}{2\pi} \sum_{n < 0} \frac{1}{n} |d[n]| + |c_f[0]| + \frac{1}{2\pi} \sum_{n > 0} \frac{1}{n} |d[n]| \\ &= \frac{1}{2\pi} \langle (1/n)_{n < 0}, (|d[n]|)_{n < 0} \rangle + |c_f[0]| + \frac{1}{2\pi} \langle (1/n)_{n > 0}, (|d[n]|)_{n > 0} \rangle < \infty. \quad \square \end{aligned}$$

So elegant das Kriterium von Theorem 3.8 auch ist, um es brauchbar zu machen, muss man noch etwas tun.

Schauen wir uns die Funktion $k(t)$ noch einmal genauer an. Ihre Fourier-Koeffizienten sind, wie schon erwähnt

$$c_k[n] = \begin{cases} 0 & \text{für } n = 0, \\ \frac{1}{2\pi i n} & \text{für } n \neq 0. \end{cases}$$

Die N -te Partialsumme der Fourierreihe ist

$$S_N^k(t) = \sum_{n=-N}^N c[n] e^{2\pi i n t} = \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \sin(2\pi n t)$$

Plottet man diese Partialsummen, siehe Abbildung 3.12, so findet man das erwartete Verhalten, wie es in Theorem 3.2 beschrieben wird: an allen Stellen $t \notin \mathbb{Z}$ konvergiert $S_N^k(t)$ für $N \rightarrow \infty$ gegen $k(t)$, an allen Stellen $k \in \mathbb{Z}$ ist offensichtlich $S_N^k(t) = 0$.

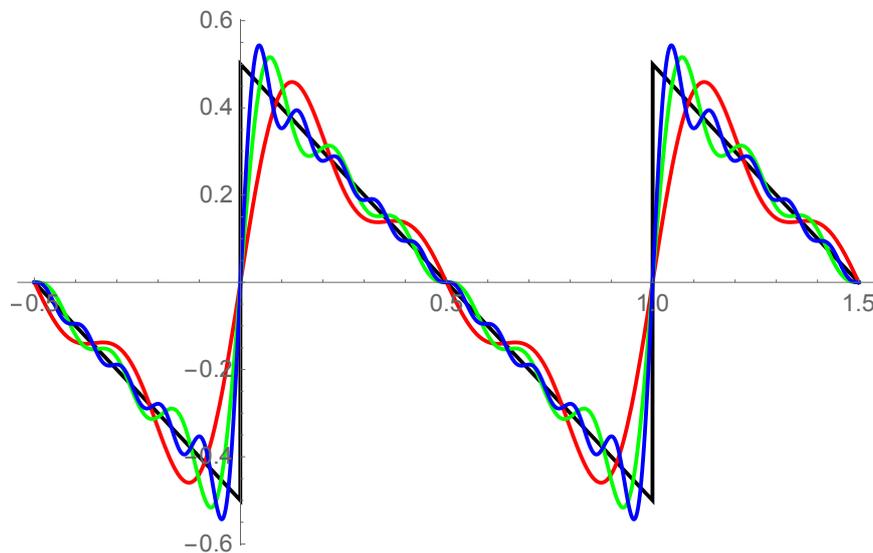


Abbildung 3.12: Die Approximationen $S_N^k(t)$ der Funktion $k(t)$ (schwarz) für $N = 3$ (rot), $N = 6$ (grün) und $N = 10$ (blau).

Nun eine spontane Idee: was wäre, wenn man in dem obigen Ausdruck für $S_N^k(t)$ die Sinusfunktionen durch Cosinusfunktionen ersetzt? Sei also für $N \geq 1$

$$C_N^k(t) = \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \cos(2\pi n t).$$

Das sind zwar nicht mehr die Partialsummen der Fourierreihe für $k(t)$, aber möglicherweise haben diese neuen Partialsummen doch noch etwas mit $f(t)$ zu tun. Das grafische Bild sieht man in den Abbildungen 3.13 und 3.14.

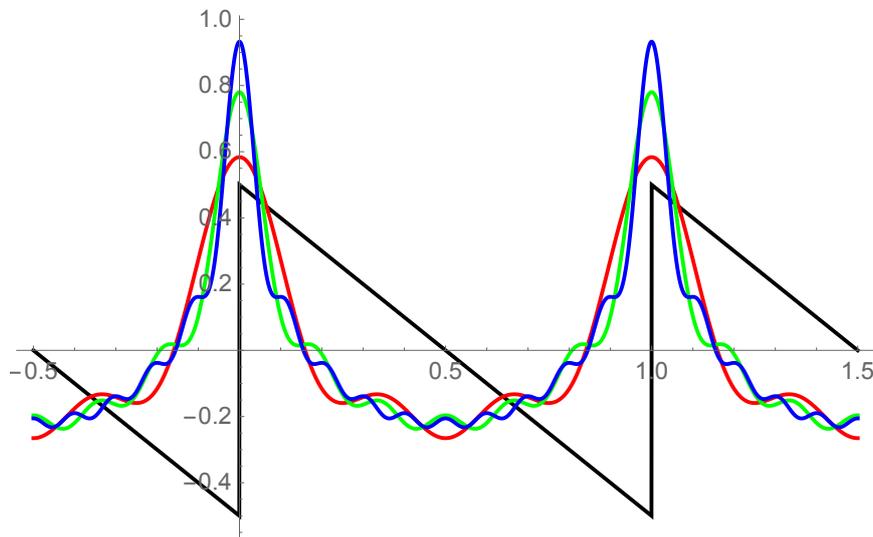


Abbildung 3.13: Die Funktion $k(t)$ (schwarz) und die Funktionen $C_N^k(t)$ für $N = 3$ (rot), $N = 6$ (grün) und $N = 10$ (blau).

Dabei zeigt sich ein ganz anderes Bild als zuvor! Für $t \in \mathbb{Z}$ divergiert $T_N^k(t) \rightarrow \infty$ für $N \rightarrow \infty$, das ist klar, denn dann ist

$$C_N^k(z) = \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \approx \log N.$$

Das Wachstum dieser harmonischen Summe ist zwar langsam, aber sie geht gegen ∞ .

Neben der Divergenz an den ganzzahligen Stellen gewinnt man den Eindruck, dass an jeder nicht-ganzzahligen Stelle, also $t \notin \mathbb{Z}$, die Werte $C_N^k(t)$ für $N \rightarrow \infty$ gegen einen endlichen Wert konvergieren, also insbesondere beschränkt sind. Das ist in der Tat der Fall, denn man kann mit einiger Rechnerei mit dem Dirichlet-Kern zeigen, dass

$$|C_N^k(t)| = \left| \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \cos(2\pi n t) \right| \leq \frac{1}{2\pi} \times \left[\frac{1}{\sin(\pi t)} \left(\frac{1}{N} + \frac{\pi^2}{6} \right) + 1 \right]$$

ist. Die Details werden hier übergangen. Wichtig ist die Folgerung:

Die Funktionen $C_N^k(t)$ verhalten sich für $N \rightarrow \infty$ an den Stetigkeits- bzw. Unstetigkeitsstellen von $k(t)$ ganz unterschiedlich:

- Für $t \in \mathbb{Z}$ (Unstetigkeit) divergiert $C_N^k(t) \rightarrow \infty$ (und zwar wie $\log N$).
- Für $t \notin \mathbb{Z}$ (Stetigkeit) bleibt $C_N^k(t)$ beschränkt.

Damit kann man einen entscheidenden Schritt weiter gehen:

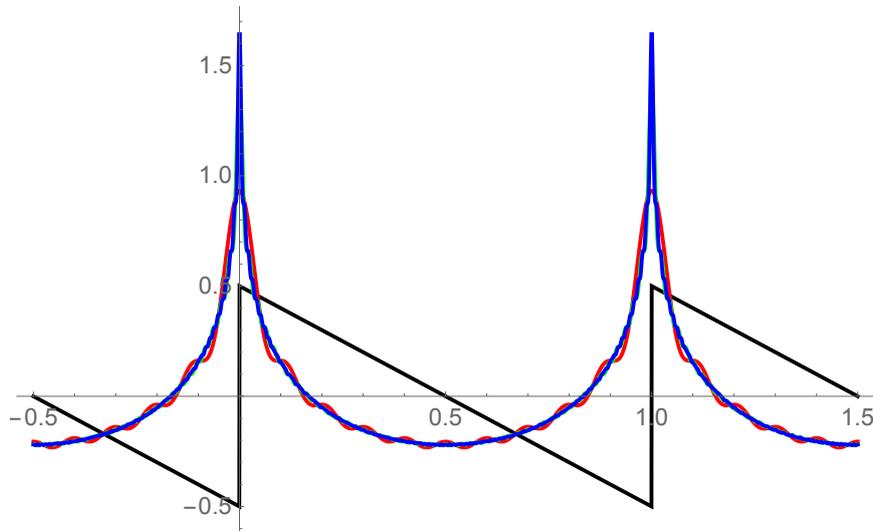


Abbildung 3.14: Die Funktion $k(t)$ (schwarz) und die Funktionen $C_N^k(t)$ für $N = 10$ (rot), $N = 50$ (grün) und $N = 100$ (blau).

Die Funktionen

$$\tilde{C}_N^k(t) = (\pi / \log(N)) \cdot C_N^k(t) = \frac{1}{\log(N)} \sum_{n=1}^N \frac{1}{n} \cos(2\pi n t)$$

verhalten sich für $N \rightarrow \infty$ an den Stetigkeits- bzw. Unstetigkeitsstellen von $k(t)$ ganz unterschiedlich:

- Für $t \in \mathbb{Z}$ (Unstetigkeit) konvergiert $\tilde{C}_N^k(t) \rightarrow 1$.
- Für $t \notin \mathbb{Z}$ (Stetigkeit) konvergiert $\tilde{C}_N^k(t) \rightarrow 0$.

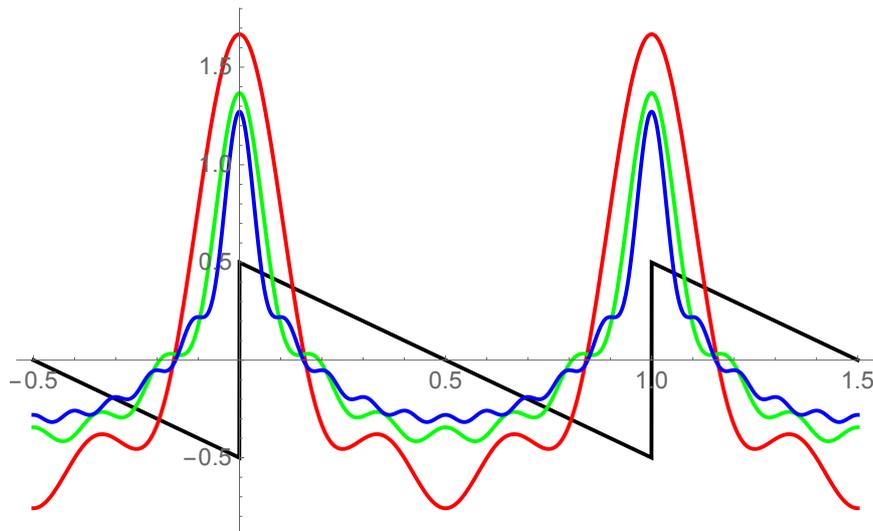


Abbildung 3.15: Die Funktion $k(t)$ (schwarz) und die Funktionen $\tilde{C}_N^k(t) = (\pi / \log(N)) \cdot C_N^k(t)$ für $N = 10$ (rot), $N = 50$ (grün) und $N = 100$ (blau).

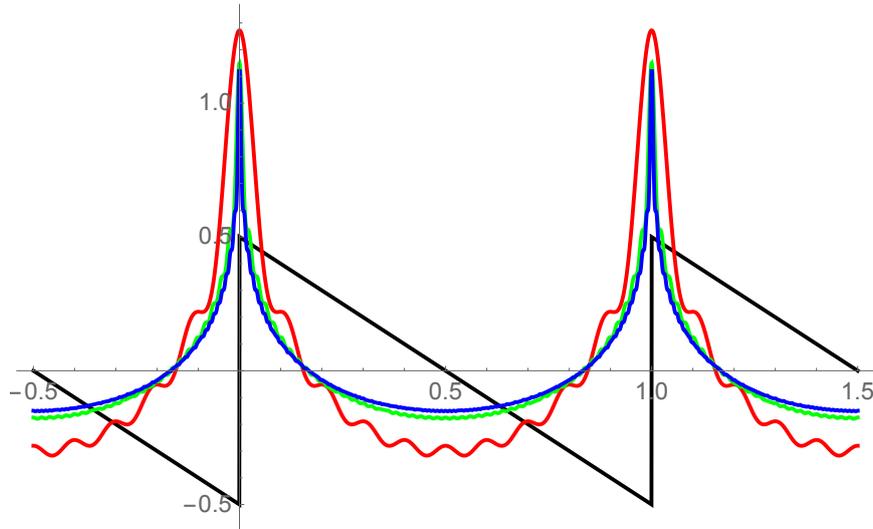


Abbildung 3.16: Die Funktion $k(t)$ (schwarz) und die Funktionen $\tilde{C}_N^k(t) = (\pi/\log(N)) \cdot C_N^k(t)$ für $N = 3$ (rot), $N = 6$ (grün) und $N = 10$ (blau).

Zugegeben: die Konvergenz ist recht langsam, weil eben die harmonische Reihe $\sum_{n \geq 1} 1/n$ sehr langsam divergiert. Man kann die Konvergenz ein wenig verbessern, indem man in der Definition von $\tilde{C}_N^k(t)$ den Nenner $\log(N)$ durch $\log(N) + \gamma$ ersetzt, wobei

$$\gamma = \lim_{N \rightarrow \infty} \left(\sum_{n=1}^N \frac{1}{n} - \log(N) \right) = 0.577215 \dots$$

die sogenannte *Eulersche Konstante* ist.

Die folgende Aussage sollte unmittelbar einleuchten und ergibt sich ohne Probleme aus den Definitionen.

Lemma 3.1 Ist $f(t)$ eine 1-periodische integrierbare Funktion, mit $c_f[n]$ ($n \in \mathbb{Z}$) als Folge ihrer Fourierkoeffizienten, sowie

$$S_N^f(t) = \sum_{n=-N}^N c[n] e^{2\pi i n t}$$

als N -te Partialsumme der formalen Fourierreihe von $f(t)$. Für $x \in \mathbb{R}$ bezeichne $T_x f$ die Translation von f um die Distanz x , also $(T_x f)(t) = f(t - x)$. Dann gilt

$$S_N^{T_x f}(t) = S_N^f(t - x),$$

also kurz: $S_N^{T_x f} = T_x(S_N^f)$, oder in Worten: Bildung von Partialsummen und Translation sind bei Fourierreihen miteinander verträglich (i.e., kommutieren miteinander).

Angewendet auf die Translate der Funktion $k(t)$ bedeutet das: alles, was über $k(t)$, $S_N^k(t)$, $C_N^k(t)$, $\tilde{C}_N^k(t)$ festgestellt wurde, überträgt sich problemlos mittels Translation.

Für festes $x \in \mathbb{R}$ gilt

$$S_N^{T_x k}(t) = S_N^k(t-x) = \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \sin(2\pi n(t-x)).$$

Definiert man

$$C_N^{T_x k}(t) = (T_x C_N^k)(t) = \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \cos(2\pi n(t-x)),$$

$$\tilde{C}_N^{T_x k} = \frac{\pi}{\log(N)} C_N^{T_x k}(t) = \frac{\pi}{\log(N)} \sum_{n=1}^N \frac{1}{n} \cos(2\pi n(t-x)),$$

so verhalten sich für $N \rightarrow \infty$ die Funktionen an den Stetigkeits- bzw. Unstetigkeitsstellen von $k(t-x)$ ganz unterschiedlich:

- Für $t \in \mathbb{Z}$ (Unstetigkeit) konvergiert $\tilde{C}_N^k(t-x) \rightarrow 1$.
- Für $t \notin \mathbb{Z}$ (Stetigkeit) konvergiert $\tilde{C}_N^k(t-x) \rightarrow 0$.

Diesen offensichtlich sehr nützlichen Schritt “ersetze sin durch cos”, der von $S_N^{T_x k}(t) = S_N^k(t-x)$ zu $C_N^{T_x k}(t) = C_N^k(t-x)$ führt, sollte man sich genauer anschauen:

$$\begin{aligned} S_N^{T_x k}(t) &= \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \sin(2\pi n(t-x)) \\ &= \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} (\sin(2\pi nt) \cos(2\pi nx) - \cos(2\pi nt) \sin(2\pi nx)) \\ C_N^{T_x k}(t) &= \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} \cos(2\pi n(t-x)) \\ &= \frac{1}{\pi} \sum_{n=1}^N \frac{1}{n} (\cos(2\pi nt) \cos(2\pi nx) + \sin(2\pi nt) \sin(2\pi nx)) \end{aligned}$$

Bezogen auf die Basisfunktionen $\cos(2\pi nt)$ und $\sin(2\pi nt)$ bedeutet das also:

- der Koeffizient von $\sin(2\pi nt)$ in $S_N^{T_x k}(t)$ wird zum Koeffizienten von $\cos(2\pi nt)$ in $C_N^{T_x k}(t)$ ($n > 0$),
- das Negative des Koeffizienten von $\cos(2\pi nt)$ in $S_N^{T_x k}(t)$ wird zum Koeffizienten von $\sin(2\pi nt)$ in $C_N^{T_x k}(t)$ ($n > 0$).

Diese Transformation kann man generell bei Fourierreihen machen! Eine kurze Rechnung zeigt, dass diese Operation im Bezug auf die Fourierkoeffizienten ganz einfach so aussieht (nachrechnen!):

- Multipliziere die Fourierkoeffizienten $c_k[n]$ mit $i \cdot \text{sign}(n)$, wobei

$$\text{sign}(n) = \begin{cases} 1 & \text{für } n > 0, \\ 0 & \text{für } n = 0, \\ -1 & \text{für } n < 0. \end{cases}$$

Diese einfache Manipulation ist so interessant, dass sie einen Namen³ verdient hat:

Definition 3.7. Ist $\mathbf{c} = (c[n])_{n \in \mathbb{Z}}$ eine komplexe Folge, so bezeichnet man die komplexe Folge

$$\mathcal{H}(\mathbf{c}) = (i \cdot \text{sign}(n) \cdot c[n])_{n \in \mathbb{Z}}$$

als die (diskrete) Hilbert-Transformierte von \mathbf{c} .

Die Transformation $\mathbf{c} \mapsto \mathcal{H}(\mathbf{c})$ ist offensichtlich linear und die Zugehörigkeit zu $\ell^1(\mathbb{Z})$ bzw. $\ell^2(\mathbb{Z})$ ändert sich durch Hilbert-Transformation nicht.

Lemma 3.2 Ist \mathbf{c} eine komplexe Folge und $\mathcal{H}(\mathbf{c}) = \mathbf{c}' = (c'[n])_{n \geq 0}$, ihre Hilbert-Transformierte, so gilt

$$\sum_{n \in \mathbb{Z}} |c'[n]| = |c[0]| + \sum_{n \in \mathbb{Z}} |c'[n]|, \quad \sum_{n \in \mathbb{Z}} |c'[n]|^2 = |c[0]|^2 + \sum_{n \in \mathbb{Z}} |c'[n]|^2.$$

Im Bezug auf die motivierenden Überlegungen gilt also:

Die Koeffizienten der Partialsummen $C_N^{T_x k}(t)$ in der Basis der Funktionen $e^{2\pi i n t}$ ($n \in \mathbb{Z}$) entstehen aus den Fourierkoeffizienten von $k(t - x)$ durch Hilbert-Transformation.

In einem nächsten Schritt macht man sich die Linearität der Fourierkoeffizienten zu Nutze, die nichts weiter ist als die Linearität der Integration.

Ist $f(t) = a \cdot g(t) + b \cdot h(t)$ eine 1-periodische Funktion (mit Konstanten a, b), so gilt

$$c_f[n] = c_{ag+bh}[n] = a \cdot c_g[n] + b \cdot c_h[n] \quad (n \in \mathbb{Z}).$$

Das überträgt sich auf beliebige endliche Linearkombinationen.

Betrachten wir nun wieder die Situation, wie sie vor Theorem 3.8 beschrieben wurde: $f(t)$ ist eine stückweise differenzierbare 1-periodische Funktion und

$$f(t) = g(t) + h(t) \quad \text{mit} \quad h(t) = \sum_{j=1}^m h_j \cdot k(t - t_j)$$

die Zerlegung in einen *stetigen Anteil* $f(t)$ und einen *Sprunganteil* $h(t)$, wobei $0 < t_1 < t_2 < \dots < t_m \leq 1$ die Sprungstellen der Funktion $f(t)$, mit Sprunghöhen h_1, h_2, \dots, h_m sind. Das Ziel ist, die Sprungstellen t_j und Sprunghöhen h_j aus der Kenntnis der Funktion $f(t)$ (bzw. ihrer Fourierkoeffizienten) allein (!) zu bestimmen. Für die Fourierkoeffizienten gilt

$$c_f[n] = c_g[n] + c_h[n] = c_g[n] + \sum_{j=1}^m h_j \cdot c_{T_{t_j} k}[n],$$

³Viel bekannter ist allerdings die kontinuierliche Hilbert-Transformation, die auch im Kontext der Wavelet-Transformationen eine Rolle spielt.

wobei

$$\lim_{N \rightarrow \infty} |c_g[n]| < \infty, \quad \lim_{N \rightarrow \infty} |c_h[n]| = \infty, \quad \lim_{N \rightarrow \infty} |c_{T_{t_j} k}[n]| = \infty \quad (1 \leq j \leq m).$$

Für die Partialsummen der Fourierreihen gilt

$$S_N^f(t) = S_N^g(t) + S_N^h(t) \quad \text{mit} \quad S_N^h(t) = \sum_{j=1}^m h_j \cdot S_N^k(t - t_j).$$

Schreibt man $C_N^f(t)$ bzw. $C_N^g(t)$ bzw. $C_N^{T_{t_j} k}(t)$ für die Partialsummen, die durch Hilbert-Transformation der ursprünglichen Fourierkoeffizienten entstehen, so erhält man entsprechend

$$C_N^f(t) = C_N^g(t) + C_N^h(t) \quad \text{mit} \quad C_N^h(t) = \sum_{j=1}^m h_j \cdot C_N^k(t - t_j).$$

Im Bezug auf den Term $C_N^h(t)$ folgt aus den früheren Betrachtungen:

Das Verhalten der Partialsummen $C_N^h(t)$ für $N \rightarrow \infty$ unterscheidet sich grundsätzlich, je nachdem, ob t Unstetigkeitsstelle oder Stetigkeitsstelle von $h(t)$ ist:

- Unstetigkeit:
Für $t \in \{t_1, t_2, \dots, t_m\}$ (und ebenso, falls $t - x \in \{t_1, t_2, \dots, t_m\}$ für ein $x \in \mathbb{Z}$) gilt Divergenz $C_N^h(t) \rightarrow \pm\infty$ (je nach dem Vorzeichen des relevanten h_j).
- Stetigkeit:
Für $t \notin \{t_1, t_2, \dots, t_m\}$ (und ebenso, falls $t - x \notin \{t_1, t_2, \dots, t_m\}$ für alle $x \in \mathbb{Z}$) bleibt die Folge der $|C_N^h(t)|$ beschränkt.

Macht man nun noch die Modifikation, die von den $C_N^k(t)$ zu den $\tilde{C}_N^k(t)$ geführt hat, also $\tilde{C}_N^k(t) = \frac{\pi}{\log(N)} C_N^k(t)$, so erhält man

Das Verhalten der Partialsummen $\tilde{C}_N^h(t)$ für $N \rightarrow \infty$ unterscheidet sich grundsätzlich, je nachdem, ob t Unstetigkeitsstelle oder Stetigkeitsstelle von $h(t)$ ist:

- Unstetigkeit:
Für $t \in \{t_1, t_2, \dots, t_m\}$ (und ebenso, falls $t - x \in \{t_1, t_2, \dots, t_m\}$ für ein $x \in \mathbb{Z}$) gilt Divergenz $\tilde{C}_N^h(t) \rightarrow h_j$ (je nach dem relevanten h_j).
- Stetigkeit:
Für $t \notin \{t_1, t_2, \dots, t_m\}$ (und ebenso, falls $t - x \notin \{t_1, t_2, \dots, t_m\}$ für alle $x \in \mathbb{Z}$) konvergiert die Folge der $\tilde{C}_N^h(t) \rightarrow 0$.

Betrachtet man nun, was die entsprechenden Operationen für eine Auswirkung auf die Partialsummen $S_N^g(t)$, $C_N^g(t)$ und $\tilde{C}_N^g(t)$ haben, so gilt wegen der Stetigkeit von $g(t)$ folgende Aussagen:

Für alle $t \in \mathbb{R}$ gilt

- $|S_N^g(t)| < M_g$ für eine nur von g abhängige Konstante M_g , da die Partialsummen gleichmäßig gegen die stetige (und somit ohnehin beschränkte) Funktion $g(t)$ konvergieren.
- $|C_N^g(t)| < M'_g$ für eine nur von g abhängige Konstante M'_g , da die Folge der Hilbert-transformierten Fourierkoeffizienten von $g(t)$ ebenso absolut summierbar ist, wie die der ursprünglichen Fourierkoeffizienten von $g(t)$.
- $\lim_{N \rightarrow \infty} |\tilde{C}_N^g(t)| = 0$, wegen der Division der Glieder beschränkter Partialsummen durch $\log(N)$.

Nun muss man nur noch das Verhalten von $g(t)$ und von $h(t)$ zusammenbringen, aber da sich alle Schritte mit der Linearität vertragen, ist das ganz einfach.

Theorem 3.9

Ist $f(t)$ eine stückweise-differenzierbare 1-periodische Funktion mit Fourierkoeffizienten $\mathbf{c}_f = (c_f[n])_{n \in \mathbb{Z}}$, mit Unstetigkeitsstellen $0 < t_1 < t_2 \dots < t_m \leq 1$ und Sprunghöhen h_1, h_2, \dots, h_m , und definiert Partialsummen mittels Hilbert-Transformation und Division durch $\log(N)$,

$$\tilde{C}_N^f(t) = \frac{\pi}{\log(N)} \sum_{n=1}^N i \operatorname{sign}(n) c_f[n] e^{2\pi i n t},$$

dann gilt für $0 \leq t \leq 1$

$$\lim_{N \rightarrow \infty} \tilde{C}_N^f(t) = \begin{cases} h_j & \text{falls } t = t_j \ (j \in \{1, 2, \dots, m\}) \quad (\text{Unstetigkeit}), \\ 0 & \text{falls } t \notin \{t_1, t_2, \dots, t_m\} \quad (\text{Stetigkeit}). \end{cases}$$

Dabei ist die Geschwindigkeit der Konvergenz $O(1/\log(N))$.

Der Witz dieser Angelegenheit sollte damit klar sein:

Die Partialsummen $\tilde{C}_N^f(t)$ kann man bilden, so weit man die Fourierkoeffizienten $c_f[n]$ (z.B. numerisch) berechnen kann – man muss dafür die Unstetigkeitsstellen von $f(t)$ nicht kennen!

Im Gegenteil: Plottet man die Partialsummen $\tilde{C}_N^f(t)$ für wachsendes N , so zeigen sich die Lage der Sprungstellen t_j (exakt) und deren Höhe h_j (approximiert). Man erhält auf diesem Weg die wesentlichen Informationen über die Unstetigkeitsstellen von $f(t)$.

■ **Beispiel 3.3** Die Funktion $f(t)$ aus Beispiel 3.2 soll nun im Licht von Theorem 3.9 untersucht werden. Die Partialsumme $\tilde{C}_{10}^f(t)$ ist

$$\begin{aligned} \tilde{C}_{10}^f(t) = & -0.159632 \sin(2\pi t) + 0.259944 \sin(4\pi t) - 0.276649 \sin(6\pi t) + 0.0510963 \sin(8\pi t) \\ & + 0.258 \sin(10\pi t) - 0.0772292 \sin(12\pi t) - 0.0701375 \sin(14\pi t) - 0.0363594 \sin(16\pi t) \\ & + 0.0702795 \sin(18\pi t) + 0.00565847 \sin(20\pi t) + 0.320651 \cos(2\pi t) + 0.51547 \cos(4\pi t) \\ & + 0.738113 \cos(6\pi t) - 0.163613 \cos(8\pi t) + 0.30511 \cos(10\pi t) + 0.115926 \cos(12\pi t) \\ & + 0.130634 \cos(14\pi t) + 0.0211273 \cos(16\pi t) + 0.18477 \cos(18\pi t) + 0.112044 \cos(20\pi t) \end{aligned}$$

Plottet man diese Funktion $\tilde{C}_{10}^f(t)$ gegen die Funktion $f(t)$ (Abbildung 3.17), so machen sie die Unstetigkeitsstellen von $f(t)$ an den Extrema von $\tilde{C}_{10}^f(t)$ schon deutlich bemerkbar.

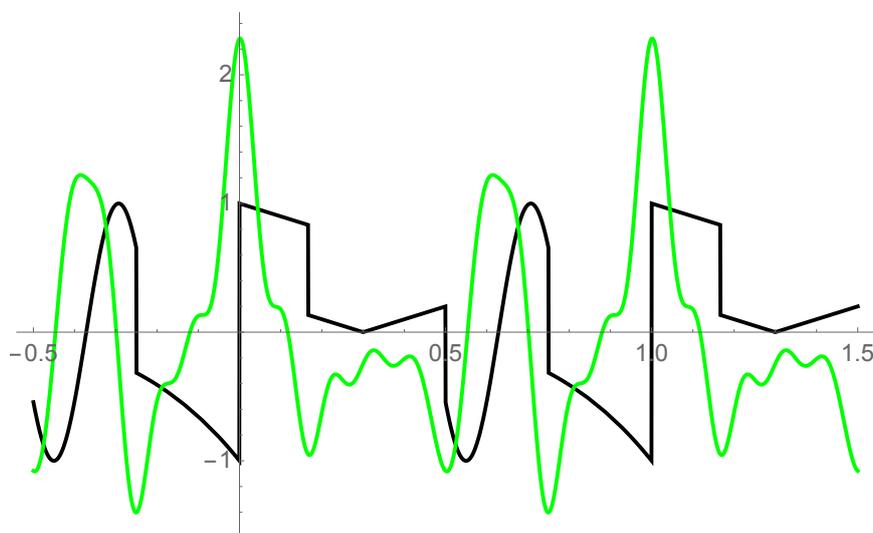


Abbildung 3.17: Die Funktion $f(t)$ (schwarz) und die Partialsumme $\tilde{C}_{10}^f(t)$ (grün)

Allerdings hat $\tilde{C}_{10}^f(t)$ auch noch einen Extremwert bei etwa $t = 0.6$, der hat aber nichts mit einer Unstetigkeit zu tun, sondern rührt von dem relativ hochfrequenten Term $\sin(20t)$ von $f(t)$ im Intervall $1/2 \leq t < 3/4$. Er verschwindet für grosses N , aber leider nur wie $1/\log(N)$, also ziemlich langsam. Immerhin sieht die Situation für $N = 100$ schon deutlicher aus.

Abschliessend in Abbildung 3.19 noch ein Vergleich der Funktion $f(t)$ mit den drei Partialsummen $\tilde{C}_{100}^f(t)$, $\tilde{C}_{100}^g(t)$ und $\tilde{C}_{100}^h(t)$. ■

Das durch Theorem 3.9 beschriebene Verfahren zur Detektierung von Sprungstellen in stückweise differenzierbaren periodischen Funktionen mittels Hilbert-Transformation der Fourierkoeffizienten hat, wie schon weiter oben erwähnt, den Nachteil einer langsamen Konvergenz. Auf der positiven Seite ist zu vermerken, dass die Höhen der “Spitzen” in den approximierenden Partialsummen $\tilde{C}_N^f(t)$ im Limes für $N \rightarrow \infty$ tatsächlich mit den Höhen der Sprünge an den jeweiligen Sprungstellen übereinstimmen. Man kann das Konvergenzverhalten deutlich verbessern, wenn man die Fourierkoeffizienten von $f(t)$ so verändert, dass hohe Frequenzen (wie sie an

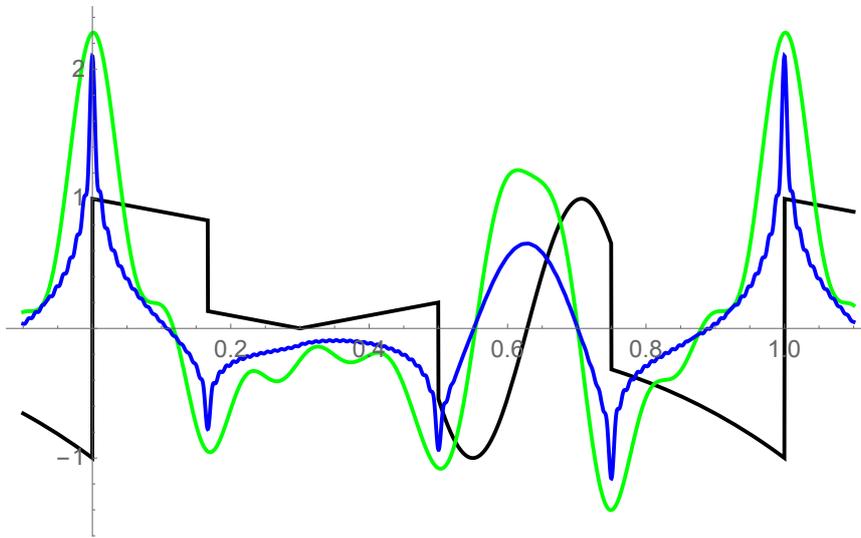


Abbildung 3.18: Die Funktion $f(t)$ (schwarz) und die Partialsummen $\tilde{C}_{100}^f(t)$ (grün) und $\tilde{C}_{100}^f(t)$ (blau).

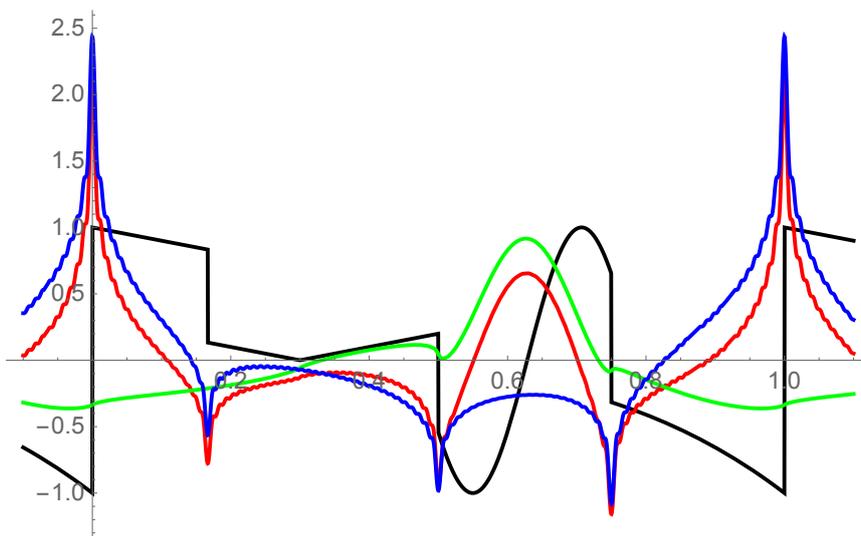


Abbildung 3.19: Die Funktion $f(t)$ (schwarz) und die Partialsummen $\tilde{C}_{100}^f(t)$ (rot), $\tilde{C}_{100}^g(t)$ (grün) und $\tilde{C}_{100}^h(t)$ (blau).

Kanten einer Funktion vorherrschen) stärker berücksichtigt werden. Das kann man durch Multiplikation erreichen:

$$c_f[n] \rightsquigarrow \widehat{c}_f[n] = -i n c_f[n] \quad (n \in \mathbb{Z}).$$

Das sind i.w. die Fourierkoeffizienten der *Ableitung* von $f(t)$.

Theorem 3.10

Ist $f(t)$ eine stückweise-differenzierbare 1-periodische Funktion mit Fourierkoeffizienten $\mathbf{c}_f = (c_f[n])_{n \in \mathbb{Z}}$, mit Unstetigkeitsstellen $0 < t_1 < t_2 \dots < t_m \leq 1$ und Sprunghöhen h_1, h_2, \dots, h_m , und definiert Partialsummen durch

$$\widehat{C}_N^f(t) = \sum_{n=-N}^N \widehat{c}_f[n] e^{2\pi i n t} = -\frac{\pi}{N} \sum_{n=-N}^N i n c_f[n] e^{2\pi i n t},$$

dann gilt für $0 \leq t \leq 1$

$$\lim_{N \rightarrow \infty} \widehat{C}_N^f(t) = \begin{cases} h_j & \text{falls } t = t_j \ (j \in \{1, 2, \dots, m\}) \quad (\text{Unstetigkeit}), \\ 0 & \text{falls } t \notin \{t_1, t_2, \dots, t_m\} \quad (\text{Stetigkeit}). \end{cases}$$

Dabei ist die Geschwindigkeit der Konvergenz $O(1/N)$.

Auf Einzelheiten soll hier nicht eingegangen werden, es sei aber vermerkt, dass auch hier die Eigenschaften des Dirichlet-Kerns eine entscheidende Rolle spielen. Aber die Auswirkungen dieses Resultats sind so frappierend, dass sie zumindest an Beispielen demonstriert werden sollen.

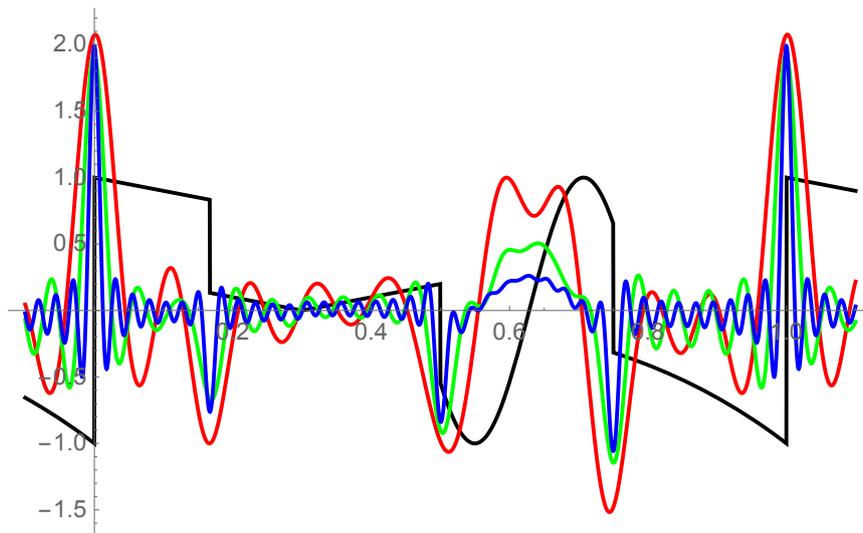


Abbildung 3.20: Die Funktion $f(t)$ (schwarz) und die Partialsummen $\widehat{C}_{10}^f(t)$ (rot), $\widehat{C}_{20}^f(t)$ (grün) und $\widehat{C}_{40}^f(t)$ (blau).

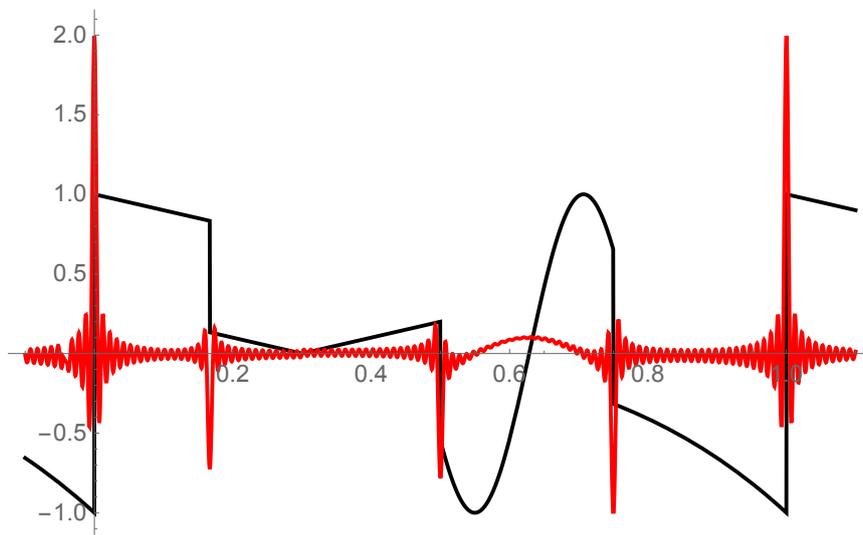


Abbildung 3.21: Die Funktion $f(t)$ (schwarz) und die Partialsummen $\widehat{C}_{100}^f(t)$ (rot).

3.5 Mathematica notebooks

Die folgenden Seiten enthalten ein Mathematica notebooks zur Illustration wichtiger Aspekte der Fourierreihen.

1. Beispiele zur Berechnung von Fourierreihen
2. Illustration des Gibbs-Phänomens in 2D

Computing Fourier series

```
In[17]:= per[fun_, a_, b_, t_]:=
If[a<=t<b, fun[t], fun[t-Floor[(t-a)/(b-a)]*(b-a)]];
per::usage="computes periodization of f(t)
as defined over the interval [a,b]";
```

```
In[19]:= minmax[fun_, a_, b_]:=
{MinValue[{fun[t], a<=t<=b}, {t}],
MaxValue[{fun[t], a<=t<=b}, {t}]};

minmax::usage="computes minimum and maximum
of f(t) over the interval [a,b]";
```

```
In[21]:= showapprox[fun_, a_, b_, t_, N_]:=
Plot[{
fun[t],
per[fun, a, b, t],
Evaluate[FourierSum[fun, a, b, t, N]]},
{t, a-1/2, b+1/2},
PlotLegends->{fun, "per"[fun], "SN"[fun]},
PlotStyle->{Black, Blue, Red}, Exclusions->None];

showapprox::usage="displays the function f(t),
defined over the interval [a,b],
its periodization per(f) and its truncated
symbolic Fourierseries approximation SN(f,t)";
```

```
In[23]:= animateapprox[fun_, a_, b_, t_, N_]:=Module[{span1, m1, FSumA},
span1=minmax[fun, a, b][[2]]-minmax[fun, a, b][[1]];
m1=minmax[fun, a, b]+{-0.3, 0.3}*span1;
FSumA=
Array[FourierSum[fun, a, b, t, #]&, N+1, 0];
Manipulate[
Plot[{
fun[t],
per[fun, a, b, t],
Evaluate[FSumA][[k+1]]}, {t, a-0.5, b+0.5},
PlotRange->m1,
PlotLegends->{fun, "per"[fun], "SN"[fun]},
PlotStyle->{Black, Blue, Red},
Exclusions->None}, {k, 0, N, 1}
];

animateapprox::usage="displays the function f(t),
defined over the interval [a,b],
its periodization per(f) and its truncated
symbolic Fourierseries approximations Sk(f,t) for k from 0 to N";
```

```
In[25]:= showNapprox[fun_,a_,b_,t_,N_]:=
Plot[{
  fun[t],
  per[fun,a,b,t],
  Evaluate[NFourierSum[fun,a,b,t,N]]},
{t,a-1/2,b+1/2},
PlotLegends->{fun,"per"[fun],"SN"[fun]},
PlotStyle->{Black,Blue,Red},
Exclusions->None];

showNapprox::usage="displays the function f(t),
defined over the interval [a,b),
its periodization per(f) and its truncated
numerical Fourier series approximation SN(f,t)";
```

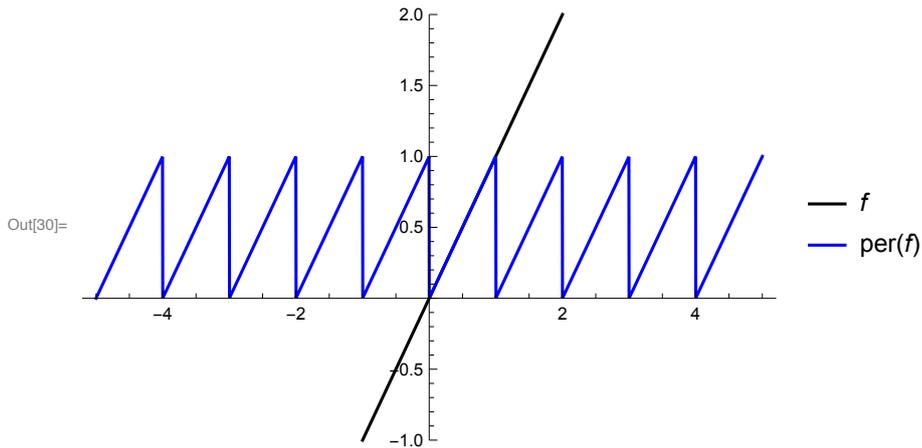
```
In[27]:= animateNapprox[fun_,a_,b_,t_,N_]:=Module[{span2,NFSumA,m2},
span2=Minmax[fun,a,b][[2]]-Minmax[f,a,b][[1]];
m2=Minmax[fun,a,b]+{-0.1,0.1}*span2;
NFSumA=
Array[NFourierSum[fun,a,b,t,#]&,N+1,0];
Manipulate[
Plot[{
  fun[t],
  per[fun,a,b,t],
  Evaluate[NFSumA][[k+1]],{t,a-0.5,b+0.5},
PlotRange->m2,
PlotLegends->{fun,"per"[fun],"SN"[fun]},PlotStyle->{Black,Blue,Red},
Exclusions->None},{k,0,N,1}];

animateNapprox::usage="displays the function f(t),
defined over the interval [a,b),
its periodization per(f) and its truncated
numerical Fourier series approximations Sk(f,t) for k from 0 to N";
```

Define a function and its periodization taking an interval [a,b)

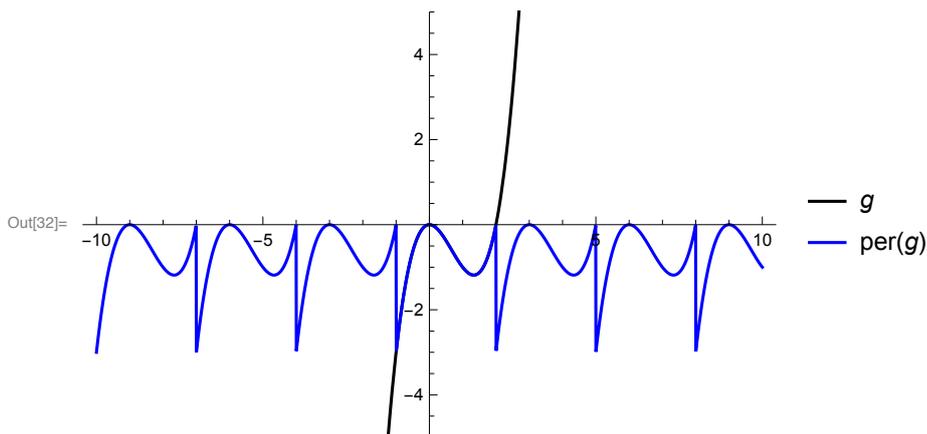
```
In[29]:= f[t_] = t;

In[30]:= Plot[{f[t], per[f, 0, 1, t]}, {t, -5, 5},
PlotStyle -> {Black, Blue},
PlotRange -> {-1, 2},
PlotLegends -> {f, "per"[f], "SN"[f]},
Exclusions -> None]
```



In[31]= `g[t_] = t^3 - 2 t^2;`

In[32]= `Plot[{g[t], per[g, -1, 2, t]}, {t, -10, 10},
PlotStyle -> {Black, Blue},
PlotRange -> {-5, 5},
PlotLegends -> {g, "per"[g], "SN"[g]},
Exclusions -> None]`



Compute Fourier coefficients symbolically (!) over an interval [a,b)

In[33]= `myFourier[fun_, a_, b_, k_] := myFourier[fun, a, b, k] = Module[{p},
p = b - a;
(1/p) * Integrate[
fun[t] Exp[-2 Pi I k t/p], {t, a, b}]
]`

In[34]= `myFourier[f, 0, 1, k]`

Out[34]= $-\left(\frac{e^{-2 i k \pi}(-1 + e^{2 i k \pi} - 2 i k \pi)}{4 k^2 \pi^2}\right)$

Assuming that the parameter k is an integer simplifies the expression considerably

In[35]= `Assuming[k ∈ Integers, Simplify[%]]`

Out[35]= $\frac{i}{2 k \pi}$

But symbolic coefficients can get very complicated!

In[36]:= Assuming[k ∈ Integers, Simplify[myFourier[g, -1, 2, k]]]

$$\text{Out[36]} = \begin{cases} \frac{1}{8 k^4 \pi^4} 3 \left((27 - 14 k^2 \pi^2) \text{Cos}\left[\frac{2 k \pi}{3}\right] + \right. & k = 0 \\ \quad \left. (-27 + 8 k^2 \pi^2) \text{Cos}\left[\frac{4 k \pi}{3}\right] - 2 k \pi (-15 + 2 k^2 \pi^2 + 24 \text{Cos}\left[\frac{2 k \pi}{3}\right]) \text{Sin}\left[\frac{2 k \pi}{3}\right] \right) & \\ \frac{1}{4 k^3 \pi^3} 3 (-1)^k (-27 + 3 i k \pi + 2 k^2 \pi^2) (i \text{Cos}\left[\frac{k \pi}{3}\right] + \text{Sin}\left[\frac{k \pi}{3}\right]) & \text{True} \end{cases}$$

In[37]:= myFourier[g, -1, 2, -5]

Out[37]= $(3 (-1)^{1/6} (-27 - 15 i \pi + 50 \pi^2)) / (500 \pi^3)$

Summing the truncated Fourier series with Fourier coefficients from -N to +N

```
In[38]:= FourierSum[fun_, a_, b_, t_, N_] :=
Sum[myFourier[fun, a, b, k] Exp[2 Pi I k t / (b - a)], {k, -N, N}]
```

In[39]:= Manipulate[FourierSum[f, 0, 1, t, n], {n, 0, 5, 1}]

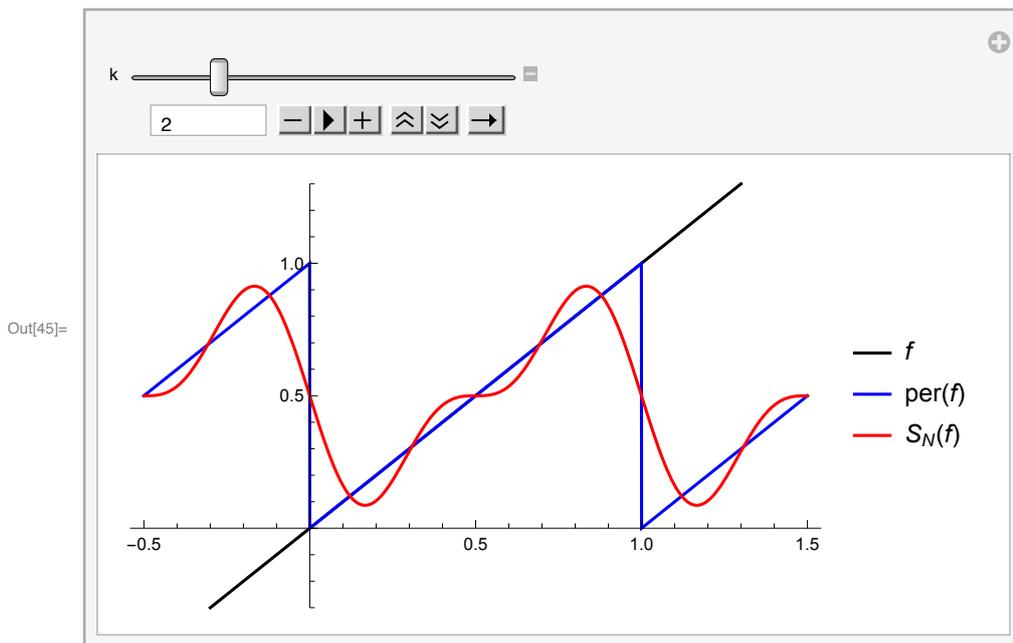
Out[39]=

```
In[40]:= Manipulate[
Collect[
ExpToTrig[
FourierSum[f, 0, 1, t, n]], Sin[k_ Pi t]], {n, 0, 5, 1}]
```

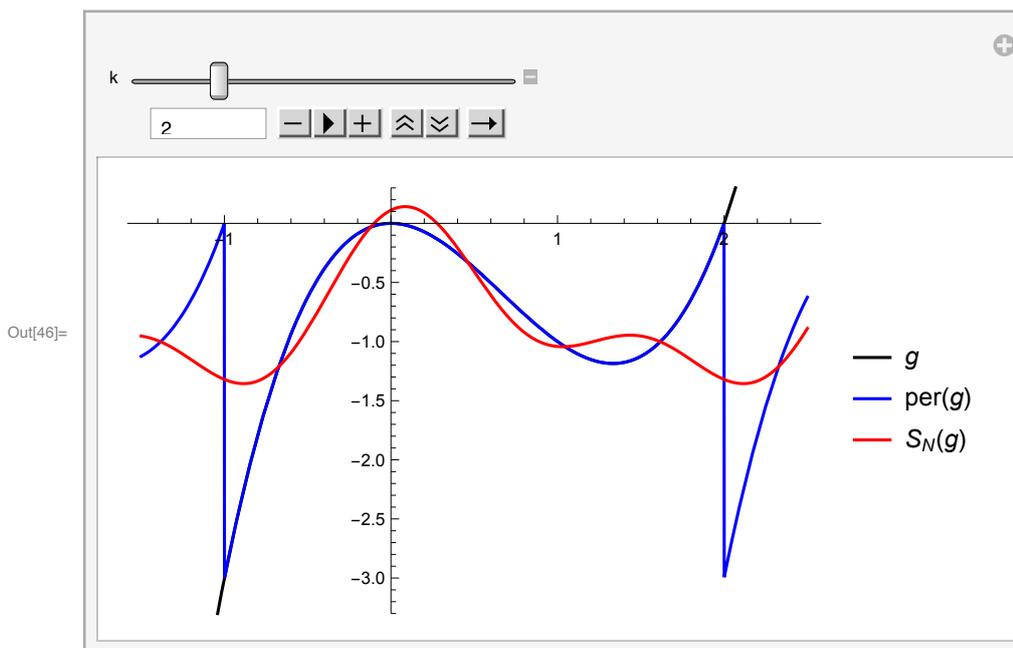
Out[40]=

In[41]:= Manipulate[FourierSum[g, -1, 2, t, n], {n, 0, 5, 1}]

Out[41]=



In[46]:= `animateapprox[g, -1, 2, t, 10]`



Second example

The procedures are now numerical! Use `NIntegrate` in place of `Integrate`

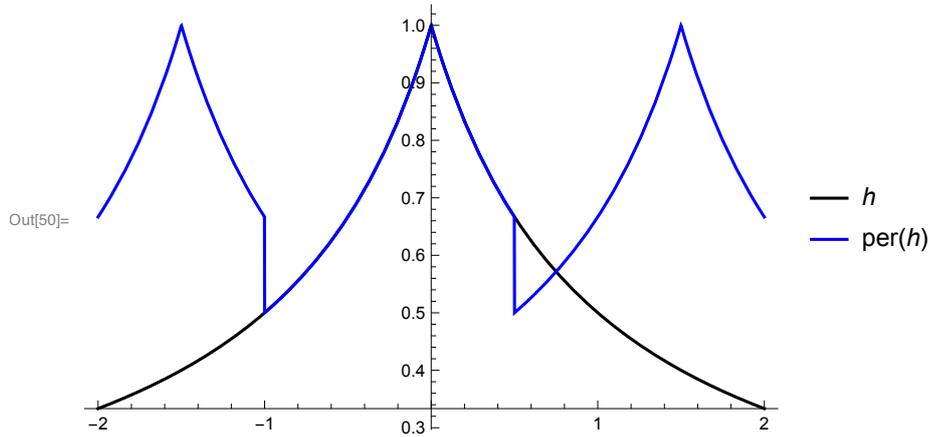
```
In[47]:= myNFourier[fun_, a_, b_, k_] := myNFourier[fun, a, b, k] = Module[{},
  p = b - a;
  (1/p) * NIntegrate[
    fun[t] Exp[-2 Pi I k t/p], {t, a, b}
  ]
]
```

```
In[48]:= NFourierSum[fun_, a_, b_, t_, N_] :=
  Sum[myNFourier[fun, a, b, k] Exp[2 Pi I k t/(b-a)], {k, -N, N}]
```

In[49]:= $h[t_] = 1 / (1 + \text{Abs}[t])$

Out[49]= $\frac{1}{1 + \text{Abs}[t]}$

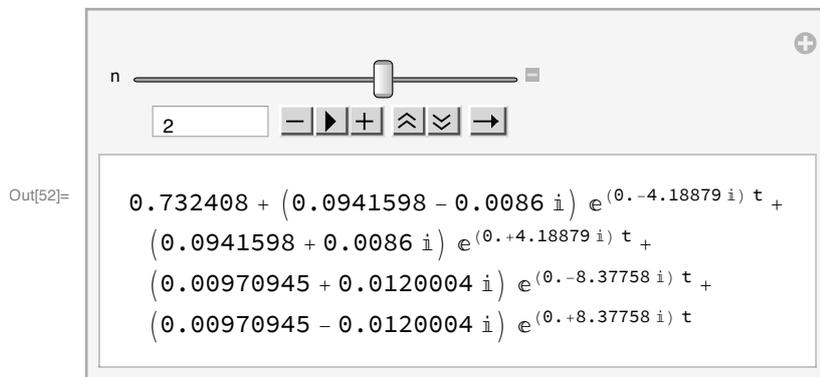
In[50]:= `Plot[{h[t], per[h, -1, 0.5, t]}, {t, -2, 2},
PlotStyle -> {Black, Blue},
PlotLegends -> {h, "per"[h], "SN"[]},
Exclusions -> None]`



In[51]:= `NF[h, -1, 0.5, 10]`

Out[51]= `NF[h, -1, 0.5, 10]`

In[52]:= `Manipulate[NFourierSum[h, -1, 0.5, t, n], {n, 0, 3, 1}]`



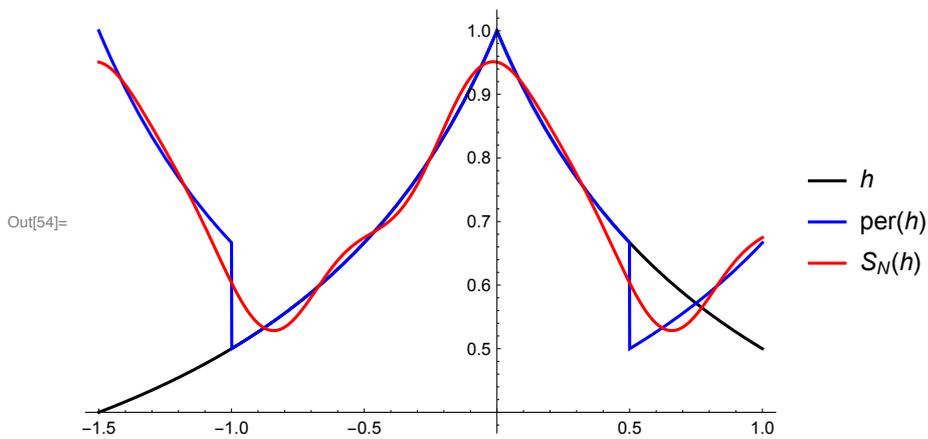
In[53]:= `Manipulate[
ExpToTrig[
NFourierSum[h, -1, 0.5, t, n]], {n, 0, 4, 1}]`

n

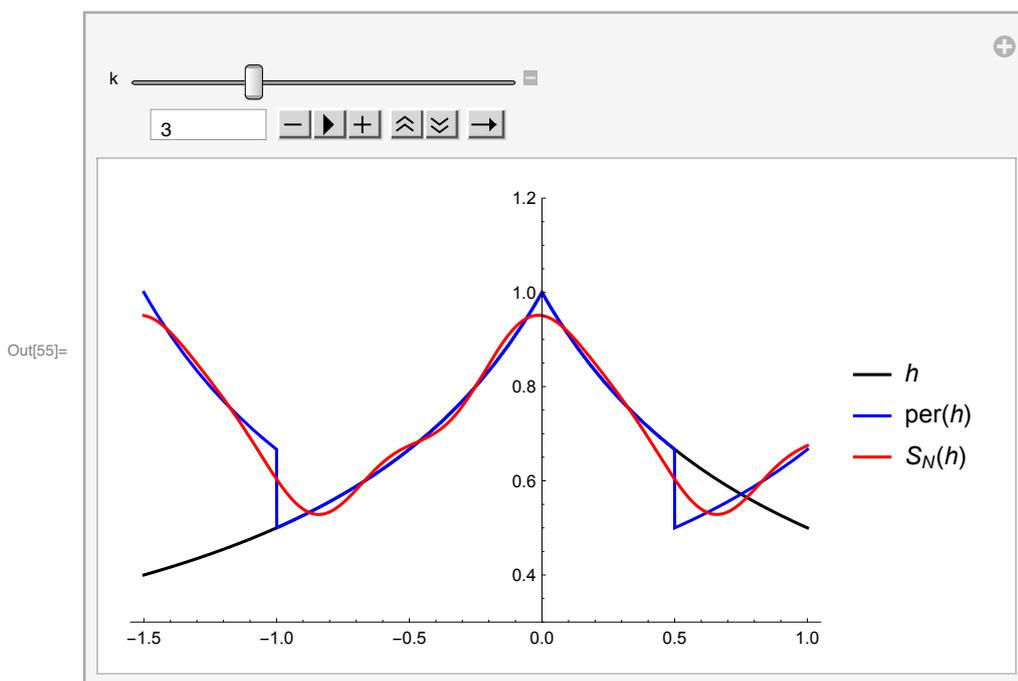
Out[53]=

$$\begin{aligned}
 & 0.732408 + \\
 & (0.0941598 - 0.0086 i) (\text{Cos}[(4.18879 + 0. i) t] - i \text{Sin}[(4.18879 + 0. i) t]) + \\
 & (0.0941598 + 0.0086 i) (\text{Cos}[(4.18879 + 0. i) t] + i \text{Sin}[(4.18879 + 0. i) t]) + \\
 & (0.00970945 + 0.0120004 i) \\
 & (\text{Cos}[(8.37758 + 0. i) t] - i \text{Sin}[(8.37758 + 0. i) t]) + \\
 & (0.00970945 - 0.0120004 i) \\
 & (\text{Cos}[(8.37758 + 0. i) t] + i \text{Sin}[(8.37758 + 0. i) t])
 \end{aligned}$$

In[54]:= showNapprox[h, -1, 0.5, t, 3]



In[55]:= animateNapprox[h, -1, 1/2, t, 10]

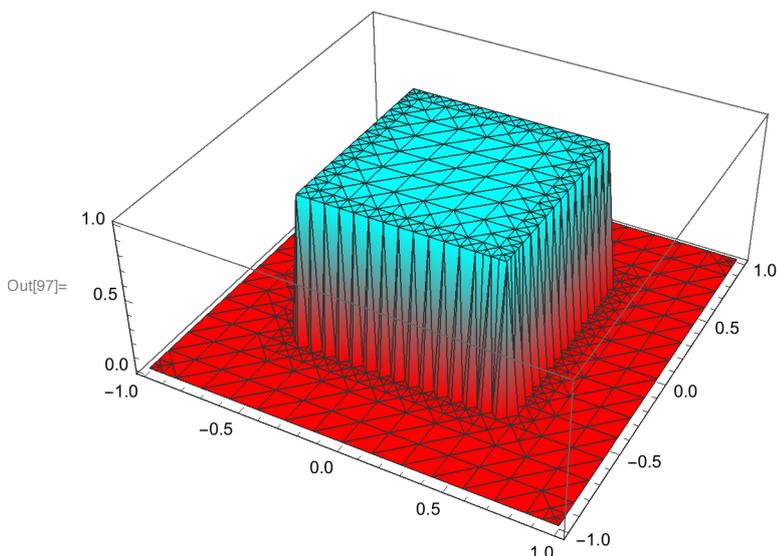


Gibbs phenomenon in 2D

The cube

```
In[96]:= cube[x_, y_] = UnitBox[x] * UnitBox[y];
```

```
In[97]:= Plot3D[cube[x, y], {x, -1, 1}, {y, -1, 1}, Mesh -> All, MeshFunctions -> Automatic,
  Exclusions -> None, ColorFunction -> Function[{x, y, z}, Hue[z/2]]]
```



In this very simple case the 2D-Fourier series is the product of the 1D-Fourier series for the x- and y-coordinates, because the object is the product of two box functions

```
In[98]:= ser1[n_, x_, y_] := ComplexExpand[FourierSeries[UnitBox[s], s, n] *
  FourierSeries[UnitBox[t], t, n]] /. {s -> x, t -> y}
```

The Fourier series terms can be computed symbolically

```
In[99]:= ser1[3, x, y]
```

Out[99]=

$$\begin{aligned} & \frac{1}{4\pi^2} + \frac{\cos[x] \sin\left[\frac{1}{2}\right]}{\pi^2} + \frac{\cos[y] \sin\left[\frac{1}{2}\right]}{\pi^2} + \frac{1}{\pi^2} 4 \cos[x] \cos[y] \sin\left[\frac{1}{2}\right]^2 + \\ & \frac{\cos[2x] \sin[1]}{2\pi^2} + \frac{\cos[2y] \sin[1]}{2\pi^2} + \frac{1}{\pi^2} 2 \cos[2x] \cos[y] \sin\left[\frac{1}{2}\right] \sin[1] + \\ & \frac{1}{\pi^2} 2 \cos[x] \cos[2y] \sin\left[\frac{1}{2}\right] \sin[1] + \frac{1}{\pi^2} \cos[2x] \cos[2y] \sin[1]^2 + \\ & \frac{\cos[3x] \sin\left[\frac{3}{2}\right]}{3\pi^2} + \frac{\cos[3y] \sin\left[\frac{3}{2}\right]}{3\pi^2} + \frac{1}{3\pi^2} 4 \cos[3x] \cos[y] \sin\left[\frac{1}{2}\right] \sin\left[\frac{3}{2}\right] + \\ & \frac{1}{3\pi^2} 4 \cos[x] \cos[3y] \sin\left[\frac{1}{2}\right] \sin\left[\frac{3}{2}\right] + \frac{1}{3\pi^2} 2 \cos[3x] \cos[2y] \sin[1] \sin\left[\frac{3}{2}\right] + \\ & \frac{1}{3\pi^2} 2 \cos[2x] \cos[3y] \sin[1] \sin\left[\frac{3}{2}\right] + \frac{1}{9\pi^2} 4 \cos[3x] \cos[3y] \sin\left[\frac{3}{2}\right]^2 \end{aligned}$$

But for display it is better to use numerical values

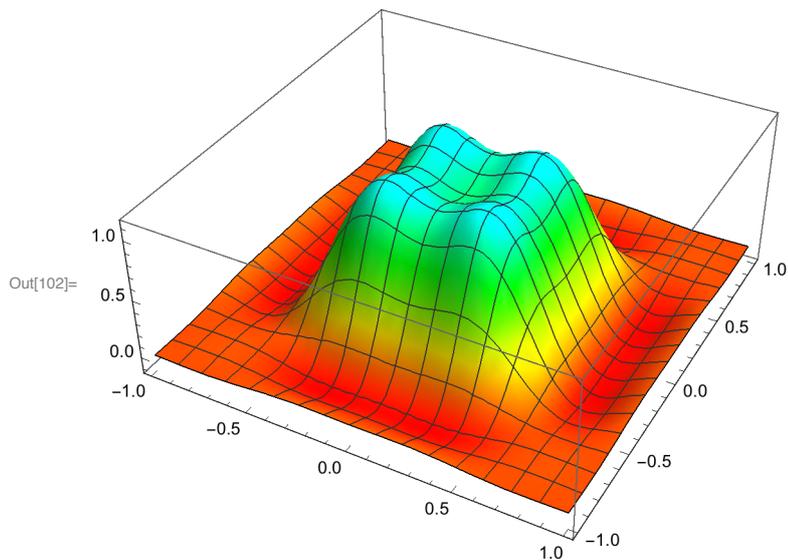
```
In[100]:= nser1[n_, x_, y_] := N[ExpToTrig[ser1[n, x, y]]]
```

In[101]:= nser1[3, x, y]

Out[101]= $0.0253303 + 0.048576 \cos[x] + 0.0426294 \cos[2. x] +$
 $0.0336891 \cos[3. x] + 0.048576 \cos[y] + 0.0931542 \cos[x] \cos[y] +$
 $0.0817505 \cos[2. x] \cos[y] + 0.0646057 \cos[3. x] \cos[y] + 0.0426294 \cos[2. y] +$
 $0.0817505 \cos[x] \cos[2. y] + 0.0717428 \cos[2. x] \cos[2. y] +$
 $0.0566968 \cos[3. x] \cos[2. y] + 0.0336891 \cos[3. y] + 0.0646057 \cos[x] \cos[3. y] +$
 $0.0566968 \cos[2. x] \cos[3. y] + 0.0448063 \cos[3. x] \cos[3. y]$

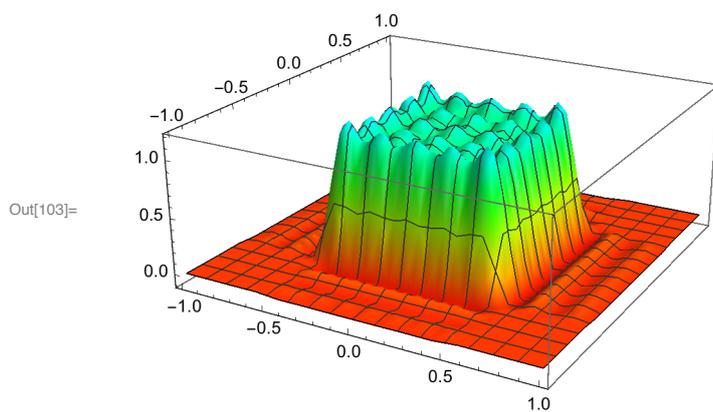
Approximation of order $N=10$

In[102]:= Plot3D[Evaluate[nser1[10, x, y]], {x, -1, 1}, {y, -1, 1},
 Exclusions \rightarrow None, ColorFunction \rightarrow Function[{x, y, z}, Hue[$\frac{z}{2}$]]]



Approximation of order $N=30$

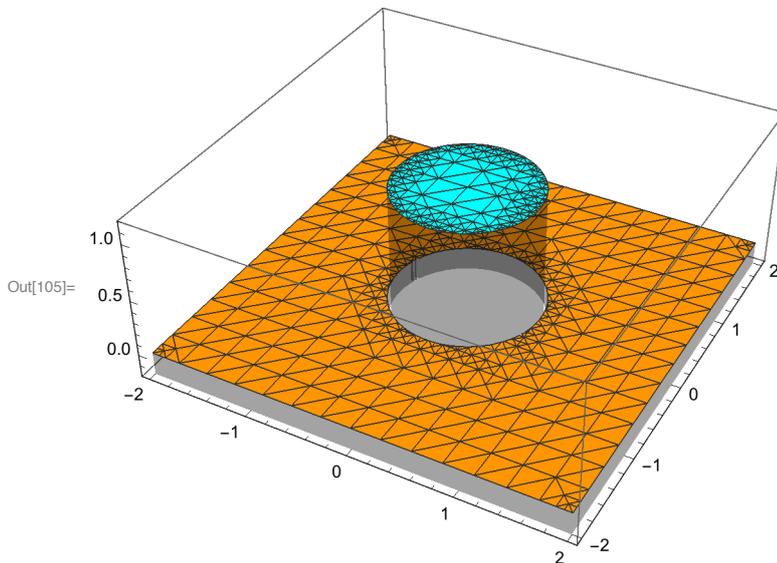
In[103]:= Plot3D[Evaluate[nser1[30, x, y]], {x, -1, 1}, {y, -1, 1},
 Exclusions \rightarrow None, ColorFunction \rightarrow Function[{x, y, z}, Hue[$\frac{z}{2}$]]]



The cylinder

```
In[104]:= cylinder[x_, y_] = UnitBox[x^2 + y^2];
```

```
In[105]:= Plot3D[cylinder[x, y], {x, -2, 2}, {y, -2, 2},
  Mesh → All, MeshFunctions → Automatic, PlotRange → {-0.2, 1.2},
  Filling → Bottom, ColorFunction → Function[{x, y, z}, Hue[z/2]]]
```



The 2D-cylinder is not a product of 1D-objects, nevertheless the computation of its Fourier series can be computed easier by passing to polar coordinates

```
In[106]:= int1 = Assuming[r ∈ Reals && s ∈ Reals && t ∈ Reals,
  Integrate[Exp[-Pi I r (s Cos[θ] + t Sin[θ])], {θ, 0, 2 Pi}]]
```

Out[106]= $2 \pi \text{BesselJ}[0, \pi \sqrt{s^2 + t^2} \text{Abs}[r]]$

```
In[107]:= int2 = Integrate[int1 * r, {r, 0, 1/2}]
```

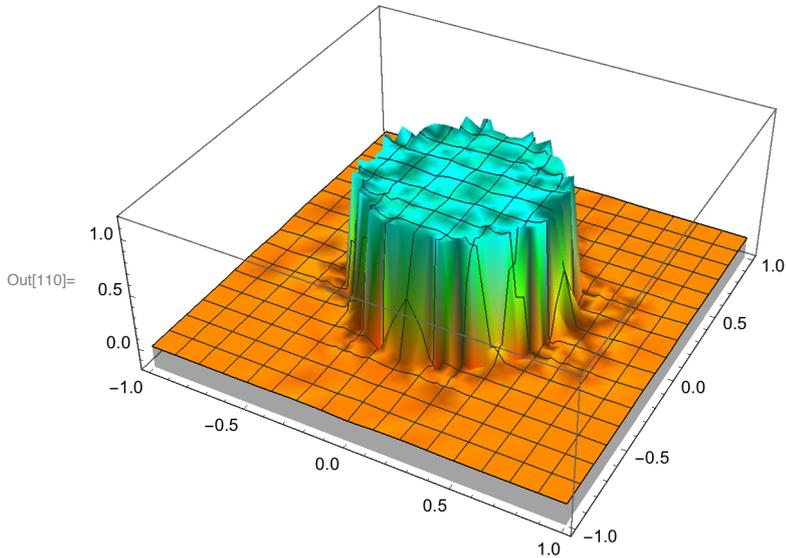
Out[107]= $\frac{1}{4} \pi \text{Hypergeometric0F1Regularized}[2, -\frac{1}{16} \pi^2 (s^2 + t^2)]$

```
In[108]:= ser2[n_, x_, y_] := Sum[
  int2 * Exp[Pi I x s + Pi I y t], {s, -n, n}, {t, -n, n}]
```

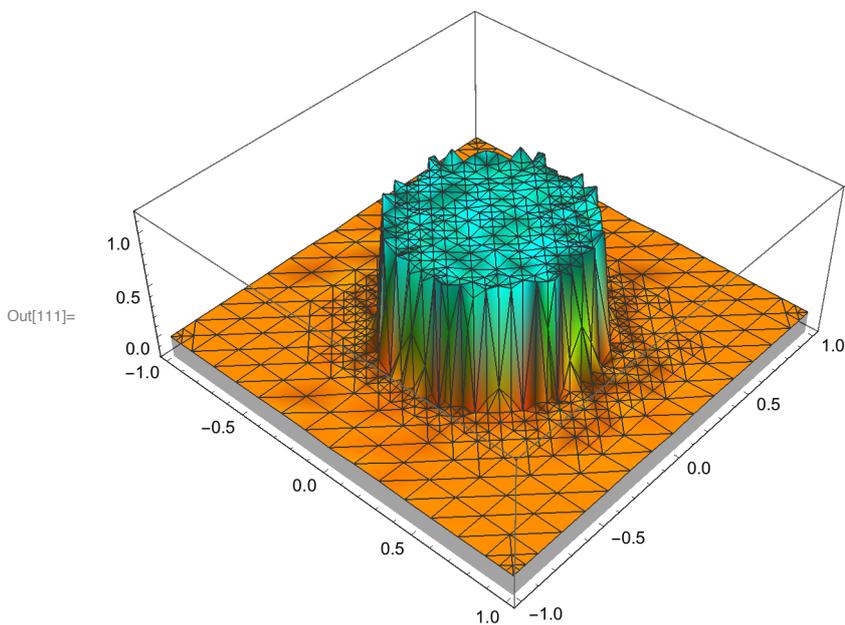
```
In[109]:= nser2[n_, x_, y_] := N[ExpToTrig[ser2[n, x, y]]]
```

Approximation with $N=50$ shows almost a fractal-like object due to the Gibbs phenomenon

```
In[110]:= Plot3D[Evaluate[nser2[50, x, y]/4], {x, -1, 1},
  {y, -1, 1}, Exclusions -> None, PlotRange -> {-0.2, 1.2},
  Filling -> Bottom, ColorFunction -> Function[{x, y, z}, Hue[z/2]]]
```

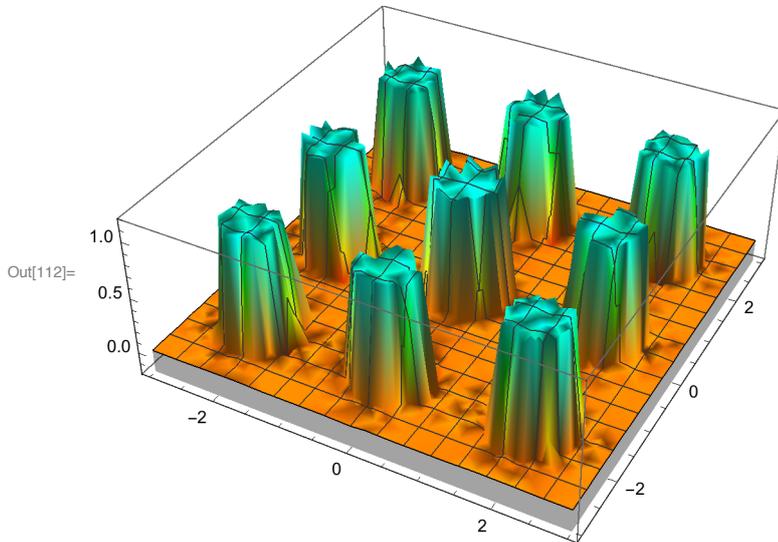


```
In[111]:= Plot3D[Evaluate[nser2[50, x, y]/4], {x, -1, 1}, {y, -1, 1}, Mesh -> All,
  MeshFunctions -> Automatic, Exclusions -> None, PlotRange -> {-0.2, 1.2},
  Filling -> Bottom, ColorFunction -> Function[{x, y, z}, Hue[z/2]]]
```



The approximation is indeed a 2-periodic function in both x- and y-directions

```
In[112]:= Plot3D[ Evaluate[nser2[50, x, y] / 4], {x, -3, 3},  
  {y, -3, 3}, Exclusions -> None, PlotRange -> {-0.2, 1.2},  
  Filling -> Bottom, ColorFunction -> Function[{x, y, z}, Hue[z/2]]]
```



4. Fouriertransformation

4.1 Heuristik

Betrachtet man eine Funktion $f : [-L, L] \rightarrow \mathbb{C}$, so stellt deren Fourierreihe

$$f(t) \sim \sum_{n \in \mathbb{Z}} c[n] e^{2\pi i n t / (2L)} \quad \text{mit} \quad c[n] = \frac{1}{2L} \int_{-L}^L f(t) e^{-2\pi i t n / (2L)} dt$$

eine $2L$ -periodische Funktion dar, die periodische Fortsetzung von $f(t)$ zu einer Funktion auf \mathbb{R} . Betrachtet man die zweiseitig unendliche Folge der Werte $\{\hat{f}[n/(2L)]\}_{n \in \mathbb{Z}}$ mit

$$\hat{f}\left[\frac{n}{2L}\right] = \int_{-L}^L f(t) e^{-2\pi i t n / (2L)} dt = 2L \cdot c[n],$$

so konvergieren diese Graphen diskreter Funktionen gegen den Graphen einer kontinuierlichen Funktion $\hat{f} : \mathbb{R} \rightarrow \mathbb{C}$, die entsprechend als

$$\hat{f}(s) = \int_{\mathbb{R}} f(t) e^{-2\pi i s t} dt$$

zu beschreiben ist. Die ursprüngliche Fourierreihe schreibt sich damit

$$f(t) \sim \frac{1}{2L} \sum_{n \in \mathbb{Z}} \hat{f}\left[\frac{n}{2L}\right] e^{2\pi i t \frac{n}{2L}}$$

und die rechte Seite kann man als Riemann-Summe für das Integral

$$\int_{\mathbb{R}} \hat{f}(s) e^{2\pi i t s} ds$$

lesen. Man kann also erwarten, dass sich aus der Funktion \widehat{f} die Funktion f wieder gewinnen lässt, symbolisch:

$$\widehat{f}(s) = \int_{\mathbb{R}} f(t) e^{-2\pi i s t} dt \iff f(t) = \int_{\mathbb{R}} \widehat{f}(s) e^{2\pi i t s} ds.$$

Das ist die Idee von Fouriertransformation und Fourier-Rücktransformation. Wenn diese Beziehung besteht, sind die Funktion $f(t)$ und ihre Fourier-Transformierte $\widehat{f}(s)$ gleichwertige Darstellungen eines Objekts, einmal im ‘‘Ortsbereich’’ (oder ‘‘Zeitbereich’’, je nachdem) und das andere Mal im ‘‘Frequenzbereich’’. $f(t)$ wird dargestellt als (kontinuierliche) Überlagerung von harmonischen Schwingungen $e^{2\pi i t s}$ mit der jeweiligen Amplitude $\widehat{f}(s)$. Daher auch die Bezeichnung *harmonische Analyse* bzw. *harmonische Synthese*.

4.2 Definition der Fouriertransformation

Definition 4.1 (Fouriertransformation).

1. Für eine Funktion $f \in \mathcal{L}^1(\mathbb{R})$ ist die durch

$$\widehat{f} : s \mapsto \int_{\mathbb{R}} f(t) e^{-2\pi i s t} dt$$

definierte Funktion die Fouriertransformierte von f .

2. Die lineare Abbildung $\mathcal{F} : f \mapsto \widehat{f} = \mathcal{F}(f)$ ist die (kontinuierliche) Fouriertransformation.

Wichtig: Die Voraussetzung $f \in \mathcal{L}^1$ garantiert, dass das Integral in der Definition von $\widehat{f}(t)$ für jedes $t \in \mathbb{R}$ existiert und endlich ist. $\widehat{f}(t)$ ist also eine auf \mathbb{R} wohldefinierte Funktion. Mehr noch, es gilt:

Satz 4.1 — Riemann-Lebesgue.

1. Für $f(t) \in \mathcal{L}^1(\mathbb{R})$ ist $\widehat{f}(s)$ eine gleichmässig stetige Funktion auf \mathbb{R} . $\widehat{f}(s)$ muss aber selbst nicht wieder zu $\mathcal{L}^1(\mathbb{R})$ gehören.
2. Für $f(t) \in \mathcal{L}^1(\mathbb{R})$ gilt $\lim_{|s| \rightarrow \infty} \widehat{f}(s) = 0$.

Die erste Aussage ergibt sich leicht aus der Definition, für die zweite muss man etwas mehr arbeiten: Man beweist diese Aussage zuerst für Funktionen des Typs $\mathbf{1}_{[a,b]}(t)$, dann für endliche Linearkombinationen (als Treppenfunktionen mit kompakten Träger), dann approximiert man mit solchen Funktionen beliebige stetige Funktionen mit kompakten Träger, und schliesslich mit C_c^0 -Funktionen alle Funktionen aus $\mathcal{L}^1(\mathbb{R})$.

Wichtige Transformationen von Funktionen sind Skalierung, Translation, Modulation, Konjugation, die \mathcal{L}^1 -Funktionen wieder in \mathcal{L}^1 -Funktionen abbilden. Skalierung und Translation sind die grundlegenden Transformationen bei der Definition von Wavelets, daher gehören Aussagen über das Zusammenspiel solcher Transformationen mit der Fouriertransformation zum elementaren Handwerkszeug.

Definition 4.2. Für Funktionen $f : \mathbb{R} \rightarrow \mathbb{C}$ und Konstante $a > 0, b, c \in \mathbb{R}$ wird definiert:

1. die a -Skalierung (oder Dilation) $D_a f$ durch

$$(D_a f)(t) = \sqrt{a} f(at)$$

2. die b -Translation $T_b f$ durch

$$(T_b f)(t) = f(t - b)$$

3. die c -Modulation $E_c f$ durch

$$(E_c f)(t) = e^{2\pi i c t} f(t)$$

4. die Konjugation f^* durch

$$(f^*)(t) = \overline{f(t)}$$

5. die Zeitumkehr f^- durch

$$(f^-)(t) = f(-t)$$

Offensichtlich ändern diese Transformationen die Norm von \mathcal{L}^1 -Funktionen nicht:

$$\|f\| = \|D_a f\| = \|T_b f\| = \|E_c f\| = \|f^*\|.$$

Die nachfolgend aufgeführten Eigenschaften ergeben sich alle unmittelbar aus der Definition.

Satz 4.2 — Eigenschaften der \mathcal{L}^1 -Fouriertransformation.

1. Skalierung:

$$\widehat{D_a f}(s) = \frac{1}{\sqrt{a}} \widehat{f}\left(\frac{s}{a}\right) = (D_{1/a} \widehat{f})(s)$$

2. Translation:

$$\widehat{T_b f}(s) = e^{-2\pi i b s} \widehat{f}(s) = E_{-b} \widehat{f}(s)$$

3. Modulation:

$$\widehat{E_c f}(s) = \widehat{f}(s - c) = T_c \widehat{f}(s)$$

4. Konjugation:

$$\widehat{(f^*)}(s) = ((\widehat{f})^*)^-(s)$$

5. Zeitumkehr:

$$\widehat{(f^-)}(s) = (\widehat{f})^-(s)$$

6. Ferner:

$$(f^*)^* = f \quad (f^-)^- = f \quad (f^-)^* = (f^*)^- \quad \widehat{(\widehat{f})} = f^-$$

Insbesondere liefert die vierfache Hintereinanderausführung der Fouriertransformation auf eine Funktion f diese Funktion f zurück: $\mathcal{F}^4(f) = f$, in perfekter

Analogie dazu, dass für die imaginäre Einheit $i = \sqrt{-1}$ ja $i^4 = 1$ gilt.

Weitere Eigenschaften, die ebenfalls leicht herzuleiten sind (wobei vorausgesetzt wird, dass die genannten inneren Produkte existieren):

$$\begin{aligned} D_a T_b f &= T_{b/a} D_a f & \langle f, D_a g \rangle &= \langle D_{1/a} f, g \rangle & \langle f, T_b g \rangle &= \langle T_{-b} f, g \rangle \\ \langle D_a f, D_a g \rangle &= \langle f, g \rangle & \langle T_b f, T_b g \rangle &= \langle f, g \rangle & \langle f, E_c g \rangle &= \langle E_{-c} f, g \rangle \end{aligned}$$

Beachte: Wie erwähnt: für $f \in \mathcal{L}^1$ existiert das Integral in der Definition und \widehat{f} ist wohldefiniert! Aber der Wertebereich dieser Transformation ist nicht wieder der Raum $\mathcal{L}^1(\mathbb{R})$ – siehe Abschnitt 4.6, d.h. es ist nicht ohne weiteres klar, wie es um eine Rücktransformation, d.h., die Rekonstruktion von f aus \widehat{f} bestellt ist. Und ausserdem möchte man ja eigentlich eine Fouriertransformation auf dem Raum $\mathcal{L}^2(\mathbb{R})$ haben... schon wegen der Hilbertraum-Eigenschaft. Wie man dazu vorgeht, wird weiter unten erläutert.

R Wichtiger Hinweis: Für *endliche* Intervalle $I \subset \mathbb{R}$ gilt $\mathcal{L}^2(I) \subset \mathcal{L}^1(I)$, aber für unendliche Intervalle, also insbesondere $I = \mathbb{R}$, gilt das nicht mehr, vielmehr ist sowohl $\mathcal{L}^2(\mathbb{R}) \not\subset \mathcal{L}^1(\mathbb{R})$, $\mathcal{L}^1(\mathbb{R}) \not\subset \mathcal{L}^2(\mathbb{R})$!

R Wichtiger Hinweis: In verschiedenen Bereichen werden leicht unterschiedliche Definitionen der Fouriertransformation benutzt. Das Schema ist immer von der Form

$$\sqrt{\frac{|b|}{(2\pi)^{1-a}}} \int_{\mathbb{R}} f(t) e^{ibst} dt$$

Die typischen Werte für die Parameter a und b sind:

- $(a, b) = (1, -1)$ in der Mathematik und Systemtheorie
- $(a, b) = (-1, 1)$ in der “klassischen” Physik
- $(a, b) = (0, 1)$ in der “modernen” Physik
- $(a, b) = (0, -2\pi)$ in der Signalverarbeitung

Bei Verwendung von Software sollte man sich immer zunächst darüber informieren, welches die jeweilige Konvention (default) ist. Beispielsweise verwendet **Mathematica** die Konvention $(a, b) = (0, 1)$ und **Maple** die Konvention $(a, b) = (1, -1)$. Besonders zu beachten ist, dass die Formeln für die Rücktransformation entsprechend anzupassen sind!

■ **Beispiel 4.1** Abbildung 4.1 enthält einige häufig verwendete Funktionen und ihre Fouriertransformierten. In Formelsammlungen zur Physik, Nachrichtentechnik, Systemtheorie etc. findet man umfangreiche Tabellen dazu. In den Beispielen ist a ein reeller Parameter mit $a > 0$. Mit $\mathbf{1}_I(t)$ wird die Indikatorfunktion eines Intervalles $I \subset \mathbb{R}$ bezeichnet.

$$\mathbf{1}_I(t) = \begin{cases} 1 & t \in I \\ 0 & t \notin I \end{cases}$$

Das erste Beispiel zeigt übrigens, dass die Fouriertransformierte \widehat{f} einer \mathcal{L}^1 -Funktion nicht wieder zu \mathcal{L}^1 gehören muss: die Funktion $\text{sinc}(s) = \sin(\pi s)/(\pi s)$ ist nicht integrierbar im \mathcal{L}^1 -Sinne.

$$\begin{array}{lll}
f(t) & \rightsquigarrow & \widehat{f}(s) \\
\mathbf{1}_{[-a,a]}(t) & \rightsquigarrow & \frac{\sin(2\pi as)}{\pi s} \\
(1 - \lfloor \frac{t}{a} \rfloor) \cdot \mathbf{1}_{[-a,a]}(t) & \rightsquigarrow & \frac{\sin^2(2\pi as)}{a(\pi s)^2} \\
\mathbf{1}_{[0,\infty)} e^{-at} & \rightsquigarrow & \frac{1}{a + (2\pi s)} \\
e^{-a|t|} & \rightsquigarrow & \frac{2a}{a^2 + (2\pi s)^2} \\
e^{-at^2} & \rightsquigarrow & \sqrt{\frac{\pi}{a}} e^{-(\pi s)^2/a}
\end{array}$$

Abbildung 4.1: Einige Beispiele für Fouriertransformierte

Man beachte bei den Beispielen, was bei Variation des Parameters $a > 0$ geschieht: wenn $f(t)$ “gestaucht” wird, wird $\widehat{f}(s)$ “auseinandergezogen”, und umgekehrt. “Lokalisierung im “Ortsbereich” und “Lokalisierung im Frequenzbereich” sind antagonistisch voneinander abhängig! Dieses grundlegende Phänomen wird im Abschnitt zur Unschärferelation quantitativ gefasst.

Zu beachten ist weiterhin, welche Funktionen hier und in ähnlichen Tabellen *nicht* auftreten: beispielweise Polynome, trigonometrische Funktionen, Logarithmus und Exponentialfunktionen usw.... Der Grund ist: diese Funktionen gehören nicht zum Raum $\mathcal{L}^1(\mathbb{R})$ (und auch nicht zu $\mathcal{L}^2(\mathbb{R})$). Tatsächlich kann man den Bereich der Fourier-transformierbaren Funktionen erheblich erweitern, das geschieht dann aber (notwendig) dadurch, dass man den betrachteten Bereich auf Objekte ausdehnt, die keine Funktionen im traditionellen Sinn mehr sind, sondern *verallgemeinerte Funktionen*, auch *Distributionen* genannt. Im Zusammenhang mit Sampling wird es hier eine kleine Kostprobe davon geben. ■

4.3 Fouriertransformation und Faltung

Definition 4.3 (Faltung).

Sind $f, g : \mathbb{R} \rightarrow \mathbb{C}$ Funktionen, so ist deren Faltung $f \star g$ definiert durch

$$f \star g : s \mapsto \int_{\mathbb{R}} f(t) g(s - t) dt,$$

sofern das Integral existiert.

Sind $f, g : \mathbb{R} \rightarrow \mathbb{C}$ zwei Funktionen, so ist die Definiertheit von $f \star g$ und spezieller die Zugehörigkeit von $f \star g$ zu einer bestimmten Klasse von Funktionen

in Abhängigkeit von den Eigenschaften von f und g nicht immer ganz einfach zu klären. Hier sind einige Fälle aufgelistet.

Theorem 4.1 — Faltung.

1. Für $f \in \mathcal{L}^\infty(\mathbb{R})$, $g \in \mathcal{L}^1(\mathbb{R})$ oder $f, g \in \mathcal{L}^2(\mathbb{R})$, ist $f \star g \in C^0(\mathbb{R})$ (und sogar gleichmässig stetig).
2. Für $f, g \in \mathcal{L}^1(\mathbb{R})$, ist auch $f \star g \in \mathcal{L}^1(\mathbb{R})$ und $\|f \star g\|_1 \leq \|f\|_1 \cdot \|g\|_1$.
3. Für $f \in \mathcal{L}^2(\mathbb{R})$, $g \in \mathcal{L}^1(\mathbb{R})$ ist $f \star g \in \mathcal{L}^2(\mathbb{R})$ und $\|f \star g\|_2 \leq \|f\|_2 \cdot \|g\|_1$.
4. Für $f, g \in \mathcal{L}^2(\mathbb{R})$ ist $f \star g \in \mathcal{L}^\infty(\mathbb{R})$ und $\|f \star g\|_\infty \leq \|f\|_2 \cdot \|g\|_2$.
5. Für $f \in \mathcal{L}^\infty(\mathbb{R})$, $g \in \mathcal{L}^1(\mathbb{R})$ ist $f \star g \in \mathcal{L}^\infty(\mathbb{R})$ und $\|f \star g\|_\infty \leq \|f\|_\infty \cdot \|g\|_1$.

Sind $f, g \in \mathcal{L}^1(\mathbb{R})$ und haben beide kompakten Träger, so ist $f \star g \in \mathcal{L}^1(\mathbb{R})$ und hat kompakten Träger. Eine analoge Aussage gilt für $\mathcal{L}^2(\mathbb{R})$.

Das folgende *Faltungstheorem* ist die Grundlage für eine Technik, die als “Filtern im Frequenzbereich” in jeglicher Signalverarbeitung weit verbreitet ist.

Theorem 4.2 — Faltungstheorem.

Sind $f, g \in \mathcal{L}^1(\mathbb{R})$, so ist auch $f \star g \in \mathcal{L}^1(\mathbb{R})$ und es gilt

$$\widehat{f \star g}(s) = \widehat{f}(s) \cdot \widehat{g}(s).$$

Der Beweis beruht wiederum “nur” auf der Vertauschung von Integralen:

$$\begin{aligned} \widehat{f \star g}(s) &= \int_{\mathbb{R}} (f \star g)(t) e^{-2\pi i s t} dt \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(x) g(t-x) e^{-2\pi i s t} dx dt \\ &= \int_{\mathbb{R}} f(x) e^{-2\pi i x s} \int_{\mathbb{R}} g(t-x) e^{-2\pi i s (t-x)} dt dx \\ &= \widehat{g}(s) \cdot \int_{\mathbb{R}} f(x) e^{-2\pi i x s} dx \\ &= \widehat{g}(s) \cdot \widehat{f}(s) \quad \square \end{aligned}$$

Eine schematische Darstellung des Faltungstheorems als “kommutatives Diagramm” sieht so aus:

$$\begin{array}{ccc} f, g & \xrightarrow{\mathcal{F}} & \widehat{f}, \widehat{g} \\ \star \downarrow & & \downarrow \cdot \\ f \star g & \xrightarrow{\mathcal{F}} & \widehat{f \star g} = \widehat{f} \cdot \widehat{g} \end{array}$$

Dabei bezeichnet \mathcal{F} den Operator *Fouriertransformation*, \star den Operator *Faltung* und \cdot den Operator *punktweises Produkt*. Das Diagramm besagt, dass die beiden Wege von “links oben” nach “rechts unten”, also entweder *erst Falten, dann Fouriertransformation*, oder *erst Fouriertransformation, dann punktweises Produkt*, gleichwertig sind.

In der Anwendung “Filtern im Frequenzbereich” liest man das Diagramm so (unter der Voraussetzung, dass die Fouriertransformation für die beteiligten Objekte (Funktionen) umkehrbar ist):

$$\begin{array}{ccc} f, g & \xrightarrow{\mathcal{F}} & \widehat{f}, \widehat{g} \\ \star \downarrow & & \downarrow \cdot \\ f \star g & \xleftarrow{\mathcal{F}^{-1}} & \widehat{f \star g} = \widehat{f} \cdot \widehat{g} \end{array}$$

Das ist so zu verstehen: in der linken Spalte befinden sich die Objekte $f, g, f \star g$ im “Ortsbereich” (oder “Zeitbereich”). Das Ziel ist, die Filteroperation $(f, g) \mapsto f \star g$ auszuführen — das ist die Situation des Filterns. Anstatt dies direkt zu berechnen, kann man den Umweg über den “Frequenzbereich” (Spektren) in der rechten Spalte des Diagramms mit den Objekten \widehat{f}, \widehat{g} und $\widehat{f \star g}$ gehen, indem man

1. mittels Fouriertransformation $\mathcal{F} : f \mapsto \widehat{f}$ und $\mathcal{F} : g \mapsto \widehat{g}$ zu f bzw. g äquivalente Objekte \widehat{f} bzw. \widehat{g} berechnet;
2. das punktweise Produkt $\widehat{f} \cdot \widehat{g}$ berechnet, das nach Faltungstheorem mit $\widehat{f \star g}$ identisch ist;
3. mittels Fourier-Rücktransformation schliesslich $\mathcal{F}^{-1} : \widehat{f \star g} \mapsto f \star g$ berechnet.

Dieser “Umweg” lohnt sich (effizientmässig betrachtet), wenn Fouriertransformation eine “schnelle” Operation – wie es beispielsweise bei der Schnellen Fouriertransformation (FFT) der Fall ist. Punktweise Produkte sind ohnehin “schnell” zu berechnen, aber die Faltung, ihrer Definition entsprechend ausgeführt, ist eine kostspielige Operation.

4.4 Inversion: Fourier-Rücktransformation

Was man sich wünscht ist, dass man eine Funktion $f \in \mathcal{L}^1$ tatsächlich mittels der Inversionsformel

$$f(t) = \int_{\mathbb{R}} \widehat{f}(s) e^{2\pi i s t} ds$$

rekonstruieren kann. Das ist im ersten Anlauf zu optimistisch gedacht, denn die Funktion $\widehat{f}(s)$ muss ja keineswegs integrierbar sein – Vorsicht ist also angebracht!

Immerhin, unter der Voraussetzung der Stetigkeit von f gilt das Gewünschte, was man unter Verwendung des “regularisierenden” Gauss-Kerns

$$K_r(t) = \frac{1}{r} \cdot e^{-\pi(t/r)^2}$$

zeigen kann, vgl. Theorem 3.6.

Satz 4.3 Für stetiges $f \in \mathcal{L}^1(\mathbb{R})$ gilt für alle $t \in \mathbb{R}$

$$f(t) = \lim_{r \downarrow 0} \int_{\mathbb{R}} \widehat{f}(s) e^{-\pi r^2 s^2} e^{2\pi i s t} ds.$$

Der Beweis dieses Satzes soll skizziert werden, da er die Verwendung von Summationskernen zeigt. Ohne hier auf die Berechtigung der Vertauschung der Integrale genauer einzugehen, geht man so vor:

$$\begin{aligned}
 \int_{\mathbb{R}} \widehat{f}(s) e^{-\pi r^2 s^2} e^{2\pi i s t} ds &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(x) e^{-2\pi i x s} dx e^{-\pi r^2 s^2} e^{2\pi i s t} ds \\
 &= \int_{\mathbb{R}} f(x) \int_{\mathbb{R}} e^{-2\pi i s(t-x)} e^{-\pi r^2 s^2} ds dx \\
 &= \int_{\mathbb{R}} f(x) \frac{1}{r} e^{-\pi(t-x)^2/r^2} dx \\
 &= \int_{\mathbb{R}} f(x) K_r(t-x) dx \xrightarrow{r \downarrow 0} f(t) \quad \square
 \end{aligned}$$

Folgerung 4.1. Für stetiges $f \in \mathcal{L}^1(\mathbb{R})$ ist $\widehat{f} \in \mathcal{L}^1(\mathbb{R})$ und für alle $t \in \mathbb{R}$ gilt

$$f(t) = \int_{\mathbb{R}} \widehat{f}(s) e^{2\pi i s t} ds.$$

Entsprechend kann man zeigen:

Satz 4.4 Für $f \in \mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$ ist $\widehat{f} \in \mathcal{L}^2(\mathbb{R})$ und es gilt

$$f(t) = \lim_{r \downarrow 0} \int_{\mathbb{R}} \widehat{f}(s) e^{-\pi r^2 s^2} e^{2\pi i s t} ds$$

im Sinne der \mathcal{L}^2 -Konvergenz.

Das ist ein wichtiger Schritt, der zu folgender fundamentalen Aussage über die Fouriertransformation als “isometrischer Transformation” führt:

Theorem 4.3 — Parseval-Plancherel.

Skalarprodukte und \mathcal{L}^2 -Norm bleiben unter Fouriertransformation erhalten, d.h., die Fouriertransformation ist eine *isometrische* (oder anders gesagt: *unitäre*) Transformation, genauer

1. Für $f \in \mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$ ist $\widehat{f} \in \mathcal{L}^2(\mathbb{R})$ und es gilt Plancherels Formel:

$$\|f\|_2 = \|\widehat{f}\|_2.$$

2. Für $f, g \in \mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$ gilt Parsevals Formel:

$$\langle \widehat{f}, \widehat{g} \rangle = \langle f, g \rangle.$$

Die \mathcal{L}^2 -Norm bezeichnet man oft auch als “Energie”, spricht von \mathcal{L}^2 -Funktionen als “Signalen mit endlicher Energie”. Plancherels Formel kann als “Energieerhaltungssatz” angesehen werden.

Zum Beweis der Formel von Plancherel:

Man definiert $\tilde{f}(t) = \overline{f(-t)}$ und bemerkt $\widehat{\tilde{f}}(s) = \overline{\widehat{f}(s)}$. Das Faltungstheorem ergibt

$$\widehat{(f \star \tilde{f})}(s) = \widehat{f}(s) \cdot \widehat{\tilde{f}}(s) = |\widehat{f}(s)|^2$$

Die Stetigkeit von $f \star \tilde{f}$ (siehe Theorem 4.1) liefert dann

$$\lim_{\tau \rightarrow 0^+} \int_{\mathbb{R}} \widehat{(f \star \tilde{f})}(s) e^{-\pi\tau^2 s^2} e^{2\pi i s t} ds = (f \star \tilde{f})(t) = \int_{\mathbb{R}} f(x) \tilde{f}(t-x) dx$$

und insbesondere für $t = 0$ ergibt das

$$\|f\|_2^2 = \lim_{\tau \rightarrow 0^+} \int_{\mathbb{R}} \widehat{(f \star \tilde{f})}(s) e^{-\pi\tau^2 s^2} ds = (f \star \tilde{f})(0) = \int_{\mathbb{R}} |f(x)|^2 dx = \|\widehat{f}\|_2^2 \quad \square$$

4.5 Fouriertransformation und Differenzierbarkeit

Die folgende fundamentale Eigenschaft der Fouriertransformation, die das ‘‘Verschwinden im Unendlichen’’ der Transformierten einer Funktion aus \mathcal{L}^1 betrifft, wurde bereits erwähnt: siehe Punkt 2 in Satz 4.1.

Satz 4.5 — Riemann-Lebesgue-Lemma.

Ist $f \in \mathcal{L}^1(\mathbb{R})$, so ist die Funktion $\widehat{f}(s)$ auf \mathbb{R} gleichmässig stetig und es gilt

$$\lim_{|s| \rightarrow \infty} \widehat{f}(s) = 0.$$

Das ist aber nur die Spitze eines Eisberges: Generell gilt, dass die Glattheitseigenschaft (Differenzierbarkeit höhere Ordnung) einer Funktion $f(t)$ mit der Schnelligkeit des Verschwindens im Unendlichen von $\widehat{f}(s)$ korrespondiert. Davon wird später im Rahmen der Wavelet-Theorie Gebrauch gemacht.

Zum Verständnis des folgenden Satzes: die Aussage ‘‘ $t^k \cdot f(t) \in \mathcal{L}^1(\mathbb{R})$ ’’ bedeutet, dass $\int_{\mathbb{R}} |t^k \cdot f(t)| dt < \infty$ ist. Je grösser hierbei k ist, desto schneller muss die Funktion $|f(t)|$ für $t \rightarrow \infty$ verschwinden, damit das genannte Integral noch endlich sein kann.

Satz 4.6 — Fouriertransformation und Ableitung.

1. Ist $t^k \cdot f(t) \in \mathcal{L}^1(\mathbb{R})$ für $0 \leq k \leq n$, dann ist $\widehat{f}(s) \in \mathcal{C}^n$ und

$$\mathcal{F}[t^k \cdot f(t)](s) = \left(\frac{i}{2\pi}\right)^k \widehat{f}^{(k)}(s) \quad (1 \leq k \leq n)$$

2. Ist $f(t) \in \mathcal{C}^n$ und $f^{(k)}(t) \in \mathcal{L}^1(\mathbb{R})$ für $0 \leq k \leq n$, dann ist

$$\mathcal{F}[f^{(k)}](s) = (2\pi i s)^k \cdot \widehat{f}(s) \quad (1 \leq k \leq n)$$

4.6 \mathcal{L}^2 -Fouriertransformation

Ein ärgerliches Problem bei der Fouriertransformation ist die Tatsache, dass die Fouriertransformierte \widehat{f} einer \mathcal{L}^1 -Funktion f selbst nicht mehr zu \mathcal{L}^1 gehören muss, z.B.

$$\mathcal{L}^1(\mathbb{R}) \ni f(t) = \mathbf{1}_{[-1/2, 1/2]}(t) \quad \rightsquigarrow \quad \widehat{f}(s) = \frac{\sin(\pi s)}{\pi s} \notin \mathcal{L}^1(\mathbb{R})$$

Man hätte natürlich gerne

$$\int_{\mathbb{R}} \frac{\sin(\pi s)}{\pi s} e^{2\pi i s t} ds = \mathbf{1}_{[-1/2, 1/2]}(t).$$

Um das zu erreichen, um also die Fouriertransformation und ihre Umkehrung auf demselben Raum, nämlich $\mathcal{L}^2(\mathbb{R})$ zu definieren, kann man wie folgt vorgehen:

1. Man betrachtet den Vektorraum $\mathcal{K} = \mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$. Dieser Unterraum von $\mathcal{L}^2(\mathbb{R})$ ist *dicht* in $\mathcal{L}^2(\mathbb{R})$, d.h., jedes $f \in \mathcal{L}^2(\mathbb{R})$ lässt sich beliebig gut in der \mathcal{L}^2 -Metrik durch Elemente von \mathcal{K} approximieren.
2. Auf \mathcal{K} ist – wegen $\mathcal{K} \subset \mathcal{L}^1(\mathbb{R})$ – die Fouriertransformation $\mathcal{F} : \mathcal{K} \rightarrow \mathcal{L}^2(\mathbb{R})$ als eine isometrische lineare Transformation definiert, siehe Theorem 4.3.
3. Wegen der Isometrieeigenschaft von \mathcal{F} ist für jede Cauchyfolge $(f_n)_{n \geq 0}$ in \mathcal{K} die Folge der Fouriertransformierten $(\mathcal{F}(f_n))_{n \geq 0}$ eine Cauchyfolge in $\mathcal{L}^2(\mathbb{R})$.
4. Wegen der Dichtheit von \mathcal{K} in $\mathcal{L}^2(\mathbb{R})$ gibt es für jedes $f \in \mathcal{L}^2(\mathbb{R})$ eine \mathcal{K} -Cauchyfolge $(f_n)_{n \geq 0} \rightarrow f$. Als $\mathcal{F}(f) \in \mathcal{L}^2(\mathbb{R})$ definiert man dann den \mathcal{L}^2 -Limes der Cauchyfolge $(\mathcal{F}(f_n))_{n \geq 0}$.

(Man muss natürlich zeigen, dass dieser Limes wohldefiniert ist, also nicht von der Wahl der Cauchyfolge $(f_n)_{n \geq 0}$ abhängt, aber das ist Routine-Technik.)

Damit hat man eine lineare Isometrie

$$\mathcal{F} : \mathcal{L}^2(\mathbb{R}) \rightarrow \mathcal{L}^2(\mathbb{R})$$

definiert, die die Eigenschaften der \mathcal{L}^1 -Fouriertransformation hat und diese in natürlicher Weise fortsetzt (und deshalb auch als Fouriertransformation bezeichnet wird, obwohl J.B. FOURIER wenig von dem mathematischen Apparat geahnt haben wird, der dafür erforderlich seine Ideen wahr werden zu lassen). Diese abstrakte Definition kann man konkretisieren, indem man Approximationen mittels Summationskernen, etwas dem Gauss-Kern, betrachtet.

Theorem 4.4 — \mathcal{L}^2 -Fouriertransformation.

Ist $f \in \mathcal{L}^2(\mathbb{R})$, so gibt es ein $\hat{f} \in \mathcal{L}^2(\mathbb{R})$ mit

$$\hat{f}(s) = \lim_{\tau \rightarrow 0^+} \int_{\mathbb{R}} f(t) e^{-\pi \tau^2 s^2} e^{-2\pi i s t} dt.$$

im Sinne der \mathcal{L}^2 -Konvergenz. In diesem Fall gilt auch die Plancherel-Formel

$$\|f\|_2 = \|\hat{f}\|_2$$

und eine Rücktransformationsformel

$$f(t) = \lim_{\tau \rightarrow 0^+} \int_{\mathbb{R}} \hat{f}(s) e^{-\pi \tau^2 s^2} e^{2\pi i s t} ds,$$

ebenfalls im Sinne der \mathcal{L}^2 -Konvergenz.

4.7 Heisenbergs Unschärferelation

Die berühmte, nach dem Physiker Werner HEISENBERG (1901–1976) benannte *Unschärferelation* ist eine grundlegende Aussage der Quantenphysik bekannt, ja populär geworden: Man kann sogenannte *komplementäre Observable* (Messgrößen), wie z.B. den Ort und den Impuls eines Teilchens, nicht simultan mit beliebiger Genauigkeit messen. Was weniger bekannt ist, ist die Tatsache, dass die mathematische Grundlage für eine solche Feststellung eine Eigenschaft der Fouriertransformation ist, die, für sich genommen, erst einmal garnichts mit Physik zu tun hat. Es geht ganz abstrakt um die Beschreibung von Objekten in zwei zueinander komplementären Bereichen, etwa dem “Ortsbereich” und dem “Frequenzbereich”, die über die Fouriertransformation mit einander verbunden sind. Beide Beschreibungen sind absolut gleichwertig – der “Teilchen–Welle-Dualismus” der Physik ist nur eine Konkretisierung davon. Die grundlegende Frage, wie scharf man Signale simultan im Ortsbereich und im Frequenzbereich lokalisieren kann, ist eine andere Erscheinungsform des gleichen Phänomens. Diese Frage ist wichtig, wenn man sich für die prinzipiell maximale “Leistungsfähigkeit” von Wavelets interessiert, die ja die Aufgabe haben, Signale sowohl örtlich (oder zeitlich), als auch in ihrer Frequenz zu lokalisieren. In diesem Zusammenhang verwendet man sogenannte *Heisenberg-Boxen* für die Beschreibung des maximal erreichbaren simultanen Auflösungsvermögens.

Ist $f \in \mathcal{L}^2(\mathbb{R})$, also $\|f\|_2^2 = \int_{\mathbb{R}} |f|^2 = 1$, so kann man $t \mapsto |f(t)|^2$ als eine Wahrscheinlichkeitsdichte auf \mathbb{R} interpretieren. Diese hat dann einen Erwartungswert

$$\mu(f) = \int_{\mathbb{R}} t \cdot |f(t)|^2 dt$$

und eine Varianz

$$\sigma^2(f) = \int_{\mathbb{R}} (t - \mu(f))^2 \cdot |f(t)|^2 dt = \int_{\mathbb{R}} t^2 \cdot |f(t)|^2 dt - \mu(f)^2.$$

Beachte:

$$\int_{\mathbb{R}} t^2 \cdot |f(t)|^2 dt = \langle t \cdot f(t) | t \cdot \bar{f}(t) \rangle = \|t \cdot f(t)\|_2^2,$$

so dass man also

$$\sigma^2(f) = \|t \cdot f(t)\|_2^2 - \mu(f)^2$$

schreiben kann.

Wegen Plancherels Identität ist auch $\|\widehat{f}\|_2^2 = \int_{\mathbb{R}} |\widehat{f}|^2 = 1$ und diese Wahrscheinlichkeitsdichte hat Erwartungswert und Varianz

$$\begin{aligned} \mu(\widehat{f}) &= \int_{\mathbb{R}} s \cdot |\widehat{f}(s)|^2 ds, \\ \sigma^2(\widehat{f}) &= \int_{\mathbb{R}} (s - \mu(\widehat{f}))^2 \cdot |\widehat{f}(s)|^2 ds = \int_{\mathbb{R}} s^2 \cdot |\widehat{f}(s)|^2 ds - \mu(\widehat{f})^2. \end{aligned}$$

Die Heisenberg-Relation besagt, dass das Produkt der Varianzen dieser Verteilungen nicht beliebig klein werden kann. Anders formuliert: eine Funktion und ihre Fouriertransformierte können nicht gleichzeitig beliebig gut um ihre Mittelwerte

konzentriert sein, oder: eine Funktion kann nicht gleichzeitig im Orts-/Zeitbereich und im Frequenzbereich beliebig gut lokalisiert sein!

Für das Nachfolgende beachte man noch, dass (wie aus der Wahrscheinlichkeitsrechnung vertraut) sich bei Translation um $b \in \mathbb{R}$ der Verteilung der Erwartungswert entsprechend verschiebt, d.h., für $T_b f(t) = f(t - b)$ gilt

$$\mu(T_b f) = \mu(f) + b$$

und sich die Varianz nicht ändert, d.h.,

$$\sigma^2(T_b f) = \sigma^2(f).$$

Bei Modulation ändern sich Erwartungswert und Varianz offensichtlich überhaupt nicht:

$$\mu(E_c f) = \mu(f), \quad \sigma^2(E_c f) = \sigma^2(f).$$

Theorem 4.5 — Heisenbergs Unschärfelrelation.

Für $f \in \mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$ mit $\|f\|_2 = 1$ gilt

$$\sigma^2(f) \cdot \sigma^2(\hat{f}) \geq \frac{1}{16\pi^2}.$$

Gleichheit gilt genau im Fall von Gaussverteilungen.

Zum *Beweis* betrachte man das Integral

$$I = \int_{\mathbb{R}} t \cdot \frac{d}{dt} |f(t)|^2 dt = \int_{\mathbb{R}} t \cdot \frac{d}{dt} (f(t) \cdot \overline{f(t)}) dt.$$

Einerseits erhält man durch Differenzieren

$$\begin{aligned} I &= \int_{\mathbb{R}} t \cdot f(t) \cdot \overline{f'(t)} + \int_{\mathbb{R}} t \cdot f'(t) \cdot \overline{f(t)} dt \\ &= 2 \Re \left(\int_{\mathbb{R}} t \cdot f(t) \cdot \overline{f'(t)} dt \right) \\ &= 2 \Re \langle t \cdot f(t) | f'(t) \rangle. \end{aligned}$$

Andererseits erhält man mittels partieller Integration¹

$$I = \underbrace{|t \cdot |f(t)|^2|_{-\infty}^{\infty}}_{=0} - \int_{\mathbb{R}} |f(t)|^2 dt = -\|f\|_2^2.$$

Für ein f mit $\|f\|_2^2 = 1$ gilt also

$$|\Re \langle t \cdot f(t) | f'(t) \rangle| = \frac{1}{2}$$

¹ An dieser Stelle wird die Eigenschaft $\lim_{t \rightarrow \pm\infty} t \cdot |f(t)|^2 = 0$ verwendet. Das muss für Funktionen $f \in \mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$ nicht unbedingt gelten, gilt aber sicher für stetige Funktionen mit kompaktem Träger. Man kann das Ergebnis dann mittels einer Grenzwertbetrachtung für alle Funktionen aus $\mathcal{L}^1(\mathbb{R}) \cap \mathcal{L}^2(\mathbb{R})$ zeigen, wenn man es erst einmal für diese eingeschränkte Funktionenklasse bewiesen hat.

und somit insbesondere

$$|\langle t \cdot f(t) | f'(t) \rangle| \geq \frac{1}{2}.$$

Die Cauchy-Schwarz-Ungleichung besagt aber

$$|\langle t \cdot f(t) | f'(t) \rangle| \leq \|t \cdot f(t)\|_2 \cdot \|f'(t)\|_2$$

und die Eigenschaften der Fouriertransformation ergeben

$$\|f'(t)\|_2 = \|\widehat{f'(s)}\|_2 = 2\pi \|s \cdot \widehat{f(s)}\|_2$$

Damit hat man schliesslich

$$\|t \cdot f(t)\|_2 \cdot \|s \cdot \widehat{f(s)}\|_2 = \frac{1}{2\pi} \|t \cdot f(t)\|_2 \cdot \|f'(t)\|_2 \geq \frac{1}{4\pi}. \quad \square$$

Damit ist die Behauptung gezeigt, falls $\mu(f) = 0 = \mu(\widehat{f})$ gilt.

Die Aussage für allgemeines f erhält man, indem man mittels Translation und Modulation f so “zentriert”, dass die Mittelwerte der Funktion und ihrer Fouriertransformierten = 0 sind. Nun hat man nur noch zu beachten (s.o.), dass sich bei Translation und Modulation die Varianzen nicht ändern.

4.8 Diracs Funktionale $\delta_a(t)$

Eine spekulative Überlegung zu der mehrfach erwähnten Tatsache, dass die komplexen Exponentialfunktionen $\varepsilon_a(t) = e^{2\pi i a t}$ weder zu $\mathcal{L}^1(\mathbb{R})$ noch zu $\mathcal{L}^2(\mathbb{R})$ gehören, also auch eine Fouriertransformierte von $\varepsilon_a(t)$ (erst einmal) nicht definiert ist:

Welche Eigenschaften müsste eine Fouriertransformierte von $\varepsilon_a(t)$ denn besitzen?

Bezeichnen wir einmal provisorisch die Fouriertransformierte von $\varepsilon_a(t)$ mit $\delta_a(s) = \widehat{\varepsilon}_a(s)$. Die “funktionale” Schreibweise δ als Funktion der Variablen s sollte man nicht so verstehen, dass es sich bei $\delta_a(s)$ um eine *Funktion* im traditionellen Sinn handelt. Tatsächlich kann das nicht der Fall sein.

Ist nun $f(t)$ eine “richtige” Funktion, die ein Fouriertransformierte $\widehat{f}(s)$ besitzt und für die die Fourier-Umkehrformel zutrifft, also

$$\widehat{f}(a) = \int f(t) e^{-2\pi i a t} dt = \langle f | \varepsilon_a \rangle.$$

Was geschieht nun, wenn man die Gültigkeit der Parseval-Plancherel Identität für $f(t)$ und $\varepsilon_a(t)$ fordert? Also

$$\widehat{f}(a) = \langle f | \varepsilon_a \rangle \stackrel{!}{=} \langle \widehat{f} | \widehat{\varepsilon}_a \rangle = \langle \widehat{f} | \delta_a \rangle = \int \widehat{f}(s) \delta_a(s) ds.$$

Dabei ist das Integral auf der rechten Seite suggestiv gemeint, nicht aber als ein “echtes” Integral.

Definition 4.4. Für $a \in \mathbb{R}$ ist $\delta_a = \delta_a(t)$ die Abbildung, die für "hinreichend gutartige" Funktionen $f : \mathbb{R} \rightarrow \mathbb{C}$ durch

$$\delta_a(t) : f(t) \mapsto f(a)$$

gegeben ist.

Man schreibt dafür häufig:

$$\delta_a : f(t) \mapsto \langle f | \delta_a \rangle = \int f(t) \delta_a(t) dt = f(a).$$

Dabei wird aber nicht integriert, die Anwendung von $\delta_a(t)$ auf eine Funktion $f(t)$ ist schlicht die Auswertung dieser Funktion an der Stelle $t = a$, und dies ist also keine *Funktion*, sondern (wie das Integral auch) ein *Funktional*, benannt als δ -Funktional von DIRAC. Für den speziellen Fall $a = 0$ schreibt man meist einfach δ statt δ_0 .

Der Frage, was mit "hinreichend gutartige Funktionen" gemeint ist, werden wir hier allerdings nicht nachgehen. Als Abkürzung wird hier \mathcal{T} für den anonymen Raum dieser sog. *Testfunktionen* geschrieben.

Satz 4.7 — Eigenschaften von δ_a .

1. $\delta_a : \mathcal{T} \rightarrow \mathbb{C}$ ist linear.
2. *Translation:* $\delta_a(t) = \delta_0(t - a) = (T_a \circ \delta_0)(t)$.
3. *Skalierung:* Für $p > 0$ und $a \in \mathbb{R}$ gilt $\delta_a(pt) = \frac{1}{p} \delta_{a/p}(t)$.
4. *Inneres Produkt:* $\langle f | \delta_a \rangle = \langle f(t) | \delta_a(t) \rangle = f(a)$
5. *Faltung:* Für $f \in \mathcal{T}$ und $a \in \mathbb{R}$ gilt

$$(f \star \delta_a)(t) = \int f(x) \delta_a(t - x) dx = f(t - a),$$

d.h., Faltung mit δ_a ist nichts anderes als die Translation mit T_a .

Speziell gilt also für jedes $f \in \mathcal{T}$ die Beziehung $f \star \delta = f$, d.h., δ verhält sich wie ein *neutrales Element* bezüglich der Faltung.

6. *Multiplikation mit Funktionen:* Für "geeignete" Funktionen $g(t)$ (z.B. falls $g(t) \cdot f(t) \in \mathcal{T}$ für alle $t \in \mathcal{T}$) ist $g(t) \cdot \delta_a(t)$ das Funktional

$$g(t) \cdot \delta_a(t) : f(t) \mapsto g(a) \cdot \delta_a(t).$$

7. Die *Fouriertransformierte* von δ_a ist $\widehat{\delta}_a = \mathcal{F}\delta_a = \varepsilon_a^- = \varepsilon_{-a} = e^{-2\pi ia}$.

Insbesondere ist die Fouriertransformierte von $\delta(t) = \delta_0(t)$ die konstante Funktion $\mathbf{1}_{\mathbb{R}}(t) \equiv 1$.

Beweis:

1. Für $f, g \in \mathbb{T}$ und $\alpha, \beta \in \mathbb{C}$ ist

$$\delta_a : \alpha f(t) + \beta g(t) \mapsto (\alpha f + \beta g)(a) = \alpha f(a) + \beta g(a) = \alpha \delta_a(f) + \beta \delta_a(g).$$

2. Offensichtlich.
3. Mittels Variablentransformation erhält man

$$\int f(t) \delta_a(pt) dt = \int f(t) \delta(pt - a) dt = \frac{1}{p} \int f\left(\frac{\tau + a}{p}\right) \delta(\tau) d\tau = \frac{1}{p} f\left(\frac{a}{p}\right),$$

also

$$\delta_a(pt) : f(t) \mapsto \frac{1}{p} f\left(\frac{a}{p}\right) = \frac{1}{p} \delta_{a/p} f(t).$$

4. Das ist nur eine Umschreibung von

$$g(t) \cdot \delta_a(t) : f(t) \mapsto \int f(t) g(t) \delta_a(t) dt = f(a) g(a).$$

5. Die Eigenschaften der Fouriertransformation \mathcal{F} sollten sich übertragen, also $\mathcal{F}^{-1}\delta_a = \varepsilon_a = (\mathcal{F}\delta_a)$. \square

4.9 Eine kurze Exkursion in eine neue Welt

Man kann die Definition von $\delta(t)$ auch so sehen:

$\delta(t)$ ist eine Wahrscheinlichkeitsverteilung auf \mathbb{R} , bei der die Punktmasse 1 im Nullpunkt konzentriert ist. Diese Verteilung hat keine Dichtefunktion, kann aber als Limes von Verteilungen mit Dichtefunktion angesehen werden. Konkreter: $\delta(t)$ ist als Limes $\lim_{\tau \rightarrow 0^+} K_\tau(t)$ eines Summationskerns zu sehen – siehe Abschnitt 3.3.2.

Eine andere Möglichkeit besteht darin, $\delta(t)$ als “Ableitung” der Sprungfunktion $\mathbf{1}_{[0, \infty)}(t)$ von Heaviside zu sehen, d.h. $\delta(t) = \mathbf{1}'_{[0, \infty)}(t)$ anzunehmen und partielle Integration auf hinreichend gutartigen Funktionen $f(t)$ mit $\lim_{t \rightarrow \pm\infty} f(t) = 0$ zu “simulieren”:

$$\begin{aligned} \int_{\mathbb{R}} f(t) \delta(t) dt &= \int_{\mathbb{R}} f(t) \mathbf{1}'_{[0, \infty)}(t) dt = [f(t) \mathbf{1}_{[0, \infty)}(t)]_{-\infty}^{\infty} - \int_{\mathbb{R}} f'(t) \mathbf{1}_{[0, \infty)}(t) dt \\ &= - \int_0^{\infty} f'(t) dt = \lim_{t \rightarrow 0} f(t) - \lim_{t \rightarrow \infty} f(t) = f(0). \end{aligned}$$

Es gilt also, symbolisch geschrieben,

$$\delta_b : f \mapsto f(b) = \int_{\mathbb{R}} \delta_b(t) f(t) dt$$

Wenn das sinnvoll sein soll, wird man erwarten, dass

$$e^{2\pi ibt} = \int_{\mathbb{R}} \delta_b(s) e^{2\pi ist} ds$$

gilt, d.h. man kann dies als eine Fourier-Rücktransformation lesen und $s \mapsto \delta_b(s)$ als die Fouriertransformierte von $t \mapsto e^{2\pi ibt}$ ansehen, die im üblichen Sinne ja garnicht existiert, da $t \mapsto e^{2\pi ibt}$ weder zu $\mathcal{L}^1(\mathbb{R})$ noch zu $\mathcal{L}^2(\mathbb{R})$ gehört!

Ein Ausblick: wenn man sich auf das Spiel mit den Funktionalen (oder verallgemeinerten Funktionen) $\delta_a(t)$ einlässt, öffnet man die Tür zu einer ganz neuen Welt,

den *Distributionen*. Dann kann man auf sinnvolle (und nützliche) Weise die Fouriertransformation auch für Funktionen definieren, die im traditionellen Verständnis weit davon entfernt sind, Fourier-transformierbar zu sein, beispielsweise die Polynome.

Das konstante Polynom $p_0(t) = \mathbf{1}_{\mathbb{R}}(t)$ hat bereits eine Fouriertransformierte erhalten:

$$\widehat{p}_0(s) = \widehat{\mathbf{1}_{\mathbb{R}}}(s) = \delta(s)$$

Betrachten wir nun das Polynom ersten Grades $p_1(t) = t$. Wegen des Zusammenhangs zwischen Ableitungen und Fouriertransformation (Theorem 4.6 mit $f(t) = p_0(t) \equiv 1$) sollte gelten

$$\mathcal{F}(p_1)(s) = \mathcal{F}(p_1 \cdot p_0)(s) = \frac{-1}{2\pi} \frac{d}{ds} \widehat{p}_0(s) = \frac{-1}{2\pi} \delta'(s).$$

Aber was ist die Ableitung von $\delta(s)$, wo dies doch keine Funktion ist? Eine Antwort liefert die Technik der partiellen Integration. Man schreibt einfach, so wie man es gewohnt ist, als hätte man es mit traditionellen Funktionen zu tun:

$$\int_{\mathbb{R}} f(s) \cdot \delta'(s) ds = f(s) \cdot \delta(s) \Big|_{-\infty}^{\infty} - \int_{\mathbb{R}} f'(s) \cdot \delta(s) ds,$$

wobei dies für “hinreichend gutartige” Funktionen $f(s)$ gelten soll – zum Beispiel für unendlich-oft differenzierbare Funktionen, die für $t \rightarrow \pm\infty$ sehr schnell gegen 0 gehen (genau: schneller als jedes reziproke Polynom). In diesem Fall verschwindet der erste Term auf der rechten Seite und man erhält

$$\int_{\mathbb{R}} f(s) \cdot \delta'(s) ds = - \int_{\mathbb{R}} f'(s) \cdot \delta(s) ds = -f'(0).$$

Das kann man so lesen: die Ableitung $\delta'(s)$ von $\delta(s)$ ist das durch

$$\delta'(s) : f(s) \mapsto -f'(0)$$

definierte Funktional. Und damit ist die Fouriertransformierte der Funktion $p_1(t) = t$ gegeben durch

$$\widehat{p}_1 : f(s) \mapsto \frac{1}{2\pi} f'(0).$$

Jetzt kann man genauso weitermachen und die Fouriertransformierte von allen Potenzen $p_n(t) = t^n$

$$\widehat{p}_n : f(s) \mapsto \frac{1}{(2\pi)^n} f^{(n)}(0)$$

und damit von allen Polynomen $p(t)$ durch lineare Fortsetzung definieren.

Dass man auf diese Weise den Bereich von “Funktionen”, für die man die Fouriertransformation mit all ihren Eigenschaften erheblich erweitern kann – und das diese Bereichserweiterung auch für Anwendungen äusserst nützlich ist, verdankt man in erste Linie dem französischen Mathematiker Laurent SCHWARTZ [Sch65].

4.10 Diracs Kamm und die Formel von Poisson

Das *Abtasttheorem* in seiner Fassung von Claude SHANNON ist fraglos einer der Eckpfeiler jeglicher Signalverarbeitung. Es beruht auf einem ganz klassischen Resultat der Fourier-Theorie, der *Formel von Poisson* und kann auch mit klassischen Methoden bewiesen werden. Es ist aber instruktiv zu sehen, die der δ -Formalismus von Dirac zu einer sehr eleganten Beschreibung des Abtastens und seiner Eigenschaften verwendet werden kann. Das wesentliche Hilfsmittel ist ein weiteres Funktional, *Diracs Kamm* genannt, dessen erstaunliche eigenschaften sich unmittelbar aus Eigenschaften von $\delta_a(t)$ ergeben.

Definition 4.5 (DIRACS Kamm). *Der Kamm von DIRAC $\mathbb{III}(t)$ ist ein Funktional, das für $f \in \mathcal{T}$ durch*

$$\mathbb{III}(t) : f(t) \mapsto \sum_{k \in \mathbb{Z}} f(k) = \int f(t) \mathbb{III}(t) dt = \langle f | \mathbb{III} \rangle$$

definiert ist, also in δ -Schreibweise

$$\mathbb{III}(t) = \sum_{k \in \mathbb{Z}} \delta_k(t) = \sum_{k \in \mathbb{Z}} \delta(t - k).$$

Allgemeiner: für $p > 0$ ist $\mathbb{III}_p(t)$ definiert durch

$$\mathbb{III}_p(t) = \sum_{k \in \mathbb{Z}} \delta_{kp}(t) = \sum_{k \in \mathbb{Z}} \delta(t - kp),$$

d.h. es gilt für $f \in \mathcal{T}$

$$\mathbb{III}_p(t) : f(t) \mapsto \sum_{k \in \mathbb{Z}} f(kp) = \int f(t) \mathbb{III}_p(t) dt = \langle f | \mathbb{III}_p \rangle.$$

Die wesentlichen Eigenschaften von $\mathbb{III}_p(t)$ ergeben sich aus den entsprechenden Eigenschaften von $\delta_a(t)$ durch lineare Erweiterung.

Satz 4.8 — Eigenschaften von \mathbb{III}_p .

1. \mathbb{III}_p ist ein lineares Funktional.
2. *Skalierung:* Für $p > 0$ ist

$$\mathbb{III}_p(t) = \frac{1}{p} \mathbb{III}\left(\frac{1}{p}t\right).$$

3. *Inneres Produkt:* Für $p > 0$ ist

$$\langle f(t) | \mathbb{III}_p(t) \rangle = \sum_{k \in \mathbb{Z}} f(kp).$$

4. *Faltung* (alias Periodisierung)

$$(f \star \mathbb{III}_p)(t) = \sum_{k \in \mathbb{Z}} (f \star \delta_{kp})(t) = \sum_{k \in \mathbb{Z}} f(t - kp)$$

5. *Multiplikation mit einer Funktion*: (alias Sampling)

$$f(t) \cdot \mathbb{III}_p(t) = \sum_{k \in \mathbb{Z}} f(t) \cdot \delta_{kp}(t) = \sum_{k \in \mathbb{Z}} f(kp) \cdot \delta_{kp}(t) = \sum_{k \in \mathbb{Z}} f(kp) \cdot \delta(t - kp)$$

6. Die *Fouriertransformierte* von \mathbb{III} ist \mathbb{III} selbst,

$$\widehat{\mathbb{III}} = \mathbb{III}, \quad \text{und allgemein gilt} \quad \widehat{\mathbb{III}_p} = \frac{1}{p} \mathbb{III}_{1/p}.$$

Beweis:

1. Ist klar wegen der Definition.
2. Folgt aus

$$\mathbb{III}(pt) = \sum_{k \in \mathbb{Z}} \delta(pt - k) = \sum_{k \in \mathbb{Z}} \delta_k(pt) = \frac{1}{p} \sum_{k \in \mathbb{Z}} \delta_{k/p}(t) = \frac{1}{p} \mathbb{III}_{1/p}(t)$$

3. Aussagen 3., 4. und 5. ergeben sich mittels Linearität aus den entsprechenden Aussagen für das δ -Funktional.
6. Ein plausibles Argument, das rigoros gemacht werden kann, geht so:
 \mathbb{III} ist 1-periodisch (zwar nicht als Funktion, aber immerhin...), so dass man auf eine Darstellung als *Fourierreihe* hoffen kann:

$$\mathbb{III}(t) = \sum_{k \in \mathbb{Z}} s[k] e^{2\pi i k t}, \quad \text{mit} \quad s[k] = \int_{-1/2}^{1/2} \mathbb{III}(t) e^{-2\pi i k t} dt = 1,$$

da von der Summe $\mathbb{III}(t) = \sum_{\ell \in \mathbb{Z}} \delta(t - \ell)$ nur der Term für $\ell = 0$ einen Beitrag liefert. Dieser evaluiert $e^{-2\pi i k t}$ an der Stelle $t = 0$, ergibt also 1. Damit hat man

$$\mathbb{III}(t) = \sum_{k \in \mathbb{Z}} e^{2\pi i k t} = \sum_{k \in \mathbb{Z}} \varepsilon_k(t),$$

und somit

$$\mathbb{III}(t) = \sum_{k \in \mathbb{Z}} \varepsilon_k(t) = \sum_{k \in \mathbb{Z}} \widehat{\delta_{-k}}(t) = \sum_{k \in \mathbb{Z}} \widehat{\delta_k}(t) = \widehat{\mathbb{III}}(t).$$

□

Das folgende berühmte Resultat, die Formel von POISSON, ist eines der zentralen Ergebnisse und Werkzeuge der Fourier-Theorie. Sie verknüpft die an sich getrennten Formalismen der *Fourierreihen* und der *Fouriertransformation* miteinander. Man kann diese Formel, so wie es POISSON (um 1830 herum) tat, auch ohne Verwendung von Diracs δ und \mathbb{III} beweisen, aber es ist interessant, die Eleganz dieser Konzepte in Aktion zu sehen.

Theorem 4.6 — Poissons Formel (1).

Für “geeignete” Funktionen $f(t)$ gilt

$$(P) \quad \sum_{k \in \mathbb{Z}} \widehat{f}(t - k) = \sum_{k \in \mathbb{Z}} f(k) e^{-2\pi i k t},$$

was speziell für $t = 0$ eine verblüffende Aussage beinhaltet:

$$\sum_{k \in \mathbb{Z}} \widehat{f}(k) = \sum_{k \in \mathbb{Z}} f(k)$$

Ein informeller Beweis, der aber rigoros gemacht werden kann:

$$\begin{aligned} \sum_{k \in \mathbb{Z}} \widehat{f}(t - k) &= (\widehat{f} \star \text{III})(t) && \text{Periodisierung} \\ &= (\widehat{f} \star \widehat{\text{III}})(t) && \text{III ist seine eigene FT} \\ &= \widehat{(f \cdot \text{III})}(t) && \text{Faltungsformel} \\ &= \widehat{(\sum_{k \in \mathbb{Z}} f(k) \delta_k)}(t) && \text{Sampling} \\ &= \sum_{k \in \mathbb{Z}} f(k) \widehat{\delta}_k(t) && \text{Linearität der FT} \\ &= \sum_{k \in \mathbb{Z}} f(k) e^{-2\pi i k t} && \text{FT von } \delta \quad \square \end{aligned}$$

Der Kern von Poissons Formel ist der fundamentale Zusammenhang zwischen Sampling (einer Funktion) und Periodisierung (ihres Spektrums), der ausgedrückt wird durch

$$\widehat{f \cdot \text{III}_a}(s) = \widehat{f}(s) \star \widehat{\text{III}_a}(s) = \frac{1}{a} \left(\widehat{f}(s) \star \text{III}_{1/a}(s) \right)$$

Man begegnet Poissons Formel meist in einer der beiden folgenden Darstellungen, die natürlich gleichwertig und auch zum vorigen Theorem gleichwertig sind.

Theorem 4.7 — Poissons Formel (2).

Für Fourier-transformierbare Funktionen f, g und $a > 0$ gilt (falls die Reihen überhaupt konvergieren)

$$\begin{aligned} \sum_{n \in \mathbb{Z}} f(t - na) &= \frac{1}{a} \sum_{n \in \mathbb{Z}} \widehat{f}\left(\frac{n}{a}\right) e^{2\pi i (n/a)t} \\ \sum_{n \in \mathbb{Z}} \widehat{g}\left(s - \frac{n}{a}\right) &= a \sum_{n \in \mathbb{Z}} g(na) e^{-2\pi i n a s} \end{aligned}$$

Hier noch einmal (jetzt für beliebiges $a > 0$) der Beweis (für die zweite Identität)

$$\begin{aligned} \sum_n \widehat{g}\left(s - \frac{n}{a}\right) &= \widehat{g}(s) \star \text{III}_{1/a}(s) \\ &= a \widehat{g}(s) \star \widehat{\text{III}_a}(s) \\ &= a \widehat{(g \cdot \text{III}_a)}(s) = a \sum_n g(na) \cdot e^{-2\pi i n a s} \end{aligned}$$



4.11 Sampling und das Abtasttheorem

Das Abtasttheorem der Signalverarbeitung hat eine glorreiche Geschichte, die in erster Linie mit den Namen von Harry NYQUIST (1889–1976) und Claude SHANNON (1916–2001) verbunden wird – was angesichts von deren grundlegenden Beiträgen [Nyq28] und [Sha48] zwar berechtigt ist, aber doch der Sache in ihrer ganzen Breite und Tiefe und Verzweigungen nicht ganz gerecht wird. In der aktuellen Forschung ist insbesondere das Gebiet des *Compressed Sensing* intensiv bearbeitet. Über Geschichte und Stand der Wissenschaft kann man sich kompetent beispielsweise in dem Buch von Yonina ELДАР [Eld15] informieren.

- Was bedeutet es mathematisch, wenn man ein Signal $f : \mathbb{R} \rightarrow \mathbb{C}$ im Abstand $a > 0$, also mit der Frequenz $1/a$, abtastet?

Man ersetzt das kontinuierliche Signal $f(t)$ durch das diskrete Signal der Folge der Abtastwerte

$$(f(k \cdot a))_{k \in \mathbb{Z}} = (\dots f(-2a), f(-a), f(0), f(a), f(2a), \dots),$$

und das schreibt sich im δ -Formalismus einfach als $f \cdot \text{III}_a$.

- Was passiert im Frequenzbereich beim Abtasten?

Das Faltungstheorem gibt die Antwort:

Bei Abtasten von $f(t)$ im Abstand a (Frequenz $1/a$) ist

$$\widehat{f \cdot \text{III}_a} = \widehat{f} \star \widehat{\text{III}_a} = \frac{1}{a} (\widehat{f} \star \text{III}_{1/a})$$

und das ist die *Periodisierung* von \widehat{f} mit Periode $1/a$!

Anders formuliert: Das Spektrum der diskretisierten, mit Frequenz $1/a$ abgetasteten Version von $f(t)$, ist eine periodische Überlagerung von Kopien von \widehat{f} mit Abstand $1/a$ zwischen den Kopien.

- Was geschieht beim Abtasten im Abstand $a > 0$ (also mit Abtastfrequenz $1/a$), wenn die abgetastete Funktion b -bandbeschränkt (Grenzfrequenz $b/2$) ist, d.h., wenn $\widehat{f}(s) = 0$ für alle s mit $|s| > b/2$ gilt?

- Wenn $1/a \geq b/2$ ist, sind die Bereiche in denen die verschiedenen Kopien von $\widehat{f}(s)$ nicht verschwinden, disjunkt, d.h., die (unendlich vielen) Kopien von $\widehat{f}(s)$ überlagern sich effektiv nicht.
- Wenn $1/a < b/2$ ist, treten Überlagerungen der Kopien von $\widehat{f}(s)$ auf, $\widehat{f}(s)$ ist nicht mehr korrekt rekonstruierbar.

- Was ist die Konsequenz für die Rekonstruktion von $f(t)$ aus den Abtastwerten?

Im Fall $1/a \geq b/2$ liegt im Frequenzbereich $-b/2 \leq s \leq b/2$ genau eine Kopie von \widehat{f} (genauer: des Bereiches, wo \widehat{f} nicht verschwindet). Aus dieser kann man durch Fourier-Rücktransformation $f(t)$ komplett rekonstruieren.

Nun etwas formaler: für $b > 0$ sei $\Pi_b(s)$ die Box-Funktion

$$\Pi_b(s) = \mathbf{1}_{-b/2, b/2}(t) = \begin{cases} 1 & \text{if } |s| \leq b/2 \\ 0 & \text{if } |s| > b/2. \end{cases}$$

Die b -Bandbeschränktheit einer Funktion $f(t)$ lässt sich durch

$$\widehat{f}(s) = \widehat{f}(s) \cdot \Pi_b(s)$$

beschreiben. Ist $f(t)$ b -bandbeschränkt, so liegen die Träger der in der Periodisierung $f \cdot \widehat{\mathbb{III}}_{1/b}$ enthaltenen Kopien von $\widehat{f}(s)$ disjunkt. Insbesondere kann man \widehat{f} selbst durch Multiplikation mit $\pi_b(s)$ erhalten:

$$\widehat{f} = \Pi_b \cdot f \cdot \widehat{\mathbb{III}}_{1/b} = b \cdot \Pi_b \cdot (\widehat{f} \star \mathbb{III}_b)$$

Nun kann man $f(t)$ mittels Fourier-Rücktransformation und unter Verwendung der Eigenschaften des \mathbb{III} -Funktional berechnen:

$$\begin{aligned} f &= \mathcal{F}^{-1}(b \cdot \Pi_b \cdot (\mathcal{F}(f) \star \mathbb{III}_b)) \\ &= b \cdot \mathcal{F}^{-1}(\Pi_b) \star \mathcal{F}^{-1}(\mathcal{F}(f) \star \mathbb{III}_b) \\ &= b \cdot \mathcal{F}^{-1}(\Pi_b) \star (f \cdot \mathcal{F}^{-1}(\mathbb{III}_b)) \\ &= \mathcal{F}^{-1}(\Pi_b) \star (f \cdot \mathbb{III}_{1/b}), \end{aligned}$$

Die Fourier-Rücktransformierte der Box-Funktion ist eine sinc-Funktion, daher ergibt sich ausgeschrieben

$$\begin{aligned} f(t) &= \text{sinc}(bt) \star \sum_{k \in \mathbb{Z}} f(k/b) \delta(t - k/b) \\ &= \sum_{k \in \mathbb{Z}} f(k/b) \text{sinc}(b(t - k/b)), \end{aligned}$$

wobei $\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}$ ist und in der letzten Gleichheit nochmal eine Eigenschaft des δ -Funktional verwendet wurde.

Theorem 4.8 — Abtasttheorem von Nyquist-Shannon.

1. (Nyquist, 1928)

Eine *b-bandbegrenzte* Funktion $f(t) \in \mathcal{L}^2(\mathbb{R})$, d.h. $\widehat{f}(s) = 0$ für alle $|s| > b/2$, ist aus den Abtastwerten $(f(n/b))_{n \in \mathbb{Z}}$ eindeutig rekonstruierbar.

2. (Shannon, 1949)

Die Rekonstruktion von $f(t)$ aus den Abtastwerten geschieht durch

$$f(t) = \sum_{k \in \mathbb{Z}} f(k/b) \operatorname{sinc}(b(t - k/b)).$$

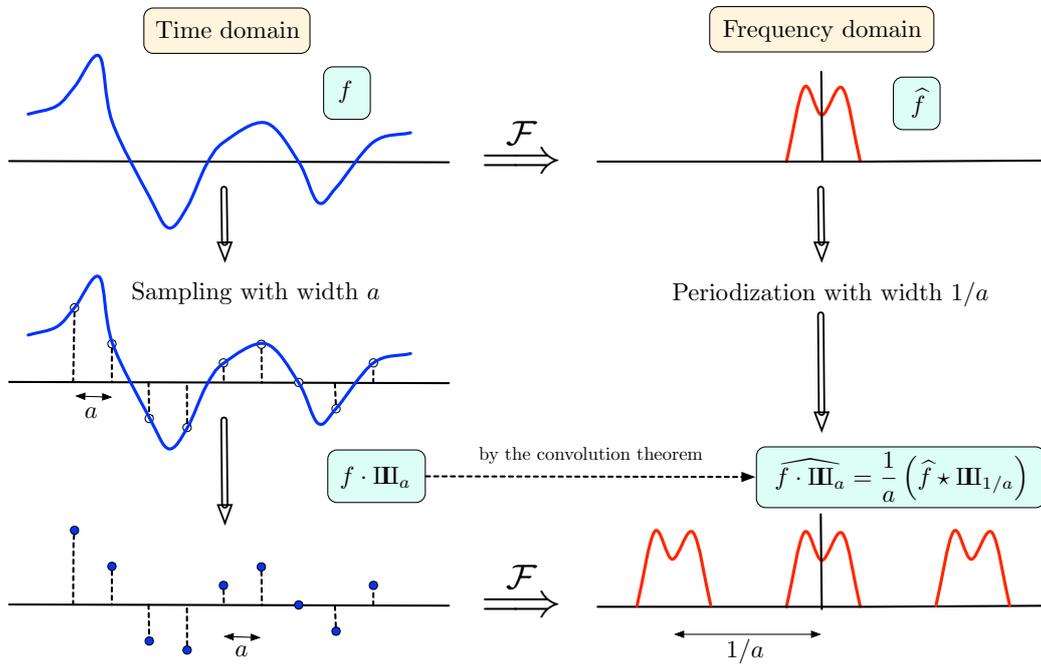


Abbildung 4.2: Sampling: Die Sichtweise der Poisson-Formel

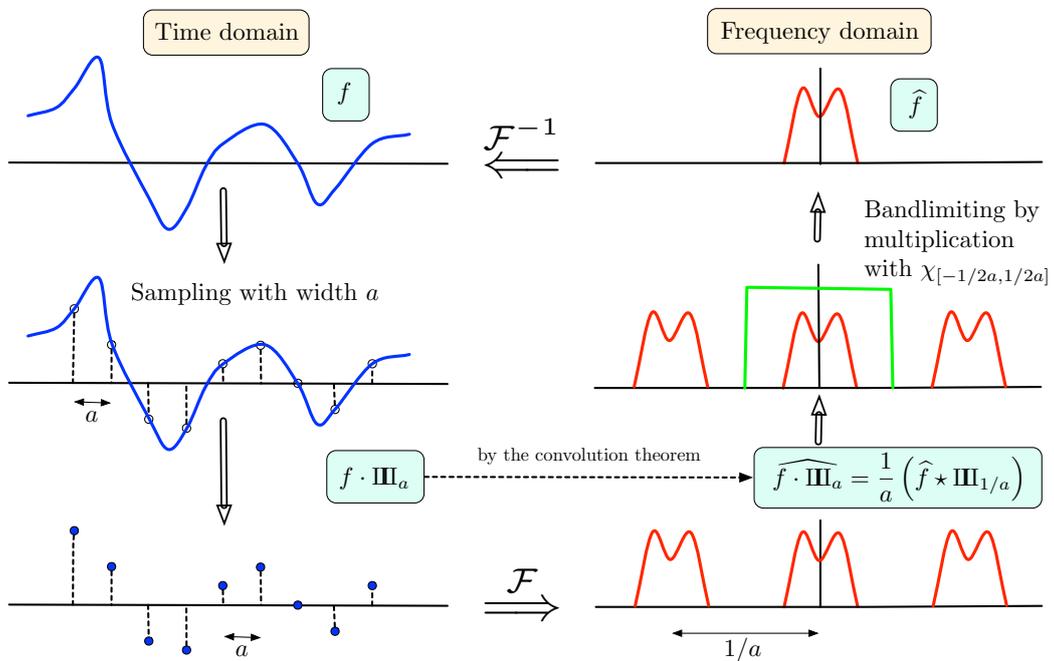


Abbildung 4.3: Sampling: Die Sichtweise der Shannon-Formel

4.12 Mathematica notebooks

Die folgenden Seiten enthalten vier Mathematica notebooks zur Illustration wichtiger Aspekte der Fouriertransformation.

1. Beispiele zur Fouriertransformation.
2. Illustration der Technik des “Filterns im Frequenzbereich” auf der Basis des Faltungstheorems.
3. Ein Demonstration des Abtasttheorems an einem einfachen Beispiel.
4. Ein Demonstration des Abtasttheorems an einem etwas komplexeren Beispiel.

Fourier transform examples

```
In[509]:= fp = FourierParameters → {0, -2 Pi}; $Assumptions = {a > 0};
```

```
In[510]:= show[f_, a_, b_, c_, d_] := Module[{t, flist, p},
  col = {Black, Blue, Red, Green};
  For[j = 1, j ≤ Length[f], j++,
    p[j] = Plot[
      f[[j]][t, a], {t, -b, b},
      Filling → Axis,
      PlotRange → {c, d},
      PlotStyle → col[[j]],
      Exclusions → None,
      ImageSize → {300, 150}];
  Show[Table[p[j], {j, 1, Length[f]}]]
]
```

The functions

1- The parametrized box function

```
In[511]:= f[t_, a_] = UnitStep[t + a] - UnitStep[t - a]
```

```
Out[511]= -UnitStep[-a + t] + UnitStep[a + t]
```

```
In[512]:= PiecewiseExpand[f[t, a]]
```

```
Out[512]= 
$$\begin{cases} -1 & a - t \leq 0 \ \&\& \ a + t < 0 \\ 1 & a - t > 0 \ \&\& \ a + t \geq 0 \\ 0 & \text{True} \end{cases}$$

```

2- The parametrized hat function

```
In[513]:= g[t_, a_] = 1 / (2 * a) * Integrate[f[x, a] * f[2 * t - x, a], {x, -Infinity, Infinity}]
```

```
Out[513]= 
$$\frac{1}{a} (-2 t \text{UnitStep}[t] + (a + t) \text{UnitStep}[a + t] + (-a + t) \text{UnitStep}[-2 a + 2 t])$$

```

```
In[514]:= PiecewiseExpand[%]
```

```
Out[514]= 
$$\begin{cases} \frac{-a-t}{a} & t \geq 0 \ \&\& \ a + t < 0 \ \&\& \ a - t \leq 0 \\ \frac{a-t}{a} & t \geq 0 \ \&\& \ a + t \geq 0 \ \&\& \ a - t > 0 \\ -\frac{2t}{a} & t \geq 0 \ \&\& \ a + t < 0 \ \&\& \ a - t > 0 \\ \frac{2t}{a} & t < 0 \ \&\& \ a + t \geq 0 \ \&\& \ a - t \leq 0 \\ \frac{-a+t}{a} & t < 0 \ \&\& \ a + t < 0 \ \&\& \ a - t \leq 0 \\ \frac{a+t}{a} & t < 0 \ \&\& \ a + t \geq 0 \ \&\& \ a - t > 0 \\ 0 & \text{True} \end{cases}$$

```

3- A parametrized function with exponential decay

```
In[515]:= h[t_, a_] = Exp[-a * Abs[t]]
```

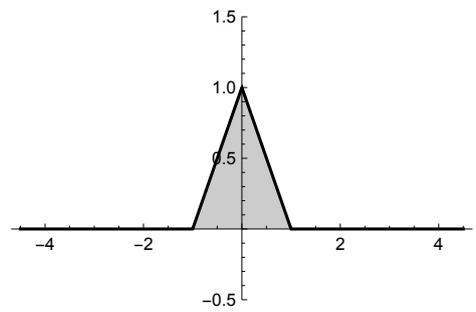
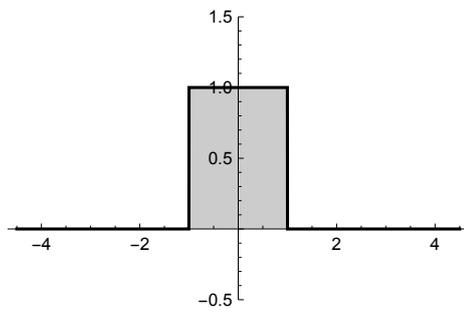
```
Out[515]= e-a Abs[t]
```

4- The parametrized gaussian function

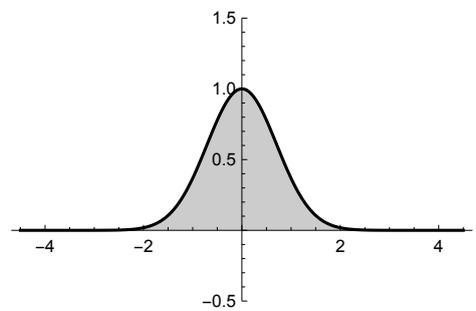
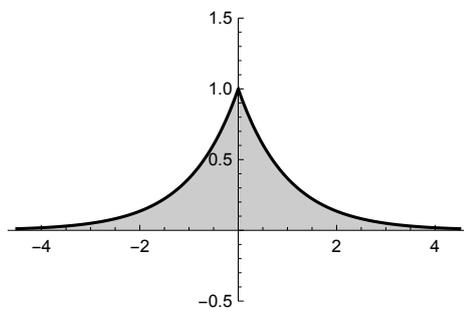
```
In[516]:= k[t_, a_] = Exp[-a * t^2]
```

```
Out[516]= e-a t2
```

```
In[517]:= GraphicsGrid[{{  
  show[{f}, 1, 4.5, -0.5, 1.5],  
  show[{g}, 1, 4.5, -0.5, 1.5]},  
{ show[{h}, 1, 4.5, -0.5, 1.5],  
  show[{k}, 1, 4.5, -0.5, 1.5]}}
```



```
Out[517]=
```



The box function and its Fourier transform

The norm of the box function

```
In[518]:= norm2f[a_] = Sqrt[Integrate[f[s, a]^2, {s, -Infinity, Infinity}]]
```

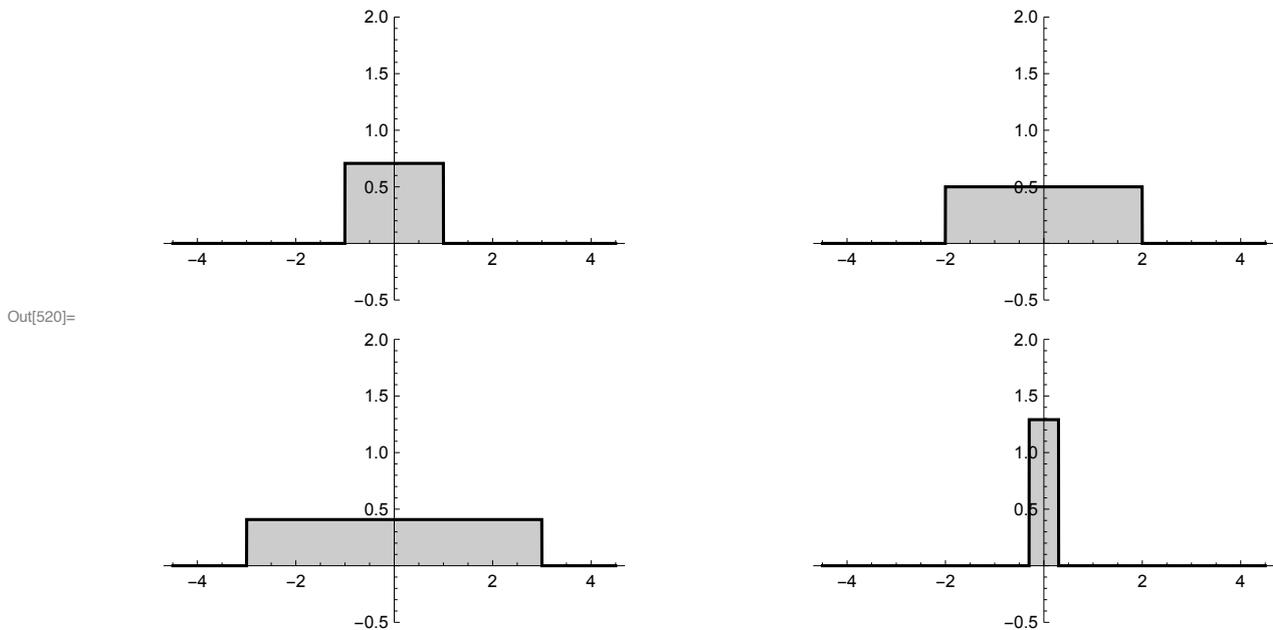
```
Out[518]:=  $\sqrt{2} \sqrt{a}$ 
```

The normalized box function

```
In[519]:= ff[t_, a_] := f[t, a] / norm2f[a]
```

Displaying the normalized box function

```
In[520]:= GraphicsGrid[{{
  show[{ff}, 1, 4.5, -0.5, 2.0],
  show[{ff}, 2, 4.5, -0.5, 2.0]},
{ show[{ff}, 3, 4.5, -0.5, 2.0],
  show[{ff}, 0.3, 4.5, -0.5, 2.0]}]}
```



The Fourier transform of the normalized box function

```
In[521]:= fhat[s_, a_] = FourierTransform[ff[t, a], t, s, fp]
```

```
Out[521]:=  $\frac{\text{Sin}[2 a \pi s]}{\sqrt{2} \sqrt{a} \pi s}$ 
```

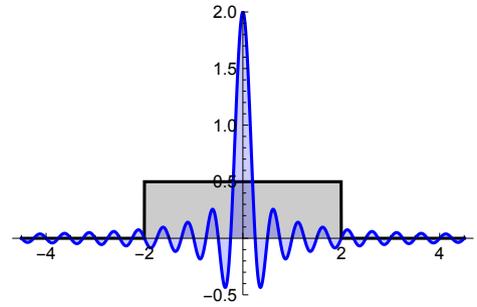
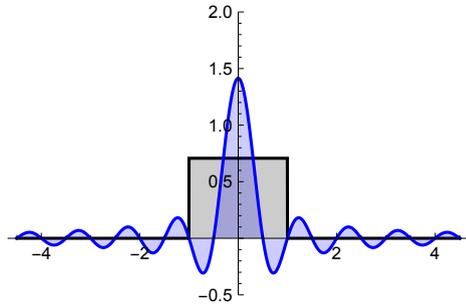
Checking that the Fourier transform is again normalized

```
In[522]:= Integrate[fhat[s, a]^2, {s, -Infinity, Infinity}, Assumptions -> a > 0]
```

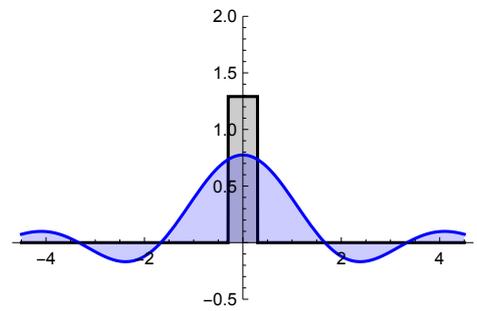
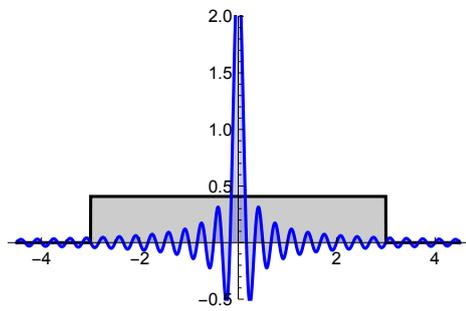
```
Out[522]:= 1
```

Comparing the normalized box function and its Fourier transform when changing the width parameter a

```
In[523]:= GraphicsGrid[{{
  show[{ff, fhat}, 1, 4.5, -0.5, 2.0],
  show[{ff, fhat}, 2, 4.5, -0.5, 2.0]},
{ show[{ff, fhat}, 3, 4.5, -0.5, 2.0],
  show[{ff, fhat}, 0.3, 4.5, -0.5, 2.0]}]}
```



Out[523]=



The hat function and its Fourier transform

The norm of the hat function

```
In[524]:= norm2g[a] = Sqrt[Integrate[PiecewiseExpand[g[t, a]]^2, {t, -Infinity, Infinity}]]
```

$$\text{Out[524]} = \sqrt{\frac{2}{3}} \sqrt{a}$$

The normalized hat function

```
In[525]:= gg[t_, a_] = g[t, a] / norm2g[a]
```

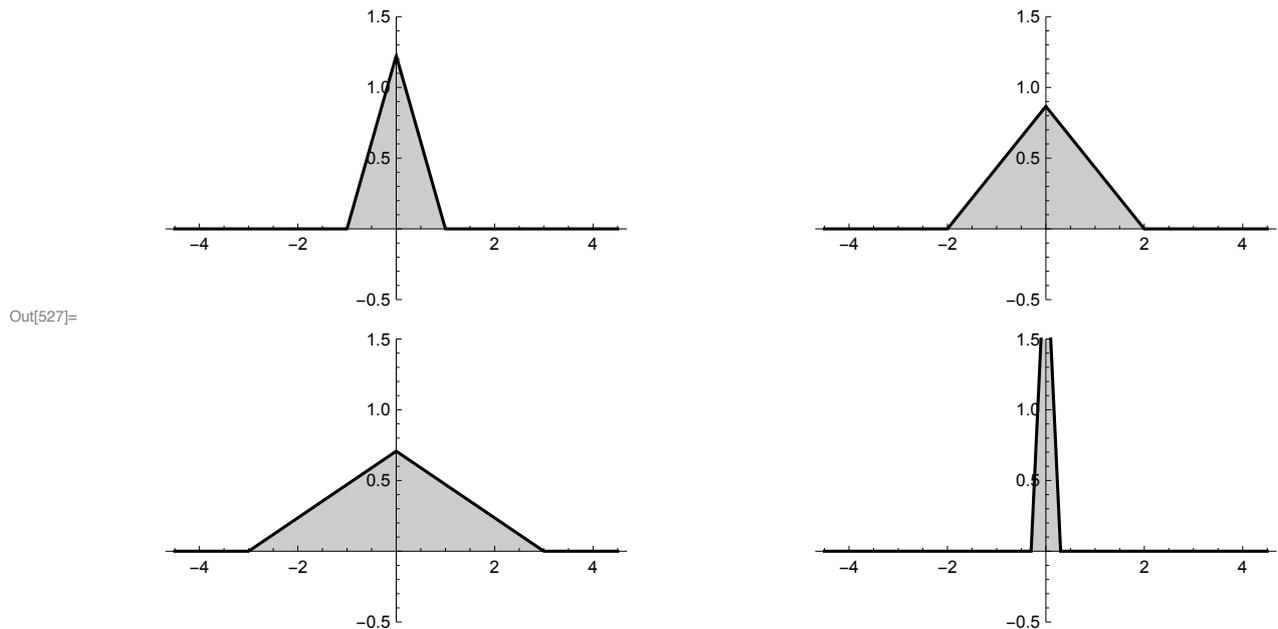
$$\text{Out[525]} = \frac{1}{a^{3/2}} \sqrt{\frac{3}{2}} \left(-2t \text{UnitStep}[t] + (a+t) \text{UnitStep}[a+t] + (-a+t) \text{UnitStep}[-2a+2t] \right)$$

... write as a piecewise defined function

```
In[526]:= PiecewiseExpand[g[t, a], a > 0]
```

$$\text{Out[526]} = \begin{cases} \frac{-a-t}{a} & t \geq 0 \ \&\& \ a+t < 0 \ \&\& \ a-t \leq 0 \\ \frac{a-t}{a} & t \geq 0 \ \&\& \ a+t \geq 0 \ \&\& \ a-t > 0 \\ -\frac{2t}{a} & t \geq 0 \ \&\& \ a+t < 0 \ \&\& \ a-t > 0 \\ \frac{2t}{a} & t < 0 \ \&\& \ a+t \geq 0 \ \&\& \ a-t \leq 0 \\ \frac{-a+t}{a} & t < 0 \ \&\& \ a+t < 0 \ \&\& \ a-t \leq 0 \\ \frac{a+t}{a} & t < 0 \ \&\& \ a+t \geq 0 \ \&\& \ a-t > 0 \\ 0 & \text{True} \end{cases}$$

```
In[527]:= GraphicsGrid[{{
  show[{gg}, 1, 4.5, -0.5, 1.5],
  show[{gg}, 2, 4.5, -0.5, 1.5]},
{show[{gg}, 3, 4.5, -0.5, 1.5],
 show[{gg}, 0.3, 4.5, -0.5, 1.5]}]}]
```



The Fourier transform of the normalized hat function

In[528]:= `FourierTransform[gg[t, a], t, s, fp]`

$$\text{Out[528]} = \frac{\sqrt{\frac{3}{2}}}{2 a^{3/2} \pi^2 s^2} - \frac{\sqrt{\frac{3}{2}} e^{-2 i a \pi s}}{4 a^{3/2} \pi^2 s^2} - \frac{\sqrt{\frac{3}{2}} e^{2 i a \pi s}}{4 a^{3/2} \pi^2 s^2}$$

... equivalently

In[529]:= `gHat[s_, a_] = ExpToTrig[%]`

$$\text{Out[529]} = \frac{\sqrt{\frac{3}{2}}}{2 a^{3/2} \pi^2 s^2} - \frac{\sqrt{\frac{3}{2}} \text{Cos}[2 a \pi s]}{2 a^{3/2} \pi^2 s^2}$$

The Fourier transform is again normalized

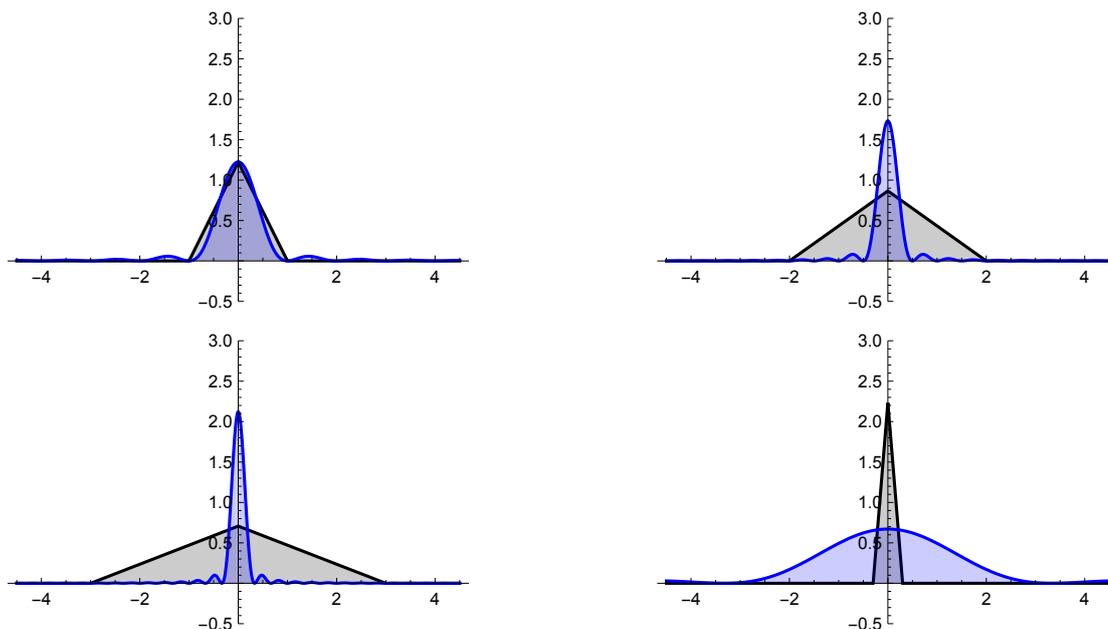
In[530]:= `Integrate[gHat[s, a]^2, {s, -Infinity, Infinity}]`

Out[530]= 1

Comparing the normalized hat function and its Fourier transform when changing the width parameter a

In[531]:= `GraphicsGrid[{{
 show[{{gg, gHat}, 1, 4.5, -0.5, 3],
 show[{{gg, gHat}, 2, 4.5, -0.5, 3]},
 { show[{{gg, gHat}, 3, 4.5, -0.5, 3], show[{{gg, gHat}, 0.3, 4.5, -0.5, 3]}}]`

Out[531]=



A parametrized function with exponential decay

The norm of this function

```
In[532]:= norm2h[a_] = Sqrt[Integrate[h[t, a]^2, {t, -Infinity, Infinity}]]
```

```
Out[532]=  $\sqrt{\frac{1}{a}}$ 
```

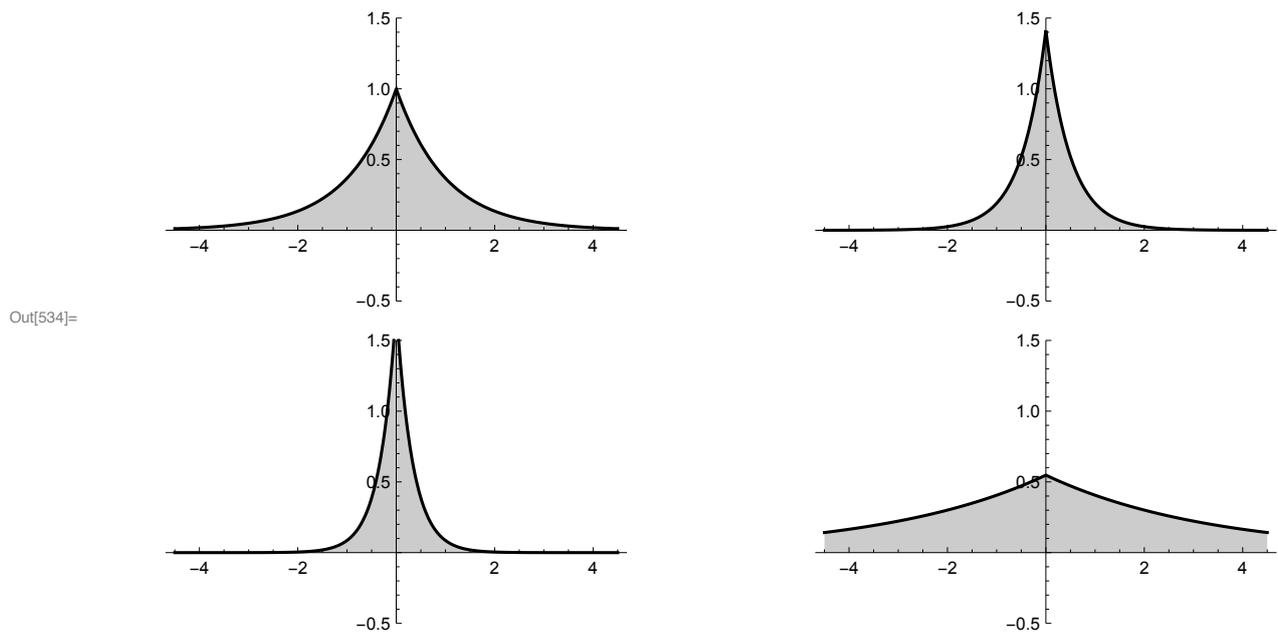
The normed version of this function

```
In[533]:= hh[t_, a_] = h[t, a] / norm2h[a]
```

```
Out[533]=  $\frac{e^{-a \text{Abs}[t]}}{\sqrt{\frac{1}{a}}}$ 
```

Displaying the normed function

```
In[534]:= GraphicsGrid[{{
  show[{hh}, 1, 4.5, -0.5, 1.5],
  show[{hh}, 2, 4.5, -0.5, 1.5]},
{show[{hh}, 3, 4.5, -0.5, 1.5],
 show[{hh}, 0.3, 4.5, -0.5, 1.5]}}}
```



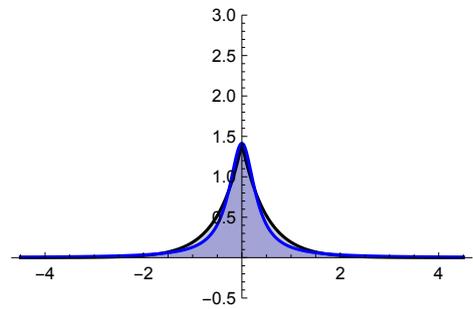
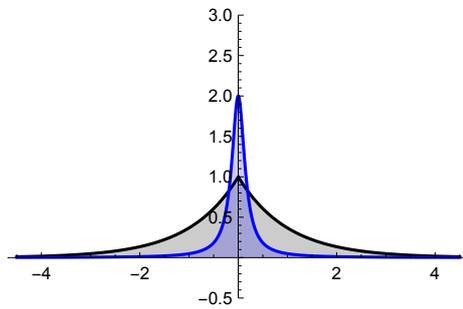
The Fourier transform of this function

```
In[535]:= hhat[s_, a_] = FourierTransform[hh[t, a], t, s, fp]
```

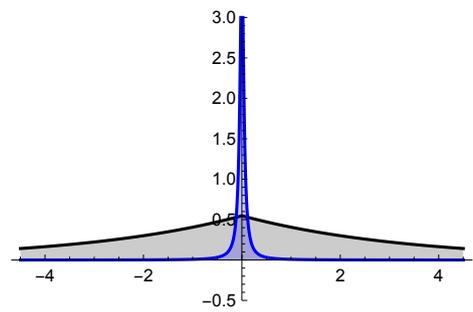
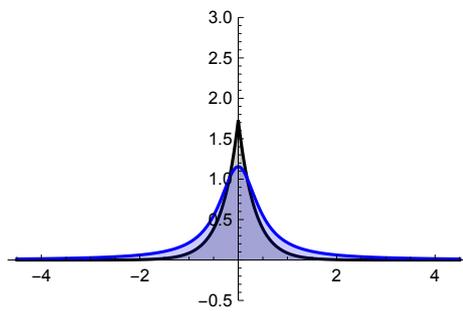
```
Out[535]=  $\frac{2}{\left(\frac{1}{a}\right)^{3/2} (a^2 + 4 \pi^2 s^2)}$ 
```

Comparing the normalized function and its Fourier transform when changing the decay parameter a

```
In[537]:= GraphicsGrid[{{
  show[{hh, hhat}, 1, 4.5, -0.5, 3],
  show[{hh, hhat}, 2, 4.5, -0.5, 3]},
{ show[{hh, hhat}, 3, 4.5, -0.5, 3], show[{hh, hhat}, 0.3, 4.5, -0.5, 3]}]}
```



Out[537]=



A parametrized gaussian function

The norm of this function

```
In[538]:= norm2k[a_] = Sqrt[Integrate[k[t, a]^2, {t, -Infinity, Infinity}]]
```

```
Out[538]=  $\frac{\left(\frac{\pi}{2}\right)^{1/4}}{a^{1/4}}$ 
```

The normalized version of this function

```
In[539]:= kk[t_, a_] = k[t, a] / norm2k[a]
```

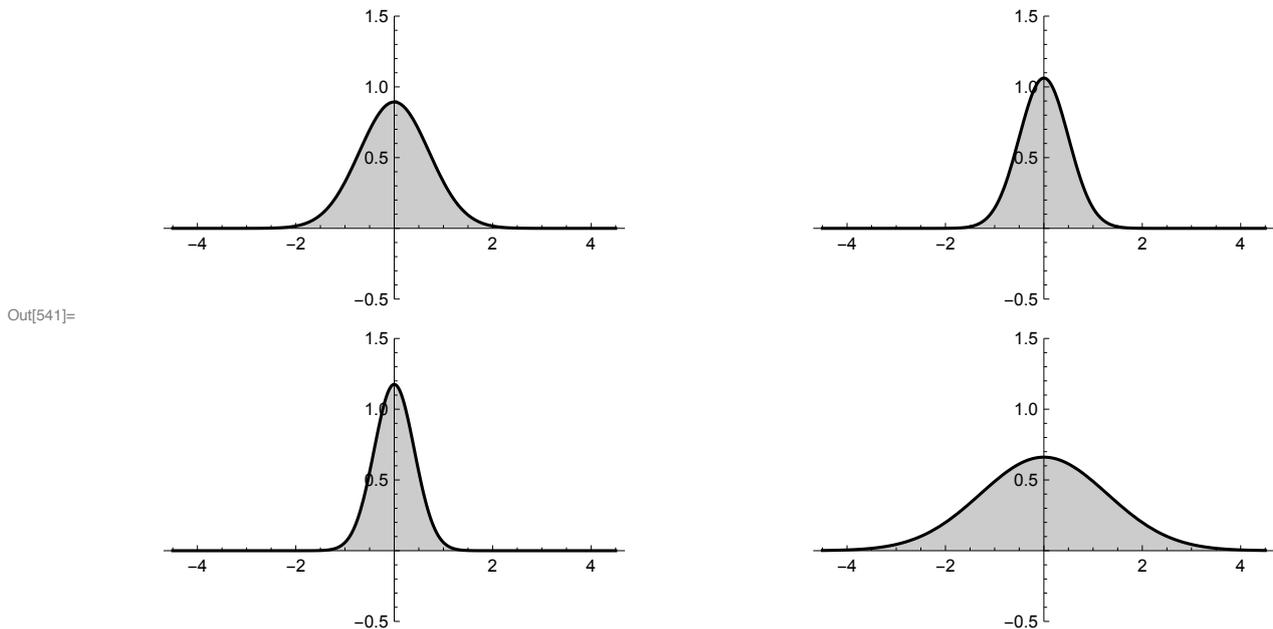
```
Out[539]=  $a^{1/4} e^{-a t^2} \left(\frac{2}{\pi}\right)^{1/4}$ 
```

```
In[540]:= Integrate[kk[t, a]^2, {t, -Infinity, Infinity}]
```

```
Out[540]= 1
```

Displaying the normalized function

```
In[541]:= GraphicsGrid[{{
  show[{kk}, 1, 4.5, -0.5, 1.5],
  show[{kk}, 2, 4.5, -0.5, 1.5]},
{show[{kk}, 3, 4.5, -0.5, 1.5],
 show[{kk}, 0.3, 4.5, -0.5, 1.5]}]}]
```



The Fourier transform of the normalized function

```
In[542]:= khat[s_, a_] = FourierTransform[kk[t, a], t, s, fp]
```

```
Out[542]=  $\frac{e^{-\frac{\pi^2 s^2}{a}} \left(2 \pi\right)^{1/4}}{a^{1/4}}$ 
```

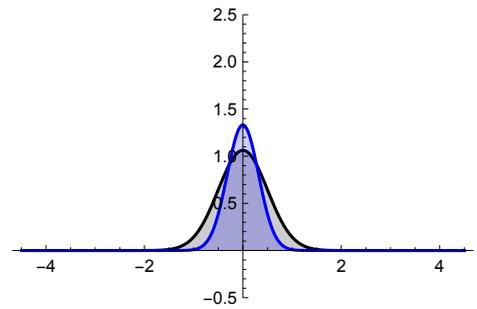
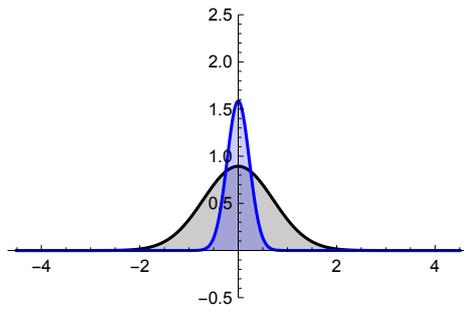
Checking the norm of the Fourier transform

```
In[543]:= Integrate[khat[t, a]^2, {t, -Infinity, Infinity}]
```

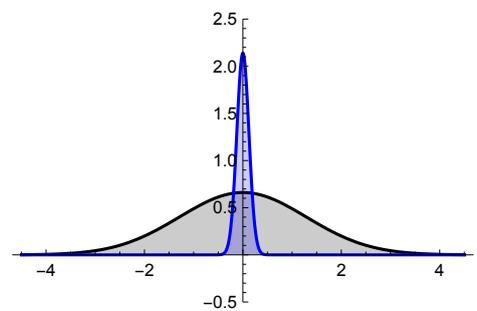
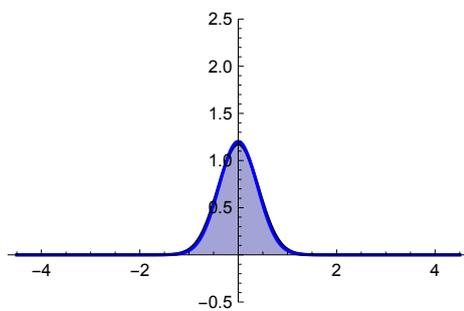
Out[543]= 1

Comparing the normalized function and its Fourier transform when changing the variance parameter a

```
In[544]:= GraphicsGrid[{{
  show[{kk, khat}, 1, 4.5, -0.5, 2.5],
  show[{kk, khat}, 2, 4.5, -0.5, 2.5]},
 {show[{kk, khat}, 3, 4.5, -0.5, 2.5], show[{kk, khat}, 0.3, 4.5, -0.5, 2.5]}]}
```



Out[544]=



Checking the uncertainty relation

for the box function

In[545]:= Integrate[ff[t, a]^2 * t^2, {t, -Infinity, Infinity}]

Out[545]= $\frac{a^2}{3}$

In[546]:= fhat[s, a]

Out[546]= $\frac{\text{Sin}[2 a \pi s]}{\sqrt{2} \sqrt{a} \pi s}$

In[547]:= Integrate[fhat[s, a]^2 * s^2, {s, -Infinity, Infinity}]

Out[547]= $\int_{-\infty}^{\infty} \frac{\text{Sin}[2 a \pi s]^2}{2 a \pi^2} ds$

for the hat function

In[548]:= Integrate[gg[t, a]^2 * t^2, {t, -Infinity, Infinity}]

Out[548]= $\frac{a^2}{10}$

In[549]:= Integrate[ghat[s, a]^2 * s^2, {s, -Infinity, Infinity}]

Out[549]= $\frac{3}{4 a^2 \pi^2}$

for the function with exponential decay

In[550]:= Integrate[hh[t, a]^2 * t^2, {t, -Infinity, Infinity}]

Out[550]= $\frac{1}{2 a^2}$

In[551]:= Integrate[hhat[s, a]^2 * s^2, {s, -Infinity, Infinity}]

Out[551]= $\frac{a^2}{4 \pi^2}$

for the gaussian function

In[552]:= Integrate[kk[t, a]^2 * t^2, {t, -Infinity, Infinity}]

Out[552]= $\frac{1}{4 a}$

In[553]:= Integrate[khat[s, a]^2 * s^2, {s, -Infinity, Infinity}]

Out[553]= $\frac{a}{4 \pi^2}$

Application of the convolution theorem: Filtering in the frequency domain

In[195]:= **fp = FourierParameters** → {0, -2 Pi}

Out[195]= FourierParameters → {0, -2 π}

In[196]:= **as = Assumptions** → {σ > 0}

Out[196]= Assumptions → {σ > 0}

In[197]:= **pra = PlotRange** → All

Out[197]= PlotRange → All

Smoothing with a Gauss filter (low-pass)

Defining the Gauss function (kernel) with variance σ^2

```
In[198]:= gauss[t_, σ_] = Module[{tmp, norm},  
    tmp = Exp[-(t)^2 / (2 σ^2)];  
    norm = Integrate[tmp, {t, -∞, ∞}, Assumptions → {σ > 0}];  
    Simplify[tmp / norm]  
    ]
```

Out[198]=
$$\frac{e^{-\frac{t^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma}$$

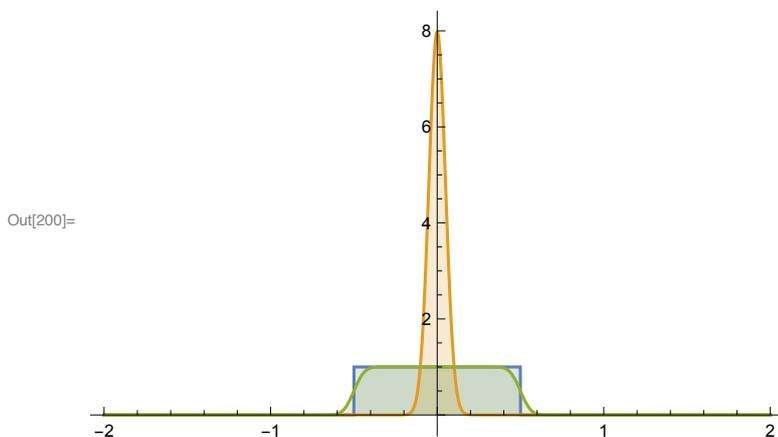
Convolve box function with gauss kernel

```
In[199]:= conv[σ_] = Convolve[UnitBox[x], gauss[x, σ], x, t]
```

Out[199]=
$$\frac{1}{2} \left(\text{Erf} \left[\frac{1-2t}{2\sqrt{2}\sigma} \right] + \text{Erf} \left[\frac{1+2t}{2\sqrt{2}\sigma} \right] \right)$$

Comparing box, gauss and convolution

```
In[200]:= Plot[{UnitBox[t], gauss[t, 0.05], conv[0.05]},  
    {t, -2, 2}, PlotRange → All, Filling → Axis]
```



Fourier transform of the Gauss kernel

In[201]:= **ftgauss**[s_, σ_] = **FourierTransform**[gauss[t, σ], t, s, fp, as]

Out[201]= $e^{-2 \pi^2 s^2 \sigma^2}$

In[202]:= **Integrate**[gauss[t, σ], {t, -∞, ∞}, as]

Out[202]= 1

In[203]:= **Integrate**[ftgauss[s, σ], {s, -∞, ∞}, as]

Out[203]= $\frac{1}{\sqrt{2 \pi} \sigma}$

Fourier transform of the box function

In[204]:= **ftbox**[s_] = **FourierTransform**[UnitBox[t], t, s, fp]

Out[204]= **Sinc**[π s]

Product of the Fourier transforms

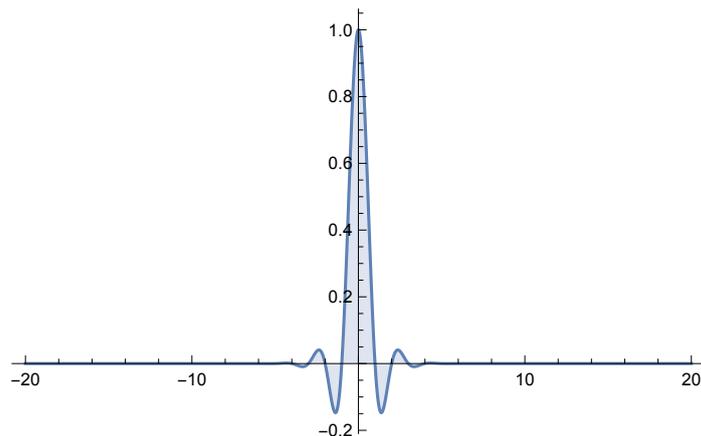
In[205]:= **prod1**[s_, σ_] = **ftbox**[s] * **ftgauss**[s, σ]

Out[205]= $e^{-2 \pi^2 s^2 \sigma^2} \text{Sinc}[\pi s]$

Plotting the product

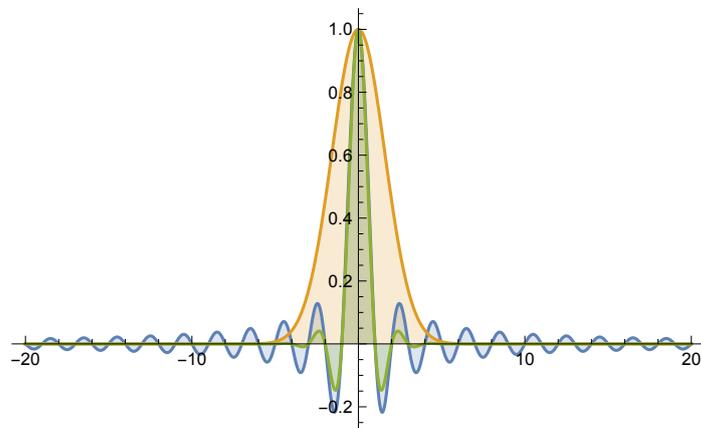
In[206]:= **Plot**[{ **prod1**[s, 0.1]}, {s, -20, 20}, **PlotRange** → **All**, **Filling** → **Axis**]

Out[206]=



In[207]:= **Plot**[{ **ftbox**[s], **ftgauss**[s, 0.1], **prod1**[s, 0.1]}, {s, -20, 20}, **PlotRange** → **All**, **Filling** → **Axis**]

Out[207]=



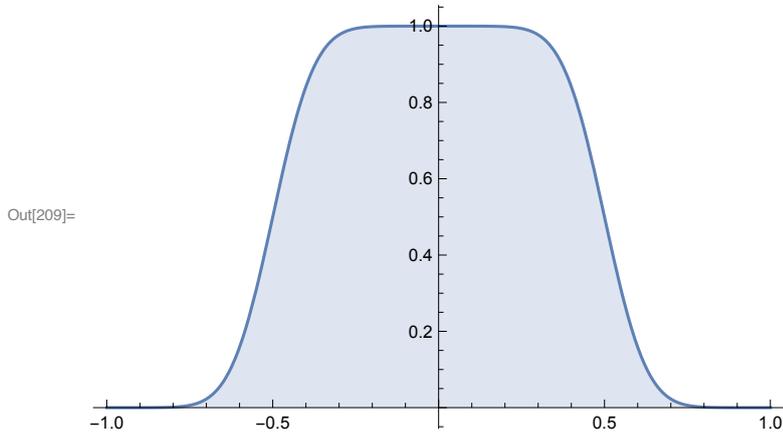
Compute the inverse Fourier transform of the product

In[208]:= **inv1[t_, σ_] = InverseFourierTransform[
prod1[s, σ], s, t, fp, as]**

$$\text{Out[208]= } \frac{1}{\pi} \left(\frac{1}{2} \pi \operatorname{Erf} \left[\frac{1}{2\sqrt{2}\sigma} - \frac{t}{\sqrt{2}\sigma} \right] + \frac{1}{2} \pi \operatorname{Erf} \left[\frac{1}{2\sqrt{2}\sigma} + \frac{t}{\sqrt{2}\sigma} \right] \right)$$

and plot the result

In[209]:= **Plot[inv1[t, 0.1], {t, -1, 1}, Filling → Axis]**



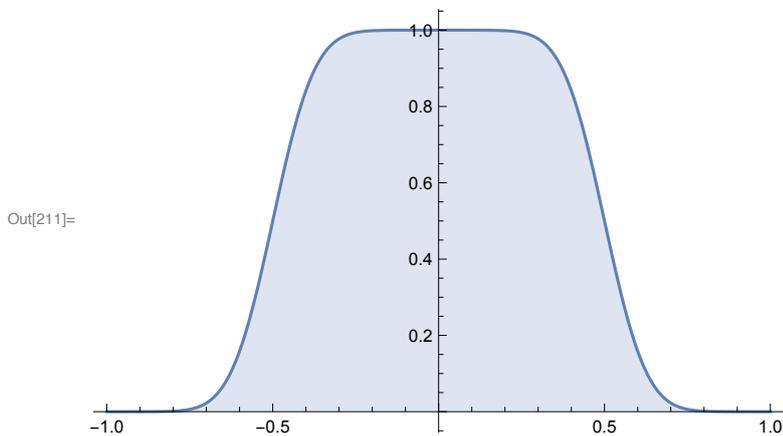
Compute the convolution directly

In[210]:= **conv1[t_, σ_] = Convolve[UnitBox[x], gauss[x, σ], x, t]**

$$\text{Out[210]= } \frac{1}{2} \left(\operatorname{Erf} \left[\frac{1-2t}{2\sqrt{2}\sigma} \right] + \operatorname{Erf} \left[\frac{1+2t}{2\sqrt{2}\sigma} \right] \right)$$

and plot the result

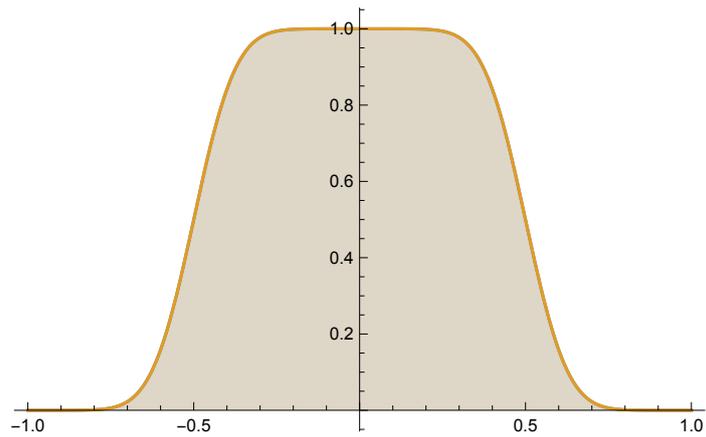
In[211]:= **Plot[{conv1[t, 0.1]}, {t, -1, 1}, PlotRange → All, Filling → Axis]**



Check the convolution theorem visually!

In[212]:= **Plot[{inv1[t, 0.1], conv1[t, 0.1]}, {t, -1, 1}, PlotRange → All, Filling → Axis]**

Out[212]=

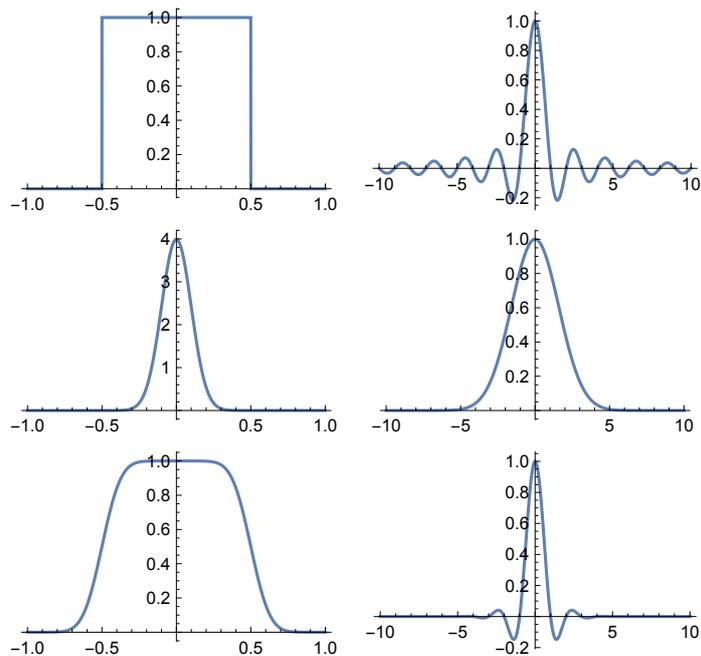


Show all relevant functions in one display

```
In[213]:= show1[σ_] := GraphicsGrid[
  {{Plot[UnitBox[t], {t, -1, 1}], Plot[ftbox[s], {s, -10, 10}, PlotRange -> All]},
  {Plot[gauss[t, σ], {t, -1, 1}, PlotRange -> All],
  Plot[ftgauss[s, σ], {s, -10, 10}]}, {Plot[conv1[t, σ], {t, -1, 1}],
  Plot[prodl[s, σ], {s, -10, 10}, PlotRange -> All]}}
```

In[214]:= show1[0.1]

Out[214]=



Detecting edges with the mexhat filter (high-pass)

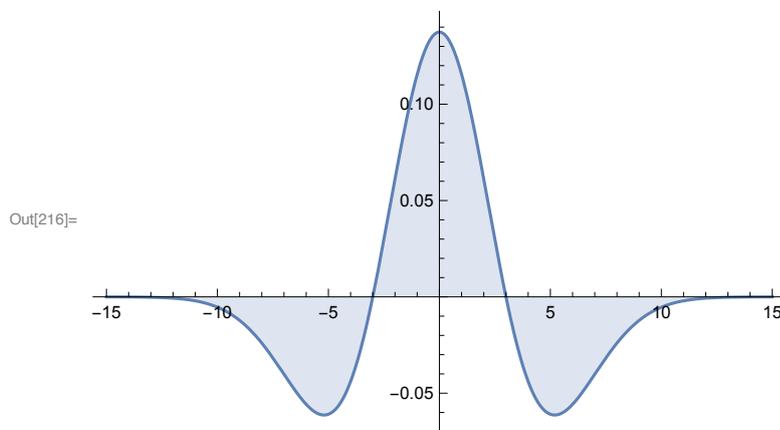
Define the mexhat filter/kernel (derivative of Gauss)

```
In[215]:= mexhat[t_, σ_] = Module[{tmp, norm},
  tmp = D[gauss[t, σ], t, t];
  norm = Integrate[Abs[tmp], {t, -∞, ∞}, as];
  Simplify[-tmp / norm]
]
```

$$\text{Out[215]= } \frac{e^{\frac{1}{2} - \frac{t^2}{2\sigma^2}} (-t^2 + \sigma^2)}{4\sigma^3}$$

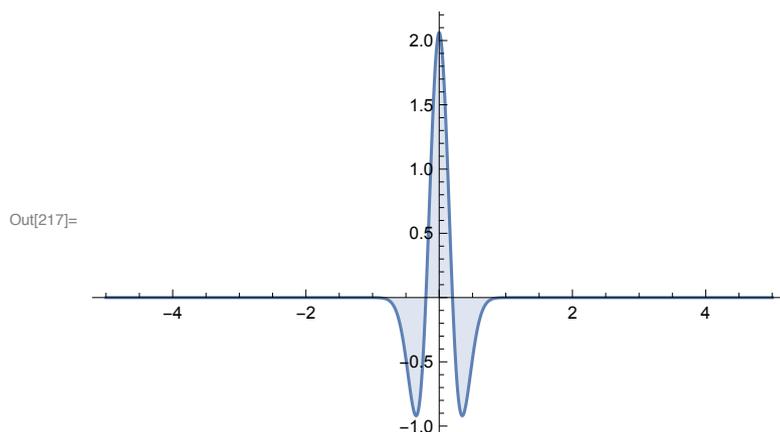
Plot the mexhat filter for $\sigma=3$

```
In[216]:= Plot[Evaluate[mexhat[t, 3]], {t, -15, 15}, PlotRange → All, Filling → Axis]
```



Plot the mexhat filter for $\sigma=0.2$

```
In[217]:= Plot[Evaluate[mexhat[t, 0.2]], {t, -5, 5}, PlotRange → All, Filling → Axis]
```



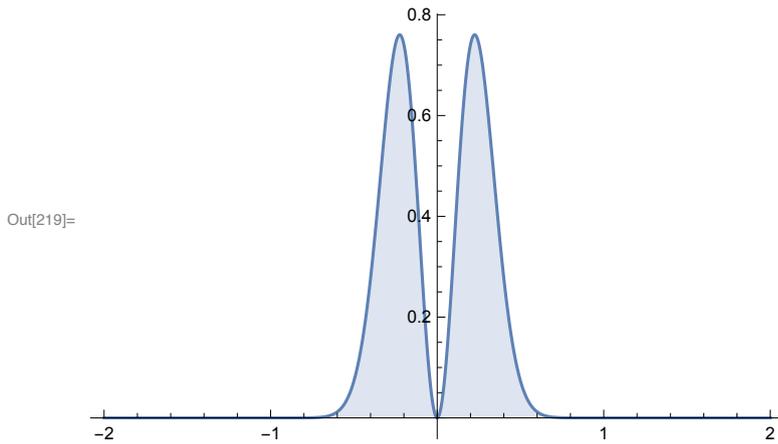
Compute the Fourier transform of the mexhat kernel

```
In[218]:= ftmexhat[s_, σ_] = FourierTransform[mexhat[t, σ], t, s, fp, as]
```

$$\text{Out[218]= } \sqrt{2} e^{\frac{1}{2} - 2\pi^2 s^2 \sigma^2} \pi^{5/2} s^2 \sigma^2$$

and plot it (for $\sigma=1$)

```
In[219]:= Plot[ftmexhat[s, 1], {s, -2, 2}, PlotRange → All, Filling → Axis]
```



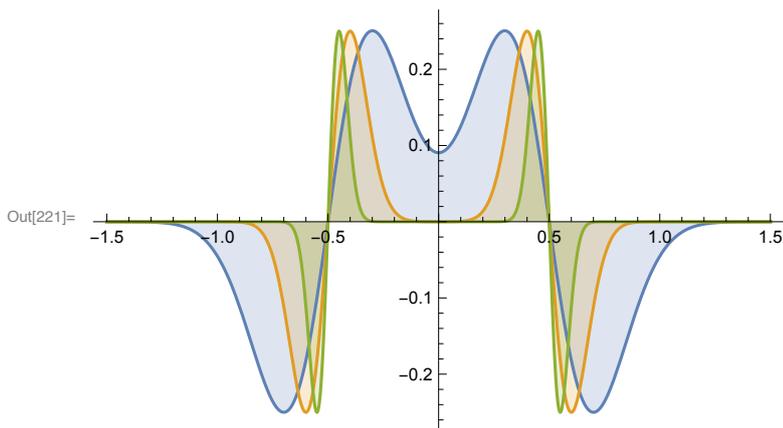
Compute convolution of the box function and the mexhat kernel directly

In[220]= `conv2[t_, σ_] = Convolve[UnitBox[x], mexhat[x, σ], x, t]`

$$\text{Out[220]= } -\frac{1}{8\sigma} e^{\frac{1}{2} - \frac{(1+2t)^2}{8\sigma^2}} \left(-1 - 2t + e^{\frac{t}{\sigma^2}} (-1 + 2t) \right)$$

Plot the result for various values of σ

In[221]= `Plot[{conv2[t, 0.2], conv2[t, 0.1], conv2[t, 0.05]},
{t, -1.5, 1.5}, PlotRange -> All, Filling -> Axis]`



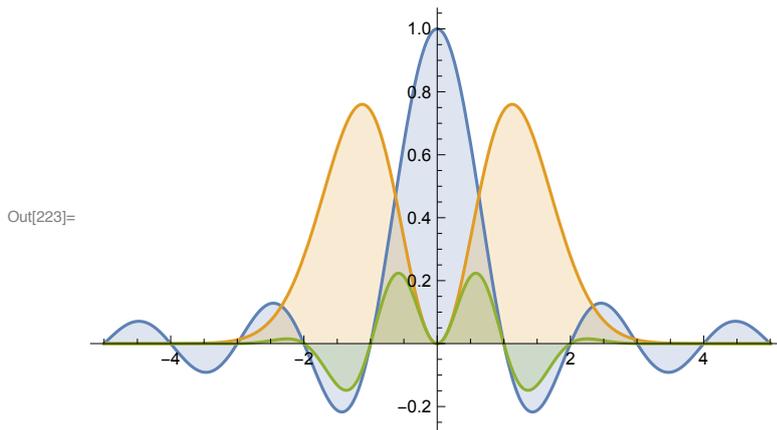
Compute the ordinary product of the Fourier transforms of the box function and the mexhat kernel

In[222]= `prod2[s_, σ_] = ftbox[s] ftmexhat[s, σ]`

$$\text{Out[222]= } \sqrt{2} e^{\frac{1}{2} - 2\pi^2 s^2 \sigma^2} \pi^{5/2} s^2 \sigma^2 \text{Sinc}[\pi s]$$

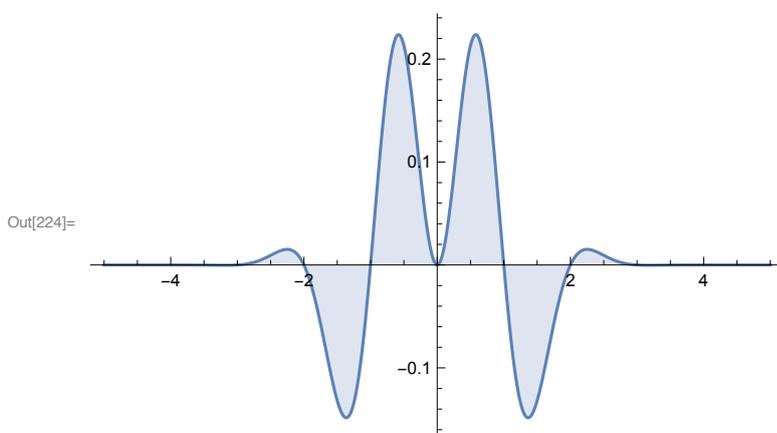
Plot the Fourier transforms and their product

In[223]= `Plot[{ftbox[s], ftmexhat[s, 0.2], prod2[s, 0.2]},
{s, -5, 5}, PlotRange -> All, Filling -> Axis]`



Plot of the product in the frequency domain

In[224]= `Plot[prod2[s, 0.2], {s, -5, 5}, PlotRange -> All, Filling -> Axis]`



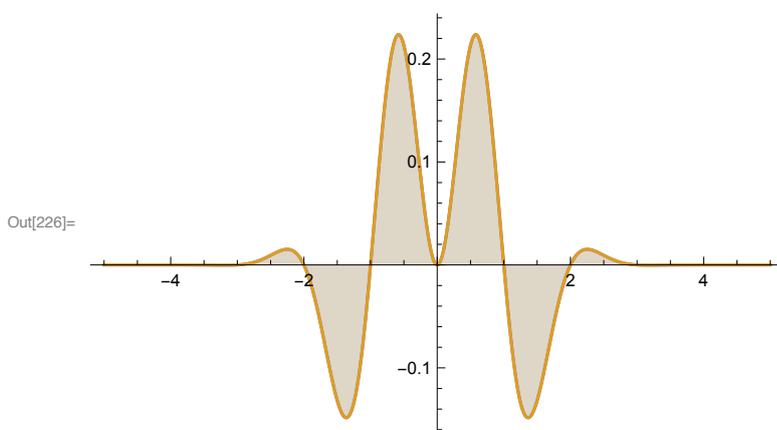
Compute the Fourier transform of the convolution

In[225]= `ftconv2[s_, σ_] = FourierTransform[conv2[t, σ], t, s, fp, as]`

$$\text{Out[225]= } -\frac{1}{\sqrt{2}} i e^{\frac{1}{2} - \pi s} (i + 2 \pi s \sigma^2) (-1 + e^{2 i \pi s}) \pi^{3/2} s \sigma^2$$

and plot it in comparison to the product of the Fourier transforms

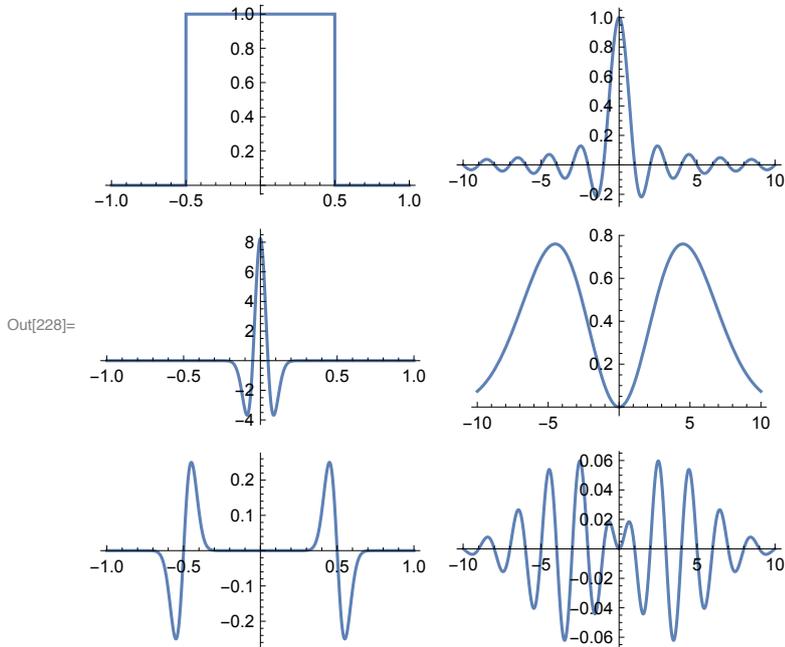
In[226]= `Plot[{ftconv2[s, 0.2], prod2[s, 0.2]}, {s, -5, 5}, PlotRange -> All, Filling -> Axis]`



Display of the relevant functions

```
In[227]:= show2[σ_] := GraphicsGrid[
  {{Plot[UnitBox[t], {t, -1, 1}], Plot[ftbox[s], {s, -10, 10}, PlotRange -> All]},
  {Plot[mexhat[t, σ], {t, -1, 1}, PlotRange -> All],
  Plot[ftmexhat[s, σ], {s, -10, 10}]}, {Plot[conv2[t, σ], {t, -1, 1},
  PlotRange -> All], Plot[prod2[s, σ], {s, -10, 10}, PlotRange -> All]}}
```

```
In[228]:= show2[0.05]
```



Checking formally that the convolution theorem is satisfied is not so easy!

```
In[229]:= s1[s_, ord_] := Series[prod2[s, σ], {s, 0, ord}]
```

```
In[230]:= s2[s_, ord_] := Series[ftconv2[s, σ], {s, 0, ord}]
```

```
In[231]:= s1[s, 6]
```

$$\text{Out[231]= } \sqrt{2} e^{-\pi^{5/2} \sigma^2} s^2 + \left(-\frac{1}{3} \sqrt{\frac{e}{2}} \pi^{9/2} \sigma^2 - 2 \sqrt{2} e^{-\pi^{9/2} \sigma^4} \right) s^4 + \frac{1}{120} \left(\sqrt{2} e^{-\pi^{13/2} \sigma^2} + 40 \sqrt{2} e^{-\pi^{13/2} \sigma^4} + 240 \sqrt{2} e^{-\pi^{13/2} \sigma^6} \right) s^6 + O[s]^7$$

```
In[232]:= s2[s, 6]
```

$$\text{Out[232]= } \sqrt{2} e^{-\pi^{5/2} \sigma^2} s^2 - \frac{1}{3 \sqrt{2}} \left(\pi^{3/2} \sigma^2 \left(\sqrt{e} \pi^3 + 12 \sqrt{e} \pi^3 \sigma^2 \right) \right) s^4 + \left(\pi^{3/2} \sigma^2 \left(\sqrt{e} \pi^5 + 40 \sqrt{e} \pi^5 \sigma^2 + 240 \sqrt{e} \pi^5 \sigma^4 \right) s^6 \right) / \left(60 \sqrt{2} \right) + O[s]^7$$

```
In[233]:= Normal[s1[s, 6] - s2[s, 6]]
```

```
Out[233]= 0
```

Sampling, undersampling, oversampling

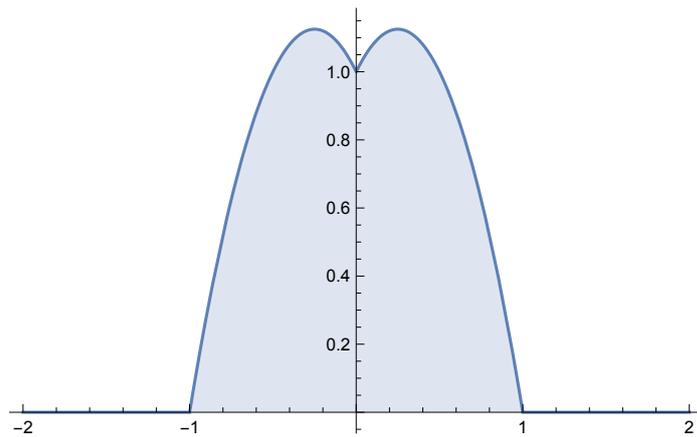
```
fp = FourierParameters → {0, -2 Pi}
```

```
FourierParameters → {0, -2 Pi}
```

Defining the spectrum of a signal. The signal is 2-bandlimited!

```
spec[s_] := Piecewise[{{1 + Abs[s] - 2 * Abs[s]^2, Abs[s] < 1}}]
```

```
Plot[spec[s], {s, -2, 2}, Filling → Axis]
```



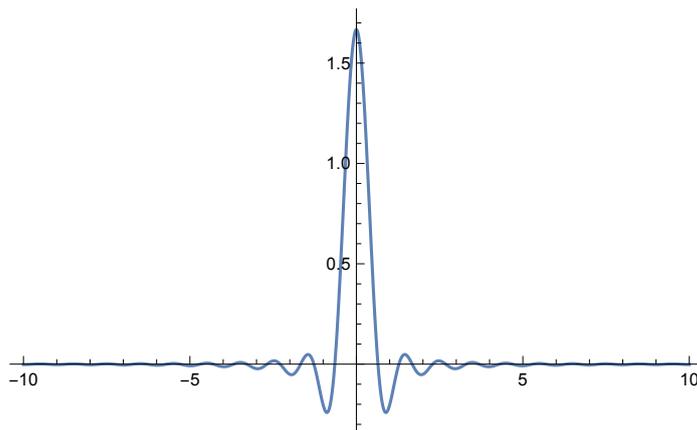
Computing the signal by using the iFT

```
sig[t_] = Simplify[ExpToTrig[InverseFourierTransform[spec[s], s, t, fp]]]
```

```
- ((pi t + 3 pi t Cos[2 pi t] - 2 Sin[2 pi t]) / (2 pi^3 t^3))
```

```
sigzero[t_] := Module[{s}, If[t == 0, Limit[sig[s], s → 0], sig[t]]]
```

```
Plot[sigzero[t], {t, -10, 10}, PlotRange -> All]
```



Taking samples of the signal in distance 1/d

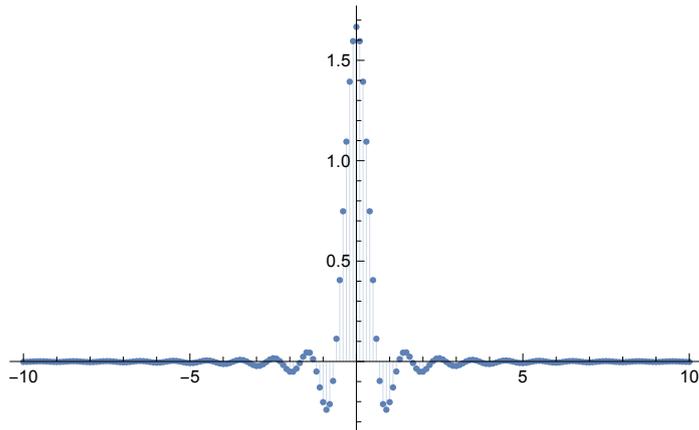
```
sample[d_] :=
```

```
Sum[sigzero[k/d] * DiracDelta[t + (k/d)], {k, -10*d, 10*d}];
```

```
showsample[d_] := ListPlot[
```

```
Table[{k/d, sigzero[k/d]}, {k, -10*d, 10*d}], PlotRange → All, Filling → Axis]
```

```
showsample[10]
```

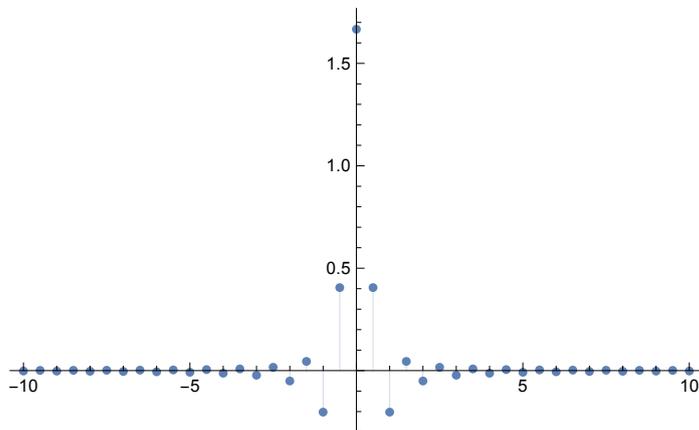


Computing the spectrum of the sampled signal

```
siginv[s_, d_] := Simplify[ExpToTrig[FourierTransform[sample[d], t, s, fp]]]/d
```

The case $d=2$. This is the limit case for sampling.

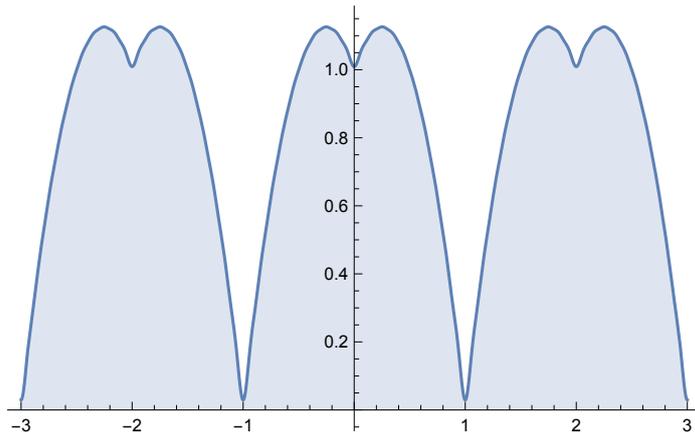
```
showsample[2]
```



```
y2 = siginv[s, 2]
```

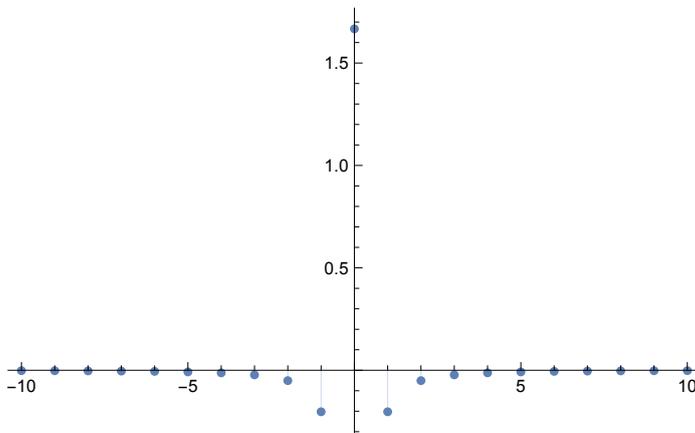
$$\frac{1}{2} \left(\frac{5}{3} + \frac{8 \cos[\pi s]}{\pi^2} - \frac{4 \cos[2 \pi s]}{\pi^2} + \frac{8 \cos[3 \pi s]}{9 \pi^2} - \frac{\cos[4 \pi s]}{\pi^2} + \frac{8 \cos[5 \pi s]}{25 \pi^2} - \frac{4 \cos[6 \pi s]}{9 \pi^2} + \frac{8 \cos[7 \pi s]}{49 \pi^2} - \frac{\cos[8 \pi s]}{4 \pi^2} + \frac{8 \cos[9 \pi s]}{81 \pi^2} - \frac{4 \cos[10 \pi s]}{25 \pi^2} + \frac{8 \cos[11 \pi s]}{121 \pi^2} - \frac{\cos[12 \pi s]}{9 \pi^2} + \frac{8 \cos[13 \pi s]}{169 \pi^2} - \frac{4 \cos[14 \pi s]}{4 \pi^2} + \frac{8 \cos[15 \pi s]}{8 \pi^2} - \frac{4 \cos[16 \pi s]}{16 \pi^2} + \frac{8 \cos[17 \pi s]}{289 \pi^2} - \frac{4 \cos[18 \pi s]}{81 \pi^2} + \frac{8 \cos[19 \pi s]}{361 \pi^2} - \frac{\cos[20 \pi s]}{25 \pi^2} \right)$$

```
p2 = Plot[y2, {s, -3, 3}, PlotRange -> All, Filling -> Axis]
```



The case $d=1$. This is a case of undersampling.

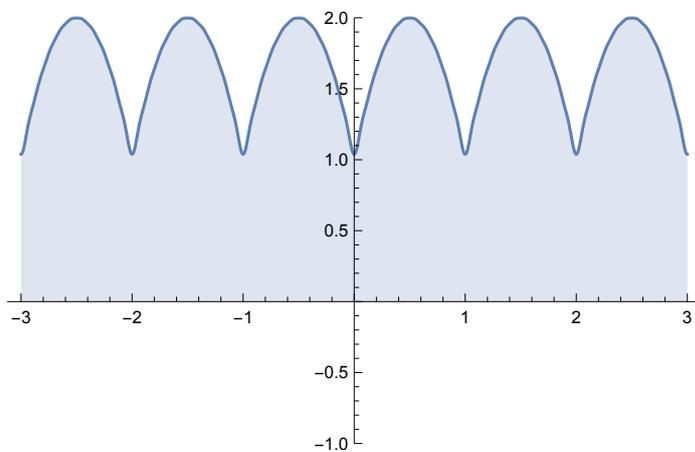
`showsample[1]`



`y1 = siginv[s, 1]`

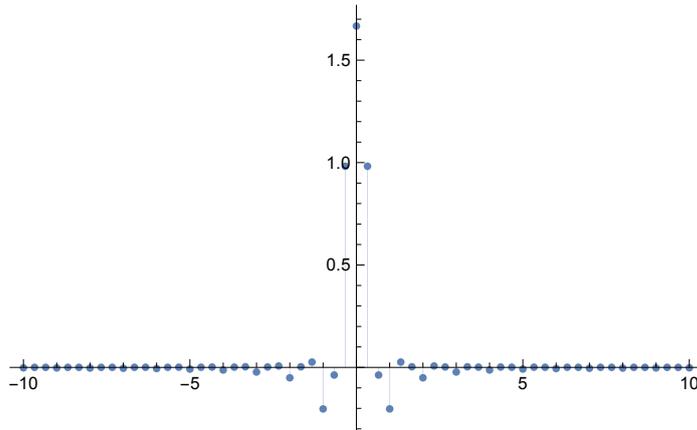
$$\frac{5}{3} \frac{4 \cos[2 \pi s]}{\pi^2} - \frac{\cos[4 \pi s]}{\pi^2} - \frac{4 \cos[6 \pi s]}{9 \pi^2} - \frac{\cos[8 \pi s]}{4 \pi^2} - \frac{4 \cos[10 \pi s]}{25 \pi^2} - \frac{\cos[12 \pi s]}{9 \pi^2} - \frac{4 \cos[14 \pi s]}{49 \pi^2} - \frac{\cos[16 \pi s]}{16 \pi^2} - \frac{4 \cos[18 \pi s]}{81 \pi^2} - \frac{\cos[20 \pi s]}{25 \pi^2}$$

`p1 = Plot[y1, {s, -3, 3}, PlotRange -> {-1, 2}, Filling -> Axis]`



The case $d=3$. This is a case of oversampling.

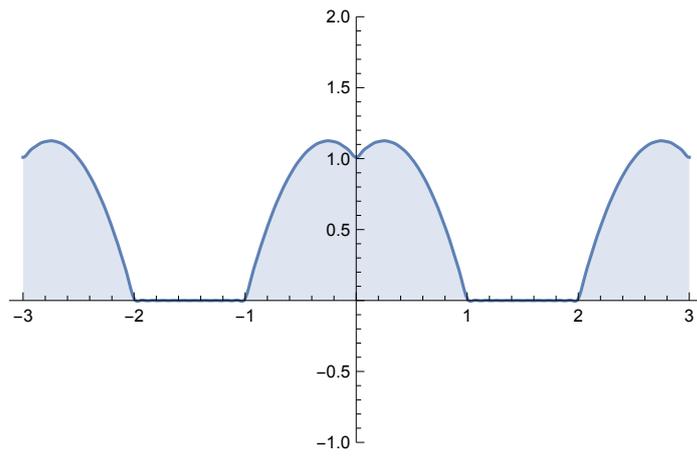
`showsample[3]`



`y3 = signv[s, 3]`

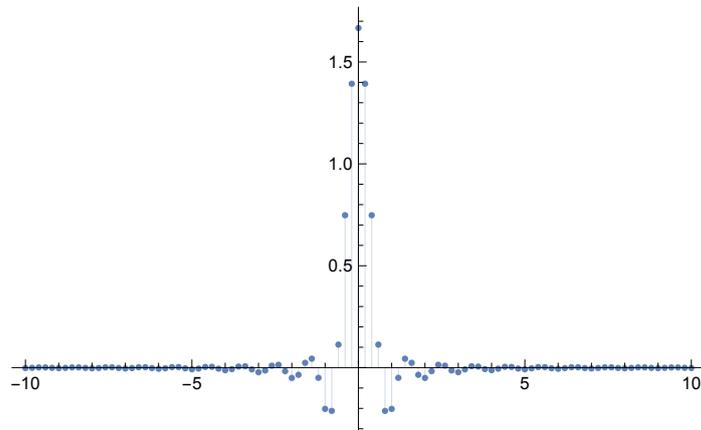
$$\frac{1}{3} \left(\frac{5}{3} + \frac{1}{2\pi^3} 9 (6\sqrt{3} + \pi) \cos\left[\frac{2\pi s}{3}\right] + \frac{1}{8\pi^3} 9 (-3\sqrt{3} + \pi) \cos\left[\frac{4\pi s}{3}\right] - \frac{4 \cos[2\pi s]}{\pi^2} + \right. \\ \frac{27\sqrt{3} \cos\left[\frac{8\pi s}{3}\right]}{64\pi^3} + \frac{9 \cos\left[\frac{8\pi s}{3}\right]}{32\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{10\pi s}{3}\right]}{125\pi^3} + \frac{9 \cos\left[\frac{10\pi s}{3}\right]}{50\pi^2} - \frac{\cos[4\pi s]}{\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{14\pi s}{3}\right]}{343\pi^3} + \frac{9 \cos\left[\frac{14\pi s}{3}\right]}{98\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{16\pi s}{3}\right]}{512\pi^3} + \frac{9 \cos\left[\frac{16\pi s}{3}\right]}{128\pi^2} - \frac{4 \cos[6\pi s]}{9\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{20\pi s}{3}\right]}{1000\pi^3} + \frac{9 \cos\left[\frac{20\pi s}{3}\right]}{200\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{22\pi s}{3}\right]}{1331\pi^3} + \frac{9 \cos\left[\frac{22\pi s}{3}\right]}{242\pi^2} - \frac{\cos[8\pi s]}{4\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{26\pi s}{3}\right]}{2197\pi^3} + \frac{9 \cos\left[\frac{26\pi s}{3}\right]}{338\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{28\pi s}{3}\right]}{2744\pi^3} + \frac{9 \cos\left[\frac{28\pi s}{3}\right]}{392\pi^2} - \frac{4 \cos[10\pi s]}{25\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{32\pi s}{3}\right]}{4096\pi^3} + \frac{9 \cos\left[\frac{32\pi s}{3}\right]}{512\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{34\pi s}{3}\right]}{4913\pi^3} + \frac{9 \cos\left[\frac{34\pi s}{3}\right]}{578\pi^2} - \frac{\cos[12\pi s]}{9\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{38\pi s}{3}\right]}{6859\pi^3} + \frac{9 \cos\left[\frac{38\pi s}{3}\right]}{722\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{40\pi s}{3}\right]}{8000\pi^3} + \frac{9 \cos\left[\frac{40\pi s}{3}\right]}{800\pi^2} - \frac{4 \cos[14\pi s]}{49\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{44\pi s}{3}\right]}{10648\pi^3} + \frac{9 \cos\left[\frac{44\pi s}{3}\right]}{968\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{46\pi s}{3}\right]}{12167\pi^3} + \frac{9 \cos\left[\frac{46\pi s}{3}\right]}{1058\pi^2} - \frac{\cos[16\pi s]}{16\pi^2} + \\ \frac{27\sqrt{3} \cos\left[\frac{50\pi s}{3}\right]}{15625\pi^3} + \frac{9 \cos\left[\frac{50\pi s}{3}\right]}{1250\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{52\pi s}{3}\right]}{17576\pi^3} + \frac{9 \cos\left[\frac{52\pi s}{3}\right]}{1352\pi^2} - \frac{4 \cos[18\pi s]}{81\pi^2} + \\ \left. \frac{27\sqrt{3} \cos\left[\frac{56\pi s}{3}\right]}{21952\pi^3} + \frac{9 \cos\left[\frac{56\pi s}{3}\right]}{1568\pi^2} - \frac{27\sqrt{3} \cos\left[\frac{58\pi s}{3}\right]}{24389\pi^3} + \frac{9 \cos\left[\frac{58\pi s}{3}\right]}{1682\pi^2} - \frac{\cos[20\pi s]}{25\pi^2} \right)$$

`p3 = Plot[y3, {s, -3, 3}, PlotRange → {-1, 2}, Filling → Axis]`



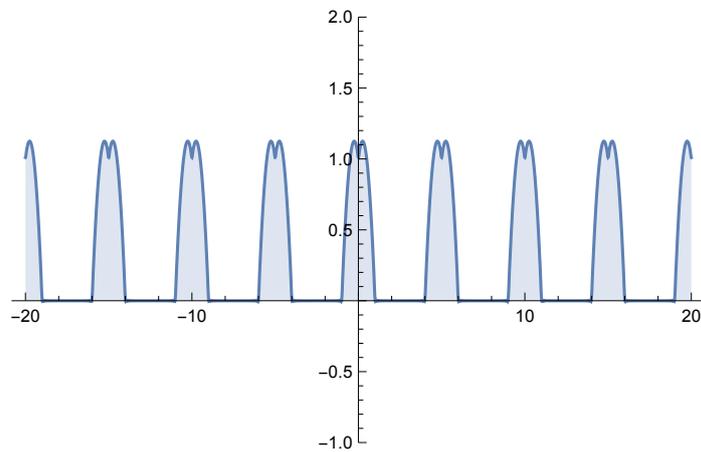
The case $d=5$ gives even more oversampling.

```
showsample[5]
```



```
y5 = siginv[s, 5];
```

```
p10 = Plot[y5, {s, -20, 20}, PlotRange -> {-1, 2}, Filling -> Axis]
```



Computing its spectrum from the signal

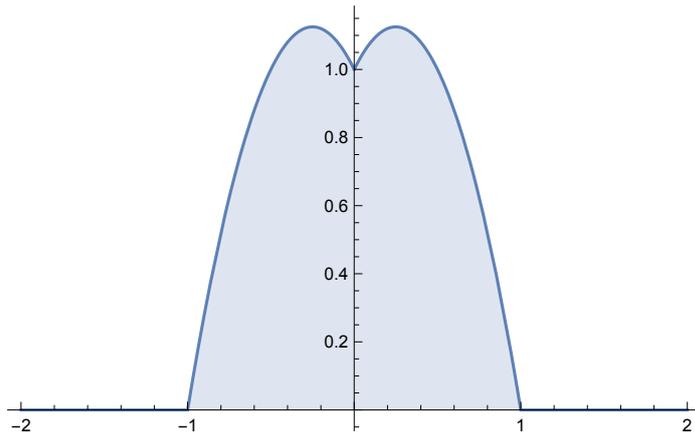
```
sig[t]
```

$$-\left(\pi t + 3\pi t \cos[2\pi t] - 2\sin[2\pi t]\right) / \left(2\pi^3 t^3\right)$$

```
spec2 = FourierTransform[sig[t], t, s, fp]
```

$$-\frac{1}{2\pi^3} \left(\pi^3 \operatorname{Sign}[-1+s] + \pi^3 s \operatorname{Sign}[-1+s] - 2\pi^3 s^2 \operatorname{Sign}[-1+s] - \right. \\ \left. 2\pi^3 s \operatorname{Sign}[s] - \pi^3 \operatorname{Sign}[1+s] + \pi^3 s \operatorname{Sign}[1+s] + 2\pi^3 s^2 \operatorname{Sign}[1+s] \right)$$

```
Plot[spec2, {s, -2, 2}, Filling -> Axis]
```



Demonstrating the Sampling Theorem

Preparation

```
In[1]:= fp = FourierParameters -> {0, -2 Pi}
```

```
Out[1]:= FourierParameters -> {0, -2 Pi}
```

Defining the components of the spectrum of a bandlimited signal

The box

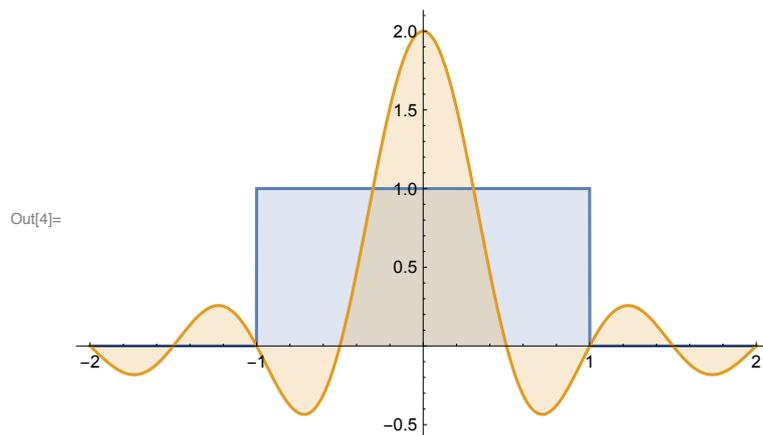
```
In[2]:= box[a_, t_] = UnitBox[t / (2 a)]
```

```
Out[2]:= UnitBox[ $\frac{t}{2 a}$ ]
```

```
In[3]:= ibox[a_, t_] = InverseFourierTransform[box[a, s], s, t, fp]
```

```
Out[3]:= 2 Abs[a] Sinc[2 a Pi t]
```

```
In[4]:= Plot[{box[1, t], ibox[1, t]}, {t, -2, 2}, Filling -> Axis]
```



The triangle

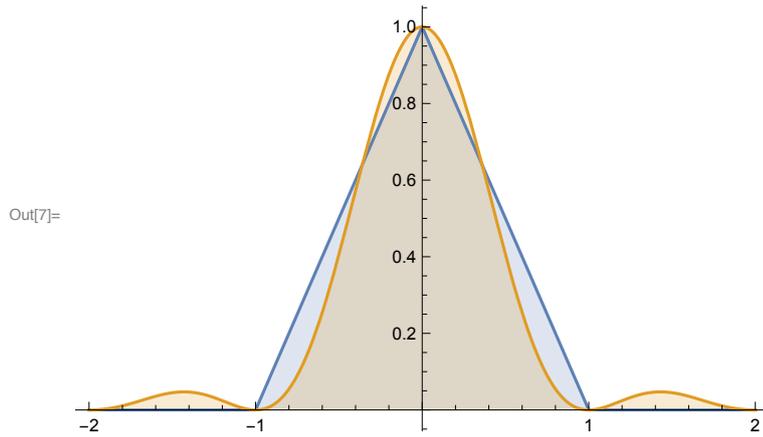
```
In[5]:= triangle[a_, t_] = a * UnitTriangle[t]
```

```
Out[5]:= a UnitTriangle[t]
```

```
In[6]:= itriangle[a_, t_] = InverseFourierTransform[triangle[a, s], s, t, fp]
```

```
Out[6]:= a Sinc[Pi t]^2
```

```
In[7]:= Plot[{triangle[1, t], itriangle[1, t]}, {t, -2, 2}, Filling -> Axis]
```



The arc

In[8]:= **arc**[a_, t_] = Piecewise[{{a * Sqrt[1 - t^2], Abs[t] ≤ 1}}

Out[8]=
$$\begin{cases} a \sqrt{1 - t^2} & \text{Abs}[t] \leq 1 \\ 0 & \text{True} \end{cases}$$

In[9]:= **iarc**[a_, t_] = InverseFourierTransform[arc[a, s], s, t, fp]

Out[9]= $(a \text{ BesselJ}[1, 2 \pi \text{ Abs}[t]]) / (2 \text{ Abs}[t])$

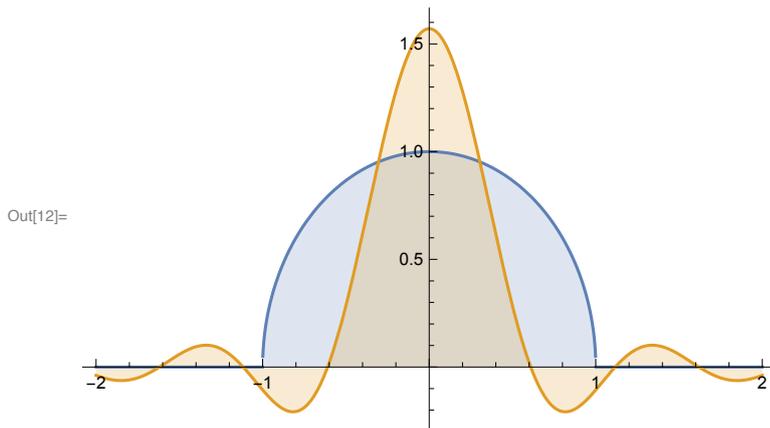
In[10]:= **limitarc**[a_] = Limit[iarc[a, t], t → 0]

Out[10]=
$$\frac{a \pi}{2}$$

In[11]:= **iarc2**[a_, t_] = If[t == 0, limitarc[a], iarc[a, t]]

Out[11]= If[t == 0, limitarc[a], iarc[a, t]]

In[12]:= **Plot**[{arc[1, t], iarc2[1, t]}, {t, -2, 2}, Filling → Axis]



The truncated cosine

In[13]:= **co**ss[a_, t_] = Piecewise[{{Cos[a * Pi * t], Abs[t] ≤ 1}}

Out[13]=
$$\begin{cases} \text{Cos}[a \pi t] & \text{Abs}[t] \leq 1 \\ 0 & \text{True} \end{cases}$$

In[14]:= **ico**ss[a_, t_] = InverseFourierTransform[co

Out[14]= $(2 (a \text{ Cos}[2 \pi t] \text{ Sin}[a \pi] - 2 t \text{ Cos}[a \pi] \text{ Sin}[2 \pi t])) / (\pi (a^2 - 4 t^2))$

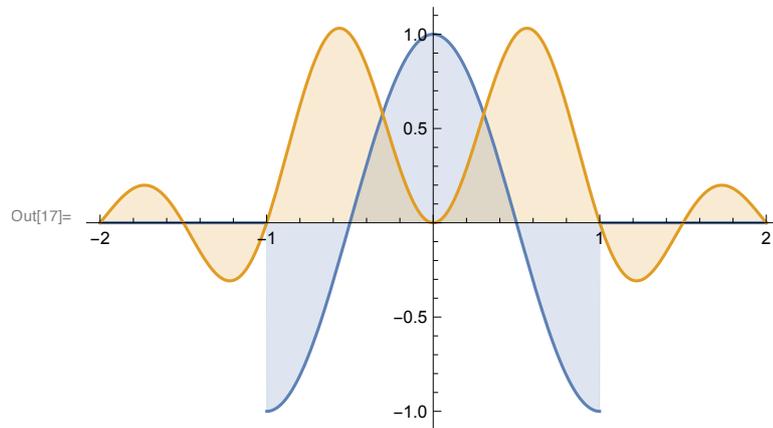
In[15]:= **limit**[a_] = Limit[ico

$$\text{Out[15]= } 1 + \frac{\text{Sin}[2 a \pi]}{2 a \pi}$$

In[16]:= **icoss2[a_, t_] = If[t == a / 2, limit[a], icoss[a, t]]**

Out[16]= **If[t == $\frac{a}{2}$, limit[a], icoss[a, t]]**

In[17]:= **Plot[{coss[1, t], icoss2[1, t]}, {t, -2, 2}, Filling -> Axis]**



The spectrum and the signal

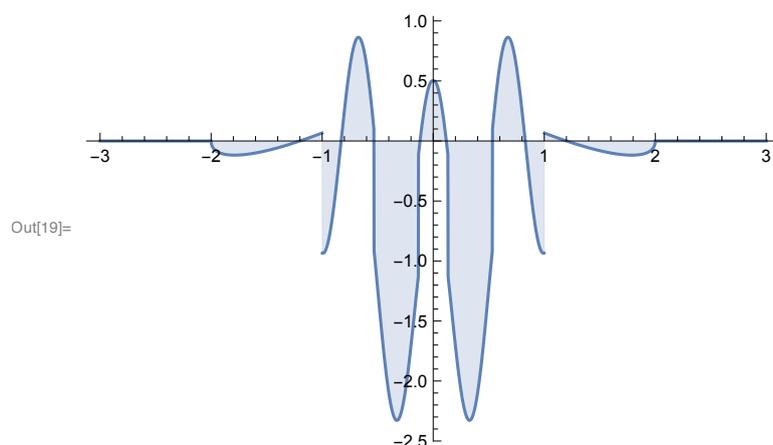
Synthesizing the spectrum

In[18]:= **spec[s_] = triangle[1 / 2, s + 1] - 1 / 2 * arc[1, s / 2] +
triangle[1 / 2, s - 1] - box[1 / 5, s - 1 / 3] - box[1 / 5, s + 1 / 3] + coss[3, s]**

Out[18]=
$$-\frac{1}{2} \left(\begin{cases} \sqrt{1 - \frac{s^2}{4}} & \frac{\text{Abs}[s]}{2} \leq 1 \\ 0 & \text{True} \end{cases} \right) + \left(\begin{cases} \text{Cos}[3 \pi s] & \text{Abs}[s] \leq 1 \\ 0 & \text{True} \end{cases} \right) - \text{UnitBox}\left[\frac{5}{2} \left(-\frac{1}{3} + s\right)\right] -$$

$$\text{UnitBox}\left[\frac{5}{2} \left(\frac{1}{3} + s\right)\right] + \frac{1}{2} \text{UnitTriangle}[1 - s] + \frac{1}{2} \text{UnitTriangle}[1 + s]$$

In[19]:= **Plot[spec[s], {s, -3, 3}, Filling -> Axis, PlotRange -> All]**



The signal by inverse Fourier transform

In[20]:= **sig1[t_] = InverseFourierTransform[spec[s], s, t, fp]**

$$\text{Out[20]= } \frac{1}{8} \left(-\frac{1}{\pi^2} \left(4 \pi^3 \text{Hypergeometric0F1Regularized}[2, -4 \pi^2 t^2] + \frac{1}{t^2} \left((-1 + e^{2 i \pi t})^2 + 8 e^{-\frac{2}{3} i \pi t} \left(1 + e^{\frac{4 i \pi t}{3}} \right) \pi t \text{Sin}\left[\frac{2 \pi t}{5}\right] + (32 \pi t^3 \text{Sin}[2 \pi t]) / (-9 + 4 t^2) \right) \right) + 4 e^{-2 i \pi t} \text{Sinc}[\pi t]^2 \right)$$

Simplification of the analytic expression for the signal

In[21]= **Expand[ExpToTrig[sig1[t]]]**

$$\text{Out[21]= } -\frac{1}{8 \pi^2 t^2} + \frac{\text{Cos}[2 \pi t]}{4 \pi^2 t^2} - \frac{\text{Cos}[2 \pi t]^2}{8 \pi^2 t^2} - \frac{1}{2} \pi \text{Hypergeometric0F1Regularized}[2, -4 \pi^2 t^2] - \frac{\text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right]}{\pi t} - \frac{1}{\pi t} \text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Cos}\left[\frac{4 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right] + \frac{i \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right]}{\pi t} + \frac{1}{\pi t} i \text{Cos}\left[\frac{4 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right] - \frac{1}{\pi t} i \text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{4 \pi t}{3}\right] - \frac{1}{\pi t} \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{4 \pi t}{3}\right] + \frac{i \text{Sin}[2 \pi t]}{4 \pi^2 t^2} - \frac{4 t \text{Sin}[2 \pi t]}{\pi (-9 + 4 t^2)} - (i \text{Cos}[2 \pi t] \text{Sin}[2 \pi t]) / (4 \pi^2 t^2) + \frac{\text{Sin}[2 \pi t]^2}{8 \pi^2 t^2} + \frac{1}{2} \text{Cos}[2 \pi t] \text{Sinc}[\pi t]^2 - \frac{1}{2} i \text{Sin}[2 \pi t] \text{Sinc}[\pi t]^2$$

In[22]= **li = ReplacePart[%, 0 -> List]**

$$\text{Out[22]= } \left\{ -\frac{1}{8 \pi^2 t^2}, \frac{\text{Cos}[2 \pi t]}{4 \pi^2 t^2}, -\frac{\text{Cos}[2 \pi t]^2}{8 \pi^2 t^2}, -\frac{1}{2} \pi \text{Hypergeometric0F1Regularized}[2, -4 \pi^2 t^2], -\frac{\text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right]}{\pi t}, -\frac{1}{\pi t} \text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Cos}\left[\frac{4 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right], \frac{i \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right]}{\pi t}, \frac{1}{\pi t} i \text{Cos}\left[\frac{4 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right], -\frac{1}{\pi t} i \text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{4 \pi t}{3}\right], -\frac{1}{\pi t} \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{4 \pi t}{3}\right], \frac{i \text{Sin}[2 \pi t]}{4 \pi^2 t^2}, -\frac{4 t \text{Sin}[2 \pi t]}{\pi (-9 + 4 t^2)}, -((i \text{Cos}[2 \pi t] \text{Sin}[2 \pi t]) / (4 \pi^2 t^2)), \frac{\text{Sin}[2 \pi t]^2}{8 \pi^2 t^2}, \frac{1}{2} \text{Cos}[2 \pi t] \text{Sinc}[\pi t]^2, -\frac{1}{2} i \text{Sin}[2 \pi t] \text{Sinc}[\pi t]^2 \right\}$$

In[23]= **FullSimplify[Total[Cases[li, Complex[0, _] ___]]]**

Out[23]= 0

In[24]= **ExpToTrig[Total>DeleteCases[li, Complex[0, _] ___]]]**

$$\text{Out[24]= } -\frac{1}{8 \pi^2 t^2} + \frac{\text{Cos}[2 \pi t]}{4 \pi^2 t^2} - \frac{\text{Cos}[2 \pi t]^2}{8 \pi^2 t^2} - \frac{1}{2} \pi \text{Hypergeometric0F1Regularized}[2, -4 \pi^2 t^2] - \frac{\text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right]}{\pi t} - \frac{1}{\pi t} \text{Cos}\left[\frac{2 \pi t}{3}\right] \text{Cos}\left[\frac{4 \pi t}{3}\right] \text{Sin}\left[\frac{2 \pi t}{5}\right] - \frac{1}{\pi t} \text{Sin}\left[\frac{2 \pi t}{5}\right] \text{Sin}\left[\frac{2 \pi t}{3}\right] \text{Sin}\left[\frac{4 \pi t}{3}\right] - \frac{4 t \text{Sin}[2 \pi t]}{\pi (-9 + 4 t^2)} + \frac{\text{Sin}[2 \pi t]^2}{8 \pi^2 t^2} + \frac{1}{2} \text{Cos}[2 \pi t] \text{Sinc}[\pi t]^2$$

In[25]= **sig2[t_] = Simplify[%]**

$$\text{Out[25]} = \frac{1}{8 \pi^2 t^2 (-9 + 4 t^2)} \left(9 - 4 t^2 - 18 \cos[2 \pi t] + 8 t^2 \cos[2 \pi t] + 9 \cos[4 \pi t] - \right. \\ \left. 4 t^2 \cos[4 \pi t] - 4 \pi^3 t^2 (-9 + 4 t^2) \text{Hypergeometric0F1Regularized}\left[2, -4 \pi^2 t^2\right] - \right. \\ \left. 72 \pi t \sin\left[\frac{4 \pi t}{15}\right] + 32 \pi t^3 \sin\left[\frac{4 \pi t}{15}\right] + 72 \pi t \sin\left[\frac{16 \pi t}{15}\right] - 32 \pi t^3 \sin\left[\frac{16 \pi t}{15}\right] - \right. \\ \left. 32 \pi t^3 \sin[2 \pi t] + 4 \pi^2 t^2 (-9 + 4 t^2) \cos[2 \pi t] \text{Sinc}[\pi t]^2 \right)$$

In[26]:= **10 = Limit[sig2[t], t → 0]**

$$\text{Out[26]} = \frac{1}{5} - \frac{\pi}{2}$$

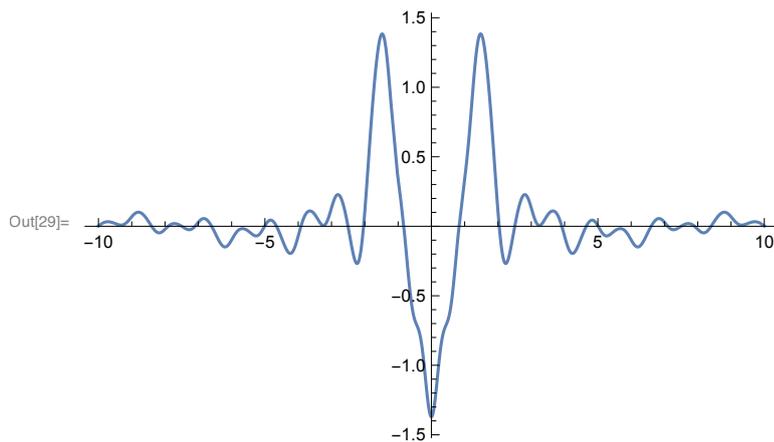
In[27]:= **132 = Limit[sig2[t], t → 3 / 2]**

$$\text{Out[27]} = \frac{1}{18 \pi^2} \left(-8 + 6 \sqrt{2 (5 + \sqrt{5})} \pi - 3 \pi^2 (-6 + \text{BesselJ}[1, 6 \pi]) \right)$$

In[28]:= **sig3[t_] = If[t^2 == 9 / 4, 132, If[t == 0, 10, sig2[t]]]**

$$\text{Out[28]} = \text{If}\left[t^2 == \frac{9}{4}, 132, \text{If}[t == 0, 10, \text{sig2}[t]]\right]$$

In[29]:= **Plot[sig3[t], {t, -10, 10}, PlotRange → All]**



Sampling and the Shannon formula

In[30]:= **sample[f_, a_, n_] := Table[N[f[k * a]], {k, -n, n}]**

In[31]:= **sample[sig3, 1, 5]**

$$\text{Out[31]} = \{0.00353768, -0.070746, 0.132314, 0.107409, 0.341363, \\ -1.3708, 0.341363, 0.107409, 0.132314, -0.070746, 0.00353768\}$$

In[32]:= **shannon[f_, t_, a_, b_] := Module[{n, sam},**
n = Ceiling[b / a];
sam = sample[f, a, n];
Sum[sam[[k + n + 1]] * Sinc[Pi * (t / a - k)], {k, -n, n}]
]

In[33]:= **shannon[sig3, t, 1, 5]**

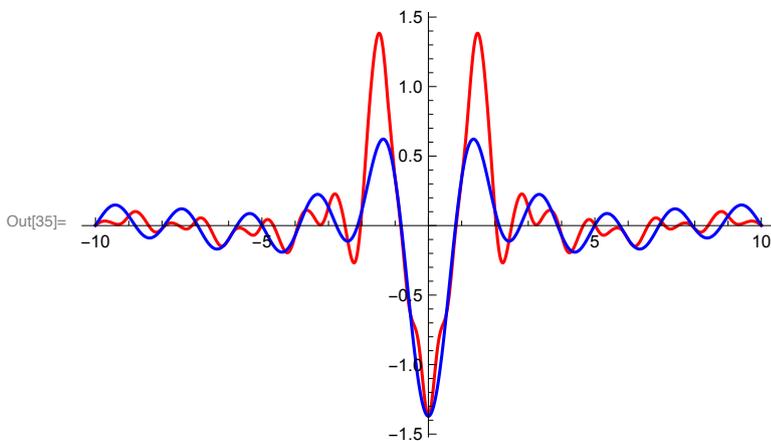
```
Out[33]= 0.00353768 Sinc[ $\pi(-5+t)$ ] - 0.070746 Sinc[ $\pi(-4+t)$ ] +
0.132314 Sinc[ $\pi(-3+t)$ ] + 0.107409 Sinc[ $\pi(-2+t)$ ] + 0.341363 Sinc[ $\pi(-1+t)$ ] -
1.3708 Sinc[ $\pi t$ ] + 0.341363 Sinc[ $\pi(1+t)$ ] + 0.107409 Sinc[ $\pi(2+t)$ ] +
0.132314 Sinc[ $\pi(3+t)$ ] - 0.070746 Sinc[ $\pi(4+t)$ ] + 0.00353768 Sinc[ $\pi(5+t)$ ]
```

A case of severe undersampling

```
In[34]= sha1 = shannon[sig3, t, 1, 10]
```

```
Out[34]= 0.00125448 Sinc[ $\pi(-10+t)$ ] + 0.0687423 Sinc[ $\pi(-9+t)$ ] - 0.0216353 Sinc[ $\pi(-8+t)$ ] +
0.0288675 Sinc[ $\pi(-7+t)$ ] - 0.0982164 Sinc[ $\pi(-6+t)$ ] + 0.00353768 Sinc[ $\pi(-5+t)$ ] -
0.070746 Sinc[ $\pi(-4+t)$ ] + 0.132314 Sinc[ $\pi(-3+t)$ ] + 0.107409 Sinc[ $\pi(-2+t)$ ] +
0.341363 Sinc[ $\pi(-1+t)$ ] - 1.3708 Sinc[ $\pi t$ ] + 0.341363 Sinc[ $\pi(1+t)$ ] +
0.107409 Sinc[ $\pi(2+t)$ ] + 0.132314 Sinc[ $\pi(3+t)$ ] - 0.070746 Sinc[ $\pi(4+t)$ ] +
0.00353768 Sinc[ $\pi(5+t)$ ] - 0.0982164 Sinc[ $\pi(6+t)$ ] + 0.0288675 Sinc[ $\pi(7+t)$ ] -
0.0216353 Sinc[ $\pi(8+t)$ ] + 0.0687423 Sinc[ $\pi(9+t)$ ] + 0.00125448 Sinc[ $\pi(10+t)$ ]
```

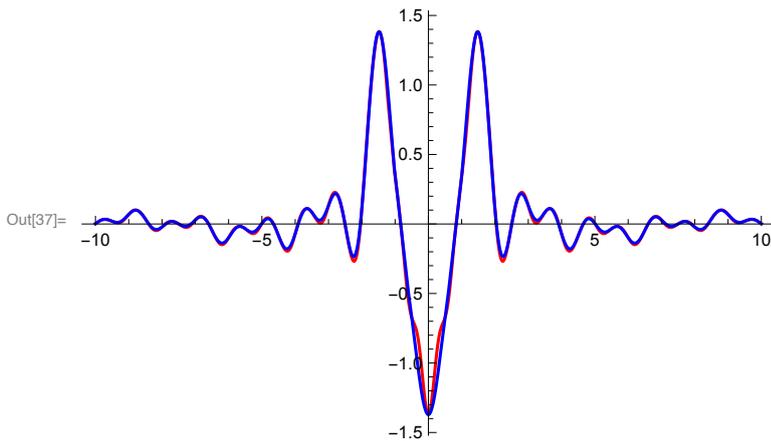
```
In[35]= Plot[{sig1[t], sha1}, {t, -10, 10}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



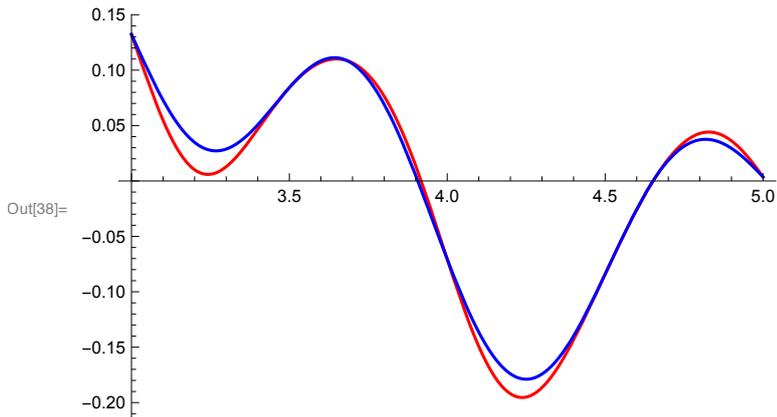
A case of weak undersampling

```
In[36]= sha2 = shannon[sig3, t, 1/2, 10];
```

```
In[37]= Plot[{sig1[t], sha2}, {t, -10, 10}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



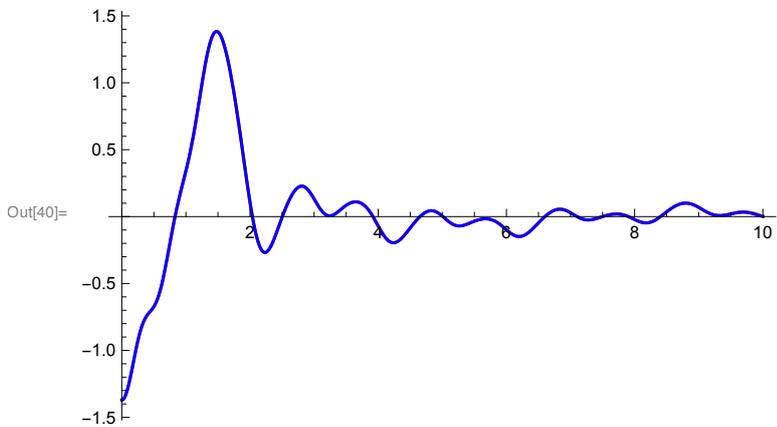
```
In[38]= Plot[{sig1[t], sha2}, {t, 3, 5}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



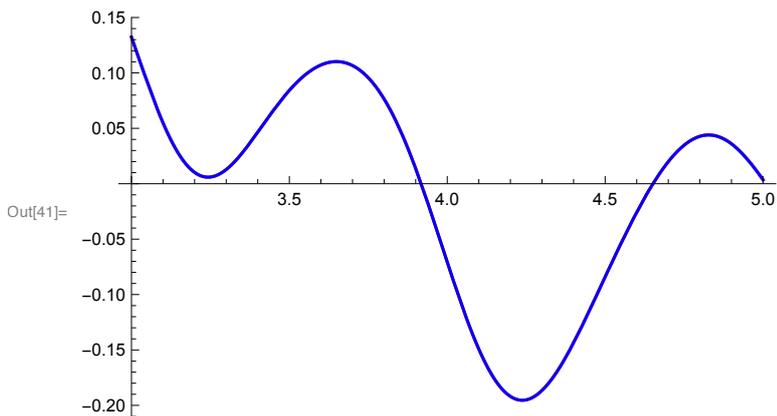
The limit case

```
In[39]:= sha4 = shannon[sig3, t, 1 / 4, 10];
```

```
In[40]:= Plot[{sig1[t], sha4}, {t, 0, 10}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



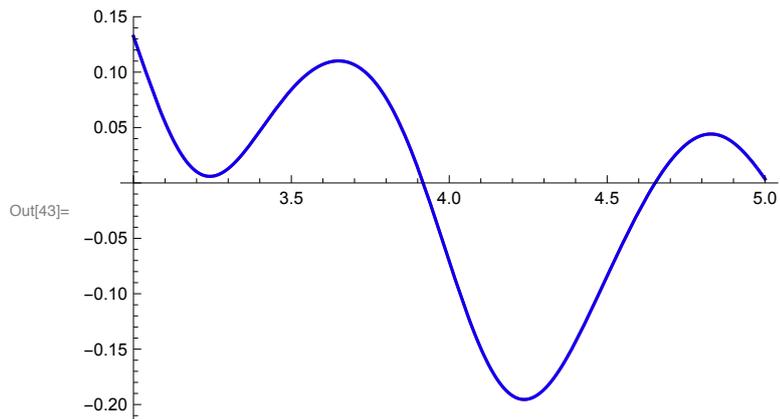
```
In[41]:= Plot[{sig1[t], sha4}, {t, 3, 5}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



A case of oversampling

```
In[42]:= sha5 = shannon[sig3, t, 1 / 5, 10];
```

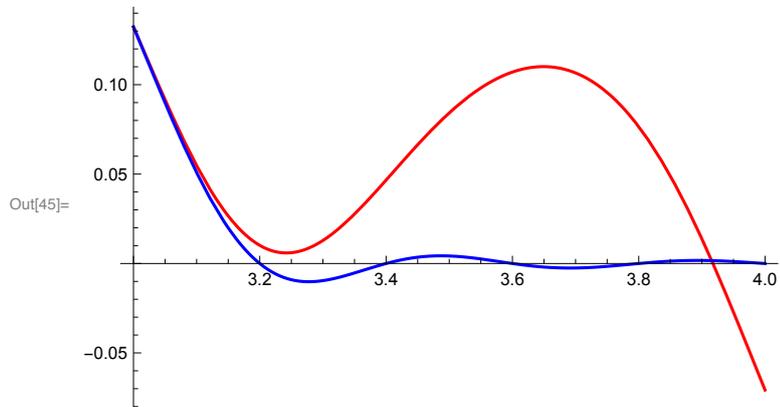
```
In[43]:= Plot[{sig1[t], sha5}, {t, 3, 5}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



Beware of the sampling interval!

```
In[44]:= sha5a = shannon[sig3, t, 1 / 5, 3];
```

```
In[45]:= Plot[{sig3[t], sha5a}, {t, 3, 4}, PlotRange -> All, PlotStyle -> {Red, Blue}]
```



5. Das Haar-Wavelet

5.1 Zur Historie

Im Jahr 1910 konstruierte Alfréd HAAR (1885–1933) in der Arbeit [Haa10] die nach ihm benannten Orthogonalbasen der Räume $\mathcal{L}^2([0, 1])$ und $\mathcal{L}^2(\mathbb{R})$. Sein Ziel war dabei, orthogonale Basen $(h_n(t))_{\geq 0}$ zu konstruieren, bezüglich derer die formale Fourierreihe

$$\langle f | h_0 \rangle h_0(t) + \langle f | h_1 \rangle h_1(t) + \langle f | h_2 \rangle h_2(t) \cdots + \langle f | h_n \rangle h_n(t) + \cdots$$

einer jeden *stetigen* Funktion $f(t) \in C^0[0, 1]$ *gleichmässig* gegen $f(t)$ auf $[0, 1]$ konvergiert. Für die Basis der trigonometrischen Funktionen hatte Paul DU BOIS-REYMOND 1873 gezeigt, dass diese Eigenschaft nicht gilt.

Im Einleitungskapitel wurde Alfréd Haars Konstruktion¹, die auf den Operationen der Skalierung (Dilation und Translation) ganz einfacher Grundfunktionen beruht, bereits vorgestellt und in Beispielen für Wavelettransformationen verwendet. Auch wenn Alfréd Haar nicht Wavelettransformationen im heutigen Sinne betrieben hat, wird seine Arbeit als frühestes Dokument der Wavelet-Theorie angesehen. Anschließend geschah aber viele Jahre wenig, was explizit in diese Richtung weist – geschweige denn, den Begriff *Wavelet* verwendet.

Auf die Geschichte der Wavelets soll hier nicht näher eingegangen werden, siehe dazu das populäre und unterhaltsam geschriebene Buch von Barbara BURKE HUBBARD [Bur96], oder – weitaus anspruchsvoller in jeder Beziehung, insbesondere mathematisch – das Buch von Stéphane JAFFARD, Yves MEYER und Robert D. RYAN [JMR01], insbesondere das zweite Kapitel daraus: *Wavelets from a Historical Perspective*. In diesem Kapitel werden Bezüge von mathematischen Entwicklungen ab den 1930er Jahren (z.B. Kurzzeit-Fourieranalyse und Gabor-Transformation, Zerlegung von Funktionen nach Basen mit Skalierung) zu dem heutigen Begriffsgebäude

¹ Tatsächlich ist Haars Lösung des Problems nur eine von unendlich-vielen Möglichkeiten, aber die bei weitem einfachste!

der Wavelets dargestellt. Aus diesem Kapitel sollen drei unterschiedliche (aber eng verwandte) Definitionen von *Wavelets* zitiert werden, die aus verschiedenen Quellen kommen:

- A. Grossmann und J. Morlet [GM84]

Ein Wavelet ist eine Funktion $\psi(t) \in \mathcal{L}^2(\mathbb{R})$, deren Fouriertransformierte $\widehat{\psi}(s)$

$$\int_0^\infty \left| \widehat{\psi}(s\xi) \right|^2 \frac{1}{s} ds = 1$$

fast überall (bez. ξ) erfüllt.

- E. M. Stein [Ste70]

Ein Wavelet ist eine Funktion $\psi(t) \in \mathcal{L}^2(\mathbb{R})$, deren Fouriertransformierte $\widehat{\psi}(s)$

$$\sum_{-\infty}^{\infty} \left| \widehat{\psi}(2^{-j}\xi) \right|^2 = 1$$

fast überall (bez. ξ) erfüllt.

- J.-O. Strömberg [Str83]

Ein Wavelet ist eine Funktion $\psi(t) \in \mathcal{L}^2(\mathbb{R})$, so dass die durch Skalierung (Dilation und Translation) entstehende Familie von Funktionen

$$\{ 2^{j/2}\psi(2^j t - k); k, j \in \mathbb{Z} \}$$

eine orthonormale Basis von $\mathcal{L}^2(\mathbb{R})$ ist.

Die ersten beiden Definitionen zeigen die enge Verflechtung der Wavelet-Theorie mit der Fourier-Theorie. Historisch ist interessant, dass Axel GROSSMANN (Physiker) und Jean MORLET (Geophysiker) aus ganz konkreten Anwendungsproblemen (Analyse von seismischen Signalen für die Lagerstättensuche von Erdöl) heraus auf Wavelets als Analysehilfsmittel gekommen sind. Die zweite Definition wurzelt in der mathematischen Disziplin der Funktionalanalysis. Beide Definitionen haben die sogenannte *Kontinuierliche Wavelettransformation* (CWT) im Sinne, die im Rahmen dieses Skriptums (noch) nicht behandelt wird (wohl aber in der Vorlesung). Die dritte Definition ist eine allgemeine Formulierung dessen, was Alfréd Haar über 70 Jahre früher in seinem Spezialfall vorgeschlagen hatte und was zur Grundlage der *Diskreten Wavelettransformation* (DWT) geworden ist. Der Fall des Haar-Wavelets wird in diesem Kapitel detailliert behandelt und später im Rahmen der Multiresolution (MRA) verallgemeinert und vertieft. Dann zeigt sich auch der enge Zusammenhang mit den ersten beiden Definitionen und den Eigenschaften der Fouriertransformierten von Waveletfunktionen. Davon werden die späteren Kapitel dieses Skripts einen Eindruck vermitteln.

Zum Abschluss dieser historischen Kurz-Skizze soll noch explizit auf drei *highlights* hingewiesen werden, die in der Geschichte der Wavelets für Theorie und Anwendung eine herausragende Rolle gespielt haben und die in diesem Skriptum behandelt werden:

- Die Konstruktion “kompakter” orthogonaler Wavelets durch Ingrid DAUBECHIES in [Dau88], [Dau93] — siehe Kapitel 8;
- Das Konzept der *Multiresolution* (MRA) von Stéphane MALLAT in [Mal89b], [Mal89a], als vereinheitlichendes Konzept für diskrete Wavelettransformationen — siehe Kapitel 9;
- Die Konstruktion “kompakter” *biorthogonaler* Wavelets durch Albert COHEN, Ingrid DAUBECHIES und Jean-Christophe FEAUVEAU in [CDF93] — siehe Kapitel 10.

5.2 Haar-Skalierungsfunktion und Haar-Waveletfunktion

Definition 5.1 (Dyadische Intervalle und Haar-Funktionen).

Für Parameter $j, k \in \mathbb{Z}$ werden die (halboffenen) dyadische Intervalle definiert als

$$\begin{aligned} I_{j,k} &= [2^{-j}k, 2^{-j}(k+1)), \\ I_{j,k}^{\ell} &= I_{j+1,2k} = [2^{-j-1}(2k), 2^{-j-1}(2k+1)), \\ I_{j,k}^r &= I_{j+1,2k+1} = [2^{-j-1}(2k+1), 2^{-j-1}(2k+2)). \end{aligned}$$

Zwei dyadische Intervalle $I_{j,k}$ und $I_{j',k'}$ sind entweder disjunkt (beispielweise immer dann, wenn $j = j'$ und $k \neq k'$ ist, oder identisch (genau dann, wenn $(j, k) = (j', k')$ ist), oder das kürzere Intervall ist echt in dem längeren enthalten. $I_{j,k}^{\ell}$ und $I_{j,k}^r$ zerlegen $I_{j,k}$ in eine linke und eine rechte Hälfte.

Wie üblich bezeichnet $\mathbf{1}_{[a,b)}(t)$ die Indikatorfunktion des Intervalles $[a, b)$.

Definition 5.2 (Haar-Funktionen).

1. Die Haar-Funktionen sind die beiden Funktionen

$$\begin{aligned} \phi(t) &= \mathbf{1}_{[0,1)}(t) && \text{Haar-Skalierungsfunktion,} \\ \psi(t) &= \mathbf{1}_{[0,1/2)}(t) - \mathbf{1}_{[1/2,1)}(t) && \text{Haar-Waveletfunktion,} \end{aligned}$$

2. sowie die Funktionen, die aus $\phi(t)$ und $\psi(t)$ durch dyadische Dilation und ganzzahlige Translation hervorgehen sind (für $j, k \in \mathbb{Z}$):

$$\begin{aligned} \phi_{j,k}(t) &= 2^{j/2} \mathbf{1}_{I_{j,k}}(t) = 2^{j/2} \phi(2^j t - k) = D_{2^j} T_k \phi(t) \\ \psi_{j,k}(t) &= 2^{j/2} \left(\mathbf{1}_{I_{j,k}^{\ell}}(t) - \mathbf{1}_{I_{j,k}^r}(t) \right) = 2^{j/2} \psi(2^j t - k) = D_{2^j} T_k \psi(t) \end{aligned}$$

Beide Haar-Funktionen $\phi(t)$ und $\psi(t)$ sind normiert: $\|\phi(t)\|_2 = 1 = \|\psi(t)\|_2$. Weiter sind die Skalierungsfaktoren $2^{j/2}$ dabei so gewählt, dass auch die Funktionen $\phi_{j,k}(t)$ und $\psi_{j,k}(t)$ normiert sind: $\|\phi_{j,k}(t)\|_2 = 1 = \|\psi_{j,k}(t)\|_2$.

Zur Erinnerung an das im Einleitungskapitel bereits Beschriebene sind hier noch einmal in den Abbildungen 5.1, 5.2 und 5.3 grafische Darstellungen von einigen Haarfunktionen dargestellt.

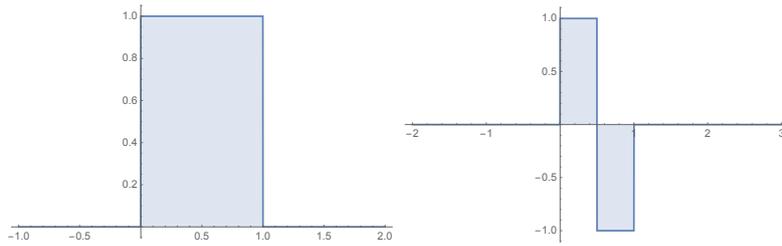


Abbildung 5.1: Haar-Skalierungsfunktion $\phi(t)$ und Haar-Waveletfunktion $\psi(t)$

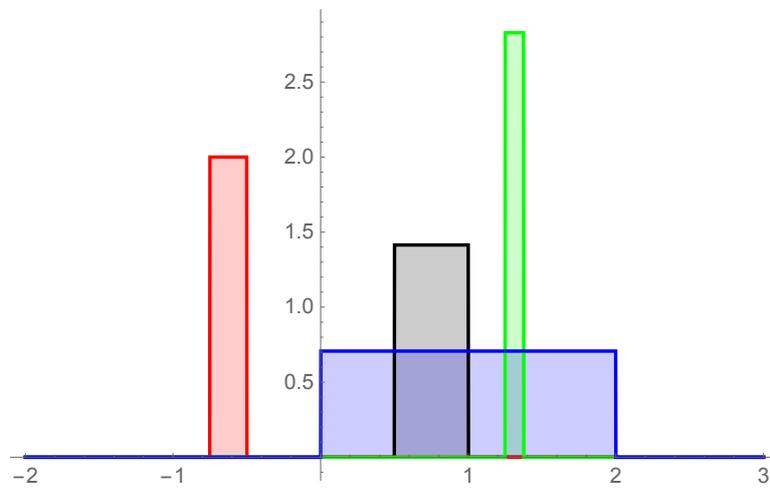


Abbildung 5.2: $\phi_{j,k}(t)$ für $(j, k) = (1, 1), (2, -3), (3, 10)$ und $(-1, 0)$

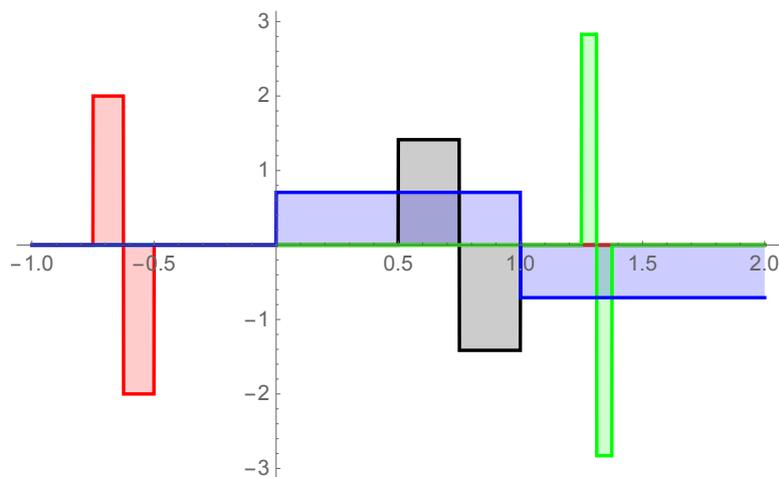


Abbildung 5.3: $\psi_{j,k}(t)$ für $(j, k) = (1, 1), (2, -3), (3, 10)$ und $(-1, 0)$

Folgende Systeme von Haar-Funktionen sind von Interesse (für $j, J \in \mathbb{Z}$):

Definition 5.3 (Familien von Haar-Funktionen).

$$\Phi = \{\phi_{j,k}\}_{j,k \in \mathbb{Z}} : \text{Haar-Skalierungsfunktionen}$$

$$\Phi_j = \{\phi_{j,k}\}_{k \in \mathbb{Z}} : \text{Haar-Skalierungsfunktionen der Stufe } j \quad (\text{Auflösung} = 2^{-j})$$

$$\Psi_j = \{\psi_{j,k}\}_{k \in \mathbb{Z}} : \text{Haar-Waveletfunktionen der Stufe } j \quad (\text{Auflösung} = 2^{-j})$$

$$\mathcal{H}_J = \Phi_J \cup \bigcup_{j \geq J} \Psi_j : \text{Haar-Funktionen der Stufe } \geq J \quad (\text{Auflösung} \leq 2^{-J})$$

$$\Psi = \mathcal{H} = \{\psi_{j,k}\}_{j,k \in \mathbb{Z}} : \text{Haar-Waveletfunktionen}$$

Definition 5.4 (Approximations- und Detailräume).

Für $j \in \mathbb{N}$ bezeichnet

$V_j = \overline{\text{span}} \Phi_j = \overline{\text{span}} \{\phi_{j,k}(t)\}_{k \in \mathbb{Z}}$ den Haar-Approximationsraum,
 – $W_j = \overline{\text{span}} \Psi_j = \overline{\text{span}} \{\psi_{j,k}(t)\}_{k \in \mathbb{Z}}$ den Haar-Detailraum
 der jeweiligen Stufe j , wobei sich $\overline{\text{span}}$ auf den Abschluss in $\mathcal{L}^2(\mathbb{R})$ bezieht.

Der Raum V_j besteht aus allen \mathcal{L}^2 -Treppenfunktionen mit Stufenbreite 2^{-j} , deren Stufen über den dyadischen Intervallen $I_{j,k}$ liegen. Deswegen sieht man direkt, dass

$$V_j \subset V_{j+1} \quad (j \in \mathbb{N})$$

gilt und deshalb bilden die Approximationsräume $\{V_j\}_{j \in \mathbb{N}}$ eine genestete Familie von Unterräumen von $\mathcal{L}^2(\mathbb{R})$:

$$\dots \subset V_{j-1} \subset V_j \subset V_{j+1} \subset \dots \subset \mathcal{L}^2(\mathbb{R}).$$

Ebenso offensichtlich ist jede Basisfunktion $\psi_{j,k}$ eine Treppenfunktion der Stufenbreite 2^{-j-1} , gehört also zu V_{j+1} und deshalb gilt

$$W_j \subset V_{j+1} \quad (j \in \mathbb{N}).$$

Die Haarfunktionen $\phi_{j,k}(t)$ und $\psi_{j,k}(t)$ sind gut lokalisiert im Zeit/Ortsbereich: sie sind ja nur auf dem Intervall $I_{j,k}$ von Null verschieden. Die Lokalisierung im Frequenzbereich erhält man mittels Fouriertransformation:

Lemma 5.1 — **Fouriertransformierte.**

1. Die Fouriertransformierte der Haar-Skalierungsfunktion $\phi(t)$ ist

$$\widehat{\phi}(s) = \frac{\sin(\pi s)}{\pi s} e^{-i\pi s}.$$

2. Die Fouriertransformierte der Haar-Waveletfunktion $\psi(t)$ ist

$$\widehat{\psi}(s) = \frac{2i}{\pi s} \sin^2(\pi s/2) e^{-i\pi s}.$$

Die Fouriertransformierten der $\phi_{j,k}(t)$ und $\psi_{j,k}(t)$ ergeben sich aus dem im vorigen Kapitel behandelten Beziehungen zwischen Dilatation, Translation und Fouriertransformation.

Die Absolutbeträge der komplexwertigen Funktionen $\widehat{\phi}(s)$ und $\widehat{\psi}(s)$ sind in Abbildung 5.4 für $-5 \leq s \leq 5$ geplottet. Die Funktion $|\widehat{\psi}(s)|$ hat ihr erstes Maximum bei $s_0 \approx 0.7420192\dots$ und ihre erste positive Nullstelle bei 2. Sie nimmt wie $1/s$ ab. Man kann also sagen, dass sie im Frequenzbereich gut bei s_0 lokalisiert ist. $\psi(t)$ selbst ist (im Ortsbereich) auch gut lokalisiert. Im Gegensatz zur Fourier-Situation hat man also simultan eine einigermaßen gute Lokalisierung sowohl im Orts- als auch im Frequenzbereich.

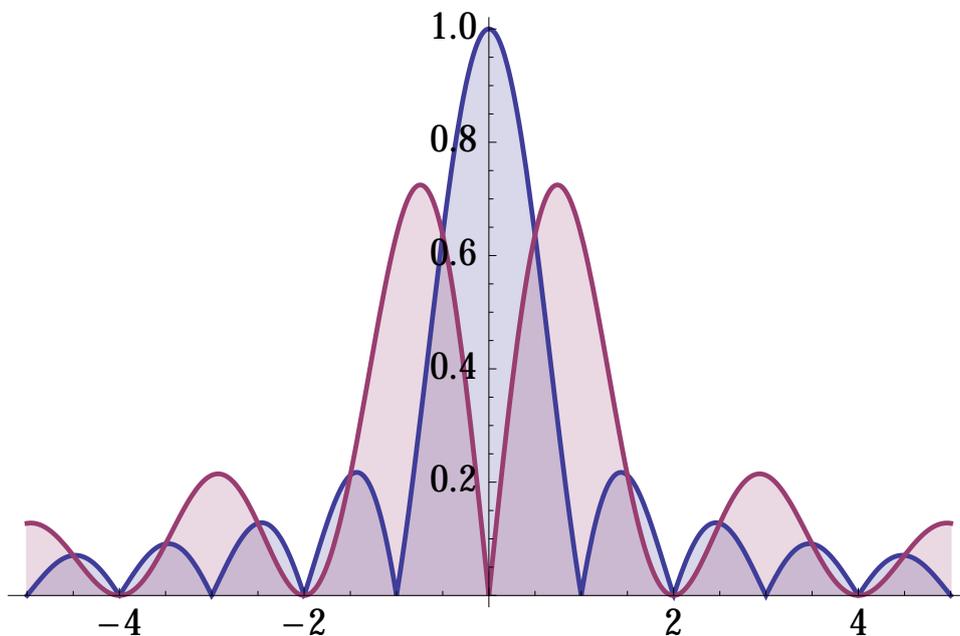


Abbildung 5.4: Absolutbetrag von $\widehat{\phi}(s)$ (blau) und $\widehat{\psi}(s)$ (rot)

Lemma 5.2 — Normierung.

Für alle $j, k \in \mathbb{Z}$ gelten die Normierungen:

$$\int_{\mathbb{R}} \phi_{j,k} = 2^{-j/2} \quad \text{und} \quad \|\phi_{j,k}\|_2^2 = \int_{\mathbb{R}} |\phi_{j,k}|^2 = 1,$$

$$\int_{\mathbb{R}} \psi_{j,k} = 0 \quad \text{und} \quad \|\psi_{j,k}\|_2^2 = \int_{\mathbb{R}} |\psi_{j,k}|^2 = 1.$$

Die folgenden Orthogonalitätsbeziehungen zwischen den $\phi_{j,k}$ und $\psi_{j',k'}$ ergeben sich unmittelbar aus der Definition.

Satz 5.1 — Orthogonalität.

Für alle $i, j, k, \ell \in \mathbb{Z}$ gelten die Orthogonalitätsbeziehungen:

$$\begin{aligned}\langle \phi_{j,k} | \phi_{j,\ell} \rangle &= \int_{\mathbb{R}} \phi_{j,k} \phi_{j,\ell} = \delta_{k,\ell}, \\ \langle \psi_{i,k} | \psi_{j,\ell} \rangle &= \int_{\mathbb{R}} \psi_{i,k} \psi_{j,\ell} = \delta_{i,j} \delta_{k,\ell}, \\ \langle \phi_{i,k} | \psi_{j,\ell} \rangle &= \int_{\mathbb{R}} \phi_{i,k} \psi_{j,\ell} = 0 \quad \text{falls } j \geq i.\end{aligned}$$

Daraus erhält man Orthogonalitätsaussagen über verschiedene Familien von Haar-Funktionen.

Theorem 5.1 — Orthonormale Systeme.

Die folgenden Familien bilden orthonormale Systeme in $\mathcal{L}^2(\mathbb{R})$:

1. Das Haar-Skalierungs-System Φ_j einer festen Stufe j ($j \in \mathbb{Z}$);
2. Das Haar-Wavelet-System Ψ ;
3. Das Haar-System \mathcal{H}_J für festes $J \in \mathbb{Z}$.

Die fundamentale Beziehung zwischen Haar-Skalierungsfunktionen und Haar-Wavelet-Funktionen, die man auch direkt an der Definition dieser Funktionen ablesen kann, lautet:

$$\begin{aligned}\phi(t) &= \phi(2t) + \phi(2t - 1), && \text{Skalierungsgleichung} \\ \psi(t) &= \phi(2t) - \phi(2t - 1), && \text{Waveletgleichung}\end{aligned}$$

und das schreibt sich zusammengefasst in Matrixform als

$$\begin{bmatrix} \phi_{0,0}(t) \\ \psi_{0,0}(t) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \phi_{1,0}(t) \\ \phi_{1,1}(t) \end{bmatrix}.$$

Die Transformationsmatrix

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

wird üblicherweise als Hadamard-Matrix² bezeichnet. Diese Matrix ist orthogonal, d.h. $H^{-1} = H$. Es gilt also auch

$$\begin{bmatrix} \phi_{1,0}(t) \\ \phi_{1,1}(t) \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{0,0}(t) \\ \psi_{0,0}(t) \end{bmatrix}.$$

Per Dilation und Translation erhält man für alle $j, k \in \mathbb{Z}$:

$$\begin{bmatrix} \phi_{j,k}(t) \\ \psi_{j,k}(t) \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{j+1,2k}(t) \\ \phi_{j+1,2k+1}(t) \end{bmatrix}, \quad \begin{bmatrix} \phi_{j+1,2k}(t) \\ \phi_{j+1,2k+1}(t) \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{j,k}(t) \\ \psi_{j,k}(t) \end{bmatrix}.$$

² Dies ist der allereinfachste Fall einer Hadamard-Matrix. Generell bezeichnet man als Hadamard-Matrizen orthogonale $(n \times n)$ -Matrizen, bei denen alle Matrixkoeffizienten $\in \{\pm 1\}$ (bis auf den globalen Normierungsfaktor) sind. Diese Matrizen spielen in Anwendungen in Geometrie, Codierungstheorie, Quantencomputing, ... eine große Rolle. Sie sind nach dem französischen Mathematiker Jacques Salomon HADAMARD (1865–1963) benannt.

Wichtige Bemerkung:

Für jedes feste $j \in \mathbb{Z}$ hat der Approximationsraum $V_{j+1} = \overline{\text{span}} \Phi_{j+1}$ die beiden (orthogonalen!) Basen

$$\Phi_{j+1} = \{\phi_{j+1,k}\}_{k \in \mathbb{Z}} \quad \text{und} \quad \Phi_j \cup \Psi_j = \{\phi_{j,k}, \psi_{j,k}\}_{k \in \mathbb{Z}}.$$

Man kann (und sollte!) die eben aufgeführten Transformationsgleichungen als Beschreibungen einer orthogonalen, weil von der orthogonalen Matrix H vermittelten, Basistransformation zwischen diesen beiden Basen verstehen.

Eine unmittelbare Folge ist die Aussage:

Satz 5.2 — Orthogonale Zerlegung der Unterräume.

Für jedes $j \in \mathbb{Z}$ ist

$$V_{j+1} = V_j \oplus W_j,$$

Für jedes $j \geq k$ gilt also

$$V_{j+1} = V_k \oplus W_k \oplus W_{k+1} \oplus \cdots \oplus W_{j-1} \oplus W_j.$$

Die Beziehung $V_{j+1} = V_j \oplus W_j$ als “direkte Summe”, die auf sich die beiden Basen Φ_{j+1} und $\Phi_j \cup \Psi_{j+1}$ bezieht, ist so zu verstehen: Für jede Funktion f in V_{j+1} gibt es also eindeutig bestimmte (und zueinander orthogonale!) Funktionen $g \in V_j$ und $h \in W_j$ mit

$$V_{j+1} \ni f = g + h \in V_j \oplus W_j.$$

- Die Abbildung $f \mapsto (g, h)$ bezeichnet man als *Analyseabbildung*, da sie f in einen “niederfrequenten” Bestandteil g und einen “hochfrequenten” Bestandteil h zerlegt.
- Die Umkehrung $(g, h) \mapsto f$ bezeichnet man als *Syntheseabbildung*, da sie f aus den nieder- und hochfrequenten Teilen rekonstruiert.

Der Vektorraum V_{j+1} hat als Basis sowohl das System Φ_{j+1} , als auch für jedes $k \leq j$ das System

$$\Phi_k \cup \Psi_k \cup \Psi_{k+1} \cup \cdots \cup \Psi_{j-1} \cup \Psi_j.$$

Definition 5.5 (Skalierungs- und Waveletkoeffizienten).

Ist $f \in \mathcal{L}^2(\mathbb{R})$, so bezeichnet man die Skalarprodukte

$$\langle f | \phi_{j,k} \rangle = 2^{j/2} \int_{I_{j,k}} f(t) dt$$

als die Haar-Skalierungskoeffizienten und die Skalarprodukte

$$\langle f | \psi_{j,k} \rangle = 2^{j/2} \left(\int_{I_{j,k}^l} f(t) dt - \int_{I_{j,k}^r} f(t) dt \right) \quad (j, k \in \mathbb{Z})$$

als die Haar-Waveletkoeffizienten von f .

Die Koeffizienten $\langle f | \phi_{j,k} \rangle$ und $\langle f | \psi_{j,k} \rangle$ hängen also nur vom Verhalten von

f auf dem dyadischen Intervall $I_{j,k}$ ab, ganz im Gegensatz zur Situation bei den Fourierkoeffizienten!

Zur Interpretation:

- $\langle f | \phi_{j,k} \rangle$ ist eine Mittelbildung und wird auch als *Approximationskoeffizient* der Funktion f bezeichnet. $\langle f | \phi_{j,k} \rangle$ “glättet” und oder “vergrößert”. Zur Abkürzung wird

$$a_{j,k} = \langle f | \phi_{j,k} \rangle$$

geschrieben.

- $\langle f | \psi_{j,k} \rangle$ gibt die Schwankung des Mittels von f bezüglich linkem und rechtem Teilintervall wieder und wird auch als *Detailkoeffizient* der Funktion f bezeichnet. $\langle f | \psi_{j,k} \rangle$ hebt Unterschiede hervor. Zur Abkürzung wird

$$d_{j,k} = \langle f | \psi_{j,k} \rangle$$

geschrieben.

Aus der Basisbeziehung $V_{j+1} = V_j \oplus W_j$ folgt unmittelbar eine Rekursionsbeziehung für die Skalierungs- und Waveletkoeffizienten:

Folgerung 5.1 (Analyse und Synthese).

Für alle $j, k \in \mathbb{Z}$ gilt für die Skalierungs- und Waveletkoeffizienten

- die Analysebeziehung

$$\begin{bmatrix} a_{j,k} \\ d_{j,k} \end{bmatrix} = H \cdot \begin{bmatrix} a_{j+1,2k} \\ a_{j+1,2k+1} \end{bmatrix}$$

- die Synthesebeziehung

$$\begin{bmatrix} a_{j+1,2k} \\ a_{j+1,2k+1} \end{bmatrix} = H \cdot \begin{bmatrix} a_{j,k} \\ d_{j,k} \end{bmatrix},$$

oder äquivalent

$$a_{j+1,2k} \cdot \phi_{j+1,2k} + a_{j+1,2k+1} \cdot \phi_{j+1,2k+1} = a_{j,k} \cdot \phi_{j,k} + d_{j,k} \cdot \psi_{j,k}.$$

Die orthogonalen Projektionen P_j bzw. Q_j von Funktionen $f \in \mathcal{L}^2(\mathbb{R})$ auf die Unterräume V_j bzw. W_j schreiben sich als

$$P_j : \mathcal{L}^2(\mathbb{R}) \rightarrow V_j : f \mapsto \sum_{k \in \mathbb{Z}} \langle f | \phi_{j,k} \rangle \phi_{j,k},$$

$$Q_j : \mathcal{L}^2(\mathbb{R}) \rightarrow W_j : f \mapsto \sum_{k \in \mathbb{Z}} \langle f | \psi_{j,k} \rangle \psi_{j,k},$$

Wobei die unendlichen Summen als konvergente Reihen im \mathcal{L}^2 -Sinne zu lesen sind. Dies sind die optimalen Approximationen im \mathcal{L}^2 -Sinne von f in den Räumen V_j bzw. W_j .

Die in der vorigen Folgerung getroffene Feststellung besagt also

$$P_{j+1} = P_j + Q_j,$$

in voller Übereinstimmung mit Satz 5.2.

5.3 Haar-Systeme auf $[0, 1]$

Betrachtet man nur Funktionen auf einem endlichen Intervall I als Definitionsbereich, so haben die Räume V_j und W_j endliche Dimension, sind also abgeschlossen unter \mathcal{L}^2 -Limesbildung und die Summen in der Beschreibung von P_j und Q_j sind endlich. Damit ist die Situation (formal) etwas einfacher, als wenn ganz \mathbb{R} der Definitionsbereich ist.

Man kann sich natürlich auf den Fall $I = [0, 1]$ beschränken. Dann sind nur Indexpaare (j, k) mit $j \geq 0$ und $0 \leq k < 2^j$ relevant, die den in $[0, 1]$ enthaltenen dyadischen Intervallen entsprechen. Alle bisherigen Ausführungen lassen sich problemlos auf diesen Fall übertragen.

Definition 5.6 (Systeme der Haar-Funktionen auf $[0, 1]$).

Für $J \geq 0$ ist $\mathcal{H}_J = \{\phi_{j,k}\}_{0 \leq k < 2^j} \cup \{\psi_{j,k}\}_{j \geq J, 0 \leq k < 2^j}$ das System der Haar-Funktionen der Stufe J auf dem Intervall $[0, 1]$.

Theorem 5.2 — Vollständigkeit der Haar-Systeme.

Für jedes $J \geq 0$ ist \mathcal{H}_J ein vollständiges ONS, also eine Hilbert-Basis für $\mathcal{L}^2[0, 1]$.

Zum Beweis: Nur die Vollständigkeit ist noch zu zeigen. Dafür genügt der Nachweis, dass $\mathcal{C}^0[0, 1]$ in $\overline{\text{span}} \mathcal{H}_J$ enthalten ist, d.h., dass sich jede stetige Funktion auf $[0, 1]$ durch Elemente von $\text{span} \mathcal{H}_J$ in der \mathcal{L}^2 -Norm beliebig gut approximieren lässt.

Nun ist jede stetige Funktion auf einem kompakten Intervall gleichmäßig stetig, d.h. man kann ein solches f gleichmäßig durch Treppenfunktionen approximieren. Zu $\varepsilon > 0$ gibt es ein $j \geq J$ und ein $g_j \in V_j$ mit

$$\|f - g_j\|_\infty < \varepsilon.$$

Dann ist aber auch

$$\|f - g_j\|_2 = \left(\int_0^1 |f - g|^2 \right)^{1/2} \leq \|f - g_j\|_\infty < \varepsilon.$$

Nun muss man nur noch g_j im Sinne von Satz 5.2 darstellen:

$$g_j = g_J + h_J + h_{J+1} + \cdots + h_{j-1} \in \text{span} \mathcal{H}_J$$

mit $g_J \in V_J$ und $h_i \in W_i$ ($J \leq i < j$). □

5.4 Haar-Systeme auf \mathbb{R}

In dieser Situation sind die Approximations- und Detailräume $V_j = \overline{\text{span}} \{\phi_{j,k}\}_{k \in \mathbb{Z}}$ bzw. $W_j = \overline{\text{span}} \{\psi_{j,k}\}_{k \in \mathbb{Z}}$ nicht mehr endlich-dimensional und in den Projektionsformeln für diese Räume treten unendliche Summen auf

An den früheren Aussagen ändert sich eigentlich nichts, ausser dass jetzt auch unendliche Summen auftreten, die aber im \mathcal{L}^2 -Sinne konvergieren.

Für die Projektionen für *Approximation* P_j und *Detail* Q_j mit

$$P_j : \mathcal{L}^2(\mathbb{R}) \rightarrow V_j : f \mapsto \sum_{k \in \mathbb{Z}} \langle f | \phi_{j,k} \rangle \phi_{j,k},$$

$$Q_j : L_c^2(\mathbb{R}) \rightarrow W_j : f \mapsto \sum_{k \in \mathbb{Z}} \langle f | \psi_{j,k} \rangle \psi_{j,k},$$

gelten dann die erwarteten Eigenschaften:

Lemma 5.3 — Eigenschaften der Projektionen.

1. Die P_j und Q_j sind lineare Transformationen.
2. Die P_j und Q_j sind Projektionen, d.h. sie sind idempotent:
 $P_j^2 = P_j, Q_j^2 = Q_j$.
3. Für $k \geq j$ ist $P_k|_{V_j} = id$.
4. Für $k \neq j$ ist $Q_k|_{W_j} = 0$.
5. $\|P_j f\|_2 \leq \|f\|_2$ und $\|Q_j f\|_2 \leq \|f\|_2$.
6. $Q_j = P_{j+1} - P_j$.
7. Für $f \in C_c^0(\mathbb{R})$ gilt $P_j f \rightarrow_{j \rightarrow \infty} f$ und $P_j f \rightarrow_{j \rightarrow -\infty} 0$ (in der \mathcal{L}^2 -Konvergenz).

Insbesondere beinhaltet der letzte Punkt die folgenden Aussagen über die Folge der genesteten Vektorräume

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots$$

- $V_\infty := \lim_{j \rightarrow \infty} V_j = \bigcup_{j \in \mathbb{Z}} V_j = \mathcal{L}^2(\mathbb{R})$,
- $V_\infty := \lim_{j \rightarrow -\infty} V_j = \bigcap_{j \in \mathbb{Z}} V_j = \{0\}$.

Dieses Schema mit seinen Eigenschaften wird im Kapitel 9 über Multiresolution gründlich betrachtet.

Theorem 5.3 — Vollständigkeit der Haar-Systeme.

Für jedes $J \in \mathbb{Z}$ ist \mathcal{H}_J ein vollständiges ONS, also eine Hilbert-Basis für $\mathcal{L}^2(\mathbb{R})$.

Zum Beweis:

Es ist $C_c^0(\mathbb{R}) \subseteq \overline{\text{span}} \mathcal{H}_J$ zu zeigen. Zu $f \in C_c^0(\mathbb{R})$ und $\varepsilon > 0$ gibt es sicher ein $N > J$ mit $\|P_N f - f\|_2 < \varepsilon$ (Argument für *kompaktes* (=endliches) Intervall, da f kompakten Träger hat). Wegen

$$\sum_{j=J}^{N-1} Q_j f = P_N f - P_J f$$

ist

$$P_N f = \sum_{j=J}^{N-1} \sum_k \langle f | \psi_{j,k} \rangle \psi_{j,k} + \sum_k \langle f | \phi_{J,k} \rangle \phi_{J,k}$$

eine endliche (!) Summe aus $\text{span} \mathcal{H}_J$. □

Theorem 5.4 — Vollständigkeit der Haar-Waveletfunktionen.

Das System $\mathcal{H} = \{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ ist ein vollständiges ONS, also eine Hilbert-Basis für $\mathcal{L}^2(\mathbb{R})$.

Zum Beweis:

Es ist $\mathcal{C}_c^0(\mathbb{R}) \subseteq \overline{\text{span}} \mathcal{H}$ zu zeigen.

Für $f \in \mathcal{C}_c^0(\mathbb{R})$ und $J \in \mathbb{N}$ schließt man mit

$$\sum_{j=-J}^J Q_j f = P_J f - P_{-J} f$$

auf

$$\|f - \sum_{j=-J}^J Q_j f\|_2 = \|f - P_J f + P_{-J} f\|_2 \leq \|f - P_J f\|_2 + \|P_{-J} f\|_2 \xrightarrow{J \rightarrow \infty} 0.$$

Es ist aber

$$\sum_{j=-J}^J Q_j f = \sum_{j=-J}^J \sum_k \langle f | \psi_{j,k} \rangle \psi_{j,k} \in \text{span} \mathcal{H},$$

wiederum deshalb, weil die Summe endlich (!) ist. □

5.5 Die Diskrete Haar-Transformation (DHT)

Für eine (reelle oder komplexe) Matrix A bezeichnet A^\dagger die *adjungierte* Matrix zur Matrix A , also die Matrix, die aus A mittels Transposition und komplexer Konjugation der Matrixkoeffizienten entsteht. Falls A eine reelle Matrix ist, ist A^\dagger einfach die transponierte Matrix. Das Zeichen \otimes steht für das Tensorprodukt (Kroneckerprodukt) von Matrizen.

Definition 5.7 (Tensorprodukt von Matrizen).

Sind $A = [a_{i,j}]_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}}$ und $B = [b_{k,\ell}]_{\substack{1 \leq k \leq p \\ 1 \leq \ell \leq q}}$ Matrizen, so ist die $(m \cdot p) \times (n \cdot q)$ -Matrix $A \otimes B$ definiert durch

$$A \otimes B = \begin{bmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{2,2}B & \dots & a_{2,n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1}B & a_{m,2}B & \dots & a_{m,n}B \end{bmatrix}$$

Wie im Abschnitt 5.2 bezeichnet auch in diesem Abschnitt H die orthogonale Hadamard-Matrix $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. I_n ist die $(n \times n)$ -Einheitsmatrix 0_n ist die $(n \times n)$ -Nullmatrix.

ist die Matrix einer einstufigen diskreten Haar-Transformation von Signalen der Länge $2n$.

Lemma 5.4 Für die $(2n \times n)$ -Matrizen

$$A_n = I_n \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}, \quad D_n = I_n \otimes \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & \\ & -1 \end{bmatrix},$$

gelten folgende Beziehungen:

$$\begin{aligned} A_n^\dagger \cdot A_n &= I_n, & A_n \cdot A_n^\dagger &= \frac{1}{2} I_n \otimes \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \\ D_n^\dagger \cdot D_n &= I_n, & D_n \cdot D_n^\dagger &= \frac{1}{2} I_n \otimes \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \\ A_n^\dagger \cdot D_n &= 0_n, & A_n \cdot D_n^\dagger &= \frac{1}{2} I_n \otimes \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}, \\ D_n^\dagger \cdot A_n &= 0_n, & D_n \cdot A_n^\dagger &= \frac{1}{2} I_n \otimes \begin{bmatrix} 1 & 1 \\ -1 & -1 \end{bmatrix}. \end{aligned}$$

Insbesondere ist $[A_n \ D_n]$ eine orthogonale Matrix, d.h.

$$[A_n \ D_n] \begin{bmatrix} A_n^\dagger \\ D_n^\dagger \end{bmatrix} = I_{2n} = \begin{bmatrix} A_n^\dagger \\ D_n^\dagger \end{bmatrix} [A_n \ D_n].$$

Die letzte Behauptung ergibt sich aus den vorigen:

$$\begin{aligned} [A_n \ D_n] \cdot [A_n \ D_n]^\dagger &= [A_n \ D_n] \begin{bmatrix} A_n^\dagger \\ D_n^\dagger \end{bmatrix} = A_n \cdot A_n^\dagger + D_n \cdot D_n^\dagger \\ &= \frac{1}{2} I_n \otimes \left(\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \right) = \frac{1}{2} I_n \otimes \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = I_{2n} \end{aligned}$$

$$[A_n \ D_n]^\dagger \cdot [A_n \ D_n] = \begin{bmatrix} A_n^\dagger \\ D_n^\dagger \end{bmatrix} [A_n \ D_n] = \begin{bmatrix} A_n^\dagger \cdot A_n & A_n^\dagger \cdot D_n \\ D_n^\dagger \cdot A_n & D_n^\dagger \cdot D_n \end{bmatrix} = \begin{bmatrix} I_n & 0_n \\ 0_n & I_n \end{bmatrix} = I_{2n}$$

Konvention: In Analogie zur Notation bei Matrizen gibt nun der Index n bei einem Vektor \mathbf{a}_n die Länge des Vektors an, also $\mathbf{a}_n = (a_1, a_2, \dots, a_n)$. Somit: \mathbf{a}_n und a_n bezeichnen verschiedene Dinge!

Für \mathbf{a}_{2n} einen (Zeilen-)Vektor der Länge $2n$ ist

$$\mathbf{a}_{2n} \mapsto \mathbf{a}_{2n} \cdot [A_n \ D_n] = [\mathbf{a}'_n \ \mathbf{d}'_n]$$

eine lineare Transformation des Vektorraumes \mathbb{C}^{2n} . Es gilt also

$$\mathbf{a}'_n = \mathbf{a}_{2n} \cdot A_n, \quad \mathbf{d}'_n = \mathbf{a}_{2n} \cdot D_n$$

Da dies eine orthogonale Transformation ist, kann man diese Beziehung ganz einfach umkehren:

$$\mathbf{a}_{2n} = \begin{bmatrix} \mathbf{a}'_n & \mathbf{d}'_n \end{bmatrix} \cdot \begin{bmatrix} A_n & D_n \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{a}'_n & \mathbf{d}'_n \end{bmatrix} \cdot \begin{bmatrix} A_n^\dagger \\ D_n^\dagger \end{bmatrix} = \mathbf{a}'_n \cdot A_n^\dagger + \mathbf{d}'_n \cdot D_n^\dagger$$

Mit dieser Notation kann die *mehrstufige* diskrete Haar-Transformation induktiv definiert werden:

Definition 5.9 (Mehrstufige Haar-Transformation).

Für $k \geq 1$ und $n \geq 1$ ist die k -stufige Haar-Transformation

$$\text{DHT}_k : \mathbb{C}^{2^k n} \rightarrow \mathbb{C}^{2^k n}$$

definiert.

– Für $k = 1$:

$$\text{DHT}_1 : \mathbb{C}^{2n} \rightarrow \mathbb{C}^{2n} : \mathbf{a}_{2n} \mapsto \begin{bmatrix} \mathbf{a}'_n & \mathbf{d}'_n \end{bmatrix} = \mathbf{a}_{2n} \cdot \begin{bmatrix} A_n & D_n \end{bmatrix}.$$

– Ist $\text{DHT}_k : \mathbb{C}^{2^k n} \rightarrow \mathbb{C}^{2^k n}$ schon definiert, so ist

$$\text{DHT}_{k+1} : \mathbb{C}^{2^{k+1}n} \rightarrow \mathbb{C}^{2^{k+1}n} : \mathbf{a}_{2^{k+1}n} \mapsto \begin{bmatrix} \text{DHT}_k(\mathbf{a}'_{2^k n}) & \mathbf{d}'_{2^k n} \end{bmatrix}$$

wobei $\text{DHT}_1(\mathbf{a}_{2^k n}) = \begin{bmatrix} \mathbf{a}'_{2^k n} & \mathbf{d}'_{2^k n} \end{bmatrix}$.

Als Kompositionen von orthogonalen Transformationen sind auch die DHT_k orthogonale Transformationen.

Suggestiv geschrieben gilt also

$$\text{DHT}_k(\mathbf{a}_{2^k n}) = \begin{bmatrix} \mathbf{a}_n^{(k)} & \mathbf{d}_n^{(k)} & \mathbf{d}_{2n}^{(k-1)} & \mathbf{d}_{4n}^{(k-2)} & \dots & \mathbf{d}_{2^{k-2}n}'' & \mathbf{d}'_{2^{k-1}n} \end{bmatrix},$$

Die Interpretation dieser Beziehung im Lichte der Vektorraumzerlegung

$$\mathcal{V}_{J+k} = \mathcal{V}_J \oplus \mathcal{W}_J \oplus \mathcal{W}_{J+1} \oplus \mathcal{W}_{J+2} \oplus \dots \oplus \mathcal{W}_{J+k-1}$$

ist die:

- sind die $\mathbf{a}_{2^k n}$ die Koeffizienten einer Funktion f bezüglich der Basis $\Phi_{J+k} = \{\phi_{J+k,m}\}$, i.e. $a_{J+k,m} = \langle f | \phi_{J+k,m} \rangle$
- so sind die Koeffizienten des Vektors $\text{DHT}_k(\mathbf{a}_{2^k n})$ die Koeffizienten von f bezüglich der Basen

$$\begin{array}{ll} - \Phi_J = \{\phi_{J,m}\} \text{ in } \mathcal{V}_J, & a_{J+k,m}^{(k)} = \langle f | \phi_{J+k,m} \rangle \\ - \Psi_J = \{\psi_{J,m}\} \text{ in } \mathcal{W}_J, & d_{J+k,m}^{(k)} = \langle f | \psi_{J,J+k} \rangle \\ - \Psi_{J+k-1} = \{\psi_{J+k-1,m}\} \text{ in } \mathcal{W}_{J+k-1}, & d_{J+k-1,m}^{(k-1)} = \langle f | \phi_{J+k-1,m} \rangle \\ - \Psi_{J+k-2} = \{\psi_{J+k-2,m}\} \text{ in } \mathcal{W}_{J+k-2}, & a_{J+k-2,m}^{(k-2)} = \langle f | \phi_{J+k-2,m} \rangle \\ - \dots & \dots \\ - \Psi_{J+1} = \{\psi_{J+1,m}\} \text{ in } \mathcal{W}_{J+1}, & d'_{J+1,m} = \langle f | \phi_{J+1,m} \rangle. \end{array}$$

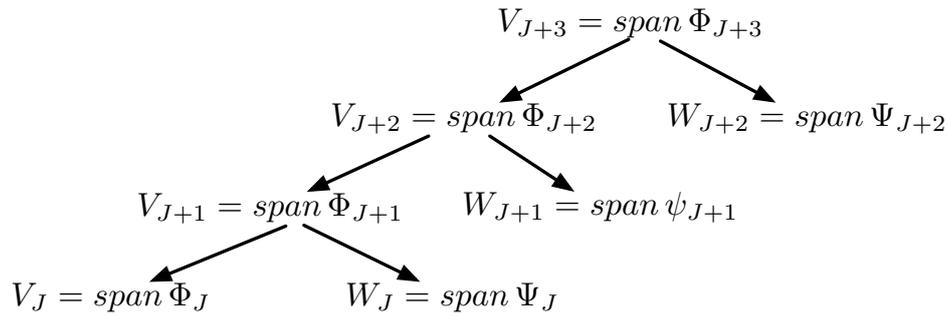


Abbildung 5.5: Schema einer 3-stufigen Haar-Transformation (Analyse)

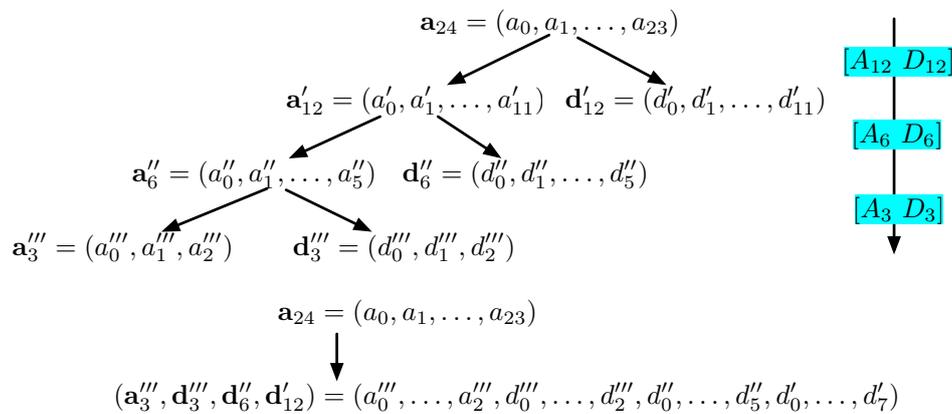


Abbildung 5.6: Schema einer 3-stufigen Haar-Transformation (Analyse) auf einem Vektor der Länge 24

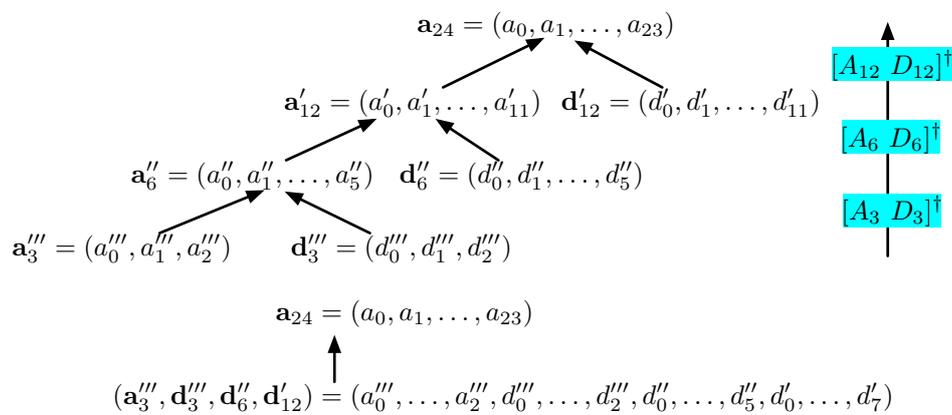


Abbildung 5.7: Schema einer 3-stufigen Haar-Transformation (Synthese) auf einem Vektor der Länge 24

Damit ergeben sich in Analogie zur Haar-Wavelettransformation die beiden *Analyse-Transformationen*

- Tiefpassfilterung mit anschliessendem downsampling:

$$\mathcal{A} \circ \downarrow_2 : \mathbf{a} \mapsto \mathbf{a} \mathcal{A} \downarrow_2 = \left(\frac{a[2n] + a[2n+1]}{\sqrt{2}} \right)_{n \in \mathbb{Z}} = \mathbf{a}'.$$

- Hochpassfilterung mit anschliessendem downsampling:

$$\mathcal{D} \circ \downarrow_2 : \mathbf{a} \mapsto \mathbf{a} \mathcal{D} \downarrow_2 = \left(\frac{a[2n] - a[2n+1]}{\sqrt{2}} \right)_{n \in \mathbb{Z}} = \mathbf{d}'.$$

Schematisch stellt man die Situation in einem Diagramm³ dar, siehe Abbildung 5.8.

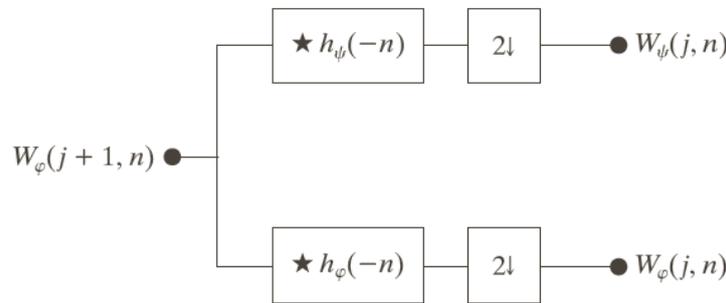


Abbildung 5.8: Haar-Analyse (1 Stufe)

Die *Synthese-Transformationen* zur Rekonstruktion von \mathbf{a} verwenden upsampling und die adjungierten Matrizen:

$$\mathbf{a}' \uparrow_2 \mathcal{A}^\dagger + \mathbf{d}' \uparrow_2 \mathcal{D}^\dagger = \mathbf{a}.$$

Um das einzusehen, muss man sich nur klarmachen, was die Operationen $\mathbf{a} \mapsto \mathbf{a} \uparrow_2 \mathcal{A}^\dagger$ bzw. $\mathbf{a} \mapsto \mathbf{a} \uparrow_2 \mathcal{D}^\dagger$ auf einer beliebigen Folge $\mathbf{a} = (a[n])_{n \in \mathbb{Z}}$ bewirken:

$$\begin{aligned} \mathbf{a} \uparrow_2 \mathcal{A}^\dagger &= \left(\frac{a[\lfloor n/2 \rfloor]}{\sqrt{2}} \right)_{n \in \mathbb{Z}}, \\ \mathbf{a} \uparrow_2 \mathcal{D}^\dagger &= \left(\frac{(-1)^n a[\lfloor n/2 \rfloor]}{\sqrt{2}} \right)_{n \in \mathbb{Z}}. \end{aligned}$$

Somit ist

$$\begin{aligned} \mathbf{a}' \uparrow_2 \mathcal{A}^\dagger + \mathbf{d}' \uparrow_2 \mathcal{D}^\dagger &= \left(\frac{a[\lfloor n/2 \rfloor]'}{\sqrt{2}} \right)_{n \in \mathbb{Z}} + \left(\frac{(-1)^n d[\lfloor n/2 \rfloor]'}{\sqrt{2}} \right)_{n \in \mathbb{Z}} \\ &= \left(\frac{a[\lfloor n/2 \rfloor]' + (-1)^n d[\lfloor n/2 \rfloor]'}{\sqrt{2}} \right)_{n \in \mathbb{Z}}. \end{aligned}$$

³ Wichtiger Hinweis: Ein großer Teil der diesem Kapitel gezeigten grafischen Darstellungen entstammen dem Buch *Digital Image Processing (3.ed.)* von R.C. Gonzalez und R.E. Woods, Pearson/Prentice Hall, 2008.

Dabei gilt für gerades n :

$$\frac{1}{\sqrt{2}} (a[\lfloor n/2 \rfloor] + (-1)^n d[\lfloor n/2 \rfloor]) = \frac{1}{\sqrt{2}} \left(\frac{a[n] + a[n+1]}{\sqrt{2}} + \frac{a[n] - a[n+1]}{\sqrt{2}} \right) = a[n],$$

und für ungerades n :

$$\frac{1}{\sqrt{2}} (a[\lfloor n/2 \rfloor] + (-1)^n d[\lfloor n/2 \rfloor]) = \frac{1}{\sqrt{2}} \left(\frac{a[n-1] + a[n]}{\sqrt{2}} - \frac{a[n-1] - a[n]}{\sqrt{2}} \right) = a[n].$$

Damit ist die Rekonstruktionseigenschaft bewiesen. Zusammengefasst kann man dies so ausdrücken:

$$\mathcal{A} \downarrow_2 \uparrow_2 \mathcal{A}^\dagger + \mathcal{D} \downarrow_2 \uparrow_2 \mathcal{D}^\dagger = Id.$$

Berücksichtigt man noch die Tatsache, dass Downsampling und Upsampling zueinander adjungierte Operationen sind, kann man das noch konzierer schreiben:

für $A = \mathcal{A} \downarrow_2, D = \mathcal{D} \downarrow_2$ gilt

$$A A^\dagger + D D^\dagger = Id.$$

Es ist leicht nachzuprüfen, dass zwischen diesen Transformationen auch die folgenden Beziehungen gelten:

$$A^\dagger A = Id, \quad D^\dagger D = Id, \quad A^\dagger D = 0 = D^\dagger A.$$

Schematisch stellt man die Situation in einem Diagramm dar:

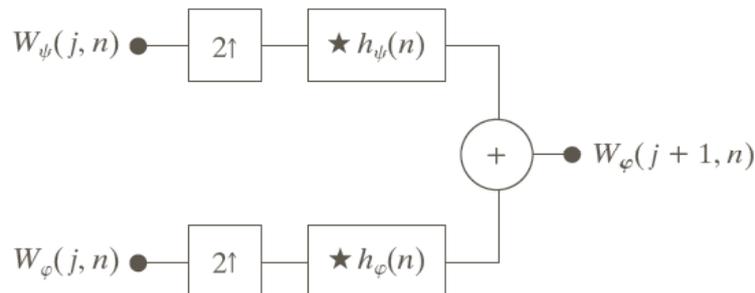


Abbildung 5.9: Haar-Synthese (1 Stufe)

5.7 Zweistufige Haar-Filterbank

Eine zweistufige Haar-*Analyse*-Transformation besteht darin, auf den tiefpassgefilterten Output einer Haar-Transformation nochmals eine Haar-Transformation anzuwenden. Der hochpassgefilterte Anteil der erste Transformation wird dagegen unverändert übernommen. Die Abbildung 5.10 zeigt das schematisch, zusammen mit einer Aufteilung in "Frequenzbänder".

Die Umkehrung dieser zweistufigen Operation, also die zweistufige *Synthese* liegt auf der Hand: man kehrt jede der einstufigen Teiltransformationen wie vorher beschrieben um. Die Abbildung 5.11 zeigt das schematisch.

Numerische Beispiele finden sich in den Abbildungen 5.12 (für die Analyse) und 5.13 (für die Synthese).

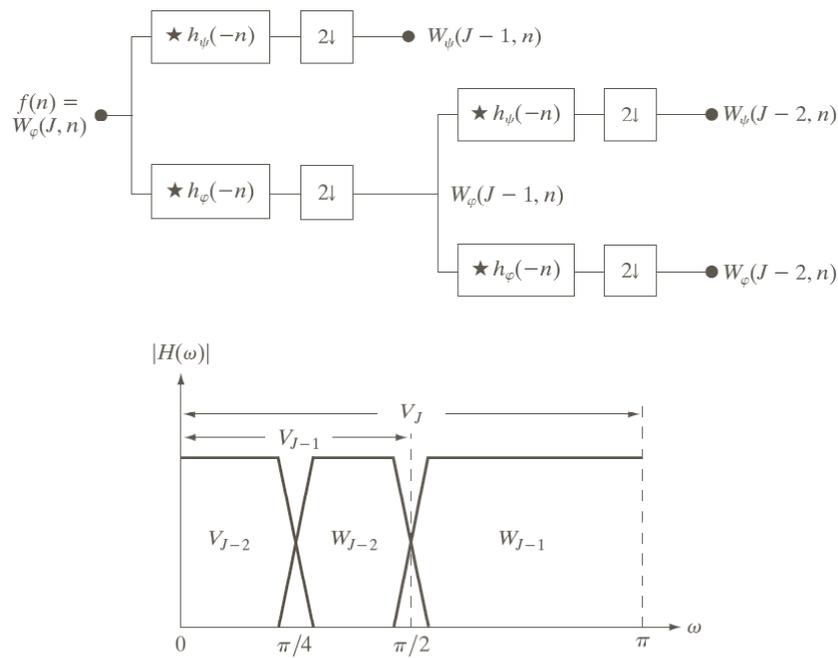


Abbildung 5.10: Haar-Analyse (2 Stufen)

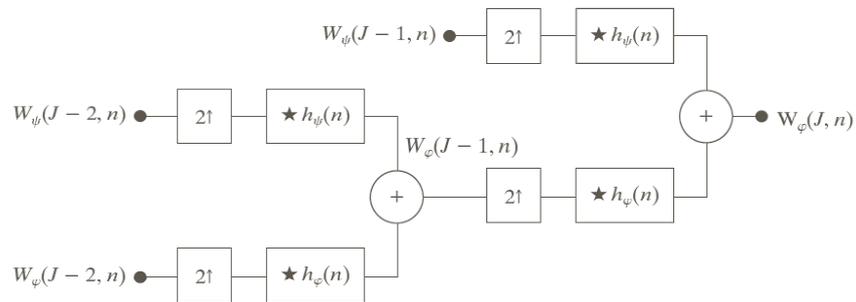


Abbildung 5.11: Haar-Synthese (2 Stufen)

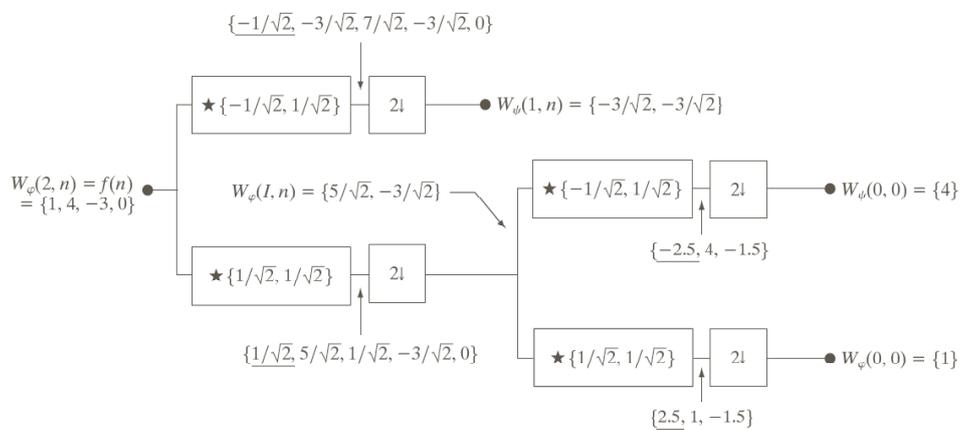


Abbildung 5.12: Haar-Analyse (2 Stufen) – Beispiel

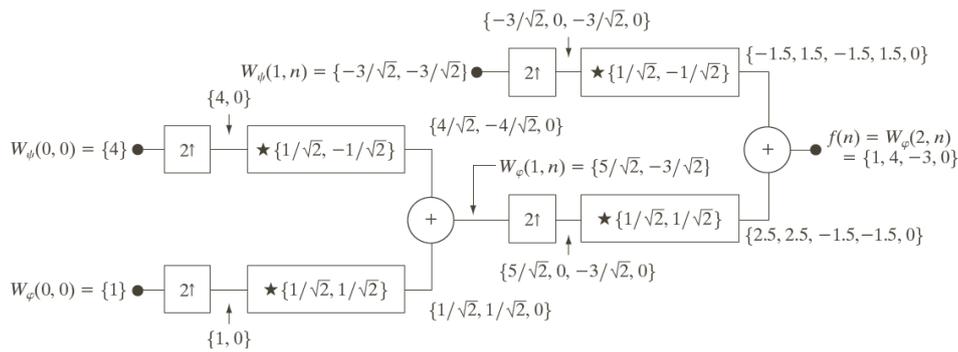


Abbildung 5.13: Haar-Synthese (2 Stufen) – Beispiel

5.8 Sichtweise der z -Transformation

Stellt man eine zweiseitig unendliche Folge reeller oder komplexer Zahlen $\mathbf{a} = (a[n])_{n \in \mathbb{Z}}$ vermöge der Potenzreihe

$$a(z) = \sum_{n \in \mathbb{Z}} a[n] z^n$$

dar, so lassen sich die Transformationen des vorigen Abschnitts bequem analytisch beschreiben. Dafür seien noch

$$h_\phi(z) = \sum_n h_\phi[n] z^n = \frac{1}{\sqrt{2}}(1+z) \quad h_\psi(z) = \sum_n h_\psi[n] z^n = \frac{1}{\sqrt{2}}(1-z)$$

die entsprechenden Potenzreihen (Polynome) der Filter h_ϕ und h_ψ . Dann gilt

$$\begin{aligned} \mathcal{A} : \mathbf{a} &\mapsto \left(\frac{a[n] + a[n+1]}{\sqrt{2}} \right)_{n \in \mathbb{Z}} && \simeq a(z) \cdot h_\phi(z^{-1}) \\ \mathcal{D} : \mathbf{a} &\mapsto \left(\frac{a[n] - a[n+1]}{\sqrt{2}} \right)_{n \in \mathbb{Z}} && \simeq a(z) \cdot h_\psi(z^{-1}) \\ \mathcal{A}^\dagger : \mathbf{a} &\mapsto \left(\frac{a[n] + a[n-1]}{\sqrt{2}} \right)_{n \in \mathbb{Z}} && \simeq a(z) \cdot h_\phi(z) \\ \mathcal{D}^\dagger : \mathbf{a} &\mapsto \left(\frac{a[n] - a[n-1]}{\sqrt{2}} \right)_{n \in \mathbb{Z}} && \simeq a(z) \cdot h_\psi(z) \\ \downarrow_2 : \mathbf{a} &\mapsto (a[2n])_{n \in \mathbb{Z}} && \simeq \frac{a(z) + a(-z)}{2} \Big|_{z^2 \rightarrow z} \\ \uparrow_2 : \mathbf{a} &\mapsto (a[n/2] \mathbf{1}_{\text{even}}(n))_{n \in \mathbb{Z}} && \simeq a(z^2) \end{aligned}$$

Nun kann man die Arbeitsweise der Haar-Filterbank analytisch nachvollziehen:

- **Analyse:** Aus einem Eingangssignal $a(z)$ wird

– im Tiefpasszweig:

* mittels \mathcal{A} :

$$a(z) \cdot h_\phi(z^{-1})$$

* mittels \downarrow_2 :

$$a'(z) = \left. \frac{a(z)h_\phi(z^{-1}) + a(-z)h_\phi(-z^{-1})}{2} \right|_{z^2 \rightarrow z}$$

– im Hochpasszweig:

* mittels \mathcal{D} :

$$a(z) \cdot h_\psi(z^{-1})$$

* mittels \downarrow_2 :

$$d'(z) = \left. \frac{a(z)h_\psi(z^{-1}) + a(-z)h_\psi(-z^{-1})}{2} \right|_{z^2 \rightarrow z}$$

- **Synthese:**

– Aus dem Signal $a'(z)$ im Tiefpasszweig wird

* mittels \uparrow_2 :

$$\frac{a(z)h_\phi(z^{-1}) + a(-z)h_\phi(-z^{-1})}{2}$$

* mittels \mathcal{A} :

$$\frac{a(z)h_\phi(z^{-1}) + a(-z)h_\phi(-z^{-1})}{2} \cdot h_\phi(z)$$

– Aus dem Signal $d'(z)$ im Hochpasszweig wird

* mittels \uparrow_2 :

$$\frac{a(z)h_\psi(z^{-1}) + a(-z)h_\psi(-z^{-1})}{2}$$

* mittels \mathcal{D} :

$$\frac{a(z)h_\psi(z^{-1}) + a(-z)h_\psi(-z^{-1})}{2} \cdot h_\psi(z)$$

und damit zusammen

$$a(z) \left[\frac{h_\phi(z)h_\phi(z^{-1}) + h_\psi(z)h_\psi(z^{-1})}{2} \right] + a(-z) \left[\frac{h_\phi(z)h_\phi(-z^{-1}) + h_\psi(z)h_\psi(-z^{-1})}{2} \right]$$

Nun gilt aber

$$\frac{h_\phi(z)h_\phi(z^{-1}) + h_\psi(z)h_\psi(z^{-1})}{2} = \frac{(1+z)(1+z^{-1}) - (1-z)(1-z^{-1})}{4} = 1$$

und

$$\frac{h_\phi(z)h_\phi(z^{-1}) + h_\psi(z)h_\psi(z^{-1})}{2} = \frac{(1+z)(1-z^{-1}) - (1-z)(1+z^{-1})}{4} = 0$$

und man erhält $a(z)$ zurück.

5.9 2D-Haar-Wavelets

Zur Erinnerung: Die eindimensionalen Haar-Funktionen sind die *Skalierungsfunktion*

$$\mathbf{1}_{[0,1)}(t)$$

und die *Waveletfunktion*

$$\psi(t) = \mathbf{1}_{[0,1/2)}(t) - \mathbf{1}_{[1/2,1)}(t)$$

Daraus abgeleitet mittels Dilation und Translation die Skalierungs- und Waveletfunktionen für die dyadischen Intervalle $I_{j,k}$ ($j, k \in \mathbb{Z}$):

$$\begin{aligned}\phi_{j,k}(t) &= 2^{j/2} \mathbf{1}_{I_{j,k}}(t) &&= 2^{j/2} \phi(2^j t - k) \\ \psi_{j,k}(t) &= 2^{j/2} \left(\mathbf{1}_{I_{j,k}^l}(t) - \mathbf{1}_{I_{j,k}^r}(t) \right) &&= 2^{j/2} \psi(2^j t - k)\end{aligned}$$

Es gelten die Beziehungen

$$\begin{bmatrix} \phi \\ \psi \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{1,0} \\ \phi_{1,1} \end{bmatrix}, \quad \begin{bmatrix} \phi_{1,0} \\ \phi_{1,1} \end{bmatrix} = H \cdot \begin{bmatrix} \phi \\ \psi \end{bmatrix}$$

und damit auch für alle $j, k \in \mathbb{Z}$

$$\begin{bmatrix} \phi_{j,k} \\ \psi_{j,k} \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{j+1,2k} \\ \phi_{j+1,2k+1} \end{bmatrix}, \quad \begin{bmatrix} \phi_{j+1,2k} \\ \phi_{j+1,2k+1} \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{j,k} \\ \psi_{j,k} \end{bmatrix}$$

Definition 5.10 (Zweidimensionale Haar-Funktionen).

Die zweidimensionalen Haar-Funktionen sind die vier Funktionen

$$\begin{aligned}\phi(x, y) &= \phi(x) \cdot \phi(y) \\ \psi^H(x, y) &= \psi(x) \cdot \phi(y) \\ \psi^V(x, y) &= \phi(x) \cdot \psi(y) \\ \psi^D(x, y) &= \psi(x) \cdot \psi(y)\end{aligned}$$

ϕ wird als Skalierungsfunktion bezeichnet, die ψ^H, ψ^V, ψ^D sind die Waveletfunktionen.

Hierbei steht “H” für “horizontal”, “V” für “vertikal”, “D” für “diagonal”, entsprechend den Richtungen, in welchen diese Funktionen Unterschiede registrieren.

Definition 5.11 (Dilation und Translation für 2D-Haar-Funktionen).

Die durch Dilation und Translation skalierten 2-dimensionalen Haar-Funktionen sind für $j, k, \ell \in \mathbb{Z}$:

$$\begin{aligned}\phi_{j,k,\ell}(x, y) &= \phi_{j,k}(x) \cdot \phi_{j,\ell}(y) = 2^j \phi(2^j x - k, 2^j y - \ell) \\ \psi_{j,k,\ell}^H(x, y) &= \psi_{j,k}(x) \cdot \phi_{j,\ell}(y) = 2^j \psi^H(2^j x - k, 2^j y - \ell) \\ \psi_{j,k,\ell}^V(x, y) &= \phi_{j,k}(x) \cdot \psi_{j,\ell}(y) = 2^j \psi^V(2^j x - k, 2^j y - \ell) \\ \psi_{j,k,\ell}^D(x, y) &= \psi_{j,k}(x) \cdot \psi_{j,\ell}(y) = 2^j \psi^D(2^j x - k, 2^j y - \ell)\end{aligned}$$

Offensichtlich gilt

$$\begin{bmatrix} \phi \\ \psi^H \\ \psi^V \\ \psi^D \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} = (H \otimes H) \begin{bmatrix} \phi_{1,0,0} \\ \phi_{1,1,0} \\ \phi_{1,0,1} \\ \phi_{1,1,1} \end{bmatrix}$$

Beachte: auch $H \otimes H$ ist eine Hadamard-Matrix, also insbesondere orthogonal.

Lemma 5.5 Für beliebige a, b, c, d gilt

$$(H \otimes H) \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} a' \\ b' \\ c' \\ d' \end{bmatrix} \iff H \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix} \cdot H = \begin{bmatrix} a' & b' \\ c' & d' \end{bmatrix}$$

Satz 5.3 Die zweidimensionalen Skalierungs- und Waveletgleichungen schreiben sich in Matrixform:

$$\begin{bmatrix} \phi_{j,k,\ell} & \psi_{j,k,\ell}^H \\ \psi_{j,k,\ell}^V & \psi_{j,k,\ell}^D \end{bmatrix} = H \cdot \begin{bmatrix} \phi_{j+1,2k,2\ell} & \phi_{j+1,2k+1,2\ell} \\ \phi_{j+1,2k,2\ell+1} & \phi_{j+1,2k+1,2\ell+1} \end{bmatrix} \cdot H$$

oder äquivalent

$$\begin{bmatrix} \phi_{j,k,\ell} \\ \psi_{j,k,\ell}^H \\ \psi_{j,k,\ell}^V \\ \psi_{j,k,\ell}^D \end{bmatrix} = (H \otimes H) \begin{bmatrix} \phi_{j+1,2k,2\ell} \\ \phi_{j+1,2k+1,2\ell} \\ \phi_{j+1,2k,2\ell+1} \\ \phi_{j+1,2k+1,2\ell+1} \end{bmatrix}$$

Definition 5.12 (2D-Haar-Detail- und Waveleträume).

Die für die zweidimensionale Waveletanalyse und -synthese relevanten Vektorräume sind:

$$\begin{aligned}V_j &= \overline{\text{span}} \{ \phi_{j,k,\ell} \}, & W_j^H &= \overline{\text{span}} \{ \psi_{j,k,\ell}^H \}, \\ W_j^V &= \overline{\text{span}} \{ \psi_{j,k,\ell}^V \}, & W_j^D &= \overline{\text{span}} \{ \psi_{j,k,\ell}^D \}.\end{aligned}$$

Für $\mathcal{L}^2(\mathbb{R}^2)$ -Haar-Wavelets laufen alle Indices über $j, k, \ell \in \mathbb{Z}$.

Für $\mathcal{L}^2([0, 1]^2)$ -Haar-Wavelets laufen alle Indices über $j \geq 0, 0 \leq k, \ell < 2^j$.

Die Vektorräume V_j werden als Approximationsräume, die Vektorräume W_j^H, W_j^V, W_j^D werden als Detail- oder Waveleträume bezeichnet.

Die Aussagen über vollständige Basen der \mathcal{L}^2 -Räume übertragen sich sinngemäß vom eindimensionalen auf den zweidimensionalen Fall, ebenso die Definition und Eigenschaften der Waveletkoeffizienten.

Satz 5.4 Für jedes $j \in \mathbb{Z}$ gilt die Zerlegung

$$V_{j+1} = V_j \oplus W_j^H \oplus W_j^V \oplus W_j^D$$

Eine Funktion $f \in V_{j+1}$ hat also eine eindeutige Zerlegung

$$f_{j+1} = f_j + g_j^H + g_j^V + g_j^D \quad \text{mit } f_j \in V_j, g_j^x \in W_j^x (x \in \{H, D, V\}).$$

Für die Haar-Waveletanalyse in $[0, 1]^2$ ordnet man die Koeffizienten dieser Funktionen bezüglich der Basen in den ihnen zugeordneten Teilräumen in einem quadratischen Schema der Seitenlänge 2^{j+1} an:

$$\boxed{f_{j+1}} \leftrightarrow \begin{array}{|c|c|} \hline f_j & g_j^H \\ \hline g_j^V & g_j^D \\ \hline \end{array}$$

Eine Phase Haar-*Waveletanalyse* besteht also darin, die rechte Seite aus den Daten der linken Seite zu berechnen:

$$\boxed{f_{j+1}} \mapsto \begin{array}{|c|c|} \hline f_j & g_j^H \\ \hline g_j^V & g_j^D \\ \hline \end{array}$$

Eine Stufe Haar-*Waveletsynthese* besteht also darin, die linken Seite aus den Daten der rechten Seite zu berechnen:

$$\boxed{f_{j+1}} \leftarrow \begin{array}{|c|c|} \hline f_j & g_j^H \\ \hline g_j^V & g_j^D \\ \hline \end{array}$$

Ein *Bild* ist eine $(M \times N)$ -Matrix $\mathbf{a}_{M,N}$ (von Grauwerten, beispielsweise). Für die Haar-Waveletanalyse sollten M und N gerade sein, also $M = 2m$ und $N = 2n$.

Ein Phase der *Waveletanalyse* besteht darin, dieses Bild $\mathbf{a}_{2m,2n}$ durch vier $(m \times n)$ -Bilder $\mathbf{a}_{m,n}, \mathbf{d}_{m,n}^H, \mathbf{d}_{m,n}^V, \mathbf{d}_{m,n}^D$ zu ersetzen nach dem Schema

$$\boxed{\mathbf{a}_{2m,2n}} \mapsto \begin{array}{|c|c|} \hline \mathbf{a}'_{m,n} & \mathbf{d}_{m,n}^H \\ \hline \mathbf{d}_{m,n}^V & \mathbf{d}_{m,n}^D \\ \hline \end{array}$$

Wiederum steht \mathbf{a} für "Approximation" und \mathbf{d} für "Detail". Die Transformation lässt sich einfach mittels der Transformationsmatrizen aus dem eindimensionalen Fall

beschreiben. Seien also wieder

$$A_n = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & & & \\ 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \quad D_n = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & & & & \\ -1 & & & & \\ & 1 & & & \\ & & -1 & & \\ & & & \ddots & \\ & & & & 1 \\ & & & & & -1 \end{bmatrix}.$$

Das sind $(2n \times n)$ -Matrizen. Dann gilt

$$\boxed{\mathbf{a}_{2m,2n}} \rightsquigarrow \begin{bmatrix} \mathbf{a}'_{m,n} & \mathbf{d}_{m,n}^H \\ \mathbf{d}_{m,n}^V & \mathbf{d}_{m,n}^D \end{bmatrix} = \begin{bmatrix} A_m^\dagger \\ D_m^\dagger \end{bmatrix} \cdot \mathbf{a}_{2m,2n} \cdot \begin{bmatrix} A_n & D_n \end{bmatrix}$$

Ausgeschrieben bedeutet das

$$\begin{aligned} \mathbf{a}'_{m,n} &= A_m^\dagger \cdot \mathbf{a}_{2m,2n} \cdot A_n \\ \mathbf{d}_{m,n}^H &= A_m^\dagger \cdot \mathbf{a}_{2m,2n} \cdot D_n \\ \mathbf{d}_{m,n}^V &= D_m^\dagger \cdot \mathbf{a}_{2m,2n} \cdot A_n \\ \mathbf{d}_{m,n}^D &= D_m^\dagger \cdot \mathbf{a}_{2m,2n} \cdot D_n \end{aligned}$$

Man kann und sollte das auch so lesen: die zweidimensionale Transformation besteht darin,

- auf dem Bild $\mathbf{a}_{2m,2n}$ erst die eindimensionale Transformation (parallel) auf den Zeilen auszuführen, das liefert $\tilde{\mathbf{a}}_{2m,2n} = \mathbf{a}_{2m,2n} \cdot \begin{bmatrix} A_n & D_n \end{bmatrix}$,
- dann die eindimensionale Transformation (parallel) auf den Spalten von $\tilde{\mathbf{a}}_{2m,2n}$, das liefert

$$\begin{bmatrix} A_m^\dagger \\ D_m^\dagger \end{bmatrix} \cdot \tilde{\mathbf{a}}_{2m,2n} = \begin{bmatrix} A_m^\dagger \\ D_m^\dagger \end{bmatrix} \cdot \mathbf{a}_{2m,2n} \cdot \begin{bmatrix} A_n & D_n \end{bmatrix}.$$

Man kann es natürlich auch umgekehrt machen: erst die Spalten und dann die Zeilen transformieren.

Für die *Synthese* hat man obige Beziehung umzukehren, was wegen der Orthogonalität der Matrizen $\begin{bmatrix} A_n & D_n \end{bmatrix}$ kein Problem ist:

$$\mathbf{a}_{2m,2n} = \begin{bmatrix} A_m & D_m \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a}'_{m,n} & \mathbf{d}_{m,n}^H \\ \mathbf{d}_{m,n}^V & \mathbf{d}_{m,n}^D \end{bmatrix} \cdot \begin{bmatrix} A_n^\dagger \\ D_n^\dagger \end{bmatrix}$$

Ausgeschrieben bedeutet das:

$$\boxed{\mathbf{a}_{2m,2n} = A_m \cdot \mathbf{a}'_{m,n} \cdot A_n^\dagger + D_m \cdot \mathbf{d}_{m,n}^V \cdot A_n^\dagger + A_m \cdot \mathbf{d}_{m,n}^H \cdot D_n^\dagger + D_m \cdot \mathbf{d}_{m,n}^D \cdot D_n^\dagger}$$

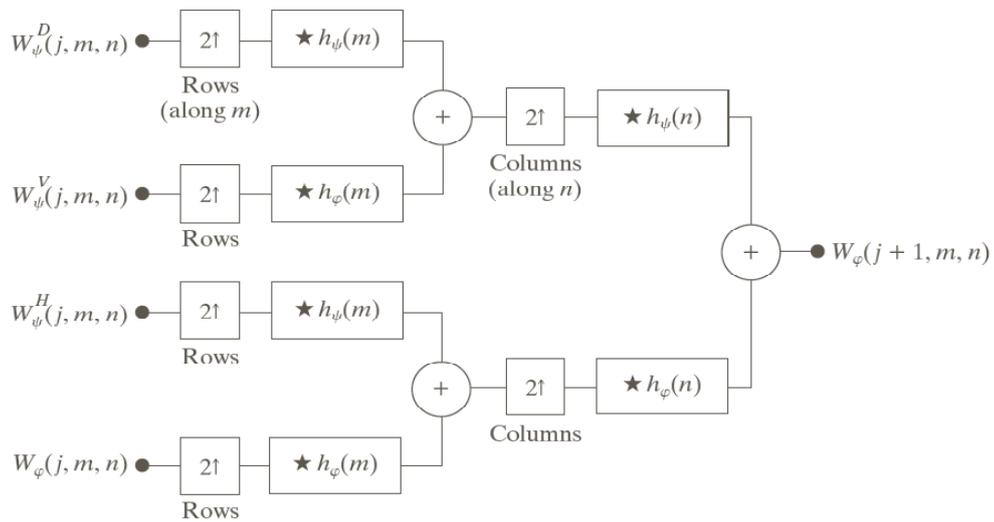
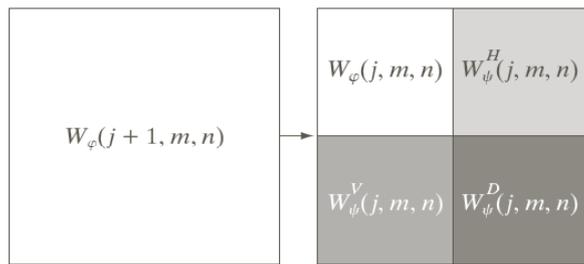
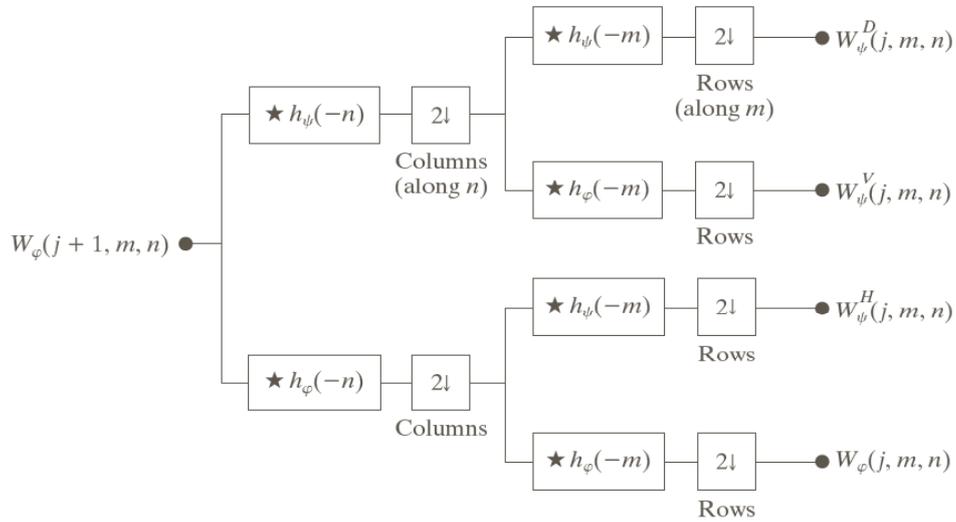


Abbildung 5.14: Einstufige 2-D Haar-Wavelettransformation als Filterbank

5.10 Mehrstufige 2D Haar-Transformation

Die einstufige 2D-Haar-Transformation lässt sich ganz analog zur Situation bei der 1D-Transformation zu einer mehrstufigen 2D-Transformation erweitern, indem man die Transformation auf die jeweils erzeugten Approximationsdaten iteriert anwendet. Das Schema ist übrigens für andere Wavelet-Transformationen genau gleich. Es soll genügen, das durch Abbildungen anzudeuten.

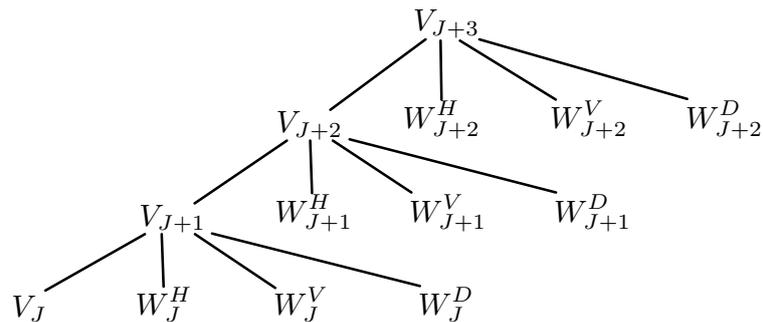


Abbildung 5.15: Zerlegungsschema einer 3-stufigen 2D-Wavelettransformation

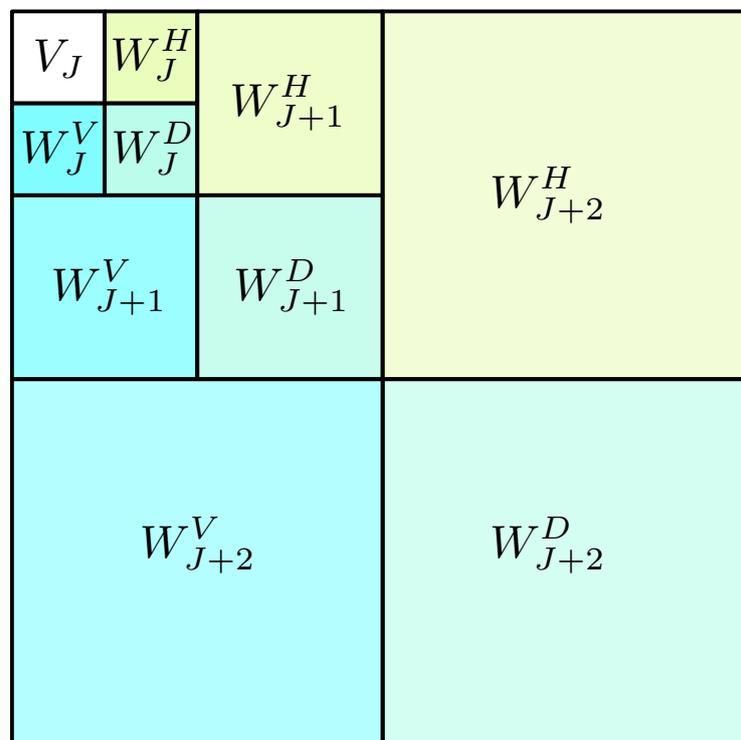


Abbildung 5.16: Koeffizientenschema einer 3-stufigen 2D-Wavelettransformation

5.11 Abschätzung der Größe von Haar-Waveletkoeffizienten

In diesem Abschnitt soll gezeigt werden, wie sich die Größe der Waveletkoeffizienten des Haar-Wavelets über mehrere Auflösungsstufen verhält, je nachdem, ob sich in dem entsprechenden dyadischen Intervall eine Sprungstelle der analysierten Funktion befindet oder nicht.

Zunächst eine Erinnerung: die dyadischen Intervalle sind definiert als

$$I_{j,k} = [2^{-j}k, 2^{-j}(k+1)) \quad (j, k \in \mathbb{Z})$$

mit den Mittelpunkten $t_{j,k} = 2^{-j}(k + 1/2)$.

Theorem 5.5

1. Ist eine Funktion $f(t)$ im Intervall $I_{j,k}$ zweimal stetig differenzierbar, so gilt

$$|\langle f | \psi_{j,k} \rangle| \approx \frac{1}{4} |f'(t_{j,k})| 2^{-3j/2}.$$

2. Ist eine Funktion $f(t)$ im Intervall $I_{j,k}$ zweimal stetig differenzierbar, ausgenommen für eine Sprung-Unstetigkeit an der Stelle $t_0 \in I_{j,k}$, so gilt

$$|\langle f | \psi_{j,k} \rangle| \approx \frac{1}{4} |f(t_0^-) - f(t_0^+)| 2^{-j/2}.$$

Dabei ist $f(t_0^+) - f(t_0^-) = \lim_{\tau \downarrow t_0} f(\tau) - \lim_{\tau \uparrow t_0} f(\tau)$ die Sprunghöhe an der Stelle t_0 . Die wesentliche Information steckt in den auflösungsabhängigen Faktoren $2^{-3j/2}$ (für den "glatten" Fall 1.) bzw. $2^{-j/2}$ (für den Fall 2. eines Sprunges). Bei wachsendem j (= zunehmender Auflösung) wird der Faktor im Fall 1. viel schneller klein als im Fall 2. Beobachtet man also die Größe von Wavelet-Koeffizienten für eine genestete Folge

$$I_{j,k} \supset I_{j+1,k'} \supset I_{j+2,k''} \supset \cdots \supset I_{j+\ell,k^\ell}$$

von dyadischen Intervallen, so kann man erkennen, ob im Intervall $I_{j+\ell,k^\ell}$ eine Sprung-Unstetigkeit von $f(t)$ vorhanden ist, oder nicht.

– Beweis von 1.

Die Taylorentwicklung an der Stelle $t_{j,k}$ für beliebige $t \in I_{j,k}$ lautet

$$f(t) = f(t_{j,k}) + f'(t_{j,k})(t - t_{j,k}) + \frac{1}{2} f''(\xi_{j,k})(t - t_{j,k})^2,$$

für ein geeignetes (von t abhängiges) $\xi_{j,k} = \xi_{j,k}(t) \in I_{j,k}$. Daraus ergibt sich für die Waveletkoeffizienten

$$\begin{aligned} \langle f | \psi_{j,k} \rangle &= \int_{I_{j,k}} f(t) \psi_{j,k}(t) dt = f(t_{j,k}) \int_{I_{j,k}} \psi_{j,k}(t) dt \\ &\quad + f'(t_{j,k}) \int_{I_{j,k}} (t - t_{j,k}) \psi_{j,k}(t) dt \\ &\quad + \frac{1}{2} \int_{I_{j,k}} f''(\xi_{j,k})(t - t_{j,k})^2 \psi_{j,k}(t) dt. \end{aligned}$$

Nun benutzt man folgende Fakten betreffend die Haar-Waveletfunktion $\psi(t)$:

$$\begin{aligned}
& - \int_{I_{j,k}} \psi_{j,k}(t) dt = 0, \\
& - \int_{I_{j,k}} t \psi_{j,k}(t) dt = -\frac{1}{4} 2^{-3j/2}, \\
& - \int_{I_{j,k}} (t - t_{j,k})^2 dt = \frac{1}{12} 2^{-3j}.
\end{aligned}$$

Aus der dritten Eigenschaft ergibt sich

$$\begin{aligned}
\left| \int_{I_{j,k}} f''(\xi_{j,k}) (t - t_{j,k})^2 \psi_{j,k}(t) dt \right| & \leq \max_{\xi \in I_{j,k}} |f''(\xi)| \int_{I_{j,k}} |(t - t_{j,k})^2 \psi_{j,k}(t)| dt \\
& \leq \max_{\xi \in I_{j,k}} |f''(\xi)| \cdot 2^{j/2} \int_{I_{j,k}} (t - t_{j,k})^2 dt \\
& \leq \max_{\xi \in I_{j,k}} |f''(\xi)| \cdot \frac{1}{12} \cdot 2^{-5j/2}
\end{aligned}$$

und damit zusammen die behauptete Abschätzung.

– Beweis von 2.

Sei nun $t_0 \in I_{j,k}^\ell$ eine Sprungstelle mit $2^{-j}k < t_0 \leq t_{j,k}$ (der Fall $t_{j,k} < t_0 < 2^{-j}k + 1k$ erledigt sich analog). Durch Taylorentwicklung von links und von rechts an der Stelle t_0 hat man

$$\begin{aligned}
f(t) &= f(t_0^-) + f'(\xi_-)(t - t_0) \quad t \in I_{j,k}, t < t_0, \xi_- \in (t, t_0) \\
f(t) &= f(t_0^+) + f'(\xi_+)(t - t_0) \quad t \in I_{j,k}, t > t_0, \xi_+ \in (t_0, t)
\end{aligned}$$

Für die Wavelekkoeffizienten ergibt sich

$$\begin{aligned}
\langle f | \psi_{j,k} \rangle &= \int_{I_{j,k}} f(t) \psi_{j,k}(t) dt = \\
& \underbrace{2^{j/2} \left(\int_{2^{-j}k}^{t_0} f(t_0^-) + \int_{t_0}^{t_{j,k}} f(t_0^+) - \int_{t_{j,k}}^{2^{-j}(k+1)} f(t_0^+) \right)}_{(t_0 - 2^{-j}k) \cdot (f(t_0^-) - f(t_0^+))} + \varepsilon_{j,k}
\end{aligned}$$

wobei

$$|\varepsilon_{j,k}| \leq \max_{\substack{\xi \in I_{j,k} \\ \xi \neq t_0}} |f'(\xi)| \int_{I_{j,k}} |t - t_0| |\psi_{j,k}(t)| dt \leq \max(\dots) \times \frac{1}{4} 2^{-3j/2}.$$

Nun ist $|t_0 - 2^{-j}k| \approx \frac{1}{4} 2^{-j}$, daher

$$\begin{aligned}
& \left| 2^{j/2} \left(\int_{2^{-j}k}^{t_0} f(t_0^-) + \int_{t_0}^{t_{j,k}} f(t_0^+) - \int_{t_{j,k}}^{2^{-j}(k+1)} f(t_0^+) \right) \right| \\
& = 2^{j/2} |t_0 - 2^{-j}k| \cdot |f(t_0^-) - f(t_0^+)| \approx \frac{1}{4} 2^{-j/2} |(f(t_0^-) - f(t_0^+))|
\end{aligned}$$

und damit ist auch dieser Fall bewiesen. \square

6. Diskrete Signale und Filter

In diesem Kapitel werden einige elementare Grundbegriffe aus dem Bereich der diskreten Signale und Filter behandelt. Auch hier geht es, wie bei der Fourier-Theorie um eine Zusammenstellung dessen, was für das Studium der Wavelets benötigt wird, nicht um eine lehrbuchmäßige Abhandlung. Beweise werden allenfalls skizziert.

6.1 2π -periodische Funktionen

1. 2π -periodische Funktionen werden identifiziert mit (komplexwertigen) Funktionen, die auf dem Intervall $I = [-\pi, \pi)$ definiert sind.
2. $\mathcal{L}^2(-\pi, \pi)$ ist der (Hilbert-)Raum der quadratintegrierbaren Funktionen auf dem Intervall I , d.h. der Funktionen $f : I \rightarrow \mathbb{C}$ mit¹ $\int_I |f(s)|^2 ds < \infty$. Dieser Raum besitzt das (komplexe) Skalarprodukt

$$\langle f | g \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \overline{g(s)} ds$$

und damit die \mathcal{L}^2 -Norm

$$\|f\|_2^2 = \langle f | f \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(s)|^2 ds.$$

3. Wie in Kapitel 3 dargestellt, bildet die Familie der Funktionen $\{\varepsilon_k(s)\}_{k \in \mathbb{Z}}$ mit $\varepsilon_k = \omega_{2\pi, k} : s \mapsto e^{iks}$ bilden eine Orthonormalbasis in $\mathcal{L}^2(-\pi, \pi)$, denn es ist

¹ Streng genommen muss man hier Integrierbarkeit im Sinne von Lebesgue fordern, was aber "für praktische Zwecke" immer gegeben ist.

insbesondere

$$\begin{aligned} \langle \varepsilon_j | \varepsilon_k \rangle &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{ijs} e^{-iks} ds \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{i(j-k)s} ds = \begin{cases} \frac{1}{2\pi} \int_{-\pi}^{\pi} 1 ds = 1 & \text{falls } j = k, \\ \frac{1}{2\pi i} \frac{1}{j-k} e^{i(j-k)s} \Big|_{-\pi}^{\pi} = 0 & \text{falls } j \neq k. \end{cases} \end{aligned}$$

4. Für eine (integrierbare) 2π -periodische Funktion f , d.h. $\int_I |f(s)| ds < \infty$, werden die Fourierkoeffizienten definiert durch

$$c_f[k] = \langle f | \varepsilon_k \rangle = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) e^{-iks} ds$$

Für hinreichend gutartige Funktionen (je nach Konvergenzbegriff) gilt dann Entwicklung in eine Fourierreihe:

$$f(s) \simeq \sum_{k \in \mathbb{Z}} c_f[k] e^{iks}.$$

5. Für Funktionen $f, g \in \mathcal{L}^2(-\pi, \pi)$ gilt die Parseval-Identität:

$$\sum_{k \in \mathbb{Z}} c_f[k] \cdot \overline{c_g[k]} = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(s) \cdot \overline{g(s)} ds = \langle f | g \rangle$$

und somit insbesondere die Plancherel-Identität

$$\sum_{k \in \mathbb{Z}} |c_f[k]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |f(s)|^2 ds = \|f\|_2^2$$

Der *Beweis* (formal hingeschrieben, ohne auf Konvergenz, Vertauschbarkeit von Summation und Integration, usw. zu achten) ist ganz einfach. Aus der Orthonormalität der Funktionen ε_k folgt:

$$\begin{aligned} \langle f | g \rangle &= \left\langle \sum_j c_f[j] \varepsilon_j \mid \sum_k c_g[k] \varepsilon_k \right\rangle \\ &= \sum_j \sum_k c_f[j] \overline{c_g[k]} \langle \varepsilon_j | \varepsilon_k \rangle = \sum_k c_f[k] \overline{c_g[k]}. \end{aligned}$$

6.2 Signale

Definition 6.1 (Diskretes Signal).

Ein (zeitdiskretes) Signal ist eine zweiseitig unendliche Folge

$$\mathbf{x} = (\dots, x[-2], x[-1], x[0], x[1], x[2], \dots) = (x[n])_{n \in \mathbb{Z}}$$

von komplexen Zahlen, also $\mathbf{x} \in \mathbb{C}^{\mathbb{Z}}$.

Bemerkungen und Bezeichnungen:

1. $\mathbb{C}^{\mathbb{Z}}$ ist ein \mathbb{C} -Vektorraum (überabzählbarer Dimension) bezüglich komponentenweiser Addition und Skalarmultiplikation.
2. $\ell^1 = \ell^1(\mathbb{Z})$: Unterraum der Signale \mathbf{x} mit $\|\mathbf{x}\|_1 = \sum_n |x[n]| < \infty$.
 ℓ^1 ist ein \mathbb{C} -Vektorraum abzählbarer Dimension mit der Norm $\|\cdot\|_1$.
3. $\ell^2 = \ell^2(\mathbb{Z})$: Unterraum der Signale \mathbf{x} mit "endlicher Energie", d.h.
 $\|\mathbf{x}\|_2^2 = \sum_{n \in \mathbb{Z}} |x[n]|^2 < \infty$.
 ℓ^2 ist ein \mathbb{C} -Vektorraum abzählbarer Dimension mit der Norm $\|\cdot\|_2$.
 ℓ^2 ist sogar ein Hilbertraum bezüglich des Skalarproduktes

$$\langle \mathbf{x} | \mathbf{y} \rangle = \sum_{n \in \mathbb{Z}} x[n] \cdot \overline{y[n]}.$$

NB: Jedes ℓ^1 -Signal ist auch ein ℓ^2 -Signal, aber nicht umgekehrt.

Definition 6.2 (Frequenzdarstellung eines Signals).

Die Frequenzdarstellung eines Signals \mathbf{x} ist die Fourierreihe

$$X(s) = \sum_{n \in \mathbb{Z}} x[n] e^{ins}.$$

Dies ist eine 2π -periodische Funktion. Die Koeffizienten erhält man aus $X(s)$ mittels Fourierintegral:

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(s) e^{-ins} ds = \langle X | \varepsilon_n \rangle.$$

Im Fall von ℓ^2 -Signalen gilt ("Energieerhaltung") die Plancherel-Formel (vgl. oben)

$$\|\mathbf{x}\|_2^2 = \sum_{n \in \mathbb{Z}} |x[n]|^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |X(s)|^2 ds = \|X(s)\|_2^2$$

und allgemeiner die Parseval-Formel

$$\langle \mathbf{x} | \mathbf{y} \rangle = \sum_{n \in \mathbb{Z}} x[n] \overline{y[n]} = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(s) \overline{Y(s)} ds = \langle X(s) | Y(s) \rangle.$$

Definition 6.3.

1. Der Impuls: $\boldsymbol{\delta} = (\delta_{n,0})_{n \in \mathbb{Z}}$, d.h. $\delta[0] = 1$ und $\delta[n] = 0$ für $n \neq 0$.
2. Translation: Für ein Signal \mathbf{x} und $k \in \mathbb{Z}$ ist

$$\tau^k \mathbf{x} = (x[n-k])_{n \in \mathbb{Z}}$$

das um die Distanz (Zeit) k verschobene Signal.

Die Abbildungen $\tau^k : \mathbb{C}^{\mathbb{Z}} \rightarrow \mathbb{C}^{\mathbb{Z}} : \mathbf{x} \mapsto \tau^k \mathbf{x}$ sind offensichtlich lineare Transformationen.

6.3 Faltung

Definition 6.4 (Faltung von Signalen).

Sind $\mathbf{x} = (x[n])_{n \in \mathbb{Z}}$ und $\mathbf{y} = (y[n])_{n \in \mathbb{Z}}$ Signale, so ist deren Faltung $\mathbf{x} \star \mathbf{y}$ definiert durch

$$(\mathbf{x} \star \mathbf{y})[n] = \sum_{k \in \mathbb{Z}} x[k] \cdot y[n - k] \quad (n \in \mathbb{Z}),$$

sofern diese Summen für alle $n \in \mathbb{Z}$ konvergieren.

Lemma 6.1 Die Faltungsoperation ist assoziativ und kommutativ und mit den Vektorraumoperationen von $\mathbb{C}^{\mathbb{Z}}$ verträglich.

Die wichtigsten Spezialfälle, in denen die Faltungsoperation sicher definiert ist, sind die beiden folgenden:

1. Ist $\mathbf{x} = (x[n])_{n \in \mathbb{Z}}$ ein *endliches* Signal, d.h. nur endlich viele $x[n]$ sind $\neq 0$, so ist $\mathbf{x} \star \mathbf{y}$ wieder ein Signal, und mit $\mathbf{y} \in \ell^1$ bzw. $\mathbf{y} \in \ell^2$ gehört auch $\mathbf{x} \star \mathbf{y}$ wieder zu ℓ^1 bzw. ℓ^2 .
2. Sind $\mathbf{x}, \mathbf{y} \in \ell^1$, so gehört auch $\mathbf{x} \star \mathbf{y}$ zu ℓ^1 . Dies ergibt sich aus:

$$\begin{aligned} \|\mathbf{x} \star \mathbf{y}\|_1 &= \sum_{n \in \mathbb{Z}} |(\mathbf{x} \star \mathbf{y})[n]| \\ &= \sum_{n \in \mathbb{Z}} \left| \sum_{k \in \mathbb{Z}} x[k] \cdot y[n - k] \right| \\ &\leq \sum_{n \in \mathbb{Z}} \sum_{k \in \mathbb{Z}} |x[k]| \cdot |y[n - k]| \\ &= \sum_{k \in \mathbb{Z}} |x[k]| \cdot \sum_{n \in \mathbb{Z}} |y[n]| = \|\mathbf{x}\|_1 \cdot \|\mathbf{y}\|_1 \end{aligned}$$

Hinweis: Für $\mathbf{x}, \mathbf{y} \in \ell^2$ ist die Angelegenheit etwas komplizierter, siehe unten.

Analog zur Situation bei der Fouriertransformation — siehe Theorem 4.2 — gibt es auch hier ein *Faltungstheorem*, das den Zusammenhang zwischen Faltung und Fourierreihen regelt. Auch dies ist eine fundamentale Tatsache, insbesondere die Realisierung von Filterungsoperationen betreffend.

Theorem 6.1 — Faltungstheorem.

Sind $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \ell^1$ mit $\mathbf{z} = \mathbf{x} \star \mathbf{y}$, so gilt für die zugehörigen Fourierreihen

$$\forall s : Z(s) = X(s) \cdot Y(s).$$

Das ergibt sich aus:

$$\begin{aligned}
 Z(s) &= \sum_{n \in \mathbb{Z}} \left(\sum_{k \in \mathbb{Z}} x[k] y[n-k] \right) e^{i n s} \\
 &= \sum_{n, k \in \mathbb{Z}} x[k] e^{i k s} y[n-k] e^{i(n-k)s} \\
 &= \sum_{k \in \mathbb{Z}} x[k] e^{i k s} \cdot \sum_{n \in \mathbb{Z}} y[n] e^{i n s} \\
 &= X(s) \cdot Y(s)
 \end{aligned}$$

6.4 Filter

Definition 6.5 (Filter).

1. Eine lineare Transformation $T : \mathbb{C}^{\mathbb{Z}} \rightarrow \mathbb{C}^{\mathbb{Z}}$ (oder von ℓ^1 bzw. ℓ^2) heisst translationsinvariant, wenn sie mit τ kommutiert:

$$\forall \mathbf{x} \in \mathbb{C}^{\mathbb{Z}} : T(\tau \mathbf{x}) = \tau(T \mathbf{x}),$$

kurz $T \circ \tau = \tau \circ T$. Es folgt dann auch $T \circ \tau^k = \tau^k \circ T$ für alle $k \in \mathbb{Z}$.

2. Eine lineare Transformation $T : \ell^1 \rightarrow \ell^1$ heisst stetig (oder stabil), wenn es eine Konstante $C > 0$ gibt mit

$$\forall \mathbf{x} \in \ell^1 : \|T \mathbf{x}\|_1 \leq C \cdot \|\mathbf{x}\|_1.$$

Äquivalent kann man formulieren: für jede Folge $(\mathbf{x})_{m \in \mathbb{N}} \in \ell^1$ und $\mathbf{x} \in \ell^1$ mit $(\mathbf{x})_m \rightarrow_{\ell^1} \mathbf{x}$ gilt $(T \mathbf{x})_m \rightarrow_{\ell^1} T \mathbf{x}$, d.h.

$$\|\mathbf{x}_m - \mathbf{x}\| \rightarrow_{m \rightarrow \infty} 0 \quad \Rightarrow \quad \|T \mathbf{x}_m - T \mathbf{x}\| \rightarrow_{m \rightarrow \infty} 0.$$

Entsprechend formuliert man das für ℓ^2 -Transformationen.

3. Ein ℓ^1 -Filter ist eine translationsinvariante und stetige lineare Transformation von ℓ^1 . Analog für ℓ^2 .

■ **Beispiel 6.1** Sei $\mathbf{h} \in \ell^1$, dann ist

$$T_{\mathbf{h}} : \ell^1 \rightarrow \ell^1 : \mathbf{x} \mapsto \mathbf{x} \star \mathbf{h}$$

ein ℓ^1 -Filter. Translationsinvarianz rechnet man direkt nach, die Stetigkeit ergibt sich aus obigem

$$\|T_{\mathbf{h}} \mathbf{x}\|_1 = \|\mathbf{x} \star \mathbf{h}\| \leq \|\mathbf{x}\|_1 \cdot \|\mathbf{h}\|_1.$$

die gesuchte Konstante C ist also gerade $\|\mathbf{h}\|_1$. ■

Theorem 6.2 — Realisierung von Filteroperationen.

Zu jedem ℓ^1 -Filter T gibt es ein $\mathbf{h} \in \ell^1$ mit $T = T_{\mathbf{h}}$.

Zum Beweis schreibt man ein Signal \mathbf{x} als Linearkombination von geschifteten Impulsen:

$$\mathbf{x} = \sum_{k \in \mathbb{Z}} x[k] \tau^k \delta.$$

Es sei nun $\mathbf{h} = T\delta$. Aus der Linearität und Translationsinvarianz von T ergibt sich

$$T\mathbf{x} = \sum_k x[k] T\tau^k \delta = \sum_k x[k] \tau^k T\delta = \sum_k x[k] \tau^k \mathbf{h}.$$

Nun ist $(\tau^k \mathbf{h})[n] = h[n - k]$, also

$$(T\mathbf{x})[n] = \sum_{k \in \mathbb{Z}} x[k] h[n - k].$$

uns somit $T\mathbf{x} = \mathbf{x} \star \mathbf{h}$.

NB: Für die Vertauschung von T mit der unendlichen Summe \sum_k benötigt man die Stetigkeit von T .

Einige Erläuterungen hierzu:

1. Zur Terminologie: $\mathbf{h} = T\delta$ heisst *Impulsantwort* des Filters. Die zugehörige Fourierreihe $H(s)$ wird als *Frequenzantwort* oder auch *Transferfunktion* des Filters bezeichnet.
2. In der Systemtheorie betrachtet man oft auch die Potenzreihe

$$H(z) = \sum_{k \in \mathbb{Z}} h[k] z^{-k},$$

die als *z-Transformierte* des Signals bzw. Filters \mathbf{h} bezeichnet wird. Demnach ist $H(s) = H(e^{-is})$. Das Überladen des Symbols H ist üblich. Schreibt man $H(s)$ (s reell) so betrachtet man $H(z)$ nur für z auf dem komplexen Einheitskreis. $H(z)$ kann man natürlich betrachten, wenn z eine komplexe Variable ist. Manche Autoren definieren übrigens $H(s) = H(e^{is})$, d.h. es wäre dann $H(s) = \sum_k h[k] e^{-iks}$.

3. Zwar gehört das Signal $\mathbf{x}_s = (e^{-ins})_{n \in \mathbb{Z}}$ nicht zu ℓ^1 , man kann aber trotzdem $T_{\mathbf{h}}\mathbf{x}_s = \mathbf{x}_s \star \mathbf{h}$ berechnen:

$$(\mathbf{x}_s \star \mathbf{h})[n] = \sum_{k \in \mathbb{Z}} e^{-iks} h[n - k] = e^{-ins} \sum_{k \in \mathbb{Z}} e^{i(n-k)s} h[n - k] = H(s) \cdot e^{-ins},$$

und das kann man so lesen: jede reine Schwingung $\mathbf{x}_s = (e^{-ins})_{n \in \mathbb{Z}}$ ist Eigenvektor von $T_{\mathbf{h}}$ zum Eigenwert $H(s)$.

4. Die entsprechende ℓ^2 -Theorie ist beweistechnisch etwas komplizierter. Es gilt aber eine analoge Aussage: Die ℓ^2 -Filter sind genau diejenigen Transformationen $T_{\mathbf{h}} : \mathbf{x} \mapsto \mathbf{x} \star \mathbf{h}$, für die $H(s) \in L^\infty(-\pi, \pi)$ ist.

5. Folgerung aus Obigem:

Ist $T = T_{\mathbf{h}}$ ein ℓ^1 -Filter mit Frequenzantwort $H(s)$, so gilt für jedes ℓ^1 -Signal \mathbf{x} und für $\mathbf{y} = T\mathbf{x} = \mathbf{x} \star \mathbf{h}$:

$$\forall s : Y(s) = X(s) \cdot H(s).$$

Definition 6.6 (FIR-Filter).

1. Ein Filter $T = T_{\mathbf{h}}$ heisst reell, wenn $h[n] \in \mathbb{R}$ sind. Für einen reellen Filter \mathbf{h} gilt

$$\overline{H(s)} = \sum_n h[n] e^{-ins} = H(-\omega)$$

und somit $|H(\omega)| = |H(-\omega)|$, d.h., die Funktion $\omega \mapsto |H(\omega)|$ ist eine gerade Funktion. Es genügt, sie auf dem Intervall $[0, \pi]$ zu kennen.

2. Ein Filter $T = T_{\mathbf{h}}$ heisst kausal, wenn $h[n] = 0$ für alle $n < 0$. Bei einem kausalen Filter \mathbf{h} gilt für $\mathbf{y} = T_{\mathbf{h}}\mathbf{x}$:

$$y[n] = \sum_{k \leq n} x[k] h[n - k] = \sum_{k \geq 0} x[n - k] h[k],$$

die Antwort (output) $y[n]$ zu einem Zeitpunkt n hängt also nur von den Eingaben (input) $x[n - k]$ zu früheren Zeitpunkten $n - k$ ab.

3. Ein Filter $T = T_{\mathbf{h}}$ heisst FIR-Filter (**f**inite **i**mpulse **r**esponse), wenn nur endlich-viele Filterkoeffizienten $h[n] \neq 0$ sind. Ein FIR-Filter ist also durch die Angabe endlich-vieler Filterkoeffizienten $(h[a], h[a+1], \dots, h[b])$ mit $a < b$ und $h[a] \neq 0 \neq h[b]$ gegeben.

6.5 Downsampling und Upsampling

Definition 6.7 (Downsampling).

Ist \mathbf{x} ein Signal, so bezeichnet $\mathbf{y} = \downarrow_2 \mathbf{x}$ das Signal, bei dem jeder zweite Koeffizient (die Koeffizienten mit ungeradem Index) eliminiert wird:

$$y[n] = x[2n] \quad (n \in \mathbb{Z}).$$

Das ist Downsampling mit Faktor 2.

Dies ist keine Filteroperation, denn sie ist nicht translationsinvariant! I.a. gilt beispielsweise

$$(\downarrow_2 \tau^k \mathbf{x})[0] = x[-k] \neq x[-2k] = (\tau^k \downarrow_2 \mathbf{x})[0].$$

Für die Frequenzdarstellung gilt wegen $(-1)^n = e^{in\pi}$:

$$Y(s) = \sum_{n \in \mathbb{Z}} x[2n] e^{ins} = \sum_{n \in \mathbb{Z}} x[n] \frac{1 + (-1)^n}{2} e^{ins/2} = \frac{1}{2} \left(X\left(\frac{s}{2}\right) + X\left(\frac{s}{2} + \pi\right) \right)$$

Definition 6.8 (Upsampling).

Ist \mathbf{x} ein Signal, so bezeichnet $\mathbf{y} = \uparrow_2 \mathbf{x}$ das Signal, bei dem zwischen je zwei Koeffizienten von \mathbf{x} eine 0 eingefügt wird:

$$y[n] = \begin{cases} x[n/2] & \text{falls } n \text{ gerade,} \\ 0 & \text{falls } n \text{ ungerade.} \end{cases}$$

Auch dies ist keine Filteroperation, da sie nicht translationsinvariant ist. Für die Frequenzdarstellung gilt

$$Y(s) = \sum_{n \in \mathbb{Z}} x[n] e^{i2ns} = X(2s).$$

Downsampling und Upsampling kommutieren nicht miteinander: es ist $\downarrow_2 \uparrow_2 \mathbf{x} = \mathbf{x}$, aber für $\mathbf{y} = \uparrow_2 \downarrow_2 \mathbf{x}$ gilt

$$y[n] = \begin{cases} x[n] & \text{falls } n \text{ gerade,} \\ 0 & \text{falls } n \text{ ungerade,} \end{cases}$$

mit der Frequenzdarstellung

$$Y(\omega) = \frac{1}{2} (X(\omega) + X(\omega + \pi)).$$

7. Orthogonale Filter

7.1 Allgemeines

Ist $\mathbf{h} = (h_0, h_1, \dots, h_L)$ ein reeller kausaler Filter endlicher Länge $L + 1$ (wobei $h_0 \neq 0 \neq h_L$ angenommen werden soll), so kann man diesem zuordnen

- das Polynom (seine “ z -Transformierte”)

$$h(z) = h_0 + h_1 z + h_2 z^2 + \dots + h_L z^L$$

- die Fourierreihe (seine “Frequenzantwort”)

$$H(\omega) = h_0 + h_1 e^{i\omega} + h_2 e^{2i\omega} \dots + h_L e^{Li\omega} = h(e^{i\omega})$$

Ein Signal $\mathbf{a} = (a[n])_{n \in \mathbb{Z}}$ kann ebenso durch

- die Potenzreihe (seine “ z -Transformierte”)

$$a(z) = \sum_{n \in \mathbb{Z}} a[n] z^n$$

- die Fourierreihe (seine “Frequenzdarstellung”)

$$A(\omega) = \sum_{n \in \mathbb{Z}} a[n] e^{in\omega} = a(e^{i\omega})$$

repräsentiert werden¹.

¹ Der Unterschied zwischen der Darstellung als Polynom bzw. Potenzreihe einerseits und als trigonometrisches Polynom bzw. Fourierreihe andererseits ist der, dass man sich im ersten Fall die Variable z als ganz allgemeine komplexe Variable vorstellt, sich im zweiten Fall mit $z = e^{i\omega}$ auf das Verhalten auf dem komplexen Einheitskreis beschränkt.

Satz 7.1 Die Zeilen der Matrix H sind orthogonal, d.h. es gilt $H \cdot H^t = I$, genau dann, wenn die folgenden $\frac{L+1}{2}$ Bedingungen erfüllt sind:

$$(O_m) \quad \sum_{k=2m}^L h_k h_{k-2m} = \delta_{m,0} \quad (0 \leq m < L/2)$$

Dies liegt daran, dass sich je zwei Zeilen der Matrix H mit ihren Filterkoeffizienten in $L + 1 - 2m$ Positionen für $m \in \{0, 1, 2, \dots, (L + 1)/2\}$ überschneiden.

- Für $m = (L+1)/2$ sind die Zeilen automatisch orthogonal, d.h. für $m \geq (L+1)/2$ ist (O_m) trivialerweise erfüllt.
- Für $1 \leq m < (L + 1)/2$ drückt (O_m) die Orthogonalität aus, wenn zwei Zeilen genau $L + 1 - 2m$ Filterpositionen gemeinsam haben.
- Speziell im Fall $m = 0$ hat man die Forderung

$$h_0^2 + h_1^2 + \dots + h_L^2 = 1,$$

d.h., dass die Zeilenvektoren von H_N ℓ^2 -Länge 1 haben.

Anschaulich stellen sich die Überlappungsmuster für die Orthogonalitätsbedingungen so dar:

- Bedingung (O_0)

h_L	h_{L-1}	h_{L-2}	\dots	h_1	h_0
h_L	h_{L-1}	h_{L-2}	\dots	h_1	h_0

- Bedingung (O_1)

h_L	h_{L-1}	h_{L-2}	h_{L-3}	\dots	\dots	h_1	h_0		
		h_L	h_{L-1}	h_{L-2}	\dots	h_3	h_2	h_1	h_0

- Bedingung (O_2)

h_L	h_{L-1}	h_{L-2}	h_{L-3}	h_{L-4}	h_{L-5}	\dots	h_1	h_0				
				h_L	h_{L-1}	\dots	h_5	h_4	h_3	h_2	h_1	h_0

- Bedingung $(O_{\frac{L-1}{2}})$

h_L	h_{L-1}	\dots	h_1	h_0			
			h_L	h_{L-1}	\dots	h_1	h_0

Die Orthogonalitätsbedingung kann man auch so beschreiben, dass man mittels H das durch $h(1/z)$ repräsentierte Signal filtert. Das schreibt sich dann³ als

$$h(z) \cdot h\left(\frac{1}{z}\right) + h(-z) \cdot h\left(-\frac{1}{z}\right) = 2,$$

beziehungsweise

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = H(\omega) \cdot \overline{H(\omega)} + H(\omega + \pi) \cdot \overline{H(\omega + \pi)} = 2.$$

³ Da die rechte Seite eine Konstante ist, kann man die Substitution $z^2 \leftarrow z$ weglassen.

schreiben. Dies lässt sich äquivalent auch mittels

$$\begin{aligned} h(z) \cdot g(1/z) + h(-z) \cdot g(-1/z) &= 0, \\ H(\omega) \cdot \overline{G(\omega)} + H(\omega + \pi) \cdot \overline{G(\omega + \pi)} &= 0 \end{aligned}$$

formulieren.

Seien nun \mathbf{h} und \mathbf{g} Filter wie beschrieben und H und G , sowie H^\dagger und G^\dagger die zugehörigen Transformationen bzw. Matrizen.

Satz 7.2 — Rekonstruktionsbedingung.

Die “Rekonstruktionsbedingung” für ein Filterpaar (\mathbf{g}, \mathbf{h}) , d.h., die Bedingung

$$W^\dagger \cdot W = \begin{bmatrix} H^\dagger & G^\dagger \end{bmatrix} \cdot \begin{bmatrix} H \\ G \end{bmatrix} = H^\dagger \cdot H + G^\dagger \cdot G = I,$$

für $W = \begin{bmatrix} H \\ G \end{bmatrix}$ gilt genau dann, wenn die beiden Bedingungen

$$\begin{aligned} h(z) \cdot h\left(\frac{1}{z}\right) + g(z) \cdot g\left(\frac{1}{z}\right) &= 2, \\ h(z) \cdot h\left(-\frac{1}{z}\right) + g(z) \cdot g\left(-\frac{1}{z}\right) &= 0, \end{aligned}$$

oder äquivalent

$$\begin{aligned} |H(\omega)|^2 + |G(\omega)|^2 &= 2, \\ H(\omega) \cdot \overline{H(\omega + \pi)} + G(\omega) \cdot \overline{G(\omega + \pi)} &= 0 \end{aligned}$$

erfüllt sind.

Man muss nur die Operationen auf den zugehörigen Potenzreihen beschreiben:

– Die Komposition $H^\dagger \circ H$ leistet:

$$H^\dagger \circ H : a(z) \mapsto \frac{1}{2} (a(z) \cdot h(z) + a(-z) \cdot h(-z)) \cdot h\left(\frac{1}{z}\right)$$

– Die Komposition $G^\dagger \circ G$ leistet:

$$G^\dagger \circ G : a(z) \mapsto \frac{1}{2} (a(z) \cdot g(z) + a(-z) \cdot g(-z)) \cdot g\left(\frac{1}{z}\right)$$

– Damit gilt

$$\begin{aligned} H^\dagger \circ H + G^\dagger \circ G : a(z) \mapsto & \frac{1}{2} \left(h(z) \cdot h\left(\frac{1}{z}\right) + g(z) \cdot g\left(\frac{1}{z}\right) \right) \cdot a(z) \\ & + \frac{1}{2} \left(h(-z) \cdot h\left(\frac{1}{z}\right) + g(-z) \cdot g\left(\frac{1}{z}\right) \right) \cdot a(-z) \end{aligned}$$

Der Koeffizient von $a(z)$ muss identisch = 1 sein, der von $a(-z)$ muss verschwinden.

Auf der Ebene der Filterkoeffizienten wird die Rekonstruktionsbedingung durch

$$\sum_{k \in \mathbb{Z}} h(m - 2k) \cdot h(n - 2k) + \sum_{k \in \mathbb{Z}} g(m - 2k) \cdot g(n - 2k) = \delta_{m,n}$$

ausgedrückt.

Theorem 7.2 — Dualer Filter.

Ist $\mathbf{h} = (h_0, \dots, h_L)$ ein orthogonaler Filter der Länge $L + 1$ mit ungeradem L , d.h. $H \cdot H^t = I$, und definiert man den zu h dualen Filter $\mathbf{g} = (g_0, \dots, g_L)$ durch

$$g_k = (-1)^k h_{L-k} \quad (0 \leq k \leq L),$$

so gilt:

1. \mathbf{g} ebenfalls ein orthogonaler Filter, d.h. $G \cdot G^t = I$;
2. die beiden Filter \mathbf{g} und \mathbf{h} sind orthogonal zueinander, d.h. $H \cdot G^t = 0 = G \cdot H^t$;
3. die Rekonstruktionsbedingung

$$H^t \cdot H + G^t \cdot G = I$$

ist erfüllt.

Zum *Beweis*: Die Definition des Filters \mathbf{g} drückt sich auf der Ebene der Polynome folgendermassen aus:

$$\begin{aligned} g(z) &= \sum_{k=0}^L g_k z^k = \sum_{k=0}^L h_{L-k} (-z)^k \\ &= \sum_{k=0}^L h_k (-z)^{L-k} = (-z)^L \sum_{k=0}^L h_k \left(-\frac{1}{z}\right)^k \\ &= (-z)^L h\left(-\frac{1}{z}\right) \end{aligned}$$

1. Orthogonalität von \mathbf{g} :

$$\begin{aligned} g(z) \cdot g\left(\frac{1}{z}\right) + g(-z) \cdot g\left(-\frac{1}{z}\right) \\ &= (-z)^L h\left(-\frac{1}{z}\right) \cdot \left(-\frac{1}{z}\right)^L h(-z) + z^L h\left(\frac{1}{z}\right) \cdot \left(\frac{1}{z}\right)^L h(z) \\ &= h\left(-\frac{1}{z}\right) \cdot h(-z) + h\left(\frac{1}{z}\right) \cdot h(z) = 2 \end{aligned}$$

2. Orthogonalität von \mathbf{g} und \mathbf{h} :

$$\begin{aligned} h(z) \cdot g\left(\frac{1}{z}\right) + h(-z) \cdot g\left(-\frac{1}{z}\right) \\ &= h(z) \cdot \left(-\frac{1}{z}\right)^L h(-z) + h(-z) \cdot \left(\frac{1}{z}\right)^L h(z) \\ &= \left(\frac{1}{z}\right)^L \left((-1)^L h(z) \cdot h(-z) + h(-z) \cdot h(z) \right) = 0, \end{aligned}$$

da L ungerade ist.

1. Das Operieren mit den Matrizen H_N , G_N , H_N^t , G_N^t ist natürlich das, was in Anwendungen konkret implementiert werden muss. Deshalb ist es angezeigt, die Multiplikationen explizit auszuschreiben:

Die Multiplikation eines Spaltenvektors $\mathbf{v} = (v_k)_{0 \leq k < N}$ mit der Matrix H_N (von links), als schreibt sich explizit als

$$H_N \cdot \mathbf{v} = \mathbf{w} = (w_j)_{0 \leq j < N/2} \quad \text{mit} \quad w_j = \sum_{k=0}^L h_k v_{2j+L-k \bmod N}.$$

Für die adjungierte Transformation multipliziert man einen Spaltenvektor $\mathbf{w} = (w_j)_{0 \leq j < N/2}$ (von links) mit der transponierten Matrix H_N^t :

$$H_N^t \cdot \mathbf{w} = \mathbf{v} = (v_j)_{0 \leq j < N} \quad \text{mit} \quad \begin{cases} v_{2j} = \sum_{k=0}^{(L-1)/2} h_{2k+1} w_{k+j-\frac{L-1}{2} \bmod N} \\ v_{2j+1} = \sum_{k=0}^{(L-1)/2} h_{2k} w_{k+j-\frac{L-1}{2} \bmod N} \end{cases}$$

2. Sind \mathbf{h} , \mathbf{g} orthogonale Filter gleicher (gerader) Länge $L+1$, die auch zueinander orthogonal sind, so hat man auf Vektoren gerader Länge $N \geq L+1$ eine orthogonale Transformation (*Analyse*)

$$\mathbf{a} \mapsto \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} = \begin{bmatrix} H_N \cdot \mathbf{a} \\ G_N \cdot \mathbf{a} \end{bmatrix} = \begin{bmatrix} H_N \\ G_N \end{bmatrix} \cdot \mathbf{a}.$$

Ist die Rekonstruktionsbedingung erfüllt, so kann man aus \mathbf{b} und \mathbf{c} den Vektor \mathbf{a} wiedergewinnen (*Synthese*)

$$\begin{aligned} \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} &\mapsto \begin{bmatrix} H_N \\ G_N \end{bmatrix}^t \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} = [H_N^t \quad G_N^t] \begin{bmatrix} \mathbf{b} \\ \mathbf{c} \end{bmatrix} \\ &= H_N^t \cdot \mathbf{b} + G_N^t \cdot \mathbf{c} = (H_N^t \cdot H_N + G_N^t \cdot G_N) \cdot \mathbf{a} = \mathbf{a}. \end{aligned}$$

Die dritte Bedingung lässt sich ganz einfach erfüllen, indem man

$$g_j = (-1)^j h_{3-j} \quad (0 \leq j \leq 3)$$

wählt. Mit dieser Wahl sind aber die Bedingungen 1. und 2. äquivalent und man muss sich nur noch um die Bedingung 1. kümmern.

Nun muss man noch eine Tiefpassbedingung für \mathbf{h} und eine Hochpassbedingung für \mathbf{g} formulieren:

- \mathbf{h} ist ein Tiefpassfilter, d.h. $H(\pi) = 0$.
- \mathbf{g} ist ein Hochpassfilter, d.h. $G(0) = 0$.

Diese beiden Bedingungen sind bei der vereinbarten Abhängigkeit zwischen den g_j und h_j äquivalent:

$$H(\pi) = h_0 - h_1 + h_2 - h_3 = 0 \Leftrightarrow G(0) = g_0 + g_1 + g_2 + g_3 = h_3 - h_2 + h_1 - h_0 = 0.$$

Damit ist $\mathbf{h} = (h_0, h_1, h_2, h_3)$ zu bestimmen mit

$$\begin{array}{ll} (O_0) & h_0^2 + h_1^2 + h_2^2 + h_3^2 = 1 \\ (O_1) & h_0 h_2 + h_1 h_3 = 0 \\ (T_0) & h_0 - h_1 + h_2 - h_3 = 0 \end{array}$$

Bemerkung: Aus diesen drei Bedingungen folgt

$$h(0) = g(\pi) = h_0 + h_1 + h_2 + h_3 = \pm\sqrt{2}$$

Man hat drei Bedingungsgleichungen für die vier zu bestimmenden Filterkoeffizienten h_0, h_1, h_2, h_3 . Man wird also mit einer einparametrischen Lösungsmenge rechnen können.

In der Tat: aus (O_1) folgt

$$(h_2, h_3) = c \cdot (-h_1, h_0)$$

für ein $c \in \mathbb{R}, c \neq 0$. Damit ergibt sich aus (O_0)

$$h_0^2 + h_1^2 = \frac{1}{1+c^2}$$

und somit

$$h_1 = \frac{1-c}{1+c} \cdot h_0$$

und weiter

$$h_0^2 = \frac{(1+c)^2}{2(1+c^2)^2}.$$

Aus den beiden Möglichkeiten

$$h_0 = \pm \frac{1+c}{\sqrt{2}(1+c^2)}$$

wählt man das positive Vorzeichen und erhält insgesamt die Lösung

$$\begin{aligned} h_0 &= \frac{1+c}{\sqrt{2}(1+c^2)} \\ h_1 &= \frac{1-c}{\sqrt{2}(1+c^2)} \\ h_2 &= \frac{-c(1-c)}{\sqrt{2}(1+c^2)} \\ h_3 &= \frac{c(1+c)}{\sqrt{2}(1+c^2)}. \end{aligned}$$

Um den Parameter c festzulegen, führt man eine

- zweite Tiefpassbedingung:

$$H'(\pi) = 0$$

ein. Dies drückt sich in den Filterkoeffizienten aus durch

$$(T_1) \quad h_1 - 2h_2 + 3h_3 = 0.$$

Mit dieser zusätzliche Forderung, die man als

$$(1+2c)h_1 + 3ch_0 = 0$$

schreiben kann, und der obigen Darstellung

$$h_1 = \frac{1-c}{1+c} \cdot h_0$$

ergibt sich die Bedingungsgleichung

$$\frac{1-c}{1+c} = -\frac{3c}{1+2c}$$

und somit

$$c^2 + 4c + 1 = 0.$$

üblicherweise wählt man die Lösung

$$c = -2 + \sqrt{3},$$

also

$$h_0 = \pm \frac{1 + \sqrt{3}}{4\sqrt{2}}.$$

Nimmt man das positive Vorzeichen, so findet man als Lösung schliesslich

$$\begin{aligned} h_0 &= \frac{1}{4\sqrt{2}}(1 + \sqrt{3}) & g_0 &= \frac{1}{4\sqrt{2}}(1 - \sqrt{3}) \\ h_1 &= \frac{1}{4\sqrt{2}}(3 + \sqrt{3}) & g_1 &= \frac{-1}{4\sqrt{2}}(3 - \sqrt{3}) \\ h_2 &= \frac{1}{4\sqrt{2}}(3 - \sqrt{3}) & g_2 &= \frac{1}{4\sqrt{2}}(3 + \sqrt{3}) \\ h_3 &= \frac{1}{4\sqrt{2}}(1 - \sqrt{3}) & g_3 &= \frac{-1}{4\sqrt{2}}(1 + \sqrt{3}) \end{aligned}$$

Das ist das D_4 -Filterpaar.

8.2 Daubechies-Filter D_6

Die Herleitung eines Filterpaares (\mathbf{h}, \mathbf{g}) von Filtern $\mathbf{h} = (h_0, h_1, \dots, h_5)$ und $\mathbf{g} = (g_0, g_1, \dots, g_5)$ der Länge 6 verläuft ganz analog. Wieder verknüpft man die Koeffizienten beider Filter:

$$\boxed{g_j = (-1)^j h_{5-j} \quad (0 \leq j \leq 5)} \quad ,$$

womit ein grosser Teil der Orthogonalitätsbedingungen automatisch erfüllt ist. Es verbleiben drei Orthogonalitätsbedingungen:

$$\begin{aligned} (O_0) \quad & h_0^2 + h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 = 1 \\ (O_1) \quad & h_0 h_2 + h_1 h_3 + h_2 h_4 + h_3 h_5 = 0 \\ (O_2) \quad & h_0 h_4 + h_1 h_5 = 0 \end{aligned}$$

die Tiefpasseigenschaften von \mathbf{h} werden so spezifiziert:

$$\begin{aligned} (T_0) \quad & H(\pi) = 0 \quad \Leftrightarrow \quad h_0 - h_1 + h_2 - h_3 + h_4 - h_5 = 0 \\ (T_1) \quad & H'(\pi) = 0 \quad \Leftrightarrow \quad h_1 + 2h_2 - 3h_3 + 4h_4 - 5h_5 = 0 \\ (T_2) \quad & H''(\pi) = 0 \quad \Leftrightarrow \quad h_1 + 4h_2 - 9h_3 + 16h_4 - 25h_5 = 0 \end{aligned}$$

Eine reelle Lösung dieser 6 Gleichungen $(O_0), (O_1), (O_2), (T_0), (T_1), (T_2)$ für die 6 Unbekannten h_0, \dots, h_5 ist

$$\begin{aligned} h_0 &= \frac{\sqrt{2}}{32} \left(1 + \sqrt{10} + \sqrt{5 + 2\sqrt{10}} \right) && \approx 0.332671 \\ h_1 &= \frac{\sqrt{2}}{32} \left(5 + \sqrt{10} + 3\sqrt{5 + 2\sqrt{10}} \right) && \approx 0.806892 \\ h_2 &= \frac{\sqrt{2}}{32} \left(10 - 2\sqrt{10} + 2\sqrt{5 + 2\sqrt{10}} \right) && \approx 0.459878 \\ h_3 &= \frac{\sqrt{2}}{32} \left(10 - 2\sqrt{10} - 2\sqrt{5 + 2\sqrt{10}} \right) && \approx -0.135011 \\ h_4 &= \frac{\sqrt{2}}{32} \left(5 + \sqrt{10} - 3\sqrt{5 + 2\sqrt{10}} \right) && \approx -0.085441 \\ h_5 &= \frac{\sqrt{2}}{32} \left(1 + \sqrt{10} - \sqrt{5 + 2\sqrt{10}} \right) && \approx 0.035226 \end{aligned}$$

Das sind die Koeffizienten des Tiefpassfilters des Daubechies-Filterpaares D_6 .

8.3 Daubechies-Filter D_{2M}

Es sei nun $L = 2M - 1$. Die Ausführung der Ideen des vorigen Abschnitts für Filterpaare

$$\mathbf{h} = (h_0, h_1, \dots, h_L), \quad \mathbf{g} = (g_0, g_1, \dots, g_L),$$

wobei diese Filter durch

$$\boxed{g_j = (-1)^j h_{L-j} \quad (0 \leq j \leq L)} \quad ,$$

miteinander verbunden sind, liefert M *Orthogonalitätsbedingungen* für die Filterkoeffizienten:

$$(O_m) \quad \sum_{k=2m}^L h_k h_{k-2m} = \delta_{m,0} \quad (0 \leq m < M)$$

In der kompakten Scheibweise der Fourierreihen ist das für die Frequenzdarstellung des Filters

$$H(\omega) = \sum_{k=0}^L h_k e^{ik\omega}$$

die Bedingung

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2.$$

Dazu kommen M *Tiefpassbedingungen*, die mittels der Ableitungen der Fourierreihe $H(\omega)$ an der Stelle $\omega = \pi$ formuliert werden:

$$(T_m) \quad H^{(m)}(\pi) = 0 \quad (0 \leq m < M).$$

Als Bedingungen für die Filterkoeffizienten sind das die *Momentenbedingungen*

$$(T_m) \quad \sum_{k=0}^L (-1)^k k^m h_k = 0 \quad (0 \leq m < M)$$

Damit hat man $2M = L + 1$ Bedingungsgleichungen für die $L + 1$ Koeffizienten h_0, h_1, \dots, h_L , von denen die Orthogonalitätsbedingungen nichtlinear (quadratisch!) sind, die Momentenbedingungen aber linear. Damit ist gleichzeitig gesagt, dass sich das Erfüllen dieser Bedingungen nicht auf ein Problem der Linearen Algebra reduziert! Dann wäre das im folgenden Theorem ausgedrückte Resultat weit weniger bemerkenswert, als es in der Tat ist.

Es gilt übrigens immer (das ist wie in den Beispielen eine Konsequenz dieser Bedingungen):

$$H(0) = \sum_{k=0}^L h_k = \pm\sqrt{2}.$$

Man kann die Tiefpassbedingungen auch algebraisch formulieren, indem man das Polynom ("z-Transformierte")

$$h(z) = \sum_{k=0}^L h_k z^k$$

betrachtet, also $H(\omega) = h(e^{i\omega})$. Die Tiefpassbedingungen sind dann äquivalent zu der Aussage:

- $h(z)$ hat für $z = -1$ eine Nullstelle mit Vielfachheit $\geq M$.

Ebenfalls äquivalent ist:

- $h(z) = (z + 1)^M \cdot q(z)$ für ein Polynom $q(z)$ vom Grad $M - 1$.

Theorem 8.1 — Daubechies.

1. Das System der Orthogonalitätsbedingungen $(O_m)_{0 \leq m < M}$ und der Tiefpassbedingungen $(T_m)_{0 \leq m < M}$ für Filter der Länge $2M$ hat $2^{\lfloor (2M+1)/4 \rfloor}$ reelle Lösungen.

2. Genau eine (!) dieser Lösungen hat die Eigenschaft, dass für alle Nullstellen z_k des zugehörigen Polynoms $q(z)$ gilt: $|z_k| > 1$. Diese Lösung definiert den Daubechies-Tiefpassfilter D_{2M} .

Der Daubechies-Tiefpassfilter D_2 mit $\mathbf{h} = (h_0, h_1)$ wird durch die Bedingungen

$$h_0^2 + h_1^2 = 1, \quad h_0 - h_1 = 0$$

bestimmt. Es gilt also:

$$\mathbf{h} = (1/\sqrt{2}, 1/\sqrt{2}), \quad \mathbf{g} = (1/\sqrt{2}, -1/\sqrt{2}).$$

Dies ist also nichts anderes als das Haar-Filterpaar.

8.4 Daubechies-Polynome

Die konkrete Bestimmung der Daubechies-Filter ist nicht einfach. Dieser Abschnitt soll einen Eindruck von dem Vorgehen vermitteln. Wiederum ist $L = 2M - 1$. Gesucht wird ein Filter $\mathbf{h} = (h_0, h_1, \dots, h_L)$, für den die Orthogonalitätsbedingung

$$h(z) \cdot h\left(\frac{1}{z}\right) + h(-z) \cdot h\left(-\frac{1}{z}\right) = 2$$

erfüllt ist. Auf dem komplexen Einheitskreis, also für z mit $|z| = 1$, gilt $\bar{z} = 1/z$. Da die Filterkoeffizienten *reell* sein sollen, kann man diese Bedingung auch schreiben als

$$|h(z)|^2 + |h(-z)|^2 = 2 \quad \text{für } |z| = 1.$$

Die Tiefpassbedingungen besagen, dass sich $h(z)$ schreiben lässt als

$$h(z) = (1+z)^M \cdot q(z)$$

für ein Polynom $q(z)$ vom Grad $M - 1$ mit reellen Koeffizienten.

Zwei kleine kosmetische Modifikationen sind nützlich: ersten sollte man an Stelle von $h(z)$ das Polynom $h(z)/\sqrt{2}$ betrachten, um die "2" auf der rechten Seite der Orthogonalitätsbedingung durch eine "1" ersetzen zu können. Das so modifizierte Polynom soll dann in der Form

$$\tilde{h}(z) = \frac{1}{\sqrt{2}} h(z) = \left(\frac{1+z}{2}\right)^M \cdot q_{M-1}(z)$$

geschrieben werden, was auch nur das Verschieben eines konstanten Faktors 2^M bedeutet, aber die Nullstellen der beteiligten Polynome überhaupt nicht berührt. Der Index $M - 1$ des rechts stehenden Polynoms soll seinen Grad angeben. Das wird auch in der Folge so gehandhabt.

Für ein z auf dem komplexen Einheitskreis, also $z = e^{i\phi}$, gilt

$$\left|\frac{1+z}{2}\right|^2 = \frac{1+\cos\phi}{2} = 1 - \sin^2\frac{\phi}{2},$$

$$\left|\frac{1-z}{2}\right|^2 = \frac{1-\cos\phi}{2} = \sin^2\frac{\phi}{2}.$$

Die Gleichung

$$|\tilde{h}(z)|^2 + |\tilde{h}(-z)|^2 = 1$$

für $|z| = 1$ schreibt sich also auch:

$$\left(1 - \sin^2 \frac{\phi}{2}\right)^M \cdot |q_{M-1}(e^{i\phi})|^2 + \left(\sin^2 \frac{\phi}{2}\right)^M \cdot |q_{M-1}(-e^{i\phi})|^2 = 1.$$

Da $q_{M-1}(z)$ ein Polynom mit *reellen* Koeffizienten ist, ist $|q_{M-1}(e^{i\phi})|^2$ ein Polynom in $\cos \phi$, also auch ein Polynom $p_{M-1}(y)$ in $1 - \sin^2 \frac{\phi}{2}$, also auch ein Polynom in $y = \sin^2 \frac{\phi}{2}$. Für spätere Zwecke sei gleich notiert, dass zwischen der neuen Variablen $y = \sin^2 \frac{\phi}{2}$ und der ursprünglichen Variablen $z = e^{i\phi}$ der Zusammenhang

$$y = \frac{1}{2} - \frac{1}{4} \left(z + \frac{1}{z}\right)$$

besteht. Mit

$$|q_{M-1}(e^{i\phi})|^2 = p_{M-1}(y)$$

gilt dann auch

$$|q_{M-1}(-e^{i\phi})|^2 = p_{M-1}(1 - y).$$

Zusammengefasst: gesucht wird ein Polynom $p_{M-1}(y)$ mit den beiden Eigenschaften:

- $(1 - y)^M \cdot p_{M-1}(y) + y^M \cdot p_{M-1}(1 - y) = 1$,
- $p_{M-1}(y) \geq 0$ für $0 \leq y \leq 1$.

Definition 8.1 (Daubechies-Polynome).

Die Daubechies-Polynome $P_M(y)$ sind definiert durch

$$P_M(y) = \sum_{m=0}^M \binom{M+m}{m} y^m.$$

■ **Beispiel 8.1** Die ersten Daubechies-Polynome sind:

$$P_0(y) = 1$$

$$P_1(y) = 1 + 2y$$

$$P_2(y) = 1 + 3y + 6y^2$$

$$P_3(y) = 1 + 4y + 10y^2 + 20y^3$$

■

Lemma 8.1 Die Daubechies-Polynome lassen sich auch folgendermaßen schreiben:

$$P_M(y) = \sum_{k=0}^M \binom{2M+1}{k} y^k (1-y)^{M-k}$$

Beweis: Aus der Binomialformel ergibt sich für $m, N \geq 0$

$$\sum_{k=0}^N \binom{N}{k} y^{m+k} (1-y)^{N-k} = y^m.$$

Für $M \geq m$ ergibt sich mit $N = M - m$:

$$y^m = \sum_{k=0}^{M-m} \binom{M-m}{k} y^{m+k} (1-y)^{M-m-k} = \sum_{k=m}^M \binom{M-m}{k-m} y^k (1-y)^{M-k}$$

Daher ist

$$\begin{aligned} P_M(y) &= \sum_{m=0}^M \binom{M+m}{m} y^m \\ &= \sum_{m=0}^M \binom{M+m}{m} \sum_{k=m}^M \binom{M-m}{k-m} y^k (1-y)^{M-k} \\ &= \sum_{k_0}^M \sum_{m=0}^k \binom{M+m}{m} \binom{M-m}{k-m} y^k (1-y)^{M-k} \\ &= \sum_{k=0}^M \binom{2M+1}{k} y^k (1-y)^{M-k} \end{aligned}$$

wegen der bekannten Identität

$$\sum_{m=0}^k \binom{M+m}{m} \binom{M-m}{k-m} = \binom{2M+1}{k}.$$

Satz 8.1 Die Daubechies-Polynome erfüllen die Gleichung

$$(1-y)^{M+1} \cdot P_M(y) + y^{M+1} \cdot P_M(1-y) = 1.$$

Offensichtlich gilt $P_M(y) \geq 0$ für $0 \leq y \leq 1$.

Für den Beweis genügt eine einfache Anwendung der Binomialformel auf die Darstellung im vorigen Lemma:

$$\begin{aligned} &(1-y)^{M+1} \cdot P_M(y) + y^{M+1} \cdot P_M(1-y) \\ &= \sum_{k=0}^M \binom{2M+1}{k} y^k (1-y)^{2M+1-k} + \sum_{k=0}^M \binom{2M+1}{k} (1-y)^k y^{2M+1-k} \\ &= \sum_{k=0}^{2M-1} \binom{2M+1}{k} y^k (1-y)^{2M+1-k} = (y + (1-y))^{2M+1} = 1. \end{aligned}$$

Es sei nun

$$\widehat{P}_{2M-1}(z) = (1-y)^M \cdot P_{M-1}(y) = \sum_{m=-2M+1}^{2M-1} a_m z^m,$$

wobei – wie oben bemerkt – zwischen y und z der Zusammenhang $y = \frac{1}{2} - \frac{1}{4} \left(z + \frac{1}{z} \right)$ bestehen soll.

Satz 8.2 Für $\widehat{P}_{2M-1}(z)$ gelten folgende Aussagen:

1. Für $z \in \mathbb{C}_{\neq 0}$:

$$\widehat{P}_{2M-1}(z) + \widehat{P}_{2M-1}(-z) = 1.$$

2. Für $z \in \mathbb{C}_{\neq 0}$:

$$\widehat{P}_{2M-1}(z) = \widehat{P}_{2M-1}(1/z).$$

3. Für $z \in \mathbb{C}$ mit $|z| = 1$ ist $\widehat{P}_{2M-1}(z) \geq 0$.

$\widehat{P}_{2M-1}(z)$ ist ein “Polynom”, in dem Monome mit negativen Exponenten auftreten. Man macht daraus ein Polynom, indem man setzt:

$$\mathbf{P}_{4M-2}(z) = z^{2M-1} \cdot \widehat{P}_{2M-1}(z).$$

Satz 8.3 $\mathbf{P}_{4M-2}(z)$ hat $z = -1$ als $2M$ -fache Nullstelle und es gilt $\mathbf{P}_{4M-2}(1) = 1$. Ist $z_0 \in \mathbb{C}_{\neq 0}$ eine Nullstelle von $\mathbf{P}_{4M-2}(z)$, so sind auch \bar{z}_0 , $1/z_0$ und $1/\bar{z}_0$ Nullstellen von $\mathbf{P}_{4M-2}(z)$ von derselben Ordnung.

Die Aussage über die Vielfachheit der Nullstelle $z = -1$ ergibt sich aus

$$\begin{aligned} \mathbf{P}_{4M-2}(z) &= z^{2M-1} \widehat{P}_{2M-1}(z) \\ &= z^{2M-1} (1-y)^M P_{M-1}(z) \\ &= z^{2M-1} \left(\frac{1}{2} + \frac{1}{4} \left(z + \frac{1}{z} \right) \right)^M P_{M-1}(z) \\ &= z^{2M-1} \frac{1}{4^M} \frac{1}{z^M} (z+1)^{2M} P_{M-1}(z) \\ &= (z+1)^{2M} \cdot \frac{1}{4^M} z^{M-1} P_{M-1}(z), \end{aligned}$$

wobei $z^{M-1} P_{M-1}(y)$ ein Polynom (!) in z ist. Damit erweist sich $z = -1$ als $2M$ -fache Nullstelle von $\mathbf{P}_{4M-2}(z)$. \square

Nun beachte man:

– Ist $z_0 \neq 0$ *reell*, so gilt für $|z| = 1$:

$$|(z - z_0)(z - z_0^{-1})| = \frac{1}{|z_0|} |z - z_0|^2.$$

– Ist $z_0 \neq 0$ *nicht reell*, so gilt für $|z| = 1$:

$$|(z - z_0)(z - \bar{z}_0^{-1})(z - \bar{z}_0)(z - z_0^{-1})| = \frac{1}{|z_0|^2} |z - z_0|^2 |z - \bar{z}_0|^2.$$

Daraus ergibt sich nun das gesuchte Resultat:

Theorem 8.2 Es gibt ein reelles Polynom $\mathbf{Q}_{M-1}(z)$ mit

$$\mathbf{P}_{2M-1}(z) = \left| \frac{1+z}{2} \right|^{2M} \cdot |\mathbf{Q}_{M-1}(z)|^2.$$

Die Aussage muss man nur für $|z| = 1$ beweisen, dann gilt sie für alle komplexen z . Für $|z| = 1$ ergibt sie sich aus dem dem vorigen Satz durch Zusammenfassen der zueinander gehörenden Nullstellen und der vorangehenden Bemerkung.

8.5 Nochmals D_4 und D_6

8.5.1 D_4 mittels Daubechies-Polynom

Es ist

$$P_2(y) = 1 + 3y + 6y^2$$

und die Substitution ergibt

$$\mathbf{P}_6(z) = -\frac{1}{32} (-z^6 + 9z^4 + 16z^3 + 9z^2 - 1).$$

Das Faktorisieren führt zu

$$-\frac{1}{32} (z^2 - 4z + 1) (z + 1)^4$$

und zeigt $z = -1$ als 4-fache Nullstelle. Der quadratische Faktor hat die (reellen) Nullstellen $z = 2 \pm \sqrt{3}$. Mit $\alpha = 2 - \sqrt{3}$ erhält man

$$h(z) = \frac{1}{4} \frac{(z+1)^2 (z-2+\sqrt{3})}{1/2\sqrt{6} - 1/2\sqrt{2}}$$

und somit als Gleitkommaapproximation

$$h(z) \approx 0.4829629128 z^3 + 0.8365163032 z^2 + 0.224143868 z - 0.1294095223$$

8.5.2 D_6 mittels Daubechies-Polynom

Es ist

$$P_3(y) = 1 + 4y + 10y^2 + 20y^3$$

und die Substitution ergibt

$$\mathbf{P}_{10}(z) = \frac{1}{512} (3z^{10} - 25z^8 + 150z^6 + 256z^5 + 150z^4 - 25z^2 + 3)$$

Das Faktorisieren führt zu

$$\frac{1}{512} (3z^4 - 18z^3 + 38z^2 - 18z + 3) (z + 1)^6$$

und zeigt $z = -1$ als 6-fache Nullstelle. Der Faktor vom Grad 4 hat die Nullstellen

$$\begin{aligned} \alpha &= 0.2872513780 + 0.1528923339i, & \alpha^{-1} &= 2.712748622 - 1.443886783i, \\ \bar{\alpha} &= 0.2872513780 - 0.1528923339i, & \bar{\alpha}^{-1} &= 2.712748622 + 1.443886783i. \end{aligned}$$

Damit erhält man

$$\begin{aligned} h(z) &= \frac{\sqrt{3}}{16|\alpha|} \cdot (z+1)^3 \cdot (z-\alpha) \cdot (z-\bar{\alpha}) \\ &\approx 0.3326705530 z^5 + 0.8068915095 z^4 + 0.4598775023 z^3 \\ &\quad - 0.1350110200 z^2 - 0.08544127389 z + 0.03522629187. \end{aligned}$$

8.6 Nicht-kausale Filter

Jetzt sollen generell endliche Filter $\mathbf{h} = (h_\ell, h_{\ell+1}, \dots, h_L)$ mit $\ell < L$ und $h_\ell \leq 0 \leq h_L$, also mit $L - \ell + 1$ Filterelementen, in Betracht gezogen werden. Dabei soll (wegen dem downsampling mit dem Faktor 2) die Filterlänge gerade sein, $L - \ell + 1 = 2M$, es muss also $\ell \not\equiv L \pmod{2}$ gelten. Man bezeichnet ℓ als den *Startindex* und L als den *Stopindex* des Filters.

Die Formulierung von Orthogonalitäts- und Tiefpasseigenschaften geschieht wieder mit Hilfe von

$$h(z) = \sum_{k=\ell}^L h_k z^k \quad \text{bzw.} \quad H(\omega) = \sum_{k=\ell}^L h_k e^{i\omega} = h(e^{i\omega}).$$

1. Die *Orthogonalitätsbedingungen* schreiben sich nach wie vor als

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2,$$

oder äquivalent

$$\sum_{k=\ell+2m}^L h_k h_{k-2m} = \delta_{m,0} \quad (0 \leq m < M).$$

2. Ist $\mathbf{g} = (g_\ell, \dots, g_L)$ ein weiterer solcher Filter mit der Frequenzdarstellung $G(\omega)$, so schreibt sich *Orthogonalität von \mathbf{g} und \mathbf{h}* als

$$H(\omega) \cdot \overline{G(\omega)} + H(\omega + \pi) \cdot \overline{G(\omega + \pi)} = 0$$

oder äquivalent

$$\sum_{k=\ell+2m}^L h_k g_{k-2m} = 0 \quad (0 \leq m < M).$$

3. Ist ein orthogonaler Filter \mathbf{h} gegeben, so kann man einen Filter \mathbf{g} mittels

$$G(\omega) = e^{i(n\omega+b)} \overline{H(\omega + \pi)}$$

definieren, der dann automatisch orthogonal ist:

$$|G(\omega)|^2 + |G(\omega + \pi)|^2 = 2.$$

Ist n eine ungerade ganze Zahl (b eine beliebige reelle Zahl), so gilt zudem

$$H(\omega) \cdot \overline{G(\omega)} + H(\omega + \pi) \cdot \overline{G(\omega + \pi)} = 0.$$

die Rekonstruktionsbedingung ist ebenfalls erfüllt. Auf der Ebene der Filterkoeffizienten heisst das

$$g_k = -e^{ib}(-1)^k h_{n-k}.$$

Normalerweise wählt man $b = \pi$, so dass sich dies zu

$$g_k = (-1)^k h_{n-k}.$$

vereinfacht. Damit die Filterkoeffizienten von \mathbf{g} auch den Startindex ℓ und den Stopindex L haben, muss man $n = L + \ell$ wählen.

8.7 Coiflet-Filter

8.7.1 Idee und Konstruktion der Coiflet-Filter

Es ist eine naheliegende Überlegung, für die Konstruktion eines Tiefpassfilters $\mathbf{h} = (h_\ell, \dots, h_L)$ neben den Orthogonalitätsbedingungen

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2$$

und den Tiefpassbedingungen bei $\omega = \pi$, nämlich

$$H^{(m)}(\pi) = 0 \quad (m = 0, 1, 2, \dots),$$

auch Tiefpassbedingungen zu formulieren, die das Verhalten der Frequenzdarstellung $H(\omega)$ für $\omega = 0$ regeln. Die Bedingung

$$H(0) = \sqrt{2}$$

besteht sowieso. Nun kann man aber zusätzlich fordern, dass auch Bedingungen

$$H^{(m)}(0) = 0 \quad (m = 1, 2, \dots)$$

gelten sollen, die den Funktionsverlauf von $H(\omega)$ in der Nähe von $\omega = 0$ in dem Sinne beeinflussen, dass dort die Funktionswerte sehr nahe bei dem Wert $\sqrt{2} = H(0)$ sein sollen. Dies ist die Idee der *Coiflet-Filter*, die von R. COIFMAN suggeriert und von Ingrid DAUBECHIES realisiert wurde (siehe die zweite der weiter vorne zitierten Arbeiten).

Ausgangspunkt für die folgende Konstruktion von solchen Filtern sind wieder die Daubechies-Polynome

$$P_K(y) = \sum_{k=0}^K \binom{K+k}{k} y^k$$

mit ihrer charakteristischen Eigenschaft

$$(*) \quad (1-y)^K \cdot P_{K-1}(y) + y^K \cdot P_{K-1}(1-y) = 1.$$

Als Ansatz für die zu bestimmende Frequenzdarstellung wählt man

$$(**) \quad H(\omega) = \sqrt{2}(1-y)^K \cdot [P_{K-1}(y) + y^K \cdot A(e^{i\omega})] \Big|_{y \leftarrow \sin^2(\omega/2)},$$

wobei $A(z) = \sum_{k=0}^{2K-1} a_k z^k$ ein Polynom vom Grad $< 2K$ sein soll. Aus der charakteristischen Eigenschaft (*) ergibt sich die alternative Schreibweise:

$$(***) \quad H(\omega) = \sqrt{2} + \sqrt{2}y^K \cdot [-P_{K-1}(1-y) + (1-y)^K \cdot A(e^{i\omega})] \Big|_{y \leftarrow \sin^2(\omega/2)}.$$

- Beachtet man

$$(1 - y)^K \Big|_{y \leftarrow \sin^2(\omega/2)} = \cos^{2K}(\omega/2) = \left[\frac{1}{2} e^{-i\omega/2} (1 + e^{i\omega}) \right]^{2K},$$

so erkennt man an (**), dass $H(\omega)$ für $\omega = \pi$ eine $2K$ -fache Nullstelle hat:
 $H^m(\pi) = 0$ ($0 \leq m < 2K$).

- Beachtet man

$$y^M \Big|_{y \leftarrow \sin^2(\omega/2)} = \sin^{2M}(\omega/2) = \left[\frac{i}{2} e^{-i\omega/2} (1 - e^{i\omega}) \right]^{2M},$$

so erkennt man an (***), dass $H(\omega)$ für $\omega = 0$ eine $2M$ -fache Nullstelle hat:
 $H(0) = \sqrt{2}$ und $H^m(\pi) = 0$ ($1 \leq m < 2K$).

Diese Aussagen gelten für jedes Polynom $A(z)$. Der entscheidende Schritt liegt nun in folgender Aussage (schwieriger, deshalb hier ohne Beweis):

- Man kann die $2K$ Koeffizienten $a_0, a_1, \dots, a_{2K-1}$ für das Polynom $A(z)$ so wählen, dass die Orthogonalitätsbedingung

$$|H(\omega)|^2 + |H(\omega + \pi)|^2 = 2$$

erfüllt ist.

Nun muss man noch klären, wieviele Filterkoeffizienten der zugehörige Filter $\mathbf{h} = (h_\ell, \dots, h_L)$ hat, was der Startindex ℓ und der Stopindex L sind. Hierfür hat man die rechte Seite von (**) als ein Polynom in $z = e^{i\omega}$ zu schreiben. Dafür sei an die Beziehungen

$$y = \sin^2(\omega/2) = \frac{1}{4} \left(2 - z - \frac{1}{z} \right) \quad 1 - y = \cos^2(\omega/2) = \frac{1}{4} \left(2 + z + \frac{1}{z} \right)$$

erinnert. Eingesetzt in (**) ergibt sich

- $(1 - y)^K$ hat Terme z^k für $-K \leq k \leq K$;
- $P_{K-1}(1 - y)$ hat Terme z^k für $-K + 1 \leq k \leq K - 1$;
- y^K hat Terme z^k für $-K \leq k \leq K$;
- $A(e^{i\omega})$ hat Terme z^k für $0 \leq k \leq 2K - 1$.

Satz 8.4 Der durch (**) mit Parameter K definierte Filter $H(\omega)$ hat Startindex $\ell = -2K$ und Stopindex $L = 4K - 1$, er hat also die Länge $2M = 6K$.

Dieses $\mathbf{h} = (h_{-2K}, \dots, h_{4K-1})$ definiert den *Coiflet*-Filter C_{6K} .

Für die Berechnung von C_{2K} relevant sind also die

- Orthogonalitätsbedingungen

$$\sum_{k=-2K+2m}^{4K-1} h_k h_{k-2m} = \delta_{m,0} \quad (0 \leq m < 3K)$$

- Tiefpassbedingungen

$$\begin{aligned} H^{(m)}(0) &= 0 & (1 \leq m < 2K) & & H(0) &= \sqrt{2} \\ H^{(m)}(\pi) &= 0 & (0 \leq m < 2K) & & & \end{aligned}$$

Schliesslich kann man, wie im vorangehenden Abschnitt beschrieben, zu dem durch $H(\omega)$ gegebenen Tiefpassfilter mittels

$$G(\omega) = e^{in+b} \cdot \overline{H(\omega + \pi)}$$

einen komplementären Hochpassfilter zuordnen.

8.7.2 C_6 mittels Daubechies-Polynom

Im Fall $K = 1$ ist $A(z)$ ein Polynom Grad $2K - 1$. Der Ansatz für $H(\omega)$ bzw. $h(z)$ ist daher

$$\begin{aligned} h(z) &= \left(\frac{1}{2} + \frac{1}{4}z + \frac{1}{4}z^{-1} \right) \left(1 + \left(\frac{1}{2} - \frac{1}{4}z - \frac{1}{4}z^{-1} \right) (a_0 + a_1z) \right) \\ &= \left(-\frac{1}{16}a_0z^{-2} + \left(-\frac{1}{16}a_1 + \frac{1}{4} \right) z^{-1} + \frac{1}{8}a_0 + \frac{1}{2} + \left(\frac{1}{8}a_1 + \frac{1}{4} \right) z - \frac{1}{16}a_0z^2 - \frac{1}{16}a_1z^3 \right) \end{aligned}$$

Damit ist $\mathbf{h} = (h_{-2}, \dots, h_3)$ mit den noch zu bestimmenden Koeffizienten a_0, a_1 durch

$$\left[-\frac{1}{16}a_0\sqrt{2}, \left(-\frac{1}{16}a_1 + \frac{1}{4} \right) \sqrt{2}, \left(\frac{1}{8}a_0 + \frac{1}{2} \right) \sqrt{2}, \left(\frac{1}{8}a_1 + \frac{1}{4} \right) \sqrt{2}, -\frac{1}{16}a_0\sqrt{2}, -\frac{1}{16}a_1\sqrt{2} \right]$$

gegeben.

Die Orthogonalitätsbedingung $\sum_k h_k^2 = 1$ liefert:

$$3a_0^2 + 3a_1^2 + 4a_1 + 48 + 16a_0 = 64$$

Die Orthogonalitätsbedingung $\sum_k h_k h_{k+2} = 0$ liefert:

$$-a_0^2 - 4a_0 - a_1^2 + 4 = 0$$

Die Orthogonalitätsbedingung $\sum_k h_k h_{k+4} = 0$ liefert:

$$a_0^2 + a_1^2 - 4a_1 = 0$$

Die Lösung dieser drei Gleichungen ist gegeben durch

$$a_0 = 1 - \alpha, \quad a_1 = \alpha = \sqrt{1 - 6z + 2z^2},$$

was zu

$$h = \sqrt{2} \cdot \left[-\frac{1-\alpha}{16}, -\frac{\alpha}{16} + \frac{1}{4}, \frac{5}{8} - \frac{\alpha}{8}, \frac{\alpha}{8} + \frac{1}{4}, -\frac{1-\alpha}{16}, -\frac{\alpha}{16} \right]$$

und approximiert zu

$$\begin{bmatrix} h_{-2} & -0.07273261950 \\ h_{-1} & 0.3378976624 \\ h_0 & 0.8525720199 \\ h_1 & 0.3848648468 \\ h_2 & -0.07273261950 \\ h_3 & -0.01565572814 \end{bmatrix}$$

führt.

8.7.3 C_6 direkt

Da man Start- und Stopindices des Filters kennt ($\ell = -2$ und $L = 3$), kann man auch einfach $\mathbf{h} = (h_{-2}, \dots, h_3)$ mit unbestimmten Koeffizienten ansetzen und versuchen, die drei Orthogonalitätsbedingungen

$$\sum_k h_k^2 = 1 \quad \sum_k h_k h_{k+2} = 0 \quad \sum_k h_k h_{k+4} = 0$$

und die vier Tiefpassbedingungen

$$H(0) = \sqrt{2} \quad H(\pi) = 0 \quad H'(0) = 0 \quad H'(\pi) = 0$$

direkt zu lösen.

Man hat also folgende Bedingungsgleichungen

$$\begin{aligned} h_{-2}^2 + h_{-1}^2 + h_0^2 + h_1^2 + h_2^2 + h_3^2 &= 1 \\ h_{-2}h_0 + h_{-1}h_1 + h_0h_2 + h_1h_3 &= 0 \\ h_{-2}h_2 + h_{-1}h_3 &= 0 \\ h_{-2} + h_{-1} + h_0 + h_1 + h_2 + h_3 &= \sqrt{2} \\ h_{-2} - h_{-1} + h_0 - h_1 + h_2 - h_3 &= 0 \\ -2h_{-2} - h_{-1} + h_1 + 2h_2 + 3h_3 &= 0 \\ 2h_{-2} - h_{-1} + h_1 - 2h_2 + 3h_3 &= 0 \end{aligned}$$

Dies führt auf zwei Lösungen:

$$\begin{bmatrix} h_{-2} & \frac{1}{32} \sqrt{2} + \frac{1}{32} \sqrt{14} & \frac{1}{32} \sqrt{2} - \frac{1}{32} \sqrt{14} \\ h_{-1} & \frac{5}{32} \sqrt{2} - \frac{1}{32} \sqrt{14} & \frac{5}{32} \sqrt{2} + \frac{1}{32} \sqrt{14} \\ h_0 & \frac{7}{16} \sqrt{2} - \frac{1}{16} \sqrt{14} & \frac{7}{16} \sqrt{2} + \frac{1}{16} \sqrt{14} \\ h_1 & \frac{7}{16} \sqrt{2} + \frac{1}{16} \sqrt{14} & \frac{7}{16} \sqrt{2} - \frac{1}{16} \sqrt{14} \\ h_2 & \frac{1}{32} \sqrt{2} + \frac{1}{32} \sqrt{14} & \frac{1}{32} \sqrt{2} - \frac{1}{32} \sqrt{14} \\ h_3 & -\frac{3}{32} \sqrt{2} - \frac{1}{32} \sqrt{14} & -\frac{3}{32} \sqrt{2} + \frac{1}{32} \sqrt{14} \end{bmatrix}$$

und in Gleitkommaapproximation ist das

$$\begin{bmatrix} h_{-2} & 0.1611209671 & -0.07273261949 \\ h_{-1} & 0.1040440758 & 0.3378976624 \\ h_0 & 0.3848648467 & 0.8525720201 \\ h_1 & 0.8525720201 & 0.3848648467 \\ h_2 & 0.1611209671 & -0.07273261949 \\ h_3 & -0.2495093147 & -0.0156557281 \end{bmatrix}$$

8.7.4 C_{12} direkt

Im Fall $K = 2$ wird ein Filter mit Startindex $\ell = -4$ und Stopindex $L = 7$ gesucht. Ein Vorgehen wie im vorigen Abschnitt führt auf das System von Bedingungsgleichungen für die Filterkoeffizienten h_{-4}, \dots, h_7 :

Die Orthogonalitätsbedingungen:

$$\begin{aligned} h_{-4}^2 + h_{-3}^2 + h_{-2}^2 + h_{-1}^2 + h_0^2 + h_1^2 + h_2^2 + h_3^2 + h_4^2 + h_5^2 + h_6^2 + h_7^2 &= 1 \\ h_{-4}h_{-2} + h_{-3}h_{-1} + h_{-2}h_0 + h_{-1}h_1 + h_0h_2 + h_1h_3 + h_2h_4 + h_3h_5 + h_4h_6 + h_5h_7 &= 0 \\ h_{-4}h_0 + h_{-3}h_1 + h_{-2}h_2 + h_{-1}h_3 + h_0h_4 + h_1h_5 + h_2h_6 + h_3h_7 &= 0 \\ h_{-4}h_2 + h_{-3}h_3 + h_{-2}h_4 + h_{-1}h_5 + h_0h_6 + h_1h_7 &= 0 \\ h_{-4}h_4 + h_{-3}h_5 + h_{-2}h_6 + h_{-1}h_7 &= 0 \\ h_{-4}h_6 + h_{-3}h_7 &= 0 \end{aligned}$$

Die Tiefpassbedingungen:

$$\begin{aligned} h_{-4} + h_{-3} + h_{-2} + h_{-1} + h_0 + h_1 + h_2 + h_3 + h_4 + h_5 + h_6 + h_7 &= \sqrt{2} \\ h_{-4} - h_{-3} + h_{-2} - h_{-1} + h_0 - h_1 + h_2 - h_3 + h_4 - h_5 + h_6 - h_7 &= 0 \\ 4h_{-4} + 3h_{-3} + 2h_{-2} + h_{-1} - h_1 - 2h_2 - 3h_3 - 4h_4 - 5h_5 - 6h_6 - 7h_7 &= 0 \\ 4h_{-4} - 3h_{-3} + 2h_{-2} - h_{-1} + h_1 - 2h_2 + 3h_3 - 4h_4 + 5h_5 - 6h_6 + 7h_7 &= 0 \\ -16h_{-4} - 9h_{-3} - 4h_{-2} - h_{-1} - h_1 - 4h_2 - 9h_3 - 16h_4 - 25h_5 - 36h_6 - 49h_7 &= 0 \\ -16h_{-4} + 9h_{-3} - 4h_{-2} + h_{-1} + h_1 - 4h_2 + 9h_3 - 16h_4 + 25h_5 - 36h_6 + 49h_7 &= 0 \\ -64h_{-4} - 27h_{-3} - 8h_{-2} - h_{-1} + h_1 + 8h_2 + 27h_3 + 64h_4 + 125h_5 + 216h_6 + 343h_7 &= 0 \\ -64h_{-4} + 27h_{-3} - 8h_{-2} + h_{-1} - h_1 + 8h_2 - 27h_3 + 64h_4 - 125h_5 + 216h_6 - 343h_7 &= 0 \end{aligned}$$

Die 8 Tiefpassbedingungen sind *linear*, deshalb wird man sie zuerst benutzen, um 8 der 12 Variablen zu eliminieren:

$$\begin{aligned} h_{-4} &= 4h_6 + h_4 \\ h_{-3} &= h_5 + 4h_7 - 1/32\sqrt{2} \\ h_{-2} &= -15h_6 - 4h_4 \\ h_{-1} &= -4h_5 - 15h_7 + \frac{9}{32}\sqrt{2} \\ h_0 &= 20h_6 + 6h_4 + 1/2\sqrt{2} \\ h_1 &= 6h_5 + 20h_7 + \frac{9}{32}\sqrt{2} \\ h_2 &= -10h_6 - 4h_4 \\ h_3 &= -4h_5 - 10h_7 - 1/32\sqrt{2} \end{aligned}$$

Damit bleibt folgendes *nichtlineare* System zu lösen:

$$\begin{aligned}
& \frac{21\sqrt{2}}{16} h_5 + \frac{51\sqrt{2}}{16} h_7 + 20\sqrt{2} h_6 + 6\sqrt{2} h_4 + 448 h_5 h_7 \\
& \quad + 448 h_4 h_6 + 742 h_7^2 + 70 h_4^2 + 70 h_5^2 + 742 h_6^2 = \frac{23}{128} \\
& - \frac{3\sqrt{2}}{8} h_5 - \frac{7\sqrt{2}}{16} h_7 - \frac{25\sqrt{2}}{2} h_6 - 4\sqrt{2} h_4 \\
& \quad - 350 h_5 h_7 - 350 h_4 h_6 - 560 h_7^2 - 56 h_4^2 - 56 h_5^2 - 560 h_6^2 = \frac{63}{512} \\
& - \frac{5\sqrt{2}}{8} h_5 - \frac{15\sqrt{2}}{8} h_7 + 2\sqrt{2} h_6 + \sqrt{2} h_4 \\
& \quad + 160 h_5 h_7 + 160 h_4 h_6 + 220 h_7^2 + 28 h_4^2 + 28 h_5^2 + 220 h_6^2 = \frac{9}{256} \\
& - 20 h_6^2 - 35 h_4 h_6 - 8 h_4^2 - 8 h_5^2 - 35 h_5 h_7 + \frac{3\sqrt{2}}{8} h_5 - 20 h_7^2 + \frac{15\sqrt{2}}{32} h_7 + \frac{\sqrt{2}}{2} h_6 = \frac{1}{512} \\
& h_4^2 + h_5^2 - \frac{\sqrt{2}}{32} h_5 - 15 h_6^2 - 15 h_7^2 + \frac{9\sqrt{2}}{32} h_7 = 0 \\
& 4 h_6^2 + h_4 h_6 + h_5 h_7 + 4 h_7^2 - \frac{\sqrt{2}}{32} h_7 = 0
\end{aligned}$$

Die Lösung erhält man als

$$\begin{aligned}
h_4 &= -\frac{1}{1024} \frac{1430 \alpha^3 + 5064 \sqrt{2} \alpha^2 + 10441 \alpha + 2590 \sqrt{2}}{338 \alpha^2 + 962 \sqrt{2} \alpha + 1369} \\
h_5 &= \frac{1}{2048} \frac{1615 \sqrt{2} \alpha + 4081 + 65 \alpha^2}{26 \alpha + 37 \sqrt{2}} \\
h_6 &= \frac{1}{1024} \alpha \\
h_7 &= -\frac{1}{2048} \frac{179 \sqrt{2} \alpha + 405 + 21 \alpha^2}{26 \alpha + 37 \sqrt{2}}
\end{aligned}$$

wobei α eine Lösung der Gleichung

$$25Z^4 - 1082 \sqrt{2} Z^3 - 32180 Z^2 - 77370 \sqrt{2} Z - 102375 = 0$$

ist. Man wird also vier verschiedene Lösungen erhalten, die auf der folgenden Seite tabellarisch aufgeführt sind.

In Gleitkommadarstellung:

$$\begin{bmatrix} -0.00135879906 & -0.02881077935 & 0.01638733604 & -0.0216835830 \\ -0.01461155251 & 0.00954232518 & -0.04146493789 & -0.04759942451 \\ -0.0074103835 & 0.1131648994 & -0.06737255304 & 0.163253958 \\ 0.2806116518 & 0.1765268828 & 0.3861100713 & 0.3765105895 \\ 0.7503363057 & 0.5425549768 & 0.8127236327 & 0.2709267760 \\ 0.5704650013 & 0.7452653006 & 0.4170051772 & 0.5167479708 \\ -0.0716382822 & 0.1027738095 & -0.07648859743 & 0.5458520919 \\ -0.1553572228 & -0.2967882834 & -0.05943441354 & -0.2397210372 \\ 0.05002351996 & -0.02049790739 & 0.02368017155 & -0.3277620898 \\ 0.02480433052 & 0.07883524141 & 0.005611433291 & 0.1360266602 \\ -0.01284557976 & -0.002078217989 & -0.001823208878 & 0.07651962671 \\ 0.001194572696 & -0.006274685605 & -0.0007205493428 & -0.03485797772 \end{bmatrix}$$

Die dritte Spalte dieser Matrix ist der Filter, den man üblicherweise als Coiflet-Filter der Länge 12 bezeichnet.

9. Multiresolution (MRA)

Die Multiresolutionsanalyse (MRA), begründet 1988 von Stephane MALLAT in seiner Ph.D. thesis *Multiresolution and Wavelets* (University of Pennsylvania), ist ein eleganter theoretischer Rahmen für das Studium von Wavelets und Wavelettransformationen und wird heutzutage als *das* zentrale Konzept angesehen, das die verschiedensten Aspekte integriert. Im Zusammenhang mit dem Haar-System auf \mathbb{R} (Abschnitt 5.4) wurden die charakteristischen Eigenschaften schon angesprochen.

9.1 Schema der Multiresolution

Definition 9.1 (Multiresolution).

Eine MRA (multiresolution analysis) ist eine Familie $\{V_j\}_{j \in \mathbb{Z}}$ von Teilräumen von $\mathcal{L}^2(\mathbb{R})$ mit den Eigenschaften:

1. "genestet": $V_j \subseteq V_{j+1}$ ($j \in \mathbb{Z}$)
2. "dicht" : $\overline{\text{span}}\{V_j\}_{j \in \mathbb{Z}} = \mathcal{L}^2(\mathbb{R})$
3. "trennend": $\bigcap\{V_j\}_{j \in \mathbb{Z}} = \{0\}$
4. "skalierend": $f \in V_0 \Leftrightarrow D_{2^j}f \in V_j$ ($f \in \mathcal{L}^2(\mathbb{R}), j \in \mathbb{Z}$)
5. Es gibt eine Skalierungsfunktion $\phi \in V_0$, für die die Menge ihrer Translate

$$\{T_k\phi\}_{k \in \mathbb{Z}} = \{\phi(t - k); k \in \mathbb{Z}\}$$

eine vollständige ON-Basis von V_0 ist, d.h. $V_0 = \overline{\text{span}}\{T_k\phi\}_{k \in \mathbb{Z}}$.

Folgerung 9.1.

1. Die Teilräume $(V_j)_{j \in \mathbb{Z}}$ bilden eine Inklusionskette

$$\{0\} \subsetneq \cdots \subseteq V_{-2} \subseteq V_{-1} \subseteq V_0 \subseteq V_1 \subseteq V_2 \subseteq \cdots \nearrow \mathcal{L}^2(\mathbb{R})$$

2. Die Funktionen $\{\phi_{j,k}\}_{k \in \mathbb{Z}}$ mit

$$\phi_{j,k}(t) = 2^{j/2} \phi(2^j t - k) = (D_{2^j} T_k \phi)(t)$$

bilden eine vollständige ON-Basis des Teilraumes $V_j = \overline{\text{span}}\{\phi_{j,k}\}_{k \in \mathbb{Z}}$ ($j \in \mathbb{Z}$)

3. Wegen $V_0 \subseteq V_1$ gibt es eine ℓ^2 -Folge $(h_k)_{k \in \mathbb{Z}}$ mit

$$\phi(t) = \sum_k h_k \phi_{1,k}(t).$$

Diese Gleichung wird als Skalierungsgleichung bezeichnet, die Folge $(h_k)_{k \in \mathbb{Z}}$ als Skalierungsfiler.

Die Eigenschaften bezüglich V_0 und $\phi(t)$ übertragen sich zwanglos auf alle Skalierungsstufen. So gilt beispielsweise

$$V_0 \ni f(t) = \sum_{k \in \mathbb{Z}} f_k(T_k \phi)(t) \Leftrightarrow V_j \ni (D_{2^j} f)(t) = \sum_{k \in \mathbb{Z}} f_k(D_{2^j} T_k \phi(t))$$

oder

$$\langle \phi_{j,k}, \phi_{j,\ell} \rangle = 2^j \int_{\mathbb{R}} \phi(2^j t - k) \overline{\phi(2^j t - \ell)} dt = \int_{\mathbb{R}} \phi(t - k) \overline{\phi(t - \ell)} dt = \delta_{k,\ell}.$$

Aus der Skalierungsgleichung und der Orthogonalität folgt sofort:

$$h_k = \langle \phi, \phi_{1,k} \rangle = \sqrt{2} \int_{\mathbb{R}} \phi(t) \overline{\phi(2t - k)} dt$$

und es ist für alle $j, \ell \in \mathbb{Z}$:

$$\begin{aligned} \phi_{j,\ell}(t) &= 2^{j/2} \phi(2^j t - \ell) \\ &= 2^{j/2} \sum_k h_k \phi_{1,k}(2^j t - \ell) \\ &= 2^{(j+1)/2} \sum_k h_k \phi(2^{j+1} t - 2\ell - k) \\ &= \sum_k h_k \phi_{j+1, 2\ell+k}(t) \\ &= \sum_k h_{k-2\ell} \phi_{j+1,k}(t). \end{aligned}$$

Definition 9.2 (Wavelet zu einer MRA).

Zu der Skalierungsfunktion $\phi(t)$ einer MRA wird die zugehörige Waveletfunktion $\psi(t)$ definiert als

$$\psi(t) = \sum_k g_k \phi_{1,k}(t) \quad \text{mit} \quad g_k = (-1)^k \overline{h_{1-k}}$$

Die Folge $(g_k)_{k \in \mathbb{Z}}$ ist der zu der MRA gehörende Waveletfilter.

Theorem 9.1 — Struktur einer MRA.

1. Die Funktionen $\{\psi_{j,k}\}_{k \in \mathbb{Z}}$ mit

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) = (D_{2^j} T_k \psi)(t)$$

bilden eine vollständige ON-Basis des Raumes $W_j = \overline{\text{span}}\{\psi_{j,k}\}_{k \in \mathbb{Z}}$ ($j \in \mathbb{Z}$)

2. Für alle $j \in \mathbb{Z}$ gilt

$$V_{j+1} = W_j \oplus V_j.$$

3. Für alle $J \in \mathbb{Z}$ gilt

$$\mathcal{L}^2(\mathbb{R}) = \bigoplus_{j \in \mathbb{Z}} W_j = V_J \oplus \bigoplus_{j \geq J} W_j.$$

4. Die Funktionen $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ bilden eine vollständige ON-Basis von $\mathcal{L}^2(\mathbb{R})$.

- Man bezeichnet die Funktionen aus V_j als die Funktionen mit einer Auflösung $\geq 2^{-j}$ und die Funktionen aus W_j als die Funktionen mit einer Auflösung zwischen 2^{-j-1} und 2^{-j} .
- Bezeichnet man mit

$$P_j : \mathcal{L}^2(\mathbb{R}) \rightarrow V_j : f \mapsto \sum_k \langle f, \phi_{j,k} \rangle \phi_{j,k}$$

die *Approximationsprojektion* auf den Raum V_j , so ist

$$Q_j : \mathcal{L}^2(\mathbb{R}) \rightarrow W_j : f \mapsto \sum_k \langle f, \psi_{j,k} \rangle \psi_{j,k}$$

mit $Q_j = P_{j+1} - P_j$ die *Detailprojektion* auf den Teilraum W_j .

- Wie bei dem Haar-Wavelet gesehen gilt für alle $j > m$:

$$V_{j+1} = V_m \oplus W_m \oplus W_{m+1} \oplus \cdots \oplus W_j.$$

Aus den Forderungen ‘‘dicht’’ und ‘‘trennend’’ an eine MRA ergeben sich die Aussagen (bezüglich \mathcal{L}^2 -Konvergenz)

$$\lim_{j \rightarrow \infty} P_j f = f \quad \text{und} \quad \lim_{j \rightarrow -\infty} P_j f = 0.$$

9.2 Beispiele

9.2.1 Haar-MRA

Die Daten sind aus dem vierten Kapitel wohlbekannt. Man hat die Skalierungsfunktion

$$\phi(t) = \chi_{[0,1)}(t)$$

und für $j \in \mathbb{Z}$ beinhalten die genesteten Vektorräume

$$V_j = \overline{\text{span}}\{ \phi_{j,k}(t) \}_{k \in \mathbb{Z}} \subseteq \mathcal{L}^2(\mathbb{R})$$

die Treppenfunktionen mit Stufenbreite 2^{-j} , soweit sie zu $\mathcal{L}^2(\mathbb{R})$ gehören. $\{ \phi_{j,k}(t) \}_{k \in \mathbb{Z}}$ ist eine ON-Basis dieses Raumes. Bekanntlich ist

$$\lim_{j \rightarrow \infty} V_j = \mathcal{L}^2(\mathbb{R}),$$

d.h., Dichtheit ist gegeben. Die Trennungseigenschaft ergibt sich ganz einfach: eine Funktion

$$f \in \bigcap_{j \in \mathbb{Z}} V_j$$

muss auf beliebig langen Intervallen konstant sein. Dann kann sie aber nur zu $\mathcal{L}^2(\mathbb{R})$ gehören, wenn sie (fast überall) identisch verschwindet.

Die Dilatationskoeffizienten sind durch

$$h_0 = \frac{1}{\sqrt{2}}, \quad h_1 = \frac{1}{\sqrt{2}}, \quad h_k = 0 \quad (k \neq 0, 1)$$

und die Waveletkoeffizienten sind durch

$$g_0 = \frac{1}{\sqrt{2}}, \quad g_1 = -\frac{1}{\sqrt{2}}, \quad g_k = 0 \quad (k \neq 0, 1)$$

gegeben. Die Waveletfunktion ist

$$\psi(t) = \frac{1}{\sqrt{2}} (\phi_{0,0}(t) + \phi_{0,1}(t)) = \phi(2t) - \phi(2t - 1) = \chi_{[0,1/2)}(t) - \chi_{[1/2,1)}(t).$$

Die Fouriertransformierten von $\phi(t)$ und $\psi(t)$ finden sich bereits in Lemma 5.1 und Abbildung 5.4.

9.2.2 Shannon-MRA

Diese MRA ist durch Shannons Abtasttheorem motiviert:

$$V_0 = \{ f \in \mathcal{L}^2(\mathbb{R}); \widehat{f}(s) = 0 \text{ für } |s| > 1/2 \}$$

Dies sind also die 1-bandbeschränkten Funktionen. Die Skalierungsfunktion ist

$$\phi(t) = \frac{\sin(\pi t)}{\pi t}$$

mit ihrer Fouriertransformierten

$$\widehat{\phi}(s) = \chi_{[-1/2, 1/2)}(s).$$

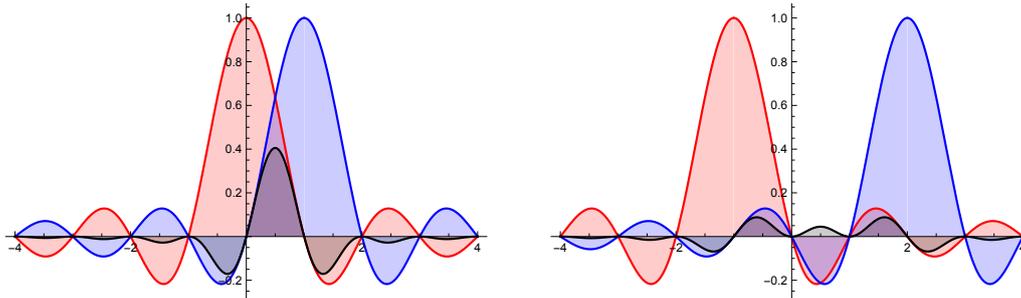


Abbildung 9.1: Orthogonalität der $T_k\phi(t)$ und $T_\ell\phi(t)$ für $(k, \ell) = (0, 1)$ und $(k, \ell) = (-1, 2)$. $T_k\phi$ in rot, $T_\ell\phi$ in blau, $T_k\phi \cdot T_\ell\phi$ in schwarz.

Aus den Eigenschaften der Fouriertransformation ergibt sich sofort, dass die Familie der Translate $\{T_k\phi(t)\}_{k \in \mathbb{Z}}$ wieder zu V_0 gehört. Man sieht leicht, dass diese Funktionen ein ONS in V_0 bilden. Das wird in der Abbildung 9.1 illustriert.

Man kann Shannons Abtasttheorem auch so verstehen:

$$V_0 = \overline{\text{span}}\{T_k\phi(t)\}_{k \in \mathbb{Z}}.$$

Die Shannon-Waveletfunktion ist

$$\psi(t) = \frac{\sin(2\pi t) - \cos(\pi t)}{\pi(t - 1/2)} = \frac{\sin(\pi(t - 1/2))}{\pi(t - 1/2)} (1 - 2 \sin(\pi t))$$

mit der Fouriertransformierten

$$\widehat{\psi}(s) = -e^{-i\pi s} (\chi_{[-1, -1/2)}(s) + \chi_{[1/2, 1)}(s)).$$

Man beachte: $\phi(t)$ und $\psi(t)$ sind analytische (unendlich-oft differenzierbare) Funktionen mit unendlicher Ausehnung; dagegen sind $\widehat{\phi}(t)$ und $\widehat{\psi}(t)$ Funktionen mit kompaktem Träger. Die Situation ist also genau umgekehrt im Vergleich zur Haar-MRA. Shannon-Skalierungsfunktion und Shannon-Waveletfunktion sind in Abbildung 9.2 dargestellt.

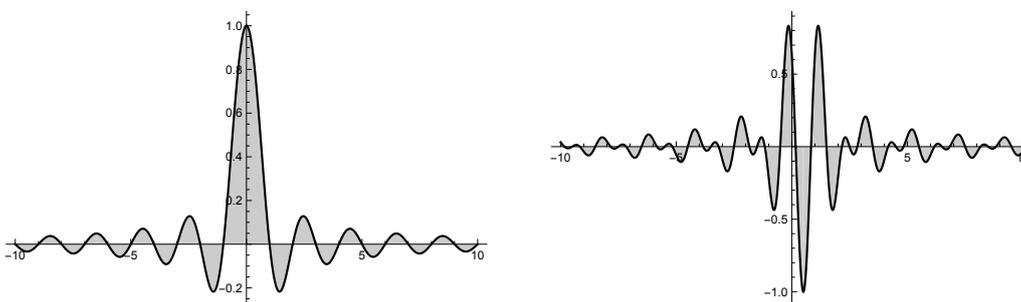


Abbildung 9.2: Shannon-Skalierungsfunktion und Shannon-Waveletfunktion

9.2.3 Stückweise lineare MRA

Eine interessante einfache Alternative zur Haar-MRA besteht darin, als Funktionen des Raumes V_0 nicht die (unstetigen) Treppenfunktionen zu nehmen, sondern stetige

\mathcal{L}^2 -Funktionen, die auf jedem Intervall der Länge 1 linear sind:

$$V_0 = \{ f \in \mathcal{L}^2(\mathbb{R}); f \text{ stetig und linear auf allen } I_{0,k} (k \in \mathbb{Z}) \},$$

und dann entsprechend für $j \in \mathbb{Z}$:

$$V_j = \{ f \in \mathcal{L}^2(\mathbb{R}); f \text{ stetig und linear auf allen } I_{j,k} (k \in \mathbb{Z}) \}.$$

Es ist leicht einzusehen, dass die Vektorräume $(V_j)_{j \in \mathbb{Z}}$ *genested* sind.

Da die *Dichtheit* gegeben ist, lässt sich ebenfalls recht einfach zeigen: man muss nur nachweisen, dass sich jede stetige Funktion mit kompaktem Träger mittels V_j -Funktionen für $j \rightarrow \infty$ gleichmässig approximieren lässt. Man kann das ganz explizit ausführen.

Die *Trennungseigenschaft* ist ebenfalls leicht einzusehen: eine \mathcal{L}^2 -Funktion $f \in \bigcap_{j \in \mathbb{Z}} V_j$ muss auf beliebig langen Intervallen linear sein. Das geht aber nur, wenn $f \equiv 0$ ist.

Die *Skalierungseigenschaft* ist Teil der Definition, darum muss man sich nicht kümmern.

Bleibt die Frage nach der Existenz einer *Skalierungsfunktion* $\phi(t) \in V_0$, deren Translate ganz V_0 erzeugen. Das ist der interessante Aspekt bei dieser MRA, denn der natürliche Kandidat

$$\phi(t) = (1 - |t|) \chi_{[-1,1]}(t)$$

hat zwar die gewünschten Erzeugungseigenschaften, aber leider sind die Translate $T_k \phi(t)$ nicht orthogonal zueinander:

$$\langle T_k \phi, T_\ell \phi \rangle = \begin{cases} 2/3 & \text{falls } k = \ell, \\ 1/6 & \text{falls } |k - \ell| = 1, \\ 0 & \text{sonst.} \end{cases}$$

Um die MRA-Eigenschaft nachzuweisen, muss man also zeigen, dass es eine andere Funktion $\tilde{\phi}(t) \in V_0$ gibt, deren Translate orthogonal zueinander sind und ganz V_0 im \mathcal{L}^2 -Sinne erzeugen. Das nachfolgend beschriebene Vorgehen lässt sich in viel allgemeinerem Rahmen als nur für dieses Beispiel durchführen. Zunächst die Erzeugungseigenschaft:

Lemma 9.1 Ist f eine stetige Funktion auf \mathbb{R} , die auf allen Intervallen $I_{0,k}$ linear ist, so gilt für jedes $t \in \mathbb{R}$:

$$f(t) = \sum_{k \in \mathbb{Z}} f(k) T_k \phi(t).$$

Das ist also eine Aussage für *punktweise Konvergenz*, deren Konvergenzeigenschaft völlig trivial ist, da für jedes $t \in \mathbb{R}$ nur zwei Summanden $\neq 0$ sind.

Lemma 9.2 Ist f eine stetige Funktion $\in \mathcal{L}(\mathbb{R})$, die auf allen Intervallen $I_{0,k}$ linear ist, so gilt in \mathcal{L}^2 -Konvergenz:

$$f(t) = \sum_{k \in \mathbb{Z}} f(k) T_k \phi(t).$$

Dies folgt aus der Abschätzung

$$\frac{1}{6} (|f(n)|^2 + |f(n+1)|^2) \leq \int_n^{n+1} |f(t)|^2 dt \leq \frac{1}{2} (|f(n)|^2 + |f(n+1)|^2)$$

für jede auf dem Intervall $[n, n+1)$ lineare Funktion.

Der Beweis der folgenden Aussage ist nun i.w. eine Routinesache:

Lemma 9.3 Es ist $V_0 = \overline{\text{span}}\{T_k \phi\}_{k \in \mathbb{Z}}$.

Eine geeignete Skalierungsfunktion $\tilde{\phi}(t)$ für diese MRA findet man durch Modifikation in der Fourierdarstellung.

Theorem 9.2 — Konstruktion eine ON-Basis.

Ist $\hat{\phi}(s)$ die Fouriertransformierte der Funktion $\phi(t)$ und die Funktion $\tilde{\phi}(t)$ definiert über ihrer Fouriertransformierte

$$\hat{\tilde{\phi}}(s) = \frac{\sqrt{3}}{\sqrt{1 + 2 \cos^2 \pi s}} \hat{\phi}(s),$$

so ist $\{T_k \tilde{\phi}\}_{k \in \mathbb{Z}}$ ist eine ON-Basis von V_0 .

Der Beweis dieses Theorems beruft sich auf die in einem folgenden Abschnitt (Theorem 9.3) gegebene Charakterisierung von sog. *Orthonormalen Translationssystemen* (ONTS). Dass die Translate von $\phi(t)$ kein ONTS bilden, sieht man im Lichte dieses Theorems an der Rechnung

$$\begin{aligned} \sum_{n \in \mathbb{Z}} \left| \hat{\phi}(s+n) \right|^2 &= \frac{1}{6} e^{-2\pi i s} + \frac{2}{3} + \frac{1}{6} e^{2\pi i s} \\ &= \frac{2}{3} + \frac{1}{3} \cos 2\pi s = \frac{1}{3} (1 + 2 \cos^2(\pi s)), \end{aligned}$$

und somit

$$\frac{1}{3} \leq \sum_{n \in \mathbb{Z}} \left| \hat{\phi}(s+n) \right|^2 \leq 1.$$

Die obige Modifikation der Fouriertransformierten leistet dann gerade das wegen Theorem 9.3 Gewünschte. Allerdings haben die Skalierungsfunktion $\tilde{\phi}(t)$ und die zugehörige Waveletfunktion keine einfache analytische Darstellung.

9.3 Approximation der Skalierungs- und Waveletfunktionen

Zu einem endlichen Filter $\mathbf{h} = (h_0, h_1, \dots, h_L)$, der den Orthogonalitätsbedingungen genügt, kann man eine *Skalierungsgleichung*

$$(S) \quad \phi(t) = \sum_{k=0}^L h_k \phi_{1,k}(t) = \sqrt{2} \sum_k h_k \phi(2t - k)$$

aufstellen und fragen, ob es eine “vernünftige” Skalierungsfunktion $\phi(t) : \mathbb{R} \rightarrow \mathbb{R}$ gibt, die dieser Gleichung genügt. Wenn es eine solche Funktion gibt, kann man nicht erwarten, dass sie durch einen mehr oder weniger komplizierten analytischen Ausdruck beschrieben wird. Man kann aber versuchen, diese Funktion $\phi(t)$ zu approximieren, um wenigstens eine grafische Anschauung zu erhalten.

Wenn man Näherungswerte für $\phi(t)$ hat, kann man die *Waveletgleichung*

$$(W) \quad \psi(t) = \sum_k g_k \phi_{1,k}(t) = \sqrt{2} \sum_{k=0}^L (-1)^{1-k} h_k \phi(2t + k - 1)$$

mit $g_k = (-1)^k h_{1-k}$ zur angenäherten Darstellung der Waveletfunktion $\psi(t)$ verwenden.

9.3.1 Der Kaskadenalgorithmus

Man kann die Gleichung *S* als eine *Fixpunktgleichung* für die zu bestimmende Funktion $\phi(t)$ auffassen und daraus ein iteratives Verfahren herleiten, bei dem Funktionen $\phi^{(n)}(t)$ für $n = 0, 1, 2, \dots$ berechnet werden:

- starte mit $\phi^{(0)}(t) = \chi_{[0,1)}(t)$;
- berechne $\phi^{(n+1)}(t)$ aus $\phi^{(n)}(t)$ mittels

$$\phi^{(n+1)}(t) = \sqrt{2} \sum_{k=0}^L h_k \phi^{(n)}(2t - k).$$

Man erwartet, dass es unter geeigneten Voraussetzungen die Folge $(\phi^{(n)}(t))$ $n \geq 0$ in der \mathcal{L}^2 -Norm gegen eine Funktion $\phi(t) \in \mathcal{L}^2(\mathbb{R})$ konvergiert. Das kann man in der Tat ohne allzu grosse Einschränkungen realisieren. Die Tatsache, dass der Filter \mathbf{h} endlich ist, führt dazu, dass die Funktion $\phi(t)$ einen kompakten Träger hat, d.h. ausserhalb des Intervalls $[0, L]$ identisch verschwindet. Die Waveletfunktion erhält man approximiert mittels

$$\psi^{(n)}(t) = \sqrt{2} \sum_k g_k \phi^{(n)}(t) \quad (n \geq 0).$$

Eine genauere Begründung (oder zumindestens Hinweise, wie es geht) dieses Verfahrens finden sich im Abschnitt 9.7.1 Grafische Beispiele dazu finden sich im im Abschnitt 9.11.

9.3.2 Dyadische Interpolation

Eine alternative Methode geht so vor, dass man zunächst die wahren Werte $\phi(t)$ für $t \in \{0, 1, 2, \dots, L\}$ aus der Skalierungsgleichung (S) bestimmt und dann stufenweise für $j = 1, 2, 3, \dots$ die wahren Werte $\phi(t)$ an den Stellen $2^{-j} \ell$ ($0 \leq \ell \leq 2^j L, \ell$ ungerade) aus (S) berechnet.

Die Gleichungen (S) für $t \in \{0, 1, 2, \dots, L\}$ sind

$$\begin{aligned}\phi(0) &= \sqrt{2} \cdot h_0 \phi(0) \\ \phi(1) &= \sqrt{2} \cdot (h_2 \phi(0) + h_1 \phi(1) + h_0 \phi(2)) \\ \phi(2) &= \sqrt{2} \cdot (h_4 \phi(0) + h_3 \phi(1) + h_2 \phi(2) + h_1 \phi(3) + h_0 \phi(4)) \\ &\vdots \\ \phi(L-1) &= \sqrt{2} \cdot (h_L \phi(L-2) + h_{L-1} \phi(L-1) + h_{L-2} \phi(L)) \\ \phi(L) &= \sqrt{2} \cdot h_L \phi(L)\end{aligned}$$

Die erste und die letzte dieser Gleichungen sind durch die Forderung $\phi(0) = \phi(L) = 0$ erfüllt. Die verbleibenden $L-1$ Gleichungen betrachtet man als Eigenwertgleichung:

$$\begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(L-2) \\ \phi(L-1) \end{bmatrix} = \sqrt{2} \cdot \begin{bmatrix} h_1 & h_0 & 0 & 0 & \dots & 0 \\ h_3 & h_2 & h_1 & h_0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & h_L & h_{L-1} & h_{L-2} & h_{L-3} \\ 0 & \dots & 0 & 0 & h_L & h_{L-1} \end{bmatrix} \cdot \begin{bmatrix} \phi(1) \\ \phi(2) \\ \vdots \\ \phi(L-2) \\ \phi(L-1) \end{bmatrix},$$

aus der man $\phi(1), \phi(2), \dots, \phi(L-1)$ exakt bestimmt. Dann interpoliert man mittels (S) für $j = 1, 2, \dots$:

$$\phi(2^{-j} \ell) = \sqrt{2} \sum_k h_k \phi(2^{-j+1} \ell - k) \quad (0 \leq \ell \leq 2^j L, \ell \text{ ungerade}).$$

Grafische Beispiele für diese Verfahren finden sich im Abschnitt 9.11 und auch im Anhang im Abschnitt 16.1.

9.4 Beweis des Theorems 9.1

Satz 9.1 — Eigenschaften des Skalierungsfilters.

Ist $(V_j)_{j \in \mathbb{Z}}$ eine MRA mit Skalierungsfunktion $\phi(t)$ und Skalierungsfilter $(h_k)_{k \in \mathbb{Z}}$, so gelten:

1. $\sum_k h_{k-2\ell} \overline{h_k} = \delta_{\ell,0}$.
2. $\sum_k |h_k|^2 = 1$.
3. $\sum_k h_k = \sqrt{2}$.
4. $\sum_k h_{2k} = \sum_k h_{2k+1} = 1/\sqrt{2}$.

Zum Beweis: Aus der Skalierung hat man

$$\phi(t - \ell) = \sum_k h_{k-2\ell} \phi_{1,k}(t)$$

und wegen der Orthogonalität der Translate von $\phi(t)$ folgt:

$$\delta_{\ell,0} = \langle \phi(t - \ell), \phi(t) \rangle = \sum_k h_{k-2\ell} \langle \phi_{1,k}(t), \phi(t) \rangle = \sum_k h_{k-2\ell} \overline{h_k}.$$

Dies ist Aussage 1., Aussage 2. ist nur der Sonderfall $\ell = 0$. Weiter ist für $\ell \neq 0$

$$\begin{aligned} \int_{\mathbb{R}} \phi(t) dt &= \int_{\mathbb{R}} \phi(t - \ell) dt = \sum_k h_{k-2\ell} \int_{\mathbb{R}} \phi_{1,k}(t) dt \\ &= \sum_k h_{k-2\ell} \sqrt{2} \int_{\mathbb{R}} \phi(2t - k) dt = \sum_k h_{k-2\ell} \frac{1}{\sqrt{2}} \int_{\mathbb{R}} \phi(t) dt. \end{aligned}$$

Später wird gezeigt, dass immer $|\int_{\mathbb{R}} \phi(t) dt| = 1$ gilt, also insbesondere $\int_{\mathbb{R}} \phi(t) dt \neq 0$. Man kann also das Integral auf beiden Seiten auskürzen und erhält Aussage 3.

Für $\alpha = \sum_k h_{2k}$ und $\beta = \sum_k h_{2k+1}$ gilt zunächst einmal $\alpha + \beta = \sqrt{2}$ wegen 1. Weiter ist

$$|\alpha|^2 + |\beta|^2 = \sum_{k,\ell} h_{2k} \overline{h_{2\ell}} + \sum_{k,\ell} h_{2k+1} \overline{h_{2\ell+1}} = \sum_{k,\ell} h_k \overline{h_{k+2\ell}} = 1$$

Aus den beiden Bedingungen für α und β folgt $|\sqrt{2}\alpha - 1|^2 = 0$ mit den beiden möglichen Lösungen $\alpha = 1/\sqrt{2}$ oder $\overline{\alpha} = 1/\sqrt{2}$. Bei reellen Filterkoeffizienten h_k muss die erste Alternative zutreffen und somit ergibt sich Aussage 4. \square

Satz 9.2 — Eigenschaften des Waveletfilters.

Ist $(V_j)_{j \in \mathbb{Z}}$ eine MRA mit Skalierungsfunktion $\phi(t)$ und Skalierungsfilter $(h_k)_{k \in \mathbb{Z}}$ und definiert man die Koeffizienten des zugehörigen Waveletfilters $(g_k)_{k \in \mathbb{Z}}$ mittels $g_k = (-1)^k \overline{h_{1-k}}$, so gelten:

1. $\sum_k g_{k-2\ell} \overline{g_k} = \delta_{\ell,0}$.
2. $\sum_k |g_k|^2 = 1$.
3. $\sum_k g_k = 0$.
4. $\sum_k g_{2k} = -\sum_k g_{2k+1} = 1/\sqrt{2}$.

Ausserdem gelten folgende Beziehungen zwischen den Filtern:

5. $\sum_k g_{k-2\ell} \overline{h_k} = 0$.
6. $\sum_k (\overline{h_{m-2k}} h_{n-2k} + \overline{g_{m-2k}} g_{n-2k}) = \delta_{m,n}$

Die Aussagen 1.-4. folgen aus den entsprechenden Aussagen des vorigen Satzes durch Einsetzen der Definition für die g_k .

Aussage 5. ergibt sich aus

$$\sum_k \overline{g_{k-2\ell}} h_k = \sum_k (-1)^{k-2\ell} \overline{h_{1-k+2\ell}} h_k = \sum_k (-1)^k \overline{h_{1-k+2\ell}} h_k = 0,$$

denn in der Summe heben sich die beiden Summanden für k und für $1 - k + 2\ell$ weg.

Für Aussage 6. schreibt man

$$\sum_k (\overline{h_{m-2k}} h_{n-2k} + \overline{g_{m-2k}} g_{n-2k}) = (-1)^{m+n} \sum_k \overline{h_{m-2k}} h_{n-2k} + \sum_{k'} h_{1-m+2k'} \overline{h_{1-n+2k'}}.$$

Ist $m+n$ ungerade, so heben sich die Summenden für $m-2k$ in der ersten Summe und für $1-n+2k'$ in der zweiten Summe weg.

Ist $m+n$ gerade, so ergibt sich, wenn man in der zweiten Summe $-n+2k'$ durch $m-2k$ ersetzt:

$$\begin{aligned} & \sum_k (\overline{h_{m-2k}} h_{n-2k} + \overline{g_{m-2k}} g_{n-2k}) \\ &= \sum_k (\overline{h_{m-2k}} h_{n-2k} + \overline{h_{m-2k+1}} g_{n-2k+1}) = \sum_k \overline{h_{m-k}} h_{n-k} = \sum_k \overline{h_k} h_{(m-n)+k} = \delta_{m,n} \end{aligned}$$

□

Folgerung 9.2. *Unter den Bedingungen des vorigen Satzes gilt für die Waveletfunktion $\psi(t) = \sum_k g_k \phi_{1,k}(t)$:*

1. Für $j \in \mathbb{Z}$ bilden die Funktionen $\{\psi_{j,k}\}_{k \in \mathbb{Z}}$ eine orthonormale Familie von Funktionen.
2. Für $j \in \mathbb{Z}$ sind die orthonormalen Familien von Funktionen $\{\psi_{j,k}\}_{k \in \mathbb{Z}}$ und $\{\phi_{j,k}\}_{k \in \mathbb{Z}}$ orthogonal zueinander, d.h. $W_j \perp V_j$.
3. Es gilt $V_1 = V_0 \oplus W_0$ und somit allgemein $V_{j+1} = V_j \oplus W_j$.
4. Für $j \neq j'$ gilt $W_j \perp W_{j'}$. Somit ist $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ eine orthogonale Familie von Funktionen.

Zum Beweis: aus der Definition von $\psi(t)$ ergibt sich zunächst einmal $g_k = \langle \psi, \phi_{1,k} \rangle$ und damit

$$\begin{aligned} \langle \psi_{j,m}, \psi_{j,n} \rangle &= 2^j \int_{\mathbb{R}} \psi(2^j t - m) \overline{\psi(2^j t - n)} dt \\ &= \int_{\mathbb{R}} \psi(t - m) \overline{\psi(t - n)} dt \\ &= \sum_{k,\ell} g_k \overline{g_\ell} \langle \phi_{1,k}(t - m), \phi_{1,\ell}(t - n) \rangle \\ &= \sum_{k,\ell} g_k \overline{g_\ell} \langle \phi_{1,k-2m}(t), \phi_{1,\ell-2n}(t) \rangle \\ &= \sum_{k,\ell} g_k \overline{g_\ell} \delta_{k-2m,\ell-2n} \\ &= \sum_k g_k \overline{g_{k-2(m-n)}} = \delta_{m,n} \end{aligned}$$

Eine ganz analoge Rechnung beweist Behauptung 2.:

$$\begin{aligned}
\langle \psi_{j,m}, \phi_{j,n} \rangle &= 2^j \int_{\mathbb{R}} \psi(2^j t - m) \overline{\phi(2^j t - n)} dt \\
&= \int_{\mathbb{R}} \psi(t - m) \overline{\phi(t - n)} dt \\
&= \sum_{k,\ell} g_k \overline{h_\ell} \langle \phi_{1,k}(t - m), \phi_{1,\ell}(t - n) \rangle \\
&= \sum_{k,\ell} g_k \overline{g_\ell} \langle \phi_{1,k-2m}(t), \phi_{1,\ell-2n}(t) \rangle \\
&= \sum_{k,\ell} g_k \overline{h_\ell} \delta_{k-2m,\ell-2n} \\
&= \sum_k g_k \overline{h_{k-2(m-n)}} = 0.
\end{aligned}$$

Für Behauptung 3. hat man zu zeigen, dass $\phi_{1,n} \in \overline{\text{span}}(W_0 \cup V_0)$ gilt, d.h. dass es Koeffizienten $(\alpha_k)_{k \in \mathbb{Z}}, (\beta_k)_{k \in \mathbb{Z}}$ gibt mit

$$\phi_{1,n} = \sum_k \alpha_k \phi_{0,k} + \sum_k \beta_k \psi_{0,k}.$$

Wenn diese Beziehung besteht, muss wegen der Orthogonalität

$$\alpha_k = \langle \phi_{1,n}, \phi_{0,k} \rangle = \overline{h_{n-2k}}, \quad \beta_k = \langle \phi_{1,n}, \psi_{0,k} \rangle = \overline{g_{n-2k}}$$

gelten. Mit dieser Wahl gilt aber wegen $\phi_{0,k} = \sum_m h_{m-2k} \phi_{1,m}$ und $\psi_{0,k} = \sum_m g_{m-2k} \phi_{1,m}$ tatsächlich

$$\begin{aligned}
\sum_k \alpha_k \phi_{0,k} + \sum_k \beta_k \psi_{0,k} &= \sum_k \alpha_k \sum_m h_{m-2k} \phi_{1,m} + \sum_k \beta_k \sum_m g_{m-2k} \phi_{1,m} \\
&= \sum_m \underbrace{\left(\sum_k (\overline{h_{n-2k}} h_{m-2k} + \overline{g_{n-2k}} g_{m-2k}) \right)}_{\delta_{m,n}} \phi_{1,m} = \phi_{1,n}
\end{aligned}$$

Schliesslich sind die Räume W_j und $W_{j'}$ für $j \neq j'$ orthogonal zueinander. Es genügt, dies am Beispiel von W_0 und W_1 nachzuweisen. Das ist aber unmittelbare Folge von $V_1 = V_0 \oplus W_0$ und der Tatsache $W_1 \perp V_1$. \square

9.5 Orthonormale Translationssysteme

Definition 9.3 (Orthonormales Translationssystem).

Eine Funktion $\phi(t) \in \mathcal{L}^2(\mathbb{R})$ definiert ein orthonormales Translationssystem (ONTS), falls die Familie ihrer Translate

$$\{(T_k\phi)(t)\}_{k \in \mathbb{Z}} = \{\phi(t - k)\}_{k \in \mathbb{Z}}$$

ein Orthonormalsystem in $\mathcal{L}^2(\mathbb{R})$ ist.

Theorem 9.3 — Charakterisierung von ONTS.

Die Familie $\{T_k\phi\}_{k \in \mathbb{Z}}$ ist genau dann ein ONTS, wenn

$$\sum_{n \in \mathbb{Z}} \left| \widehat{\phi}(s + n) \right|^2 = 1,$$

d.h., die Summe hängt nicht von $s \in \mathbb{R}$ ab, sondern ist konstant = 1.

Zum Beweis benutzt man die Translationseigenschaft der Fouriertransformation:

$$\widehat{T_k\phi}(s) = e^{-2\pi i k s} \widehat{\phi}(s).$$

Damit folgt für die Skalarprodukte mit $\phi(t)$:

$$\begin{aligned} \langle \phi | T_k\phi \rangle &= \langle \widehat{\phi} | \widehat{T_k\phi} \rangle \\ &= \int_{\mathbb{R}} \widehat{\phi}(s) \overline{\widehat{\phi}(s)} e^{2\pi i k s} ds \\ &= \sum_{n \in \mathbb{Z}} \int_n^{n+1} \left| \widehat{\phi}(s) \right|^2 e^{2\pi i k s} ds \\ &= \sum_{n \in \mathbb{Z}} \int_0^1 \left| \widehat{\phi}(s + n) \right|^2 e^{2\pi i k s} ds \\ &= \int_0^1 \sum_{n \in \mathbb{Z}} \left| \widehat{\phi}(s + n) \right|^2 e^{2\pi i k s} ds. \end{aligned}$$

Liest man die Folge $(\langle \phi | T_k\phi \rangle)_{k \in \mathbb{Z}}$ als die Koeffizienten einer Fourierreihe, so besagt dies gerade

$$\sum_{k \in \mathbb{Z}} \langle \phi | T_k\phi \rangle e^{-2\pi i k s} = \sum_{n \in \mathbb{Z}} \left| \widehat{\phi}(s + n) \right|^2.$$

Daraus ergibt sich die Behauptung. \square

Aus der letzten Gleichung im vorigen Beweis ergibt sich sofort: hat die Skalierungsfunktion $\phi(t)$ einen kompakten Träger, so sind nur endlich-viele der Koeffizienten $\langle \phi | T_k\phi \rangle$ in der Summe auf der linken Seite $\neq 0$. Es handelt sich bei der Summe also um ein trigonometrisches Polynom.

Eine weitere Beobachtung zu dem Beweis ist diese: setzt man in dem ersten Ausdruck $\langle \phi | T_k \phi \rangle$ der Gleichungskette die Funktion ϕ links durch eine Funktion ψ , so erhält man ganz entsprechend

$$\sum_{k \in \mathbb{Z}} \langle \psi | T_k \phi \rangle e^{-2\pi i k s} = \sum_{n \in \mathbb{Z}} \widehat{\psi}(s+n) \overline{\widehat{\phi}(s+n)}$$

und somit ein Kriterium dafür, wann ψ orthogonal zu allen $T_k \phi$ ist.

Theorem 9.4 — Charakterisierung von V_0 .

Ist $\{T_k \phi\}_{k \in \mathbb{Z}}$ ein ONTS, so gilt

$$f \in V_0 \Leftrightarrow \exists (c[n])_{n \in \mathbb{Z}} \in \ell^2 \text{ mit } \widehat{f}(s) = \widehat{\phi}(s) \cdot \left(\sum_{n \in \mathbb{Z}} c[n] e^{-2\pi i n s} \right).$$

Beweisskizze: Für $f \in V_0$ ist

$$f(t) = \sum_{n \in \mathbb{Z}} \langle f | T_n \phi \rangle T_n \phi(t).$$

Daraus erhält man mittels Fouriertransformation

$$\begin{aligned} \widehat{f}(s) &= \sum_{n \in \mathbb{Z}} \langle f | T_n \phi \rangle \widehat{T_n \phi}(s) \\ &= \widehat{\phi}(s) \sum_{n \in \mathbb{Z}} \langle f | T_n \phi \rangle e^{-2\pi i n s}. \end{aligned}$$

Bessels Ungleichung

$$\sum_{n \in \mathbb{Z}} |\langle f | T_n \phi \rangle|^2 \leq \|f\|_2^2 < \infty$$

zeigt, dass die Folge $(c[n])_{n \in \mathbb{Z}}$ mit $c[n] = \langle f, T_n \phi \rangle$ das Gewünschte leistet.

Gilt umgekehrt

$$\widehat{f}(s) = \widehat{\phi}(s) \left(\sum_{n \in \mathbb{Z}} c[n] e^{-2\pi i n s} \right),$$

so kann man für $N \in \mathbb{N}$ die Funktionen

$$f_N(t) = \sum_{|n| \leq N} c[n] T_n \phi(t) \in \text{span}\{T_n \phi\}_{n \leq N}$$

definieren. Fouriertransformation liefert

$$\widehat{f}_N(s) = \widehat{\phi}(s) \left(\sum_{|n| \leq N} c[n] e^{-2\pi i n s} \right).$$

Mit Hilfe der Formel von Plancherel kann man dann zeigen, dass $f_N(t) \rightarrow f(t)$ in der \mathcal{L}^2 -Norm gilt und dass damit auch $f(t) \in \overline{\text{span}}\{T_n \phi\}_{n \in \mathbb{N}}$ gilt. \square

Theorem 9.5 — ONTS-Konstruktion.

Ist $\phi(t) \in \mathcal{L}(\mathbb{R})$ mit kompaktem Träger und gibt es Konstanten A, B mit

$$0 < A < \sum_{n \in \mathbb{Z}} \left| \widehat{\phi}(s+n) \right|^2 \leq B$$

so gibt es ein $\widetilde{\phi}(t) \in \mathcal{L}(\mathbb{R})$, dessen Translate $\left\{ T_k \widetilde{\phi} \right\}_{k \in \mathbb{Z}}$ ein ONTS bilden, das den gleichen Raum V_0 wie $\left\{ T_k \phi \right\}_{k \in \mathbb{Z}}$ erzeugt.

Beweisskizze: Wenn $\phi(t)$ einen kompakten Träger hat, ist, wie oben bemerkt, $\sum_{n \in \mathbb{Z}} \left| \widehat{\phi}(s+n) \right|^2$ ein trigonometrisches Polynom, das nirgends verschwindet. Man kann also unbedenklich die Funktion

$$\Phi(s) = \left(\sum_{n \in \mathbb{Z}} \left| \widehat{\phi}(s+n) \right|^2 \right)^{-1/2}$$

definieren, die 1-periodisch und unendlich-oft differenzierbar ist. Sie besitzt eine Darstellung als Fourierreihe

$$\Phi(s) = \sum_{n \in \mathbb{N}} c[n] e^{-2\pi i n s}$$

mit $\sum_n |c_n|^2 < \infty$. Man definiert nun die Funktion $\widetilde{\phi}(t)$ vermöge ihrer Fouriertransformierten:

$$\widehat{\widetilde{\phi}}(s) = \widehat{\phi}(s) \cdot \Phi(s),$$

also

$$\widetilde{\phi}(t) = \sum_n c[n] \phi(t-n).$$

Man kann zeigen, dass $\widehat{\widetilde{\phi}}(s)$ und $\widetilde{\phi}(t)$ zu $\mathcal{L}^2(\mathbb{R})$ gehören. Natürlich ist

$$\left| \widehat{\widetilde{\phi}}(s+k) \right|^2 = \left| \widehat{\phi}(s+k) \right|^2 \Phi^2(s+k) = \left| \widehat{\phi}(s+k) \right|^2 \Phi(s)^2$$

und somit

$$\sum_k \left| \widehat{\widetilde{\phi}}(s+k) \right|^2 = \Phi(s)^2 \sum_k \left| \widehat{\phi}(s+k) \right|^2 = 1$$

nach Definition von $\Phi(s)$. Damit ist das ONTS-Kriterium von Theorem 9.3 erfüllt.

Es ist leicht zu sehen, dass die Familien $\left\{ T_k \widetilde{\phi} \right\}_{k \in \mathbb{Z}}$ und $\left\{ T_k \phi \right\}_{k \in \mathbb{Z}}$ den gleichen linearen Raum erzeugen: wegen $\widetilde{\phi}(t) = \sum_n c[n] \phi(t-n)$ gilt eine Inklusion offensichtlich, und indem man die Rollen von $\phi(t)$ und $\widetilde{\phi}(t)$ vertauscht, erhält man die andere Inklusion. \square

9.6 Weitere Eigenschaften von MRAs (I)

Theorem 9.6 — Eigenschaften von Skalierung und Wavelet einer MRA.

Die Skalierungsfunktion $\phi(t)$ und die Waveletfunktion $\psi(t)$ einer MRA haben folgende Eigenschaften:

1. $|\widehat{\phi}(0)| = \left| \int_{\mathbb{R}} \phi(t) dt \right| = 1$.
2. Für $n \in \mathbb{Z} \setminus \{0\}$ ist $\widehat{\phi}(n) = 0$.
3. $\sum_{n \in \mathbb{Z}} \phi(t+n) \equiv 1$.
4. $\widehat{\psi}(0) = \int_{\mathbb{R}} \psi(t) dt = 0$.

Der *Beweis* von Punkt 1. ist technischer Natur. Sei dafür $f \in \mathcal{L}^2(\mathbb{R})$ eine \mathcal{L}^2 -normierte und “genügend gutartige” Funktion. Konkret soll f $2R$ -bandbegrenzt sein und \widehat{f} soll stetig sein. Für festes $j \in \mathbb{Z}$ gilt dann unter Berücksichtigung der Translationseigenschaft der Fouriertransformation:

$$\begin{aligned} \|P_j f\|_2^2 &= \sum_{k \in \mathbb{Z}} |\langle f | \phi_{j,k} \rangle|^2 \\ &= \sum_{k \in \mathbb{Z}} |\langle \widehat{f} | \widehat{\phi}_{j,k} \rangle|^2 \\ &= \sum_{k \in \mathbb{Z}} \left| \int_{\mathbb{R}} \widehat{f}(s) 2^{-j/2} \widehat{\phi}(2^{-j}s) e^{-2\pi i k 2^{-j}s} ds \right|^2. \end{aligned}$$

Die Funktionen $2^{-j/2} e^{-2\pi i k 2^{-j}s}$ ($k \in \mathbb{Z}$) bilden ein vollständiges ONS in $\mathcal{L}^2(-2^{j-1}, 2^{j-1})$, bezüglich dem die letzte Summe eine Quadratsumme von Fourierkoeffizienten ist. Ist nun noch $2^{j-1} > R$, so folgt aus der Plancherelformel:

$$\|P_j f\|_2^2 = \dots = \int_{-R}^R |\widehat{f}(s)|^2 \cdot |\widehat{\phi}(2^{-j}s)|^2 ds.$$

Dabei gilt $\lim_{j \rightarrow \infty} \widehat{\phi}(2^{-j}s) = \widehat{\phi}(0)$ gleichmässig für $-R \leq s \leq R$, sodass man

$$\lim_{j \rightarrow \infty} \|P_j f\|_2^2 = |\widehat{\phi}(0)|^2 \cdot \|f\|_2^2$$

erhält. Andererseits hat man aber

$$\lim_{j \rightarrow \infty} \|P_j f\|_2^2 = \|f\|_2^2.$$

Daraus ergibt sich die Behauptung.

Für Teil 2. der Behauptung wird an den Fall $s = 0$ des vorigen Theorems erinnert:

$$\sum_{n \in \mathbb{Z}} |\widehat{\phi}(n)|^2 = 1.$$

Die Behauptung ergibt sich zusammen mit der eben bewiesenen Aussage von Teil 1.

Für Teil 3. betrachte man $f(t) = \sum_{n \in \mathbb{Z}} \phi(t+n)$ als eine 1-periodische Funktion. Für deren Fourierkoeffizienten gilt:

$$\int_0^1 f(t) e^{-2\pi i k t} dt = \dots = \int_{\mathbb{R}} \phi(t) e^{-2\pi i k t} dt = \widehat{\phi}(k) = \delta_{k,0}$$

nach dem bereits Bewiesenen. Damit ist aber $f(t) \equiv 1$ klar.

Schliesslich zu Teil 4. der Behauptung. Dazu betrachte man die Skalierungsgleichung

$$\phi(t) = \sum_k h_k \phi_{1,k}(t) = \sum_k h_k \sqrt{2} \phi(2t - k)$$

und die Waveletgleichung

$$\psi(t) = \sum_k g_k \phi_{1,k}(t) = \sum_k g_k \sqrt{2} \phi(2t - k)$$

der MRA, wobei $g_k = (-1)^k \overline{h_{1-k}}$ ($k \in \mathbb{Z}$) ist.

Mit den früheren Bezeichnungen

$$H(\omega) = \sum_k h_k e^{i k \omega}, \quad G(\omega) = \sum_k g_k e^{i k \omega},$$

ist dann

$$G(\omega) = \sum_k (-1)^k \overline{h_{1-k}} e^{i k \omega} = \dots = e^{i(\omega+\pi)} \overline{H(\omega + \pi)}.$$

Früher notierte Eigenschaften besagen:

$$\begin{aligned} H(0) &= \sum_k h_k = \sqrt{2}, \\ H(\pi) &= \sum_k h_k e^{i\pi k} = \sum_k h_{2k} - \sum_k h_{2k+1} = 0. \end{aligned}$$

Daraus folgt:

$$\begin{aligned} G(0) &= (-1) \overline{H(\pi)} = 0, \\ G(\pi) &= \overline{H(2\pi)} = \overline{H(0)} = \sqrt{2}. \end{aligned}$$

Per Fouriertransformation erhält man zu Skalierungs- und Waveletgleichung äquivalente Aussagen im Frequenzbereich:

$$\begin{aligned} \widehat{\phi}(\omega) &= \frac{1}{\sqrt{2}} \sum_k h_k e^{-i\pi k \omega} \widehat{\phi}(\omega/2) = \frac{1}{\sqrt{2}} H(-\pi\omega) \cdot \widehat{\phi}(\omega/2), \\ \widehat{\psi}(\omega) &= \frac{1}{\sqrt{2}} \sum_k g_k e^{-i\pi k \omega} \widehat{\phi}(\omega/2) = \frac{1}{\sqrt{2}} G(-\pi\omega) \cdot \widehat{\phi}(\omega/2). \end{aligned}$$

Insbesondere ist also

$$\left| \widehat{\psi}(0) \right| = \frac{1}{\sqrt{2}} |G(0)| \cdot \left| \widehat{\phi}(0) \right| = 0. \quad \square$$

9.7 Weitere Eigenschaften von MRAs (II)

In diesem Abschnitt werden spezielle Eigenschaften einer MRA $\{V_j\}_{j \in \mathbb{Z}}$ mit Skalierungsfunktion $\phi(t)$, Skalierungsfilter $(h_k)_{k \in \mathbb{Z}}$, Waveletfunktion $\psi(t)$, und Waveletfilter $(g_k)_{k \in \mathbb{Z}}$ untersucht. Ausgangspunkt sind die fundamentalen Beziehungen

$$\begin{aligned}\phi(t) &= \sum_k h_k \phi_{1,k} = \sum_k h_k \sqrt{2} \phi(2t - k) \\ \psi(t) &= \sum_k g_k \phi_{1,k} = \sum_k g_k \sqrt{2} \phi(2t - k)\end{aligned}$$

und ihre Fouriertransformierten

$$\begin{aligned}\widehat{\phi}(s) &= \frac{1}{\sqrt{2}} \sum_k h_k e^{-2\pi i k(s/2)} \widehat{\phi}(s/2) = m_0(s/2) \widehat{\phi}(s/2) \\ \widehat{\psi}(s) &= \frac{1}{\sqrt{2}} \sum_k g_k e^{-2\pi i k(s/2)} \widehat{\phi}(s/2) = m_1(s/2) \widehat{\phi}(s/2),\end{aligned}$$

wobei nun die ankürzende Schreibweise

$$\begin{aligned}m_0(s) &= \frac{1}{\sqrt{2}} h(e^{-2\pi i s}) \\ m_1(s) &= \frac{1}{\sqrt{2}} g(e^{-2\pi i s})\end{aligned}$$

für die Frequenzdarstellungen (Fourierreihen) der Skalierungs- und Waveletfilter benutzt wird.

Aus der Beziehung $g_k = (-1)^k \overline{h_{1-k}}$ ergibt sich

$$m_1(s) = e^{-2\pi i(s+\frac{1}{2})} \overline{m_0(s+1/2)}.$$

Spezielle Werte sind

$$\begin{aligned}m_0(0) &= \frac{1}{\sqrt{2}} h(1) = \frac{1}{\sqrt{2}} \sum_k h_k = 1, \\ m_0(1/2) &= \frac{1}{\sqrt{2}} h(e^{-\pi i}) = \frac{1}{\sqrt{2}} h(-1) = 0.\end{aligned}$$

Die Orthogonalität des Skalierungsfilters $(h_k)_{k \in \mathbb{Z}}$:

$$\sum_n h_n \overline{h_{n-2k}} = \delta_{0,k}$$

drückt sich nun durch

$$|m_0(s)|^2 + |m_0(s+1/2)|^2 \equiv 1$$

aus. Aus der Definition des Waveletfilters $(g_k)_{k \in \mathbb{Z}}$ ergibt sich die analoge Aussage

$$|m_1(s)|^2 + |m_1(s+1/2)|^2 \equiv 1.$$

Die Orthogonalität von Skalierungs- und Waveletfilter ist automatisch gegeben und drückt sich durch

$$m_0(s) \overline{m_0(s+1/2)} + m_1(s) \overline{m_1(s+1/2)} = 0$$

aus. Insgesamt kann man zusammenfassen:

Theorem 9.7 — Filter zu einer MRA.

Sind Filter $(h_k)_{k \in \mathbb{Z}}$ und $(g_k)_{k \in \mathbb{Z}}$ mittels $g_k = (-1)^k \overline{h_{1-k}}$ ($k \in \mathbb{Z}$) verbunden, so sind die folgenden Aussagen äquivalent:

1. $\sum_n h_n \overline{h_{n-2k}} = \delta_{0,k}$
2. $|m_0(s)|^2 + |m_0(s + 1/2)|^2 \equiv 1$
3. $\sum_n g_n \overline{g_{n-2k}} = \delta_{0,k}$
4. $HH^\dagger + GG^\dagger = Id$
5. $H^\dagger H = Id = G^\dagger G$

Dabei sind H und G bzw. H^\dagger und G^\dagger die durch die Filter $(h_k)_{k \in \mathbb{Z}}$ und $(g_k)_{k \in \mathbb{Z}}$ (incl. downsampling!) definierten Transformationen auf Signalen bzw. deren Adjungierte.

Die motiviert folgende Definition:

Definition 9.4 (QMF).

Ein Filter $(h_k)_{k \in \mathbb{Z}}$ ist ein quadrature mirror filter (QMF), wenn die beiden Bedingungen

$$\begin{aligned} |m_0(s)|^2 + |m_0(s + 1/2)|^2 &\equiv 1 \\ m_0(0) &= 1 \end{aligned}$$

für seine Frequenzdarstellung erfüllt sind.

9.7.1 Konvergenz der Fixpunktiteration

Eine Iteration der Fouriertransformierten der Skalierungsgleichung liefert “formal”

$$\widehat{\phi}(s) = \prod_{j \geq 1} m_0(s/2^j) \cdot \underbrace{\widehat{\phi}(0)}_{=1}.$$

Für das unendliche Produkt kann man die Konvergenz präzisieren:

Theorem 9.8 — Konvergenz der Produktkonstruktion eines QMF.

Ist $(h_k)_{k \in \mathbb{Z}}$ ein QMF, so konvergiert

$$\prod_{j \geq 1} m_0(s/2^j) = \lim_{\ell \rightarrow \infty} \prod_{1 \leq j \leq \ell} m_0(s/2^j)$$

absolut und gleichmässig in jedem kompakten Intervall (aber keine \mathcal{L}^2 -Konvergenz!)

\mathcal{L}^2 -Konvergenz erreicht man mit dem Hilfsmittel der Bandbeschränkung.

Theorem 9.9 — Bandbeschränkte Approximation.

Für die "bandbeschränkte Approximation" gilt in $\mathcal{L}^2(\mathbb{R})$:

$$\widehat{\mu}_\ell(s) = \prod_{j=1}^{\ell} m_0(s/2^j) \cdot \chi_{[-2^{\ell-1}, 2^{\ell-1}]}(s) \xrightarrow{\ell \rightarrow \infty} \widehat{\phi}(s),$$

falls gilt: $|s| \leq 1/4 \Rightarrow |m_0(s)| > c > 0$. (*)

Wie im Abschnitt ?? bereits ausgeführt, suggeriert die Skalierungsgleichung eine Fixpunktiteration um aus einem Filter $(h_k)_{k \in \mathbb{Z}}$ mit QMF ϕ zu approximieren:

$$\eta_0(t) = \chi_{[-1/2, 1/2]}(t) \quad \eta_{\ell+1}(t) = \sum_k h_k \sqrt{2} \eta_\ell(2t - k) \quad (\ell \geq 0)$$

in der Erwartung, dass $(\eta_\ell) \rightarrow \phi$ in \mathcal{L}^2 .

Theorem 9.10 — \mathcal{L}^2 -Konvergenz.

Ist $(h_k)_{k \in \mathbb{Z}}$ ein endlicher QMF und definiert man $\phi(t)$ durch

$$\widehat{\phi}(s) = \prod_{j \geq 1} m_0(s/2^j)$$

so gilt tatsächlich $\eta_\ell \rightarrow \phi$ in \mathcal{L}^2 . Zudem ist $\{T_k \phi\}_{k \in \mathbb{Z}}$ ein ONS in $\mathcal{L}^2(\mathbb{R})$.

9.7.2 Endliche Filter und endlicher Träger

Ist $\phi(t) = \sum_k h_k \phi_{1,k}(t)$ Skalierungsfunktion einer MRA und hat $\phi(t)$ kompakten Träger, so ist wegen

$$h_k = \langle \phi, \phi_{1,k} \rangle = \sqrt{2} \int \phi(t) \phi(2t - k) dt$$

$(h_k)_{k \in \mathbb{Z}}$ ein endlicher Filter. Genauer gilt:

Theorem 9.11 — Filterlänge für MRA.

Für eine MRA mit ϕ , $(h_k)_{k \in \mathbb{Z}}$ und Eigenschaft (*) (siehe Theorem 9.9) sind äquivalent:

1. $(h_k)_k$ hat Länge $2N$
2. $\text{supp}(\phi) \subseteq$ Intervall der Länge $N - 1$

Beweis mittels Kaskadenalgorithmus!

NB: Die Länge eines QMF muss immer gerade sein.

9.7.3 Momente, Glattheit, Rekonstruktionseigenschaften

Eine generelle Vorbemerkung über den Zusammenhang von Glattheit und Verschwinden im Unendlichen, die schon im Abschnitt 4.5 erwähnt wurde:

Theorem 9.12 — Glattheit und Verschwinden im Unendlichen.

1. Aus $f(t) \in \mathcal{L}^1(\mathbb{R})$ und $t \cdot f(t) \in \mathcal{L}^1(\mathbb{R})$ folgt $\widehat{f}(s) \in \mathcal{C}^1(\mathbb{R})$ und

$$t \cdot \widehat{f}(s) = D_s \widehat{f}(s),$$

wobei $D_s = -\frac{1}{2\pi i} \frac{d}{ds}$ im wesentlichen der Ableitungsoperator ist.

2. Allgemein: Aus $f(t) \in \mathcal{L}^1(\mathbb{R})$ und $t^N f(t) \in \mathcal{L}^1(\mathbb{R})$ folgt $\widehat{f}(s) \in \mathcal{C}^N(\mathbb{R})$ und

$$t^j \widehat{f}(s) = D_s^j \widehat{f}(s) \quad (0 \leq j \leq N),$$

“und umgekehrt”.

Definition 9.5 (Momente).

1. Für $k \geq 0$ bezeichnet man das Integral

$$\int_{\mathbb{R}} t^k f(t) dt$$

als das k -te Moment von f .

2. Gilt für eine Funktion $f(t)$ die Eigenschaft

$$\widehat{f}^{(k)}(0) = \int_{\mathbb{R}} t^k f(t) dt = 0 \quad (0 \leq k < N),$$

so sagt man: f hat N verschwindende Momente.

Beachte hierzu: falls $t^k f(t) \in \mathcal{L}^1$ ist, so gilt

$$\int_{\mathbb{R}} t^k f(t) dt = 0 \iff \widehat{f}^{(k)}(0) = 0.$$

Satz 9.3 Es sei $\psi \in \mathcal{L}^2(\mathbb{R})$ und $\{\psi_{j,k}\}$ sei ein ONS in $L^2(\mathbb{R})$. Dann gilt:

1. Sind $\psi, \widehat{\psi} \in \mathcal{L}^1(\mathbb{R})$, so ist $\int_{\mathbb{R}} \psi = 0$.
2. Sind $t^N \psi(t), s^{N+1} \widehat{\psi}(s) \in \mathcal{L}^1(\mathbb{R})$, so gilt

$$\int_{\mathbb{R}} t^m \psi(t) dt = \widehat{\psi}^{(m)}(0) = 0 \quad (0 \leq m \leq N).$$

NB: $\widehat{\psi} \in \mathcal{L}^1 \Rightarrow \psi \in \mathcal{C}^{N+1}$

Die Aussage 1. wurde bereits in Theorem 9.6 bewiesen. Hier ein weiterer Beweis dieser grundlegenden Aussage, der lediglich voraussetzt, dass die Familie $\{\psi_{j,k}(t)\}_{j,k \in \mathbb{Z}}$ ein Orthonormalsystem in $\mathcal{L}^2(\mathbb{R})$ ist. Weiter wird vorausgesetzt, dass $\psi(t) \in \mathcal{L}^1(\mathbb{R})$ gilt, damit ist die Existenz von $\int_{\mathbb{R}} \psi(t) dt$ garantiert und $\widehat{\psi}$ ist gleichmässig stetig auf \mathbb{R} . Setzt man noch $\widehat{\psi} \in \mathcal{L}^1(\mathbb{R})$ voraus, so ist auch $\psi(t)$ gleichmässig stetig auf \mathbb{R} und verschwindet für $t \rightarrow \pm\infty$.

Zunächst die Bemerkung, wie sich die Fouriertransformierten der skalierten Waveletfunktionen $\psi_{j,k}$ mittels $\widehat{\psi}$ ausdrücken:

$$\widehat{\psi_{j,k}}(s) = 2^{-j/2} e^{-2\pi i s 2^{-j} k} \widehat{\psi}(2^{-j} s).$$

Aus der Orthogonalität der Familie $\{\psi_{j,k}(t)\}_{j,k \in \mathbb{Z}}$ ergibt sich für $(j,k) \neq (0,0)$:

$$0 = \langle \psi | \psi_{j,k} \rangle = \langle \widehat{\psi} | \widehat{\psi_{j,k}} \rangle = 2^{-j/2} \int_{\mathbb{R}} \widehat{\psi}(s) e^{2\pi i s 2^{-j} k} \overline{\widehat{\psi}(2^{-j} s)} ds$$

und somit

$$\int_{\mathbb{R}} \widehat{\psi}(s) e^{2\pi i s 2^{-j} k} \overline{\widehat{\psi}(2^{-j} s)} ds = 0 \quad ((j,k) \neq (0,0)).$$

Wegen der gleichmässigen Stetigkeit von $\psi(t)$ gibt es einen dyadischen Punkt $x = 2^{-j_0} k_0$ mit $\psi(x) \neq 0$, ansonsten wäre $\psi(t) \equiv 0$. Dann ist

$$\int_{\mathbb{R}} \widehat{\psi}(s) e^{2\pi i s x} \overline{\widehat{\psi}(2^{-j} s)} ds = 0$$

für alle $j \geq j_0$, wobei $x = 2^{-j} \cdot 2^{j-j_0} k_0$.

Der Grenzübergang $\lim_{j \rightarrow \infty}$ liefert dann (gleichmässige Stetigkeit von $\widehat{\psi}(s)$ beachten):

$$\overline{\widehat{\psi}(0)} \cdot \psi(x) = \overline{\widehat{\psi}(0)} \int_{\mathbb{R}} \widehat{\psi}(s) e^{2\pi i s x} ds = \int_{\mathbb{R}} \widehat{\psi}(s) e^{2\pi i s x} \overline{\widehat{\psi}(0)} ds = 0$$

und das hat wegen $\psi(x) \neq 0$ die gewünschte Aussage $\widehat{\psi}(0) = 0$ zur Folge.

Aussage 2. ergibt sich per Induktion. Das ist technisch und langwierig und wird hier nicht ausgeführt. \square

Das vorhergehende Theorem beschreibt also die Situation, dass unter Annahmen der Glattheit bzw. des Verschwindens einer Waveletfunktion $\psi(t)$ einer MRA diese N verschwindende Momente hat. Aus der Fouriertransformierten der Waveletgleichung

$$\widehat{\psi}(s) = m_1(s/2) \cdot \widehat{\phi}(s/2)$$

ergibt sich durch Differenzieren, dass die Aussage

$$m_0^{(k)}(1/2) = 0 \quad (0 \leq k < N)$$

äquivalent zur Aussage “ $\psi(t)$ hat N verschwindende Momente” ist.

Differenziert man die Fouriertransformierte der Skalierungsgleichung

$$\widehat{\phi}(s) = m_0(s/2) \cdot \widehat{\phi}(s/2),$$

so ergibt sich daraus auch die Aussage

$$\widehat{\phi}^{(k)}(m) = 0 \quad \begin{cases} 0 \leq k < N \\ m \in \mathbb{Z} \setminus \{0\} \end{cases} .$$

9.8 Die Größe von Waveletkoeffizienten

Die Frage, wie sich die analytischen Eigenschaften einer mittels Wavelets analysierten Funktion in der Größe von Waveletkoeffizienten sichtbar werden, wurde schon im Fall des Haar-Wavelets im Abschnitt ... untersucht. Die folgende Aussage präzisiert die Konsequenzen davon, was es generell für die Größe von Waveletkoeffizienten bedeutet, wenn eine Waveletfunktion $\psi(t)$ N verschwindende Momente hat:

Theorem 9.13 — Abschätzung der Waveletkoeffizienten.

Ist $\psi \in L^2(\mathbb{R})$ eine Waveletfunktion mit kompaktem Träger $[0, a)$, mit $\|\psi\| = 1$ und mit N verschwindenden Momenten, so gibt es für jedes $f \in \mathcal{C}^N(\mathbb{R})$, für das $f^{(N)}$ auf \mathbb{R} beschränkt ist, eine Konstante $C = C(N, f)$, so dass “global” gilt:

$$|\langle f | \psi_{j,k} \rangle| \leq C \cdot 2^{-jN} \cdot 2^{-j/2} \quad (j, k \in \mathbb{Z}).$$

Genauer gilt die “lokale” Aussage:

$$|\langle f | \psi_{j,k} \rangle| \leq 2^{-jN} \cdot 2^{-j/2} \frac{1}{N!} f^{(N)}(\xi_{j,k}) \times \int_{-a/2}^{a/2} t^N \overline{\psi(t+a/2)} dt, \quad (j, k \in \mathbb{Z}),$$

wobei $\xi_{j,k}$ der Mittelpunkt des Intervalls $\tilde{I}_{j,k} = [2^{-j}k, 2^{-j}(k+a))$ ist.

Die pauschale Interpretation dieser Aussage ist:

- Wavelet-Koeffizienten, die zu Bereichen gehören, in denen $f(t)$ “glatt” ist, werden mit wachsender Auflösung $j \nearrow \infty$ sehr schnell klein! Diese Eigenschaft macht man sich bei Kompressionsverfahren zu Nutze.

Die “lokale” Aussage von Theorem 9.13 ist so zu verstehen: Ist $f(t)$ N -fach stetig differenzierbar in einem Intervall $I \ni t_0$, so gilt die Abschätzung für alle j, k mit $\tilde{I}_{j,k} \subseteq I$.

Zum *Beweis* des Theorems.

Die skalierte und translatierte Waveletfunktion $\psi_{j,k}(t)$ “lebt” auf dem Intervall $\tilde{I}_{j,k} = [2^{-j}k, 2^{-j}(k+a))$. Ist $p(t)$ ein Polynom von einem Grad $< N$, so folgt aus den N verschwindenden Momenten:

$$\langle p | \psi_{j,k} \rangle = \int_{\mathbb{R}} p(t) \overline{\psi_{j,k}(t)} dt = \int_{\tilde{I}_{j,k}} p(t) \overline{\psi_{j,k}(t)} dt = 0$$

Die Taylorentwicklung von $f(t)$ an der Stelle $\xi_{j,k}$ ist

$$f(t) = f(\xi_{j,k}) + (t - \xi_{j,k})f'(\xi_{j,k}) + \dots + \frac{1}{(N-1)!} (t - \xi_{j,k})^{N-1} f^{(N-1)}(\xi_{j,k}) + R_N(t),$$

mit dem Restglied

$$R_N(t) = \frac{1}{N!} (t - \xi_{j,k})^N f^{(N)}(\eta)$$

für ein zwischen $\xi_{j,k}$ und t gelegenes η . Dieses Restglied lässt sich für $t \in \tilde{I}_{j,k}$ abschätzen durch

$$|R_N(t)| \leq \frac{1}{N!} 2^{-N(j+1)} \cdot a^N \cdot \max_{\eta \in \tilde{I}_{j,k}} |f^{(N)}(\eta)|.$$

Alles zusammen ergibt sich

$$\begin{aligned} \langle f | \psi_{j,k} \rangle &= \int_{\tilde{I}_{j,k}} R_N(t) \overline{\psi_{j,k}} dt \\ \text{und } |\langle f | \psi_{j,k} \rangle| &= \left| \int_{\tilde{I}_{j,k}} R_N(t) \overline{\psi_{j,k}} dt \right| \\ &\leq \frac{1}{N!} 2^{-N(j+1)} \cdot a^N \cdot \underbrace{\max_{\eta \in \tilde{I}_{j,k}} |f^{(N)}(\eta)|}_{\leq \|f^{(N)}\|_\infty} \cdot \underbrace{\int_{\tilde{I}_{j,k}} |\psi_{j,k}(t)| dt}_{\leq \sqrt{a} 2^{-j}} \\ &\leq 2^{-Nj} 2^{-j/2} \times \underbrace{\frac{1}{N!} a^{N+1/2} 2^{-N}}_{C(f,N)} \cdot \|f^{(N)}\|_\infty. \end{aligned}$$

Das beweist die ‘globale’ Aussage. Die ‘lokale’ Aussage ergibt sich entsprechend. \square

■ **Beispiel 9.1** Die Waveletfunktion $\psi(t)$ des Daubechies-Wavlets D_4 hat $N = 2$ verschwindende Momente. Es gilt

$$\int_{\mathbb{R}} \psi(t) dt = 0, \quad \int_{\mathbb{R}} t \psi(t) dt = 0, \quad \int_{\mathbb{R}} t^2 \psi(t) dt = -\frac{1}{8} \sqrt{\frac{3}{2\pi}}.$$

Für $f \in \mathcal{C}^2(\mathbb{R})$ gilt (da $\psi(t)$ ausserhalb des Intervalles $[0, 3]$ verschwindet)

$$\begin{aligned} \langle f | \psi_{j,k} \rangle &= \int_{\mathbb{R}} f(t) 2^{j/2} \psi(2^j t - k) dt \\ &= \int_0^{3 \cdot 2^{-j}} f(t + 2^{-j} k) 2^{j/2} \psi(2^j t) dt \end{aligned}$$

Entwickelt man $f(t)$ an der Stelle $t = 2^{-j} k$ in eine Taylorreihe, so ergibt sich

$$\langle f | \psi_{j,k} \rangle \approx -\frac{1}{16} \sqrt{\frac{3}{2\pi}} 2^{-5j/2} f''(2^{-j} k),$$

mit Gleichheit, falls f konstant oder linear oder quadratisch ist. Insbesondere verschwinden alle Waveletkoeffizienten $\langle f | \psi_{j,k} \rangle$ in Bereichen, wo f linear ist. ■

9.9 Exakte Darstellung von Polynomen

Die guten Approximationseigenschaften von Wavelets mit vielen verschwindenden Momenten werden belegt durch folgende Aussage, die angesichts der ‘bizarren Graphen’ von Waveletfunktionen zunächst überraschend erscheint:

Theorem 9.14 — Exakte Darstellung von Polynomen durch Wavelets.

Sind ϕ, ψ Skalierung und Wavelet zu einer MRA, wobei

- ϕ kompakten Träger hat,
- ψ N verschwindende Momente hat,

so gibt es für jedes k mit $0 \leq k < N$ Konstante $q_{n,k}$ ($n \in \mathbb{Z}$) mit

$$\sum_n q_{n,k} \phi(t+n) = t^k.$$

Das bedeutet: jedes Polynom vom Grad $< N$ kann durch Linearkombination der Translate der Skalierungsfunktion $T_n \phi$ ($n \in \mathbb{Z}$) exakt dargestellt werden!

Beweis: Da $\phi(t)$ einen kompakten Träger hat, gehört die Funktion $(2\pi it)^k \phi(t)$ zu $\mathcal{L}^1(\mathbb{R})$ und man hat

$$\widehat{\phi}^{(k)}(s) = [\widehat{(2\pi it)^k \phi(t)}](s).$$

Für $m \in \mathbb{Z}$ gilt dann

$$\begin{aligned} \widehat{\phi}^{(k)}(m) &= \int_{\mathbb{R}} (2\pi it)^k \phi(t) e^{-2\pi imt} dt \\ &= \sum_{n \in \mathbb{Z}} \int_n^{n+1} (2\pi it)^k \phi(t) e^{-2\pi imt} dt \\ &= \sum_{n \in \mathbb{Z}} \int_0^1 (2\pi i(t+n))^k \phi(t+n) e^{-2\pi imt} dt \\ &= (2\pi i)^k \int_0^1 \sum_{n \in \mathbb{Z}} (t+n)^k \phi(t+n) e^{-2\pi imt} dt \end{aligned}$$

Beachte, dass $\sum_{n \in \mathbb{N}} (t+n)^k \phi(t+n)$ eine 1-periodische Funktion ist, deren Fourierkoeffizienten (bis auf den Faktor $(2\pi i)^k$) gerade die $\widehat{\phi}^{(k)}(m)$ sind.

Nun folgt, wie oben in Bemerkung 9.5 festgestellt, dass mit $\widehat{\psi}^{(k)} = 0$ für $0 \leq k < N$ auch $\widehat{\phi}^{(k)}(m) = 0$ für $0 \leq k < N$ und $m \in \mathbb{Z} \setminus \{0\}$ gilt. Als Fourierreihe interpretiert gelesen ergibt sich:

$$(2\pi i)^k \sum_{n \in \mathbb{Z}} (t+n)^k \phi(t+n) = \widehat{\phi}^{(k)}(0) \quad (0 \leq k < N).$$

Als Folgerung aus diesen Beziehungen erhält man nun für jedes $0 \leq k < N$:

$$\begin{aligned} \sum_{j=0}^k \binom{k}{j} \frac{\widehat{\phi}^{(j)}(0)}{(2\pi i)^j} (-t)^{k-j} &= \sum_{j=0}^k \binom{k}{j} \sum_{n \in \mathbb{Z}} (t+n)^j \phi(t+n) (-t)^{k-j} \\ &= \sum_{n \in \mathbb{Z}} \phi(t+n) \sum_{j=0}^k \binom{k}{j} (t+n)^j (-t)^{k-j} \\ &= \sum_{n \in \mathbb{Z}} \phi(t+n) n^k. \end{aligned}$$

Nun ist die linke Seite ein Polynom $p_k(t)$ vom Grad k in t , es gibt also Konstante $\pi_{k,j}$ für $0 \leq k < N$ mit

$$t^k = \sum_{0 \leq j \leq k} \pi_{k,j} p_j(t) = \sum_{n \in \mathbb{Z}} \phi(t+n) \sum_{0 \leq j \leq k} \pi_{k,j} n^j = \sum_{n \in \mathbb{Z}} \phi(t+n) q_k(n),$$

wobei also $q_k(n) = \sum_{0 \leq j \leq k} \pi_{k,j} n^j =: q_{n,k}$ die gesuchten Konstanten sind. \square

9.10 Zusammenfassung

Theorem 9.15 — MRA, QMF und Momente.

Sind ϕ, ψ Skalierung und Wavelet einer MRA und $(h_n)_{n \in \mathbb{Z}}$ der zugehörige Skalierungs-QMF, so sind folgende Aussagen äquivalent:

1. ψ hat N verschwindende Momente:

$$\int_{\mathbb{R}} t^k \psi(t) dt = 0 \quad (0 \leq k < N);$$

2. Der QMF (h_n) erfüllt die N Tiefpassbedingungen

$$m_0^{(k)}(1/2) = 0 \quad (0 \leq k < N);$$

3. Die Frequenzdarstellung $m_0(s)$ von (h_n) faktorisiert

$$m_0(s) = \left(\frac{1 + e^{-2\pi i s}}{2} \right)^N L(s),$$

wobei L trigonometrisches Polynom mit Periode 1 ist;

4. Der QMF (h_n) erfüllt die N Momentenbedingungen

$$\sum_n (-1)^n h_n n^k = 0 \quad (0 \leq k < N).$$

9.11 Mathematica-Notebooks mit Beispielen zum Kaskadenalgorithmus und zur dyadischen Interpolation

Die folgenden Seiten geben zwei Mathematica notebooks zur iterativen Approximation von Skalierungs- und Waveletfunktionen mittels Fixpunktiteration (Kaskadenalgorithmus) bzw. dyadischer Interpolation wieder. Diese Verfahren sind in den Abschnitten 9.3.1 bzw. 9.3.2 beschrieben worden. Weitere Beispiele zur dyadischen Interpolation finden sich im Abschnitt 16.1 des Anhangs.

Approximation of the scaling and wavelet functions using the cascade algorithm

Procedures

```
mypstyle = {Filling → Axis, FillingStyle → Red,  
  PlotRange → All, ImageSize → Scaled[0.6], PlotPoints → 200};
```

```
In[473]:= cascادهipw[h_,n_]:=Module[{degrees,min, max,right,hpol,hpols,clist,tbl,z},  
degrees=Map[First[#]&,h];  
min=Min[degrees];  
max=Max[degrees];  
hpol:=Apply[Plus,Map[Last[#] z^First[#]&,h]];  
hpols:=Expand[Product[hpol/.z→z^(2^t),{t,0,n-1}]];  
clist= 2^n CoefficientList[z^(-min (2^n-1))hpols,z];  
right=(1-2^(-n)) (max-min)+2^(-n);  
tbl=Table[{clist[[k]],(k-1)2^(-n) ≤t<k 2^(-n)},{k,1,Length[clist]}];  
Plot[Piecewise[tbl],{t,0,right},Evaluate[mypstyle]]  
]
```

```
In[474]:= cascادهipw::usage"level-n iteration for approximating  
the scaling function belonging to the filter h  
using the cascade algorithm.  
Display as a step function(piecewise)";
```

```
In[475]:= cascادهsipw[h_,n_]:=Module[{degrees,min, max,right,hpol,clist,tbl,z},  
degrees=Map[First[#]&,h];  
min=Min[degrees];  
max=Max[degrees];  
hpol:=Apply[Plus,Map[Last[#] z^First[#]&,h]];  
gpols:= hpol/.(z→-1/z);  
hpols:=Expand[ Product[(hpol/.z→z^(2^t)),{t,0,n-2}] (gpols/.z→z^(2^(n-1)))];  
clist= 2^n CoefficientList[z^(-min (2^(n-1)-1)+max 2^(n-1))hpols,z];  
right=(1-2^(-n)) (max-min)+2^(-n);  
tbl=Table[{clist[[k]],(k-1)2^(-n) ≤t<k 2^(-n)},{k,1,Length[clist]}];  
Plot[Piecewise[tbl],{t,0,right}, Evaluate[mypstyle]]  
]
```

```
In[476]:= cascادهsipw::usage"level-n iteration for approximating  
the wavelet function belonging to the filter h  
using the cascade algorithm.  
Display as a step function(piecewise)";
```

Example Daubechies-4

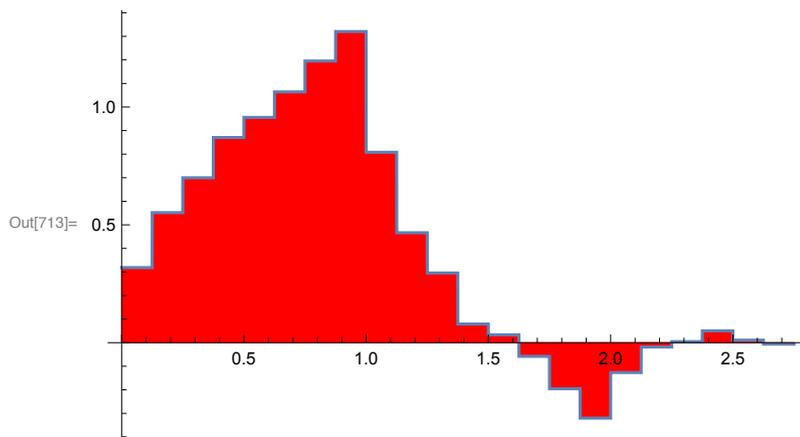
```
In[477]:= db4 = WaveletFilterCoefficients[DaubechiesWavelet[2]];
```

```
TableForm[db4, ta]
```

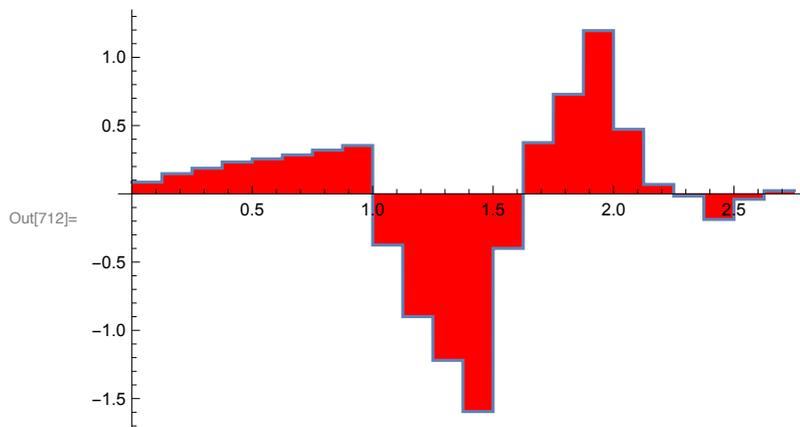
```
Out[478]//TableForm=
```

0	0.341506
1	0.591506
2	0.158494
3	-0.0915064

```
In[713]:= cascadephiw[db4, 3]
```

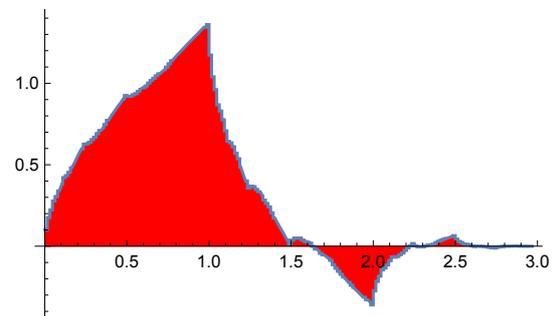
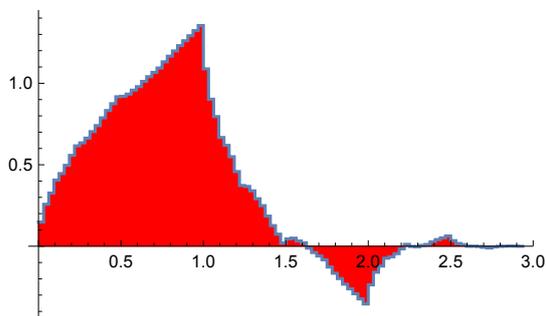
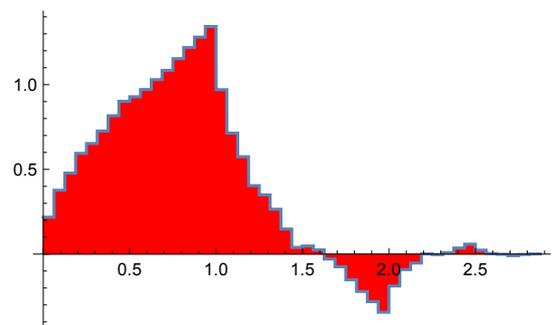
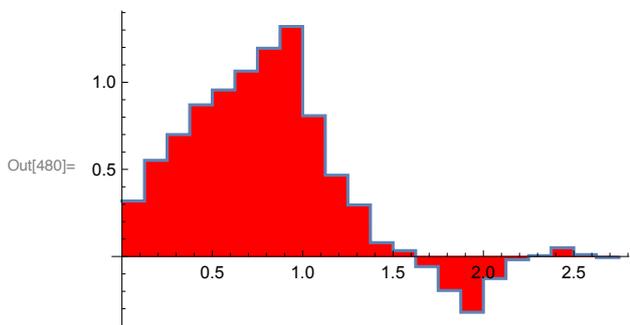
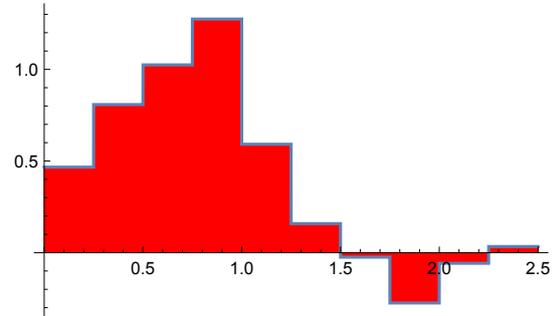
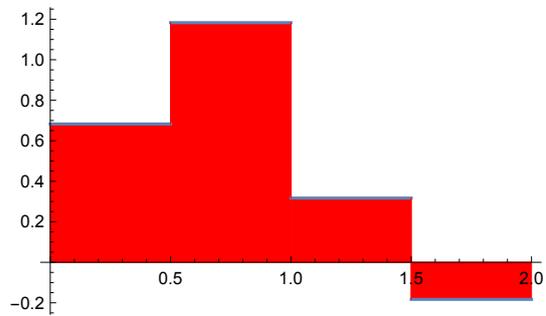


```
In[712]:= cascadepsiw[db4, 3]
```



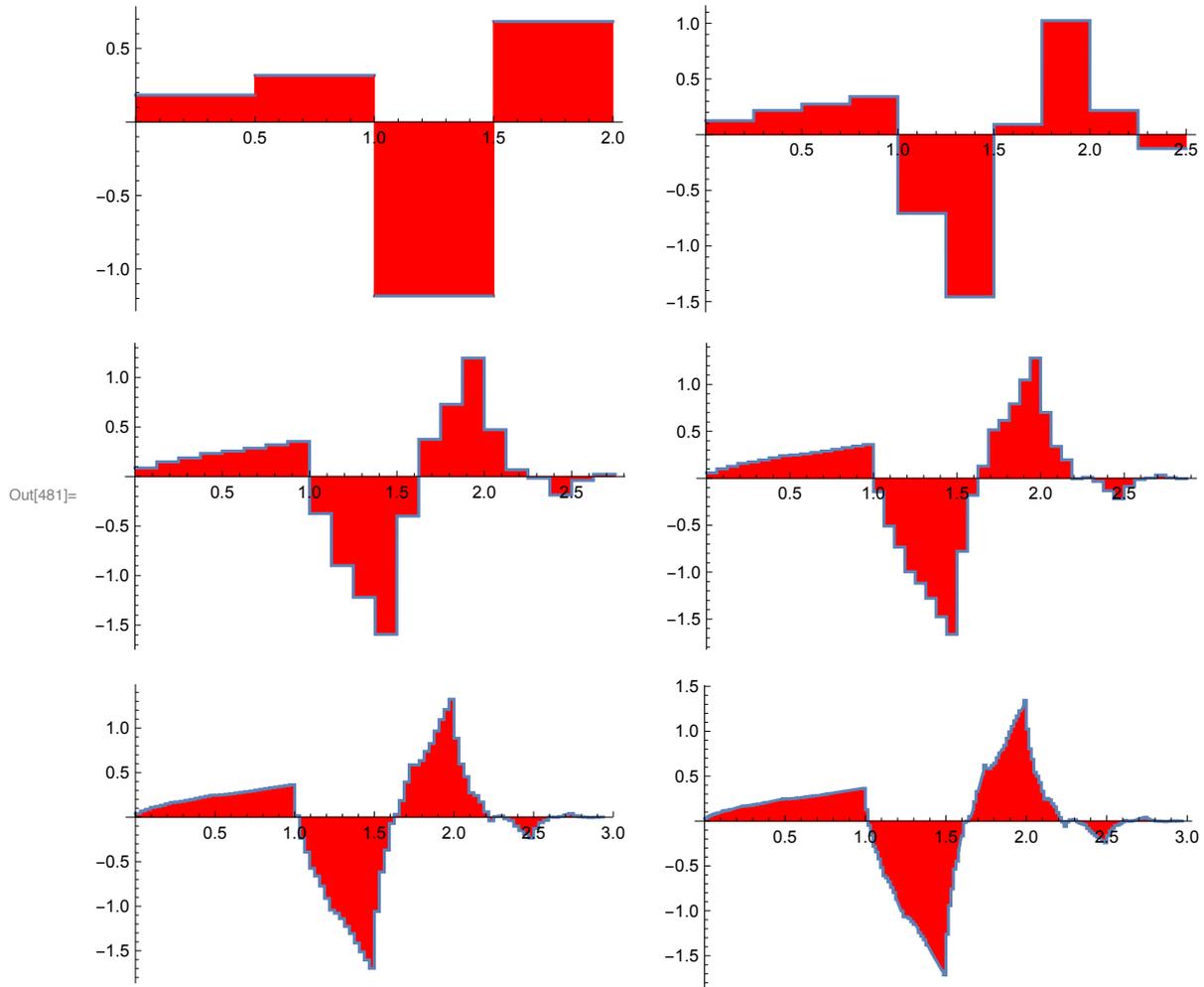
Approximating the scaling function

```
In[480]:= GraphicsGrid[Partition[Table[
  cascadehipw[db4, k], {k, 1, 6}], 2]]
```



Approximating the wavelet function

```
In[481]:= GraphicsGrid[Partition[Table[
  cascadepsipw[db4, k], {k, 1, 6}], 2]]
```



Example Daubechies-6

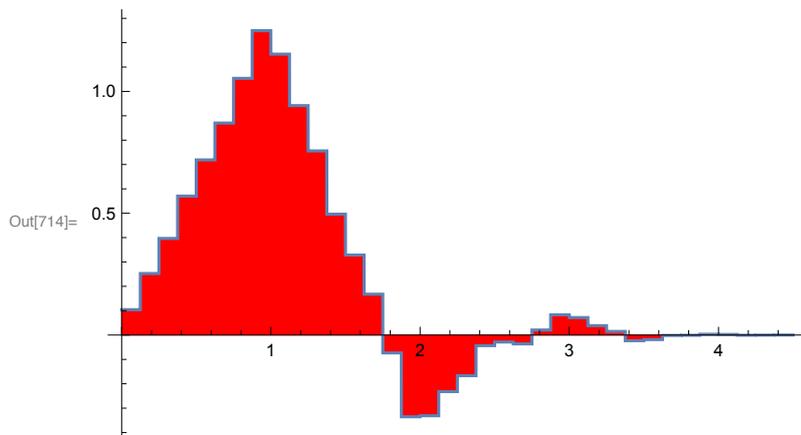
```
In[482]:= db6 = WaveletFilterCoefficients[DaubechiesWavelet[3]];
```

```
TableForm[db6, ta]
```

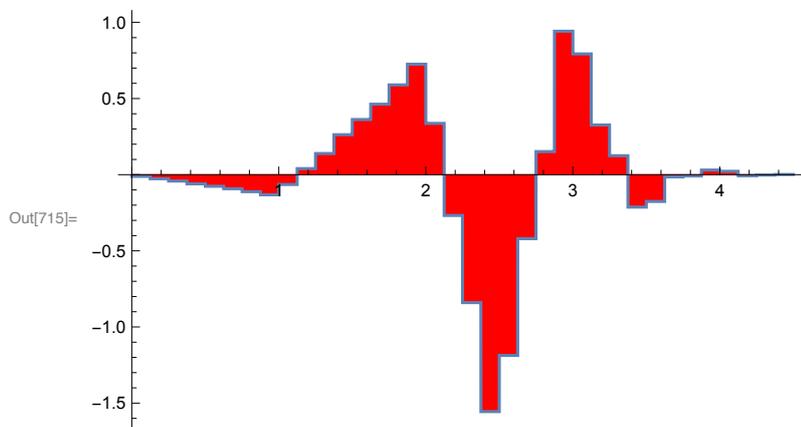
```
Out[483]//TableForm=
```

0	0.235234
1	0.570558
2	0.325183
3	-0.0954672
4	-0.0604161
5	0.0249087

```
In[714]:= cascadephiw[db6, 3]
```

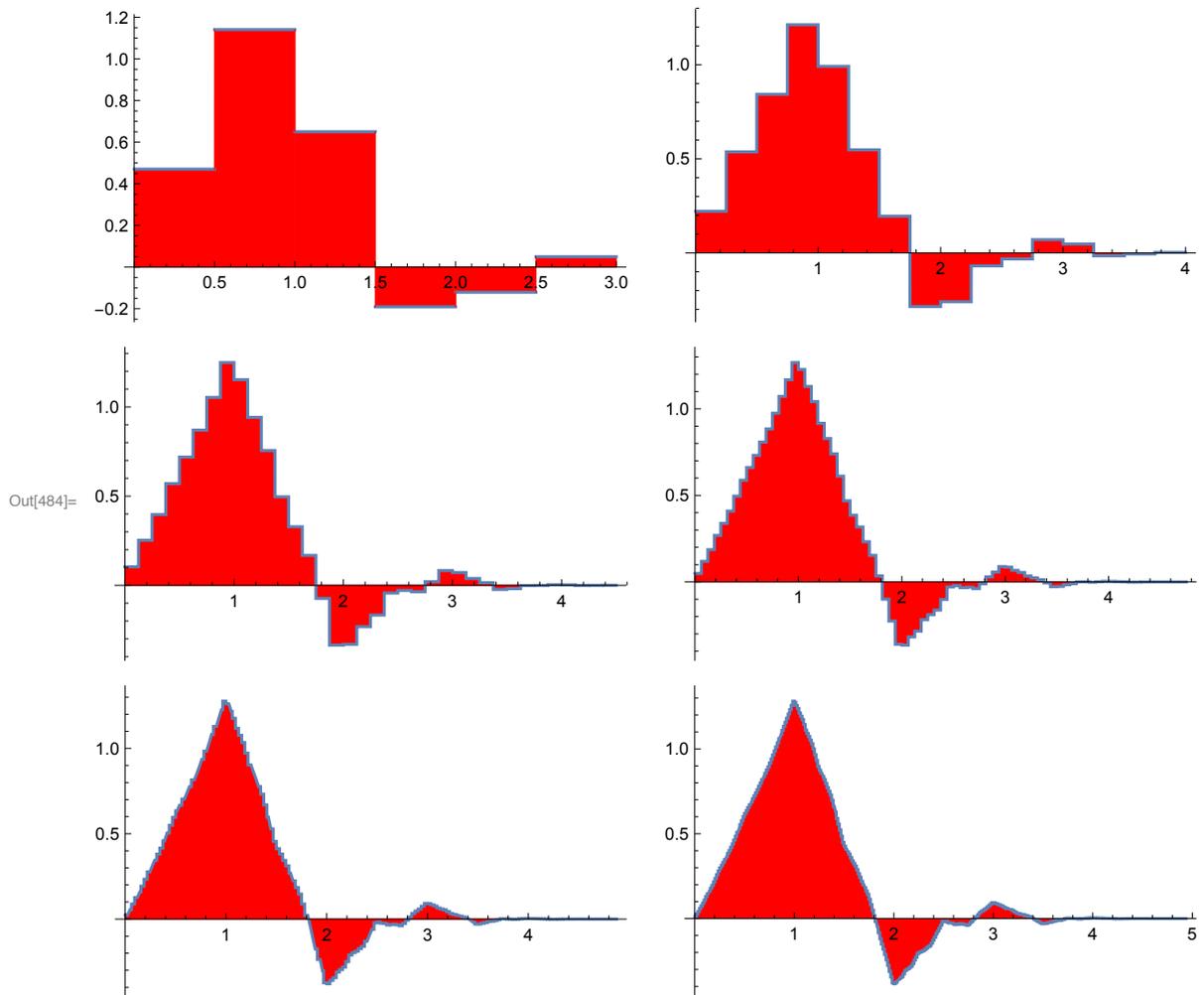


```
In[715]:= cascadepsiw[db6, 3]
```



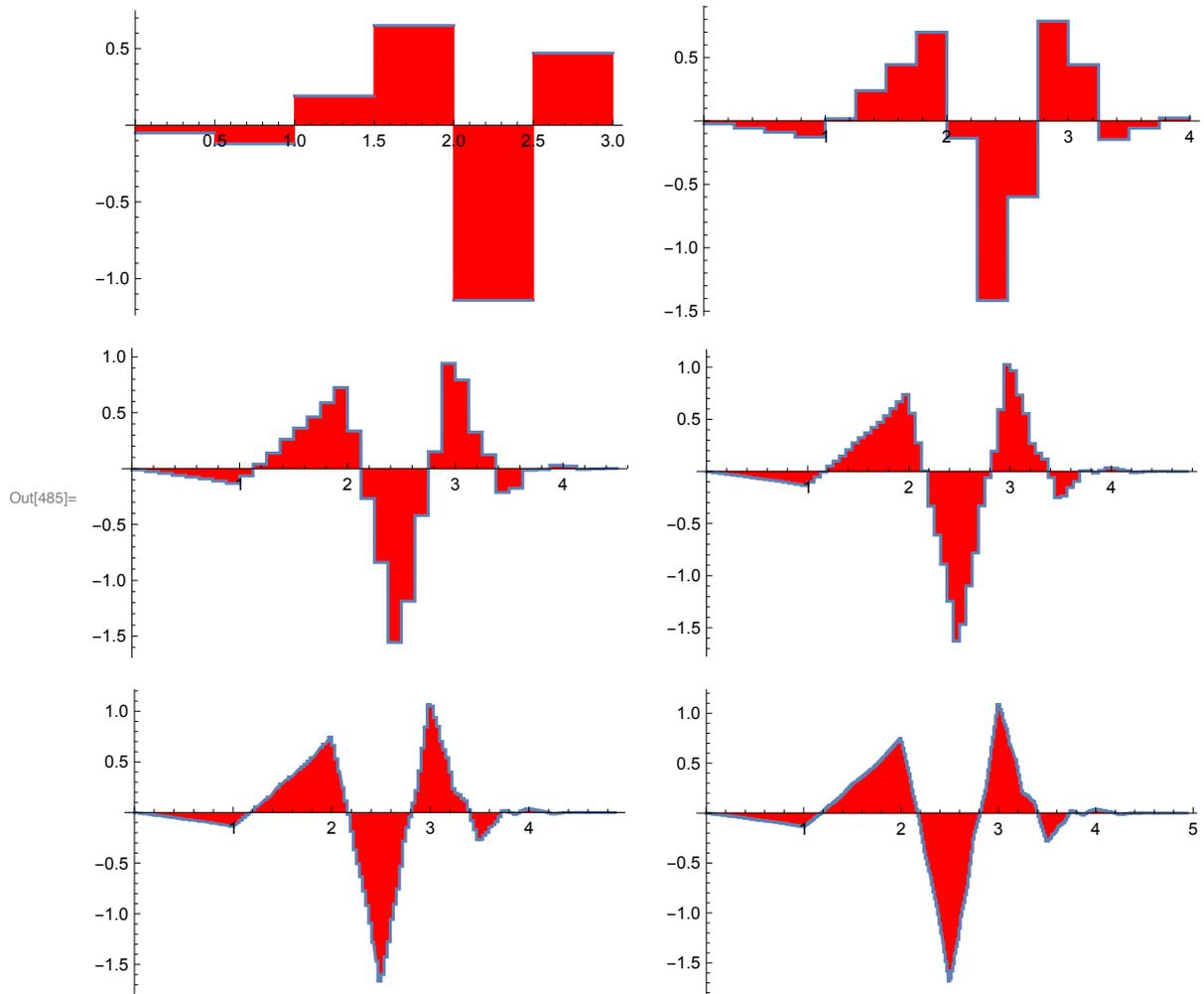
Approximating the scaling function

```
In[484]:= GraphicsGrid[Partition[Table[
  cascadehipw[db6, k], {k, 1, 6}], 2]]
```



Approximating the wavelet function

```
In[485]:= GraphicsGrid[Partition[Table[
  cascadepspw[db6, k], {k, 1, 6}], 2]]
```



Example Daubechies-8

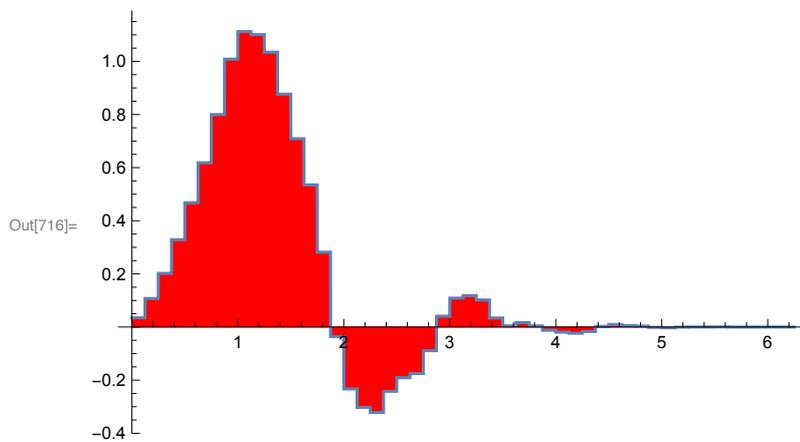
```
In[486]:= db8 = WaveletFilterCoefficients[DaubechiesWavelet[4]];
```

```
TableForm[db8, ta]
```

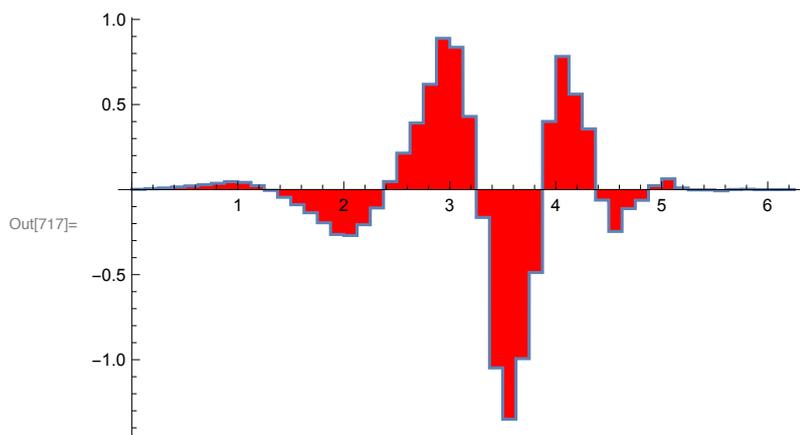
```
Out[487]//TableForm=
```

0	0.162902
1	0.505473
2	0.4461
3	-0.0197875
4	-0.132254
5	0.0218082
6	0.0232518
7	-0.00749349

```
In[716]:= cascadehipw[db8, 3]
```

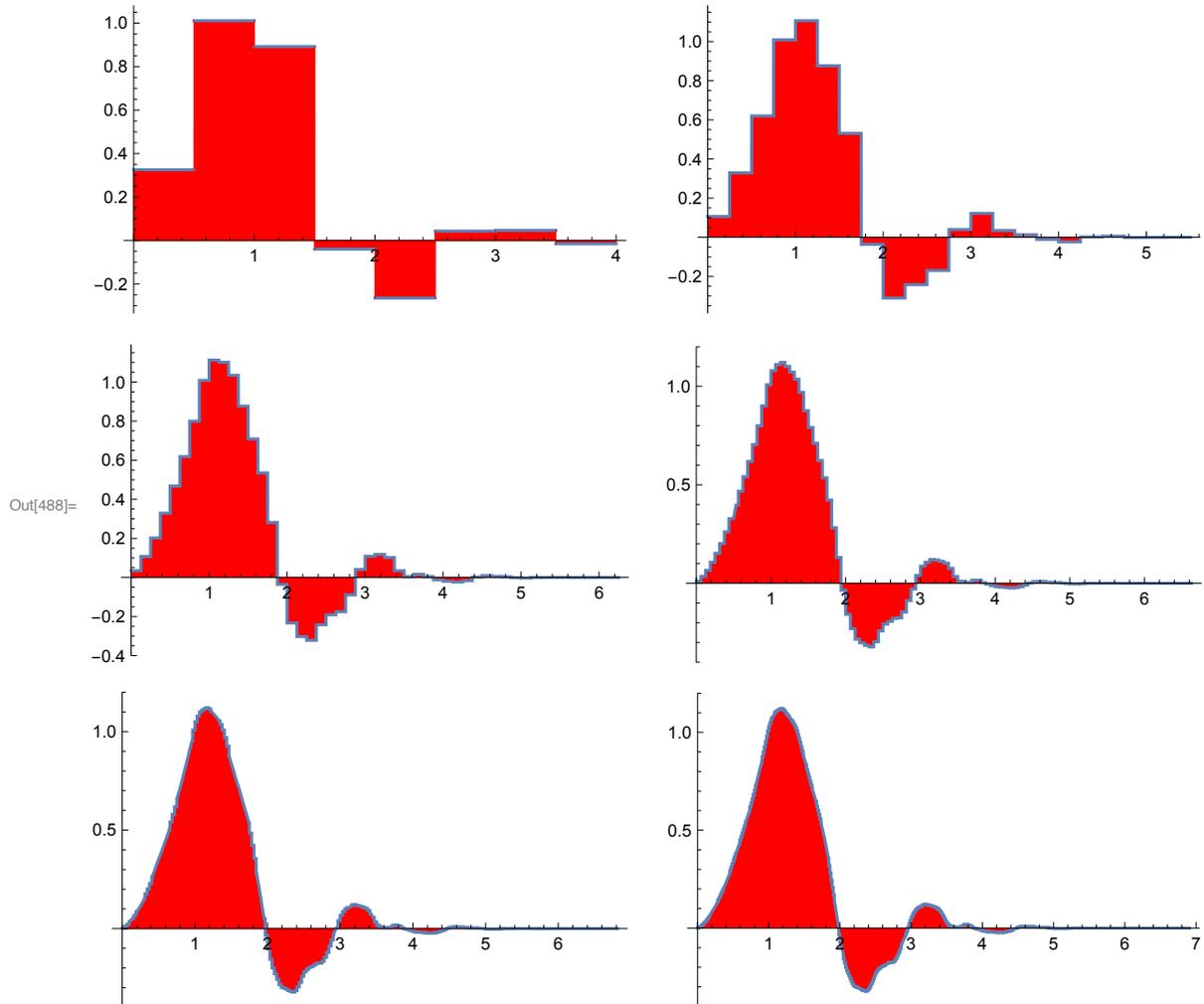


```
In[717]:= cascadepsipw[db8, 3]
```



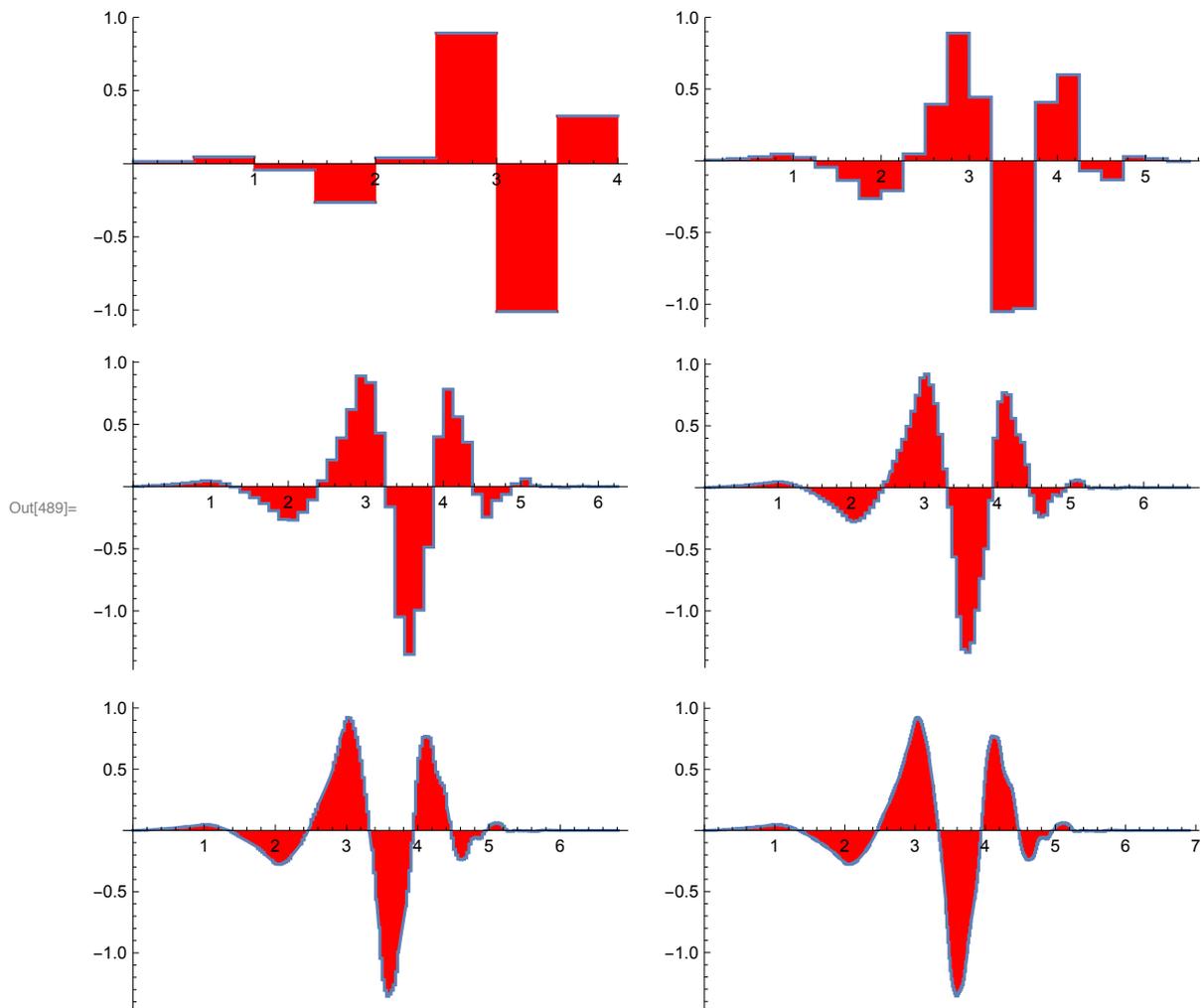
Approximating the scaling function

```
In[488]:= GraphicsGrid[Partition[Table[
  cascadehipw[db8, k], {k, 1, 6}], 2]]
```



Approximating the wavelet function

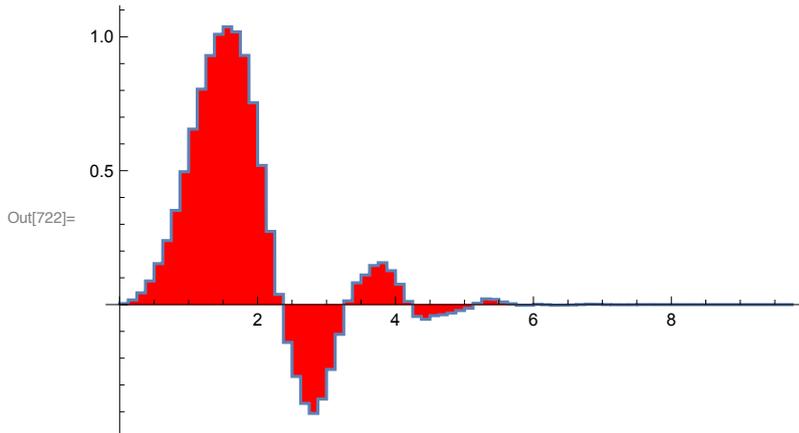
```
In[489]:= GraphicsGrid[Partition[Table[
  cascadepspw[db8, k], {k, 1, 6}], 2]]
```



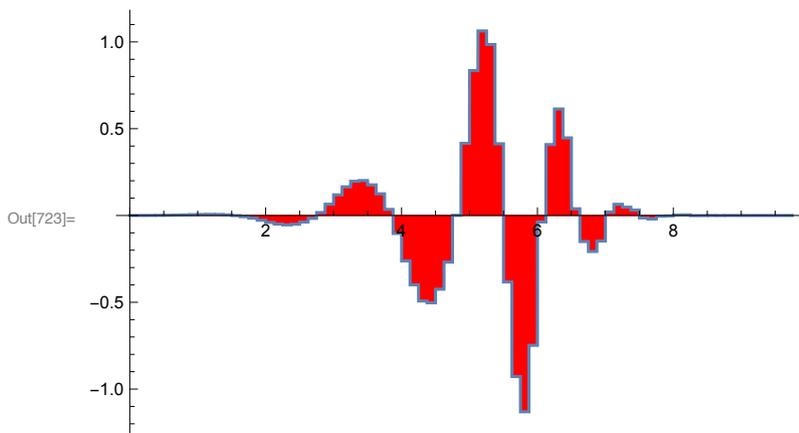
Example Daubechies-I2

```
In[718]:= db12 = WaveletFilterCoefficients[DaubechiesWavelet[6]];
Row[Table[TableForm[ db12[[3 k - 2 ;; 3 k]], ta], {k, 1, 4}], " "]
Out[719]:=
  0  0.0788712  3  0.222916  6  0.068944  9  0.000391626
  1  0.349752  4  -0.159993  7  0.0194616 10  0.00337803
  2  0.531132  5  -0.091759  8  -0.0223319 11  -0.000761767
```

```
In[722]:= cascadehipw[db12, 3]
```

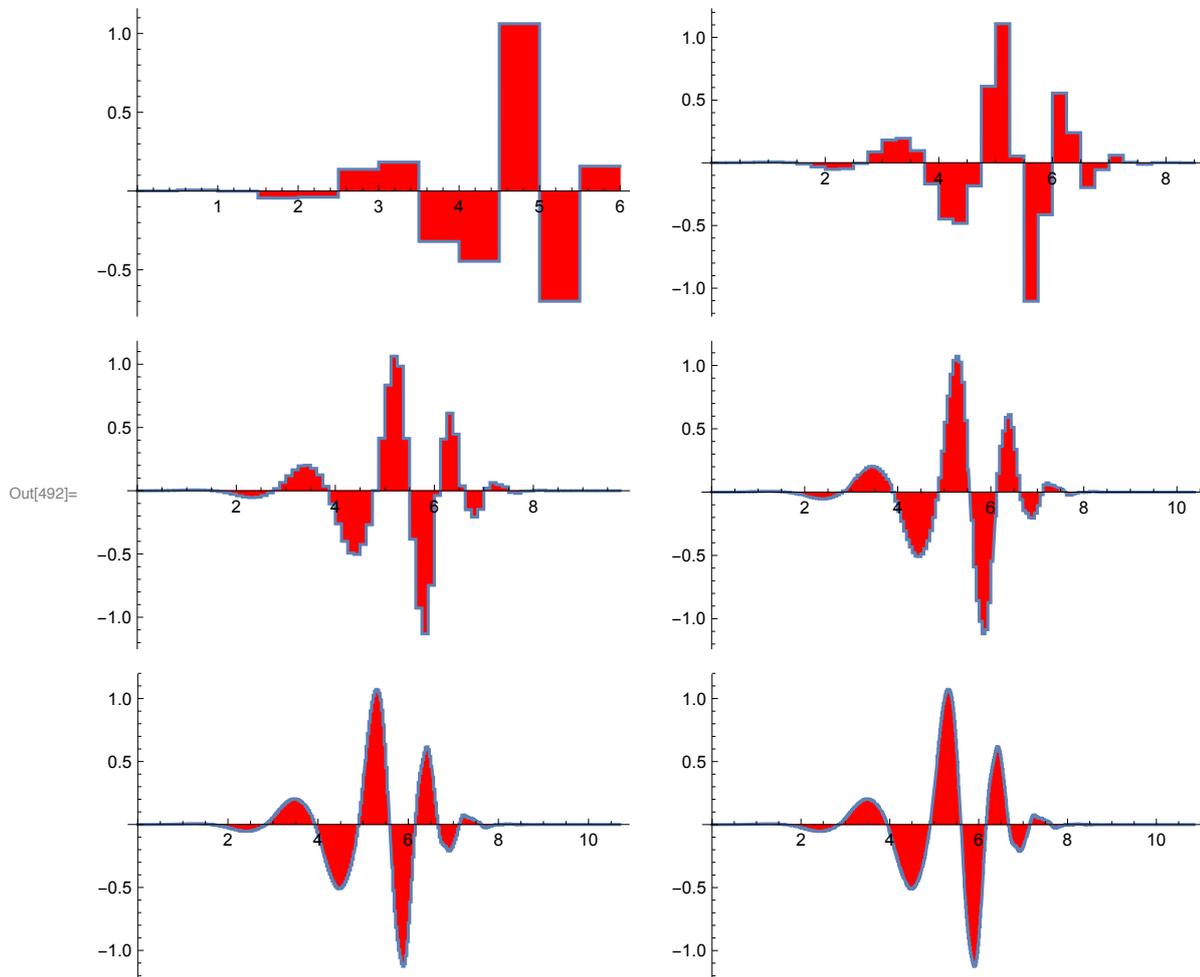


```
In[723]:= cascadepsipw[db12, 3]
```



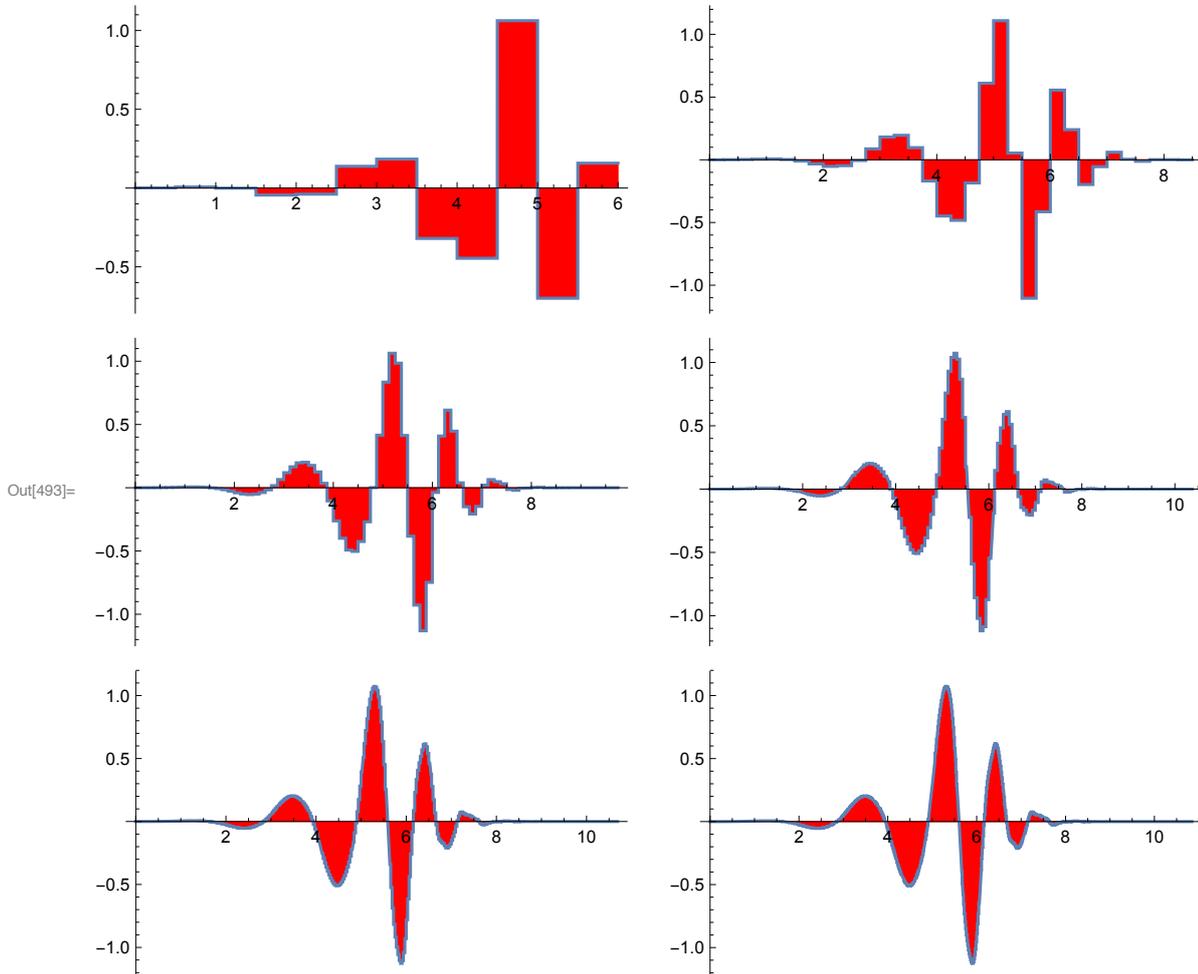
Approximating the scaling function

```
In[492]:= GraphicsGrid[Partition[Table[
  cascadepsipw[db12, k], {k, 1, 6}], 2]]
```



Approximating the wavelet function

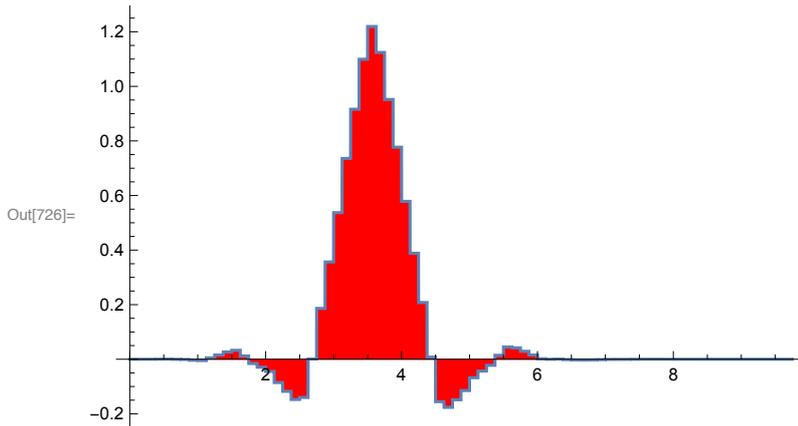
```
In[493]:= GraphicsGrid[Partition[Table[
  cascadepsipw[db12, k], {k, 1, 6}], 2]]
```



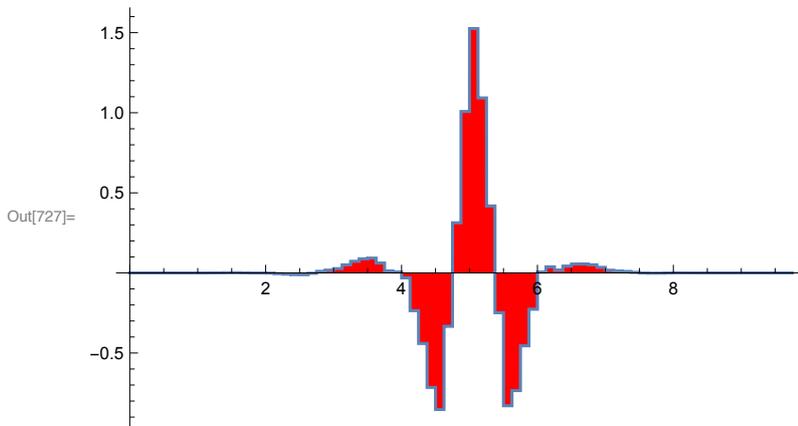
Example Coiflet-12

```
In[724]:= coif12 = WaveletFilterCoefficients[CoifletWavelet[2]];
Row[Table[TableForm[ coif12[[3 k - 2 ;; 3 k]], ta], {k, 1, 4}], "  "]
Out[725]:=
-4    0.0115876   -1    0.273021   2    -0.0540856   5    0.00396788
-3    -0.0293201   0    0.574682   3    -0.0420265   6    -0.0012892
-2    -0.0476396   1    0.294867   4    0.0167444   7    -0.000509505
```

```
In[726]:= cascadehipw[coif12, 3]
```

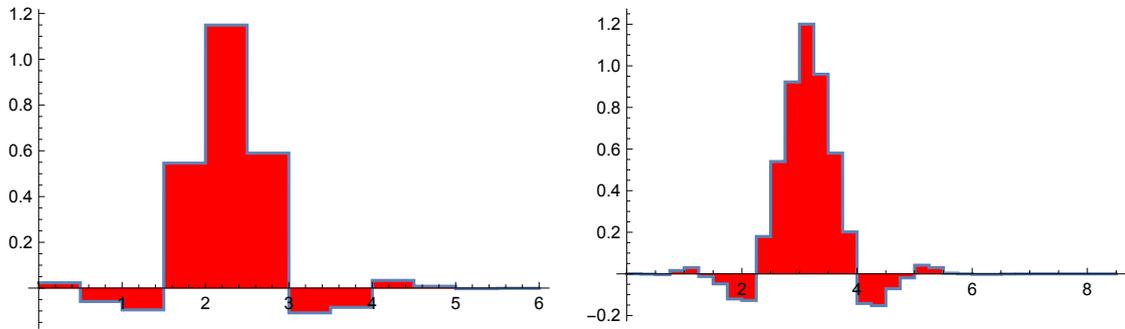


```
In[727]:= cascadepsipw[coif12, 3]
```

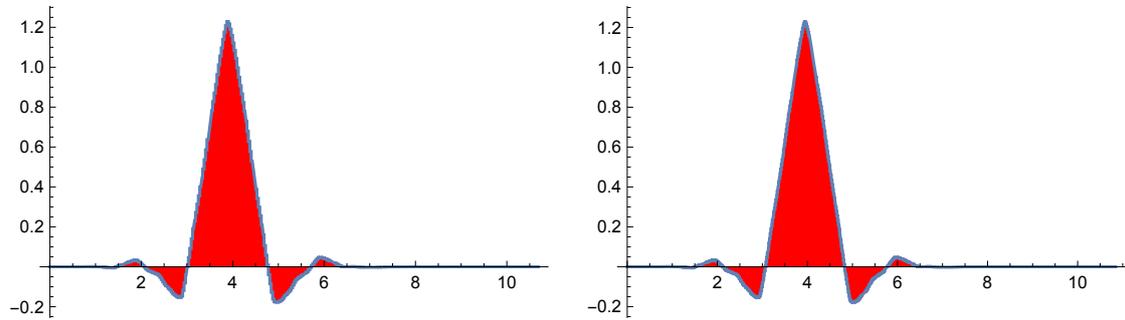
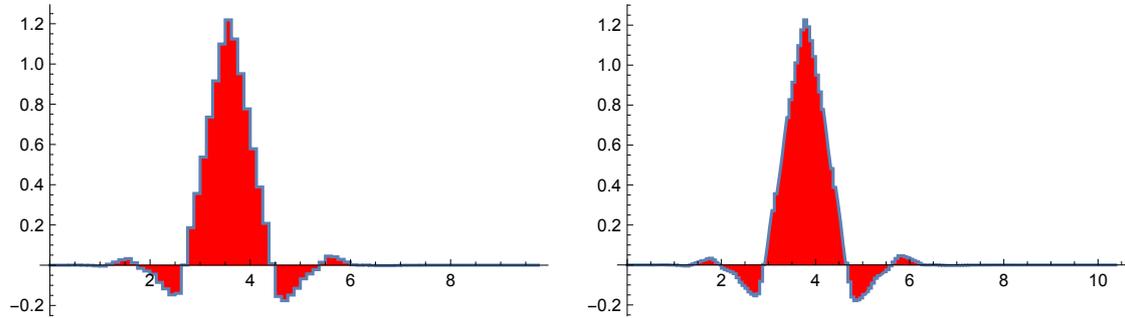


Approximating the scaling function

```
In[496]:= GraphicsGrid[Partition[Table[
  cascadehipw[coif12, k], {k, 1, 6}], 2]]
```

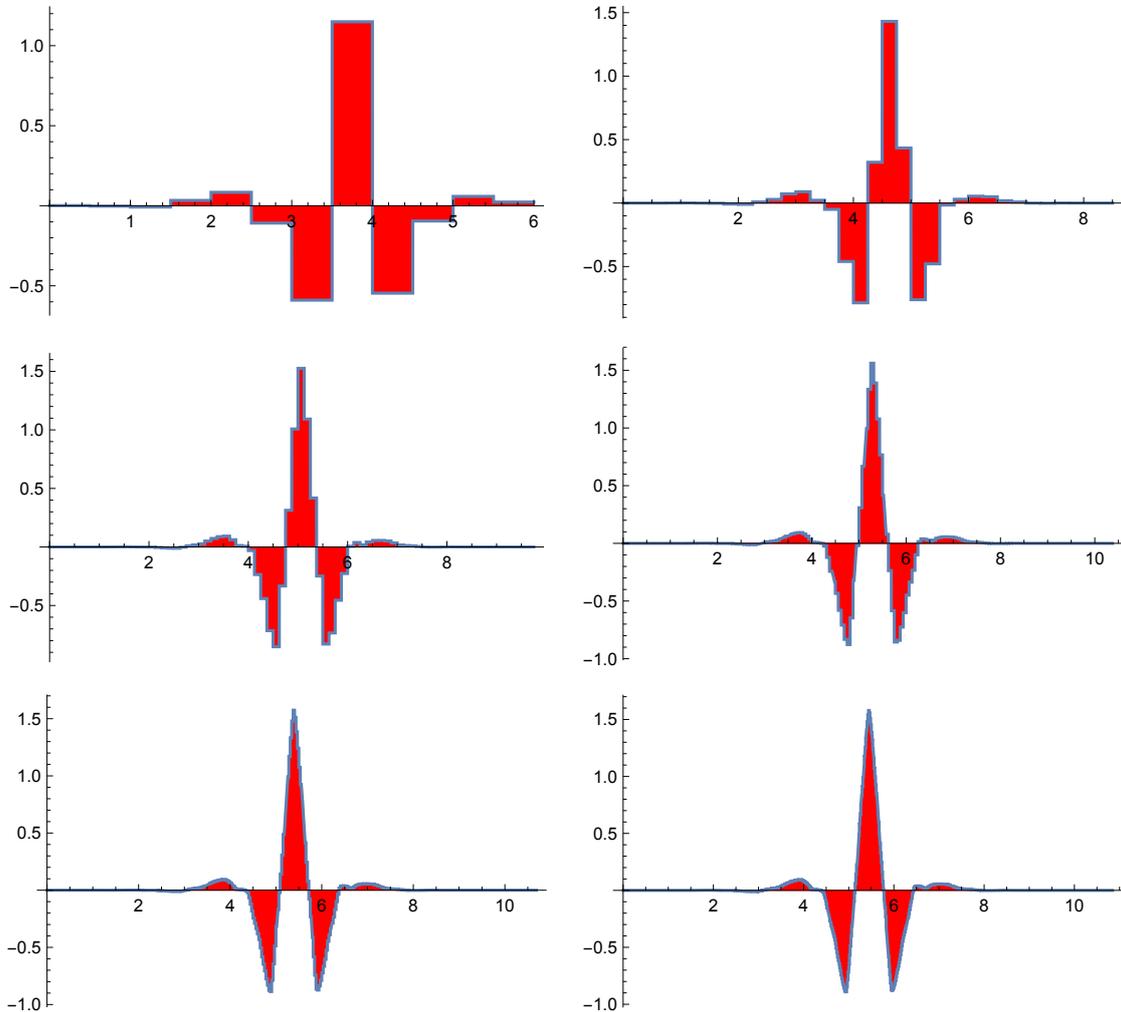


Out[496]=



Approximating the wavelet function

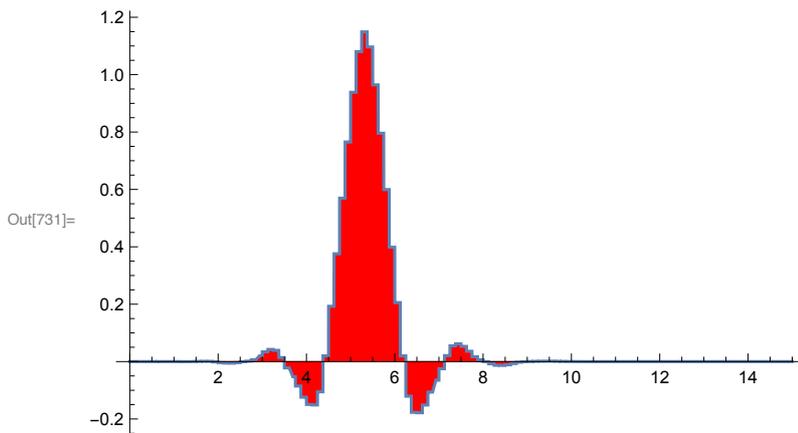
```
In[497]:= GraphicsGrid[Partition[Table[
  cascadepsipw[coif12, k], {k, 1, 6}], 2]]
```



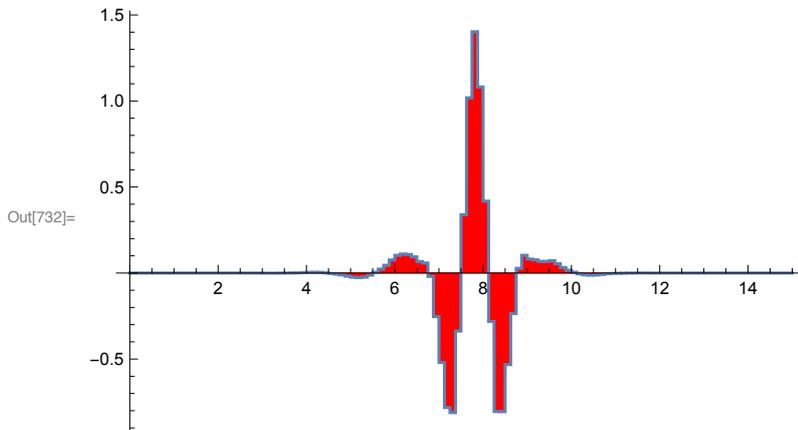
Example Coiflet-18

```
In[728]:= coif18 = WaveletFilterCoefficients[CoifletWavelet[3]];
          Row[Table[TableForm[ coif18[[6 k - 5 ;; 6 k]], ta], {k, 1, 3}], "  "]
          -6   -0.00268242   0   0.561285   6   -0.0063696
          -5   0.00550313   1   0.302984   7   -0.00182046
Out[729]=  -4   0.0165836   2   -0.0507701   8   0.000790205
          -3   -0.0465078   3   -0.0581963   9   0.000329665
          -2   -0.0432208   4   0.0244341   10  -0.0000501928
          -1   0.286503     5   0.0112292   11  -0.0000244657
```

```
In[731]:= cascadehipw[coif18, 3]
```

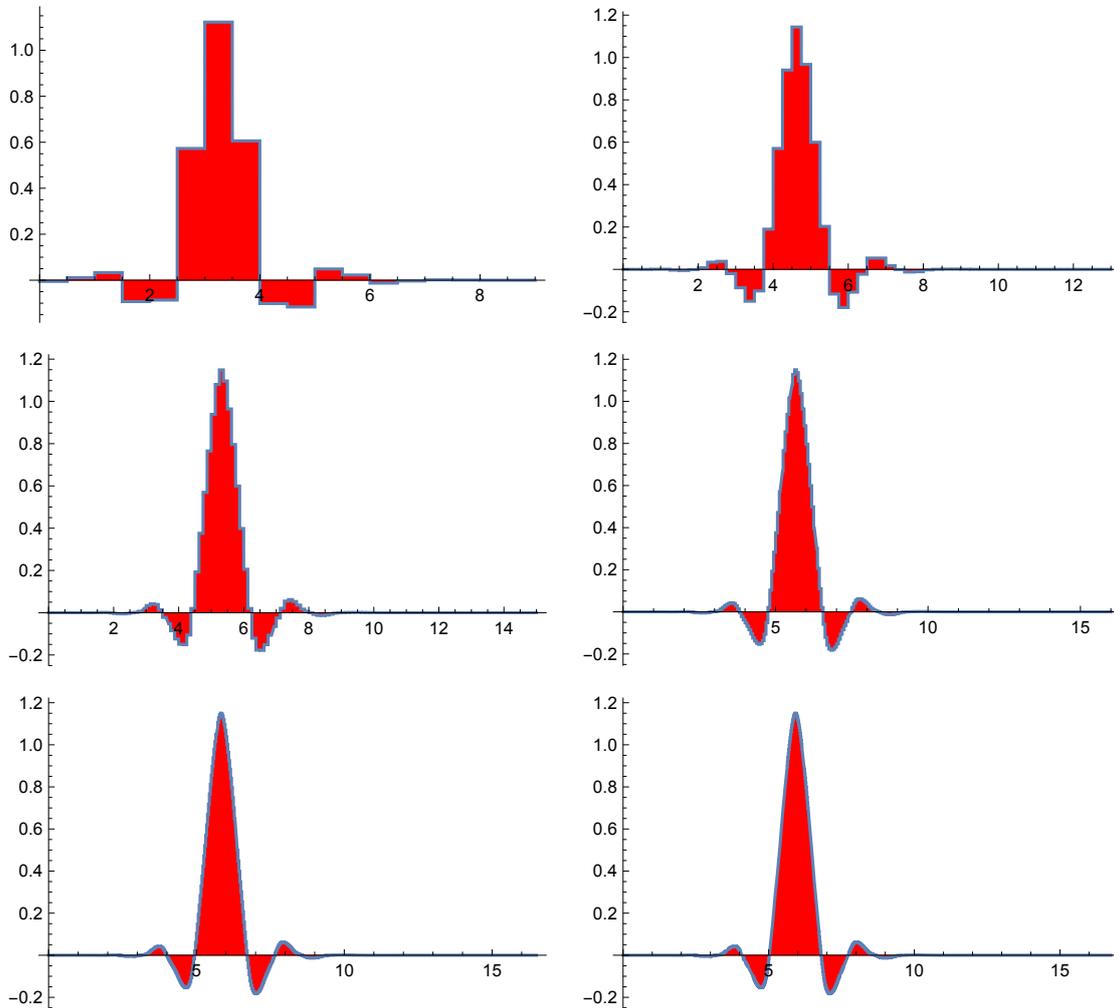


```
In[732]:= cascadepsipw[coif18, 3]
```



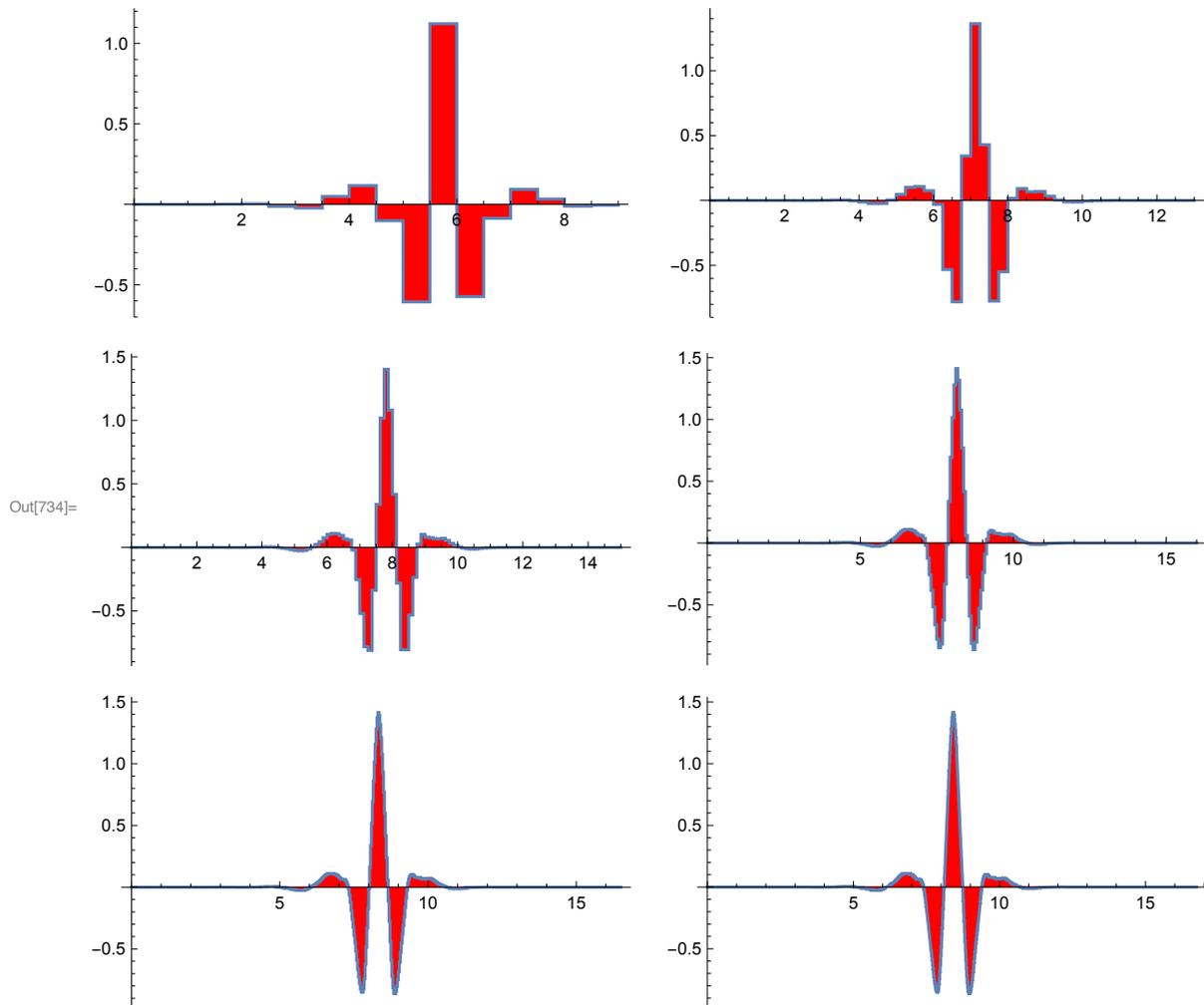
Approximating the scaling function

```
In[733]:= GraphicsGrid[Partition[Table[
  cascadehipw[coif18, k], {k, 1, 6}], 2]]
```



Approximating the wavelet function

```
In[734]:= GraphicsGrid[Partition[Table[
  cascadepsipw[coif18, k], {k, 1, 6}], 2]]
```



Dyadic approximation of the scaling and wavelet functions

The procedures

Solving the eigenvalue problem

```
eigen[h_] := Module[{A, vec, sum, hlen, len},
  hlen = Length[h];
  len = hlen - 2;
  A = Table[
    If[1 <= 2 i + 1 - j <= hlen, h[[2 i + 1 - j]], 0], {i, 1, len} ,
    {j, 1, len}];
  vec = Eigenvectors[A, 1][[1]];
  sum = Total[vec];
  vec / sum
]
```

`eigen::usage` = "Solves the eigenvalue problem belonging to the scaling equation given by the filter `h`.
Needed for starting the dyadic interpolation for the true values of the scaling function belonging to `h`";

Interpolation by iterating the scaling equation

```
dphi[h_,j_] :=
Module[{hlen,hvec,vec,tmp,vlen},
hlen=Length[h];
hvec=Chop[eigen[h]];
vec[0]=Join[{0},hvec,{0}];
Do[
vlen=Length[vec[s-1]];
tmp=Table[
Sum[h[[n]]
(vec[s-1])[[Min[Max[2 k-2^(s-1) (n-1),1],vlen]]],
{n,1, hlen}],
{k,1,vlen-1}];
vec[s]=Riffle[vec[s-1],tmp,{s,1,j}];
vec[j]
]
```

```
dyadicphi[h_,j_]:=
Module[{out},
hlen=Length[h];
out=Table[{k/2^j,dphi[h,j][[k+1]]},{k,0,2^(j) (hlen-1)}];
ListPlot[out,mylpstyle[j]]
]
```

```
dyadicphi::usage = "level-j dyadic approximation
of the scaling function belonging to the filter h";
```

```
dyadicpsi[h_,j_]:=
Module[{hlen,vec,vlen,out},
If[j==0,Return[{0}]];
hlen=Length[h];
vec=dphi[h,j-1];
vlen=Length[vec];
out=Table[{k/2^j,
Sum[(-1)^n h[[hlen-n]]
vec[[Min[Max[k+1-2^(j-1) n,1],vlen]]],
{n,0, hlen-1}],
{k,0,2^(j) (hlen-1)}];
ListPlot[out,mylpstyle[j]]
]
```

```
dyadicpsi::usage = "level-j dyadic approximation
of the wavelet function belonging to the filter h";
```

Plotstyle options

```
mylpstyle[t_]={
Filling→Axis,
FillingStyle→Red,
PlotRange→All,
PlotStyle→{Black,PointSize[0.02-0.002*t]},
ImageSize→Scaled[0.6]};
```

```
ta = TableAlignments -> ".";
```

Example Daubechies-4

```
db4m = Map[#[[1]], 2 #[[2]]] &, WaveletFilterCoefficients[DaubechiesWavelet[2]];
TableForm[db4m, ta]
```

```
0    0.683013
1    1.18301
2    0.316987
3    -0.183013
```

```
db4m = db4m[[All, 2]]
```

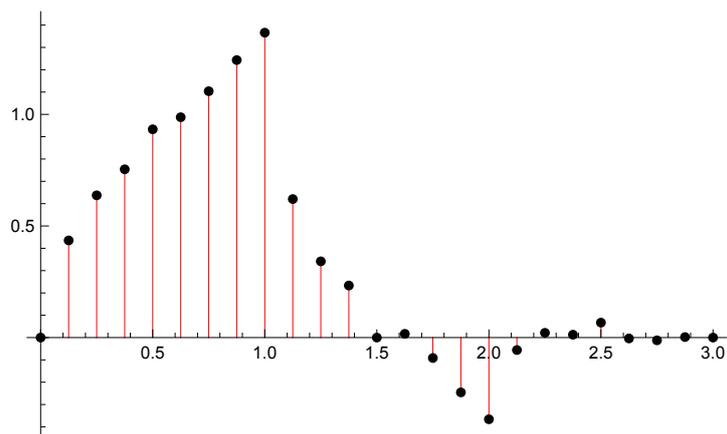
```
{0.683013, 1.18301, 0.316987, -0.183013}
```

Solving the eigenvalue problem

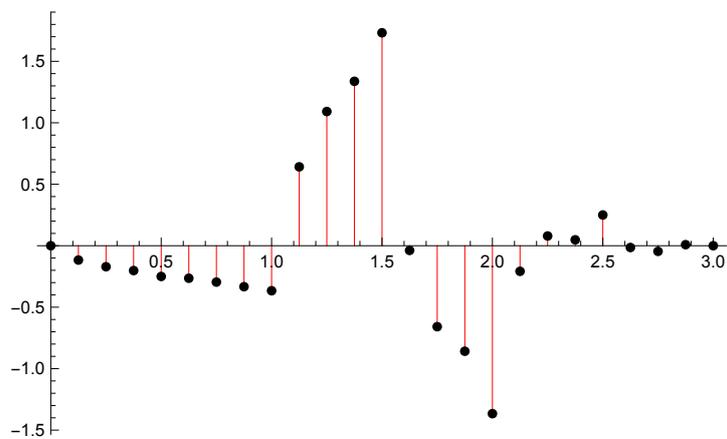
```
eigen[db4m]
```

```
{1.36603, -0.366025}
```

```
dyadicphi[db4m, 3]
```

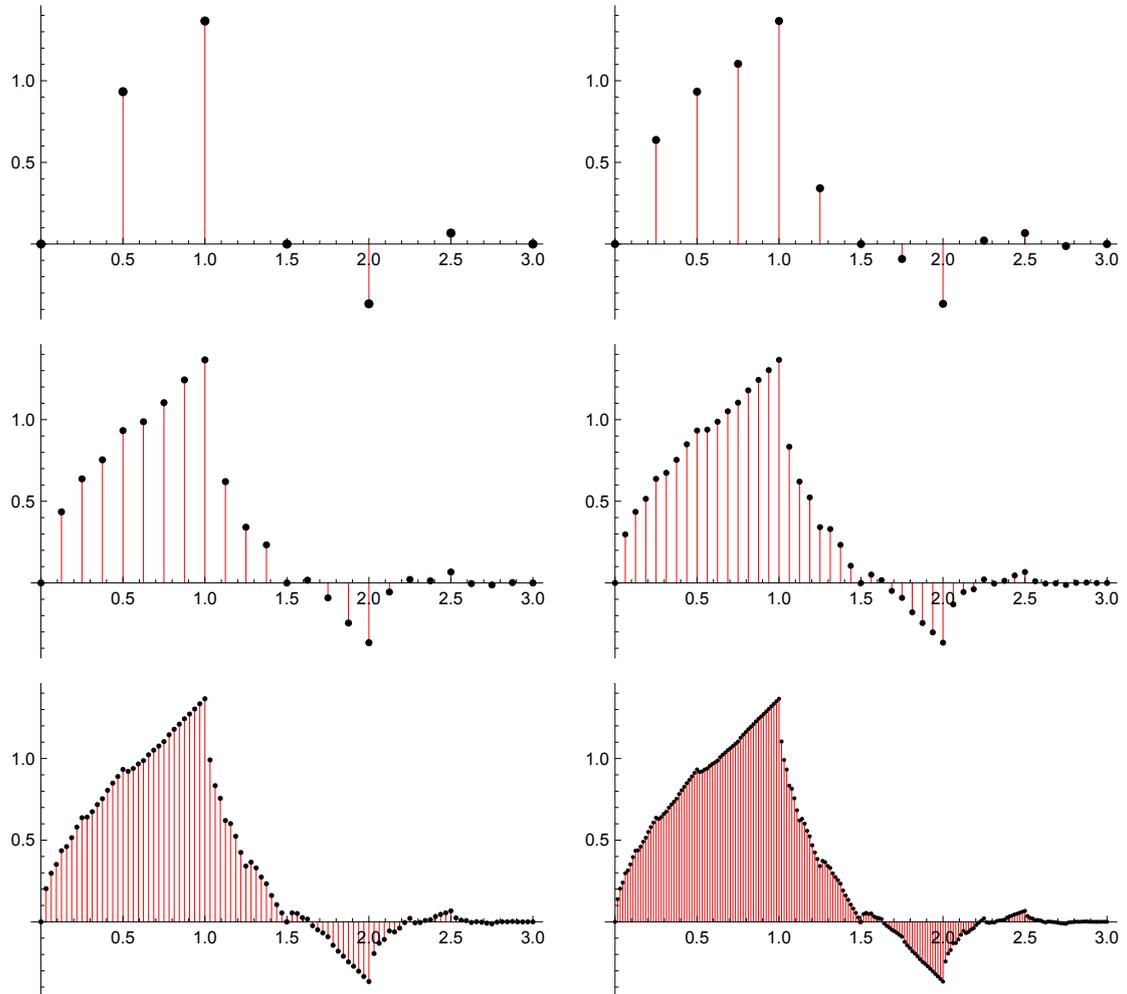


```
dyadicpsi[db4m, 3]
```



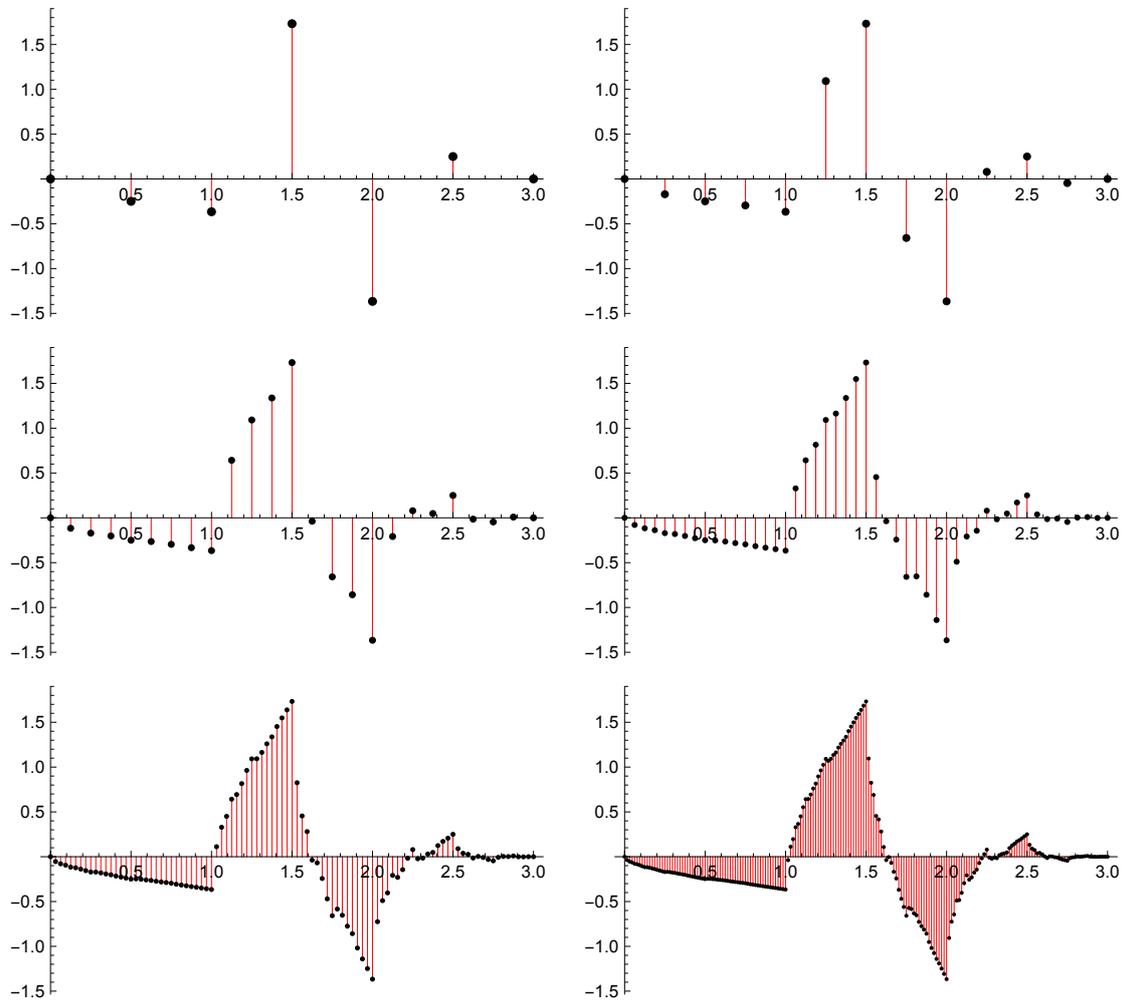
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[db4m, t], {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[db4m, t], {t, 1, 6}], 2]]
```



Example Daubechies-6

```

db6m =
  Map[#[[1]], 2 #[[2]]] &, WaveletFilterCoefficients[DaubechiesWavelet[3]];
  Row[Table[TableForm[ db6m[[3 k - 2 ;; 3 k]], ta], {k, 1, 2}], "  "]
0   0.470467   3   -0.190934
1   1.14112   4   -0.120832
2   0.650365   5   0.0498175

db6m = db6m[[All, 2]];

```

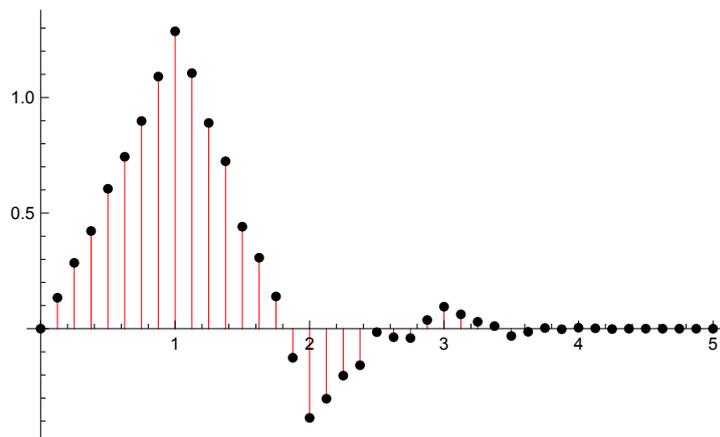
Solving the eigenvalue problem

```

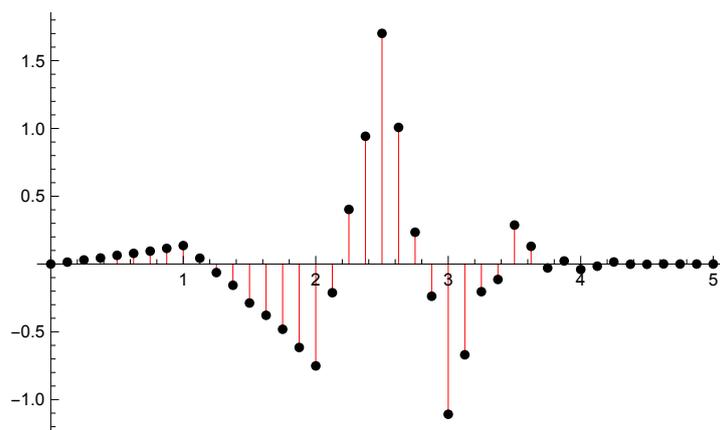
eigen[db6m]
{1.28634, -0.385837, 0.0952675, 0.00423435}

```

```
dyadicphi[db6m, 3]
```

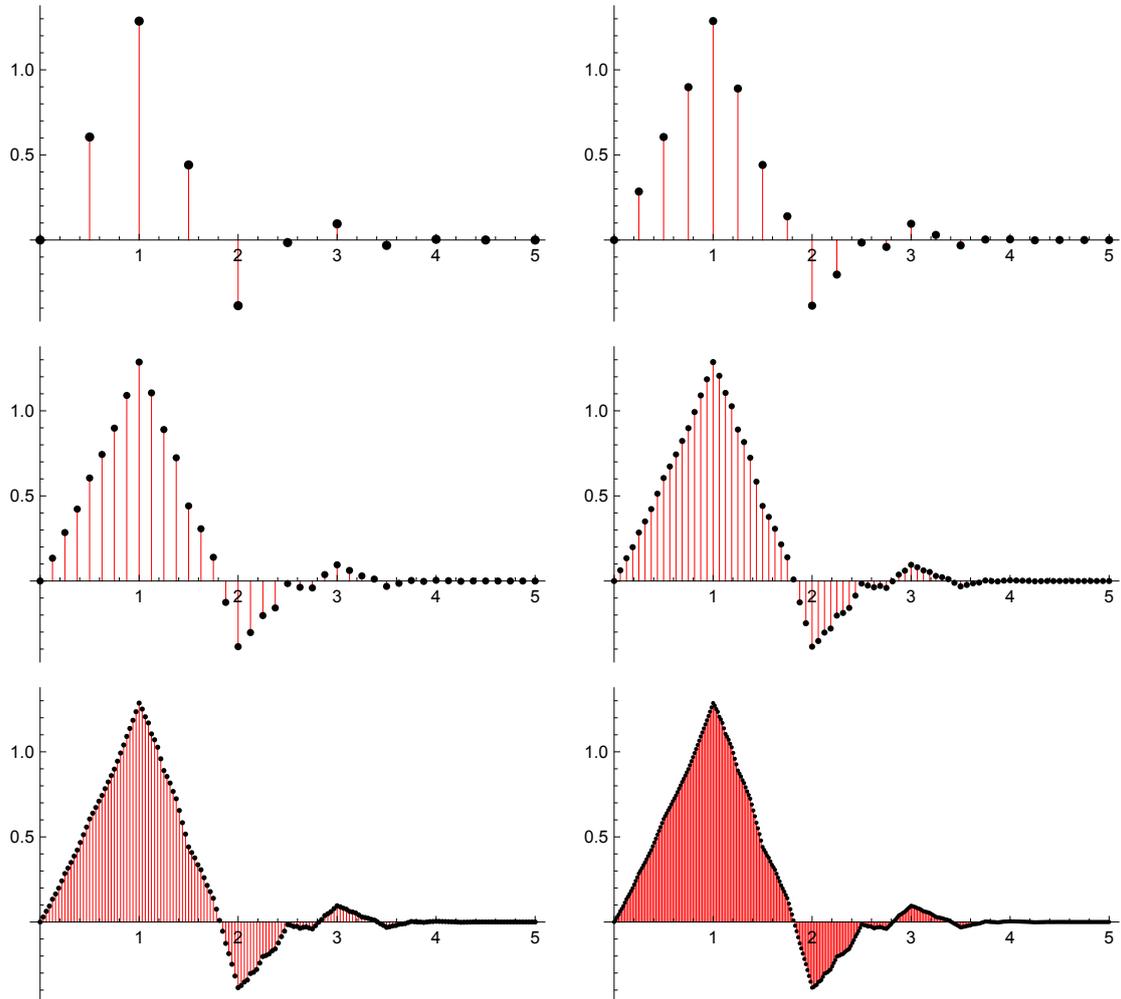


```
dyadicpsi[db6m, 3]
```



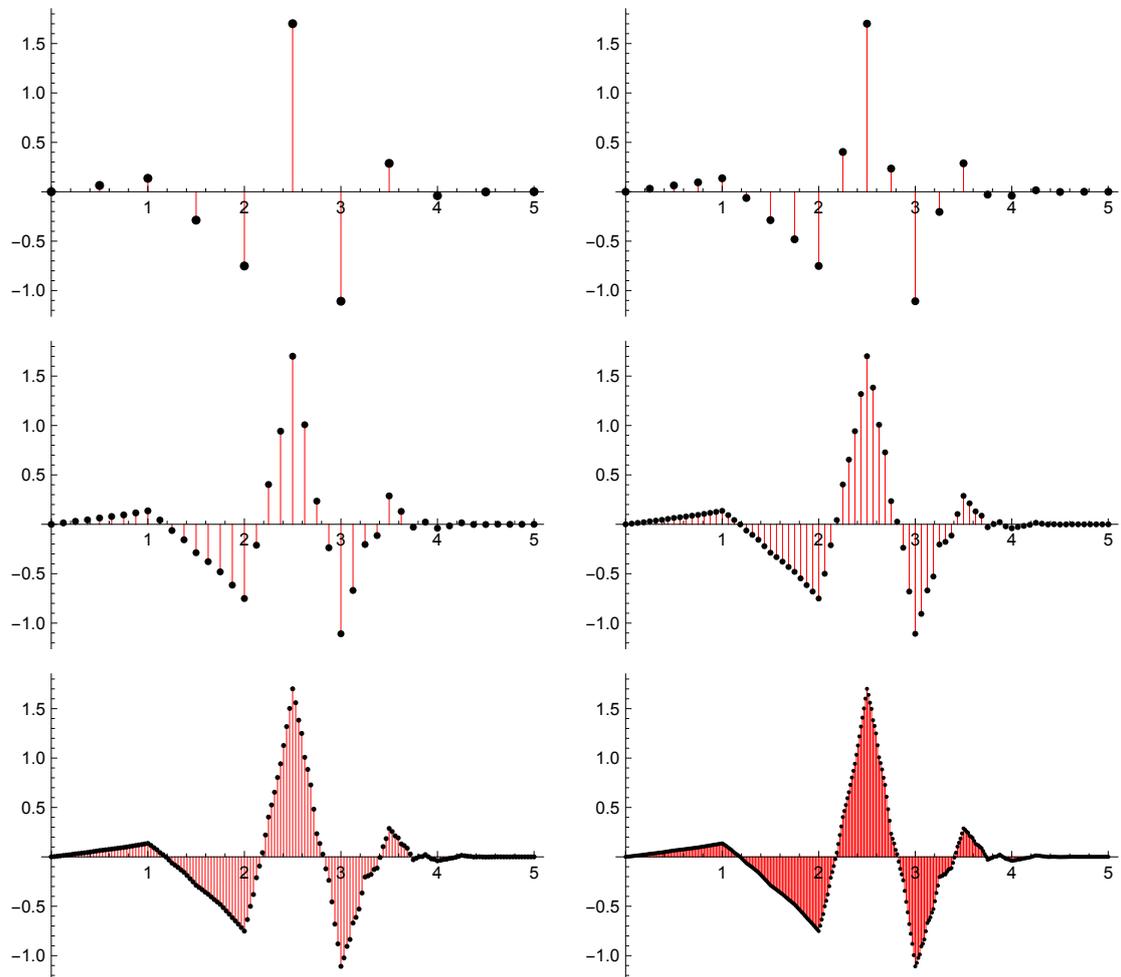
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[db6m, t], {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[db6m, t], {t, 1, 6}], 2]]
```



Example Daubechies-8

```

db8m =
  Map[#[[1]], 2 #[[2]]] &, WaveletFilterCoefficients[DaubechiesWavelet[4]];
Row[Table[TableForm[ db8m[[4 k - 3 ;; 4 k]], ta], {k, 1, 2}], "  " ]
0      0.325803   4      -0.264507
1      1.01095   5       0.0436163
2      0.8922    6       0.0465036
3     -0.039575  7     -0.014987

db8m = db8m[[All, 2]];

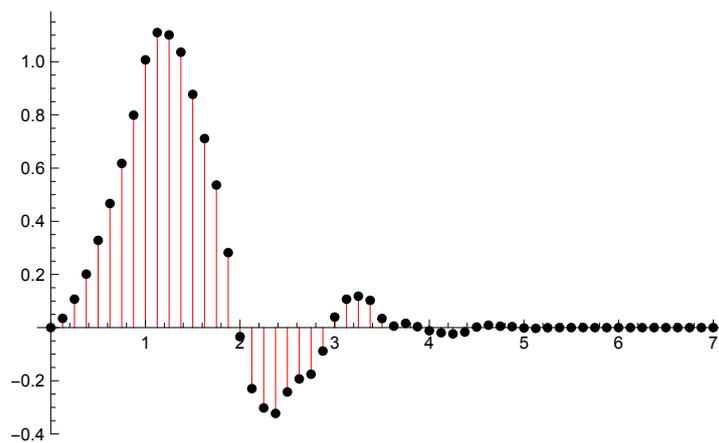
```

Solving the eigenvalue problem

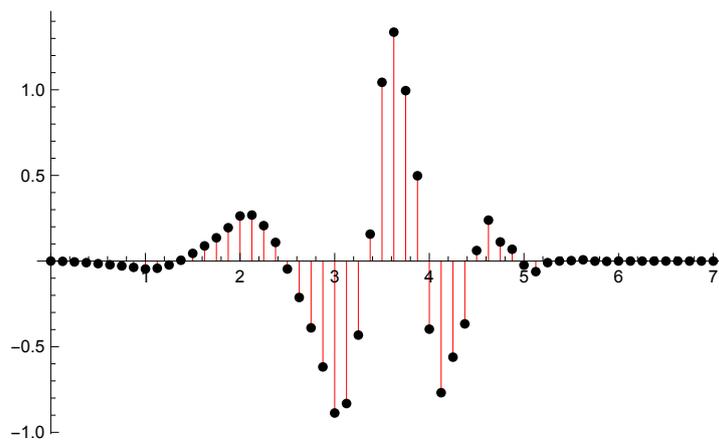
```
eigen[db8m]
```

```
{1.00717, -0.033837, 0.0396105, -0.0117644, -0.00119796, 0.0000188294}
```

```
dyadicphi[db8m, 3]
```

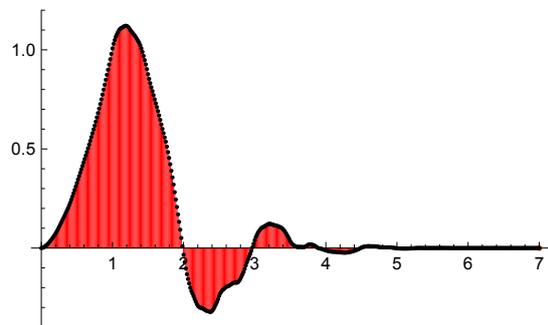
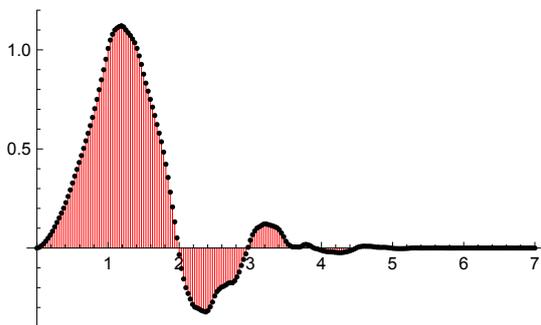
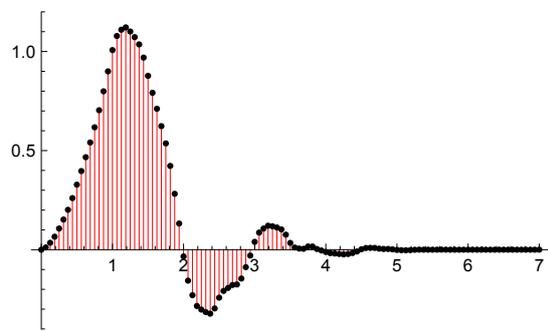
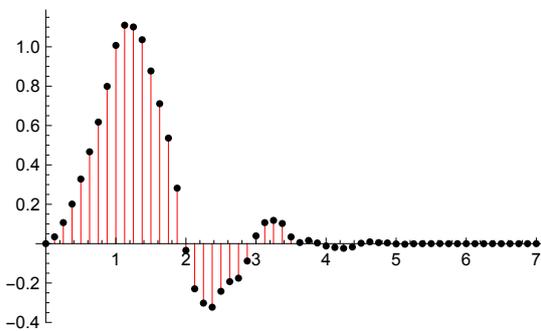
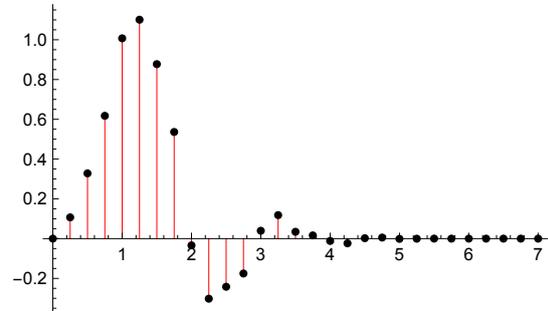
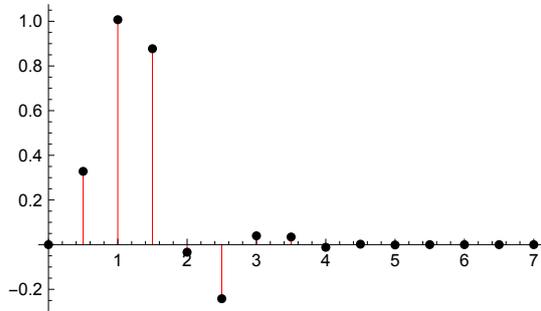


```
dyadicpsi[db8m, 3]
```



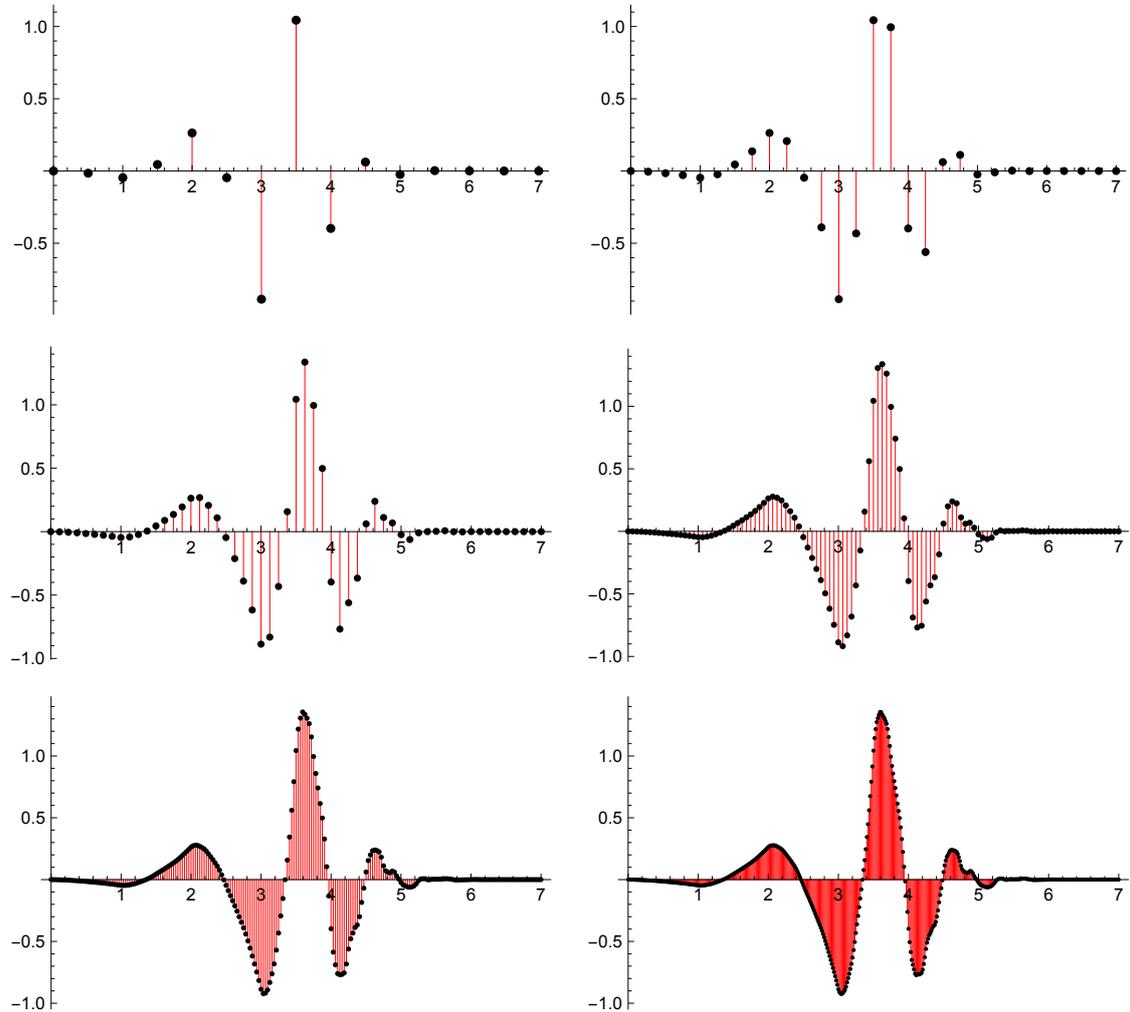
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[db8m, t], {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[db8m, t], {t, 1, 6}], 2]]
```



Example Daubechies-I2

```

db12m =
  Map[#[[1]], 2#[[2]]] &, WaveletFilterCoefficients[DaubechiesWavelet[6]];
Row[Table[TableForm[ db12m[[4 k - 3 ;; 4 k]], ta], {k, 1, 3}], "  "]
0   0.157742   4   -0.319987   8   -0.0446637
1   0.699504   5   -0.183518   9   0.000783251
2   1.06226    6   0.137888   10  0.00675606
3   0.445831   7   0.0389232  11  -0.00152353

db12m = db12m[[All, 2]];

```

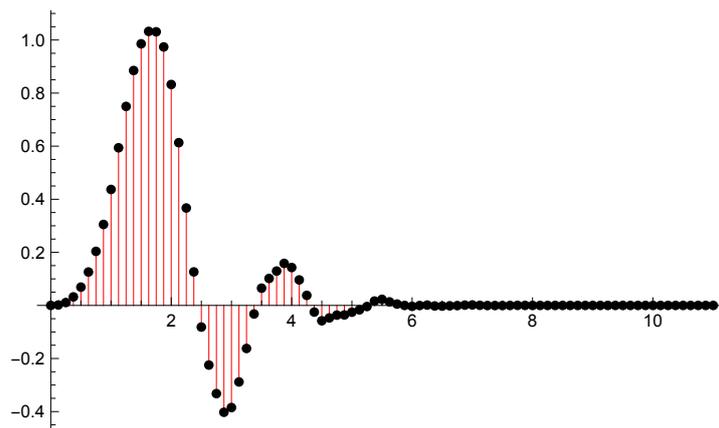
Solving the eigenvalue problem

```

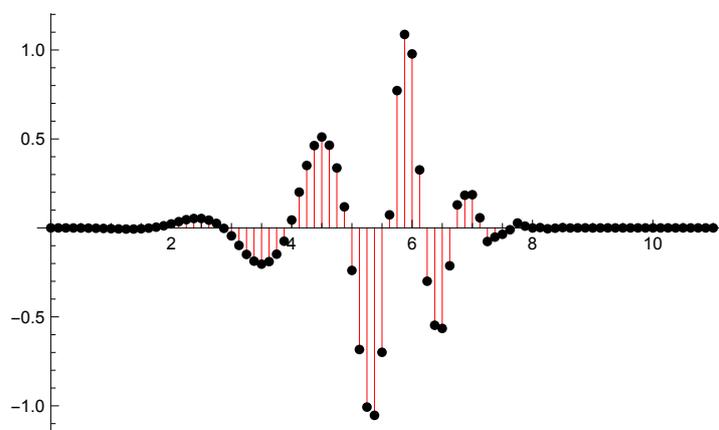
eigen[db12m]
{0.436912, 0.832309, -0.384754, 0.142798, -0.0255074,
 -0.00353052, 0.00175977, 0.0000155912, -2.57792 × 10-6, 3.95427 × 10-9}

```

```
dyadicphi[db12m, 3]
```

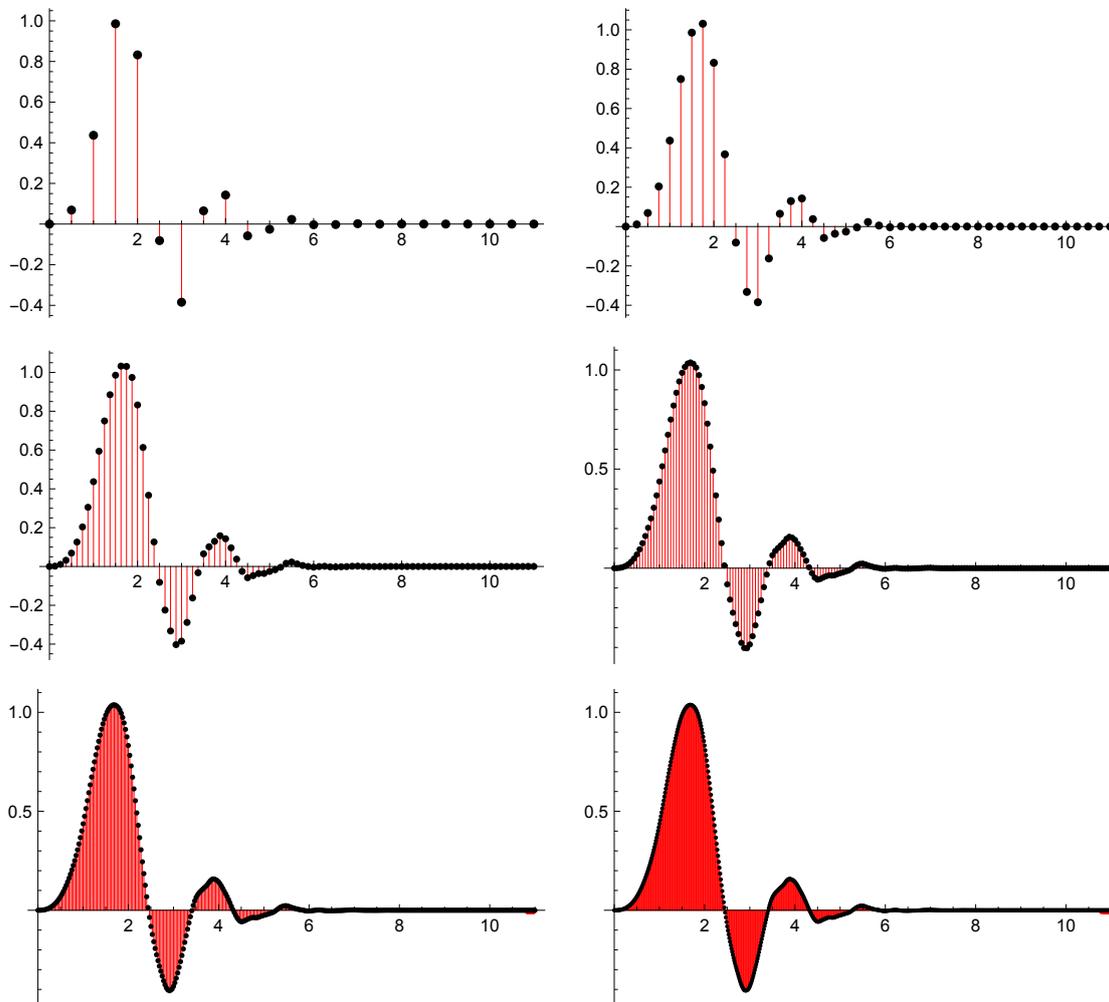


```
dyadicpsi[db12m, 3]
```



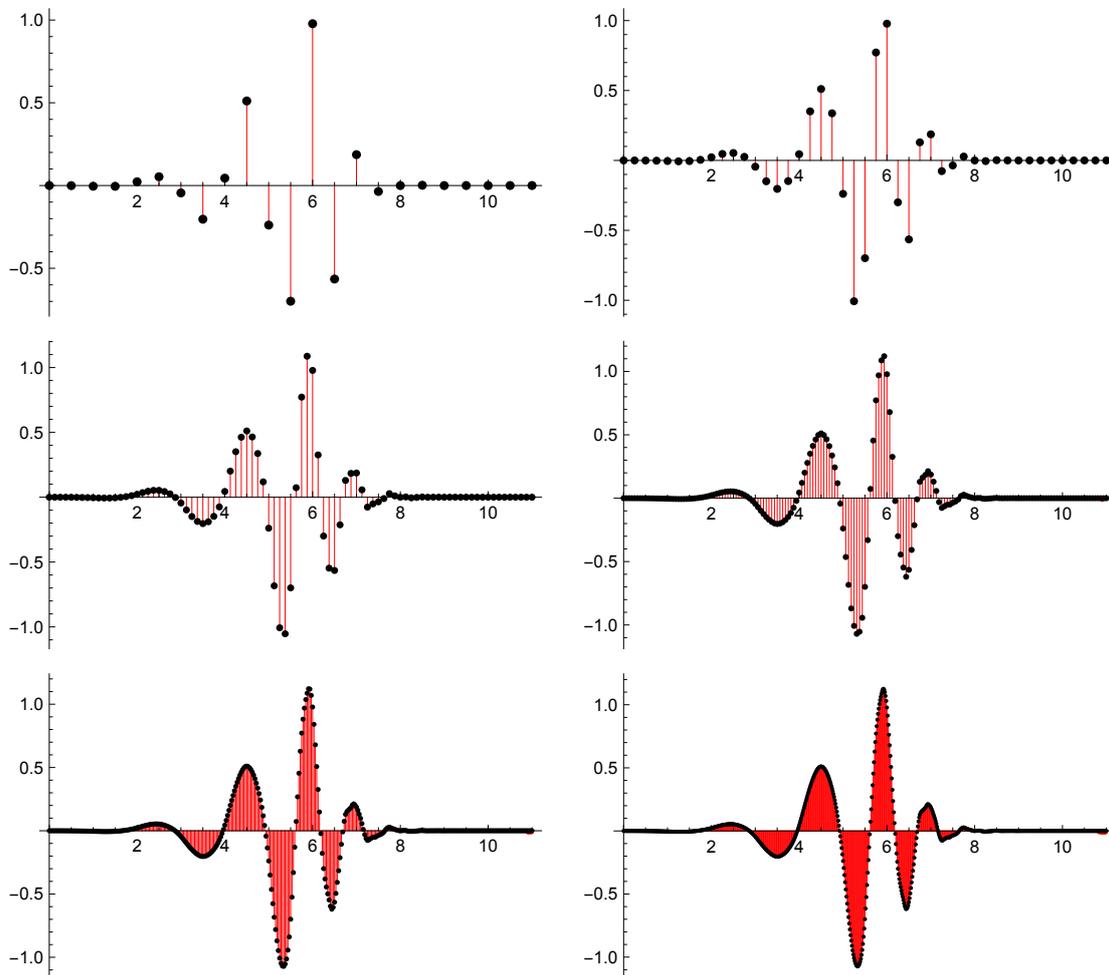
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[db12m, t], {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[db12m, t], {t, 1, 6}], 2]]
```



Example Coiflet-12

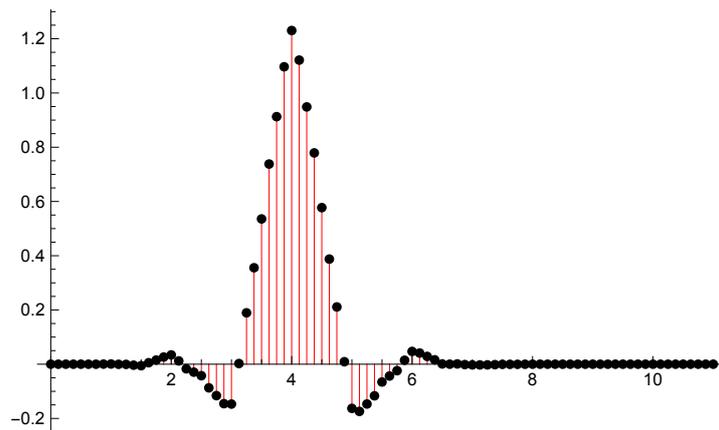
```
coif12m =
  Map[#[[1]], 2 #[[2]]] &, WaveletFilterCoefficients[CoifletWavelet[2]]];
Row[Table[TableForm[coif12m[[4 k - 3 ;; 4 k]], ta], {k, 1, 3}], "  "]
-4    0.0231752    0    1.14936    4    0.0334888
-3   -0.0586403    1    0.589734   5    0.00793577
-2   -0.0952792    2   -0.108171   6   -0.00257841
-1    0.546042     3   -0.084053   7   -0.00101901

coif12m = coif12m[[All, 2]];
```

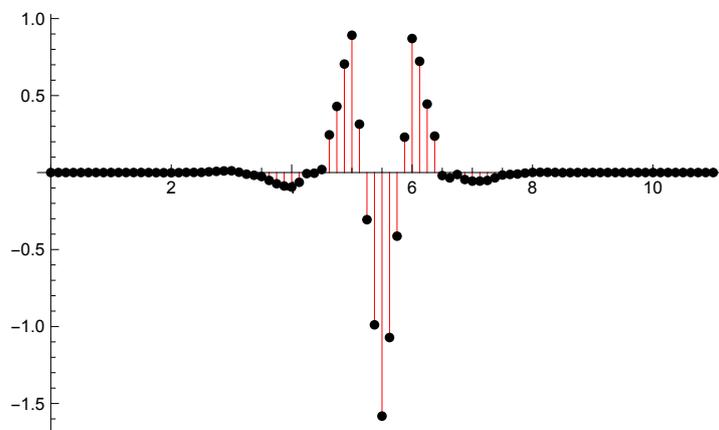
Solving the eigenvalue problem

```
eigen[coif12m]
{0.000750387 + 0. i, 0.0342776 + 0. i, -0.147015 + 0. i,
 1.23031 + 0. i, -0.162912 + 0. i, 0.0470858 + 0. i, -0.00252724 + 0. i,
 0.0000251784 + 0. i, 2.53036 × 10-6 + 0. i, -2.57183 × 10-9 + 0. i}
```

```
dyadicphi[coif12m, 3]
```

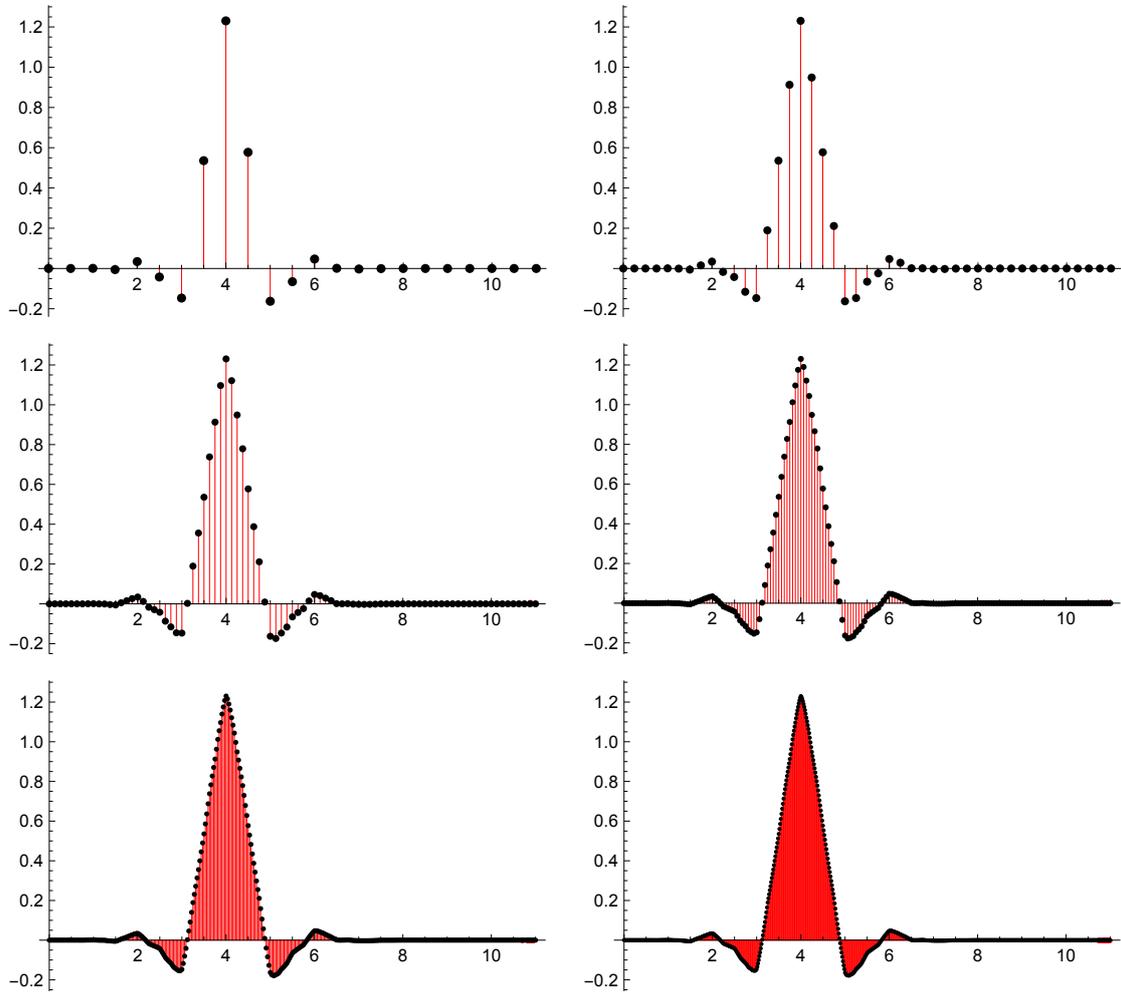


```
dyadicpsi[coif12m, 3]
```



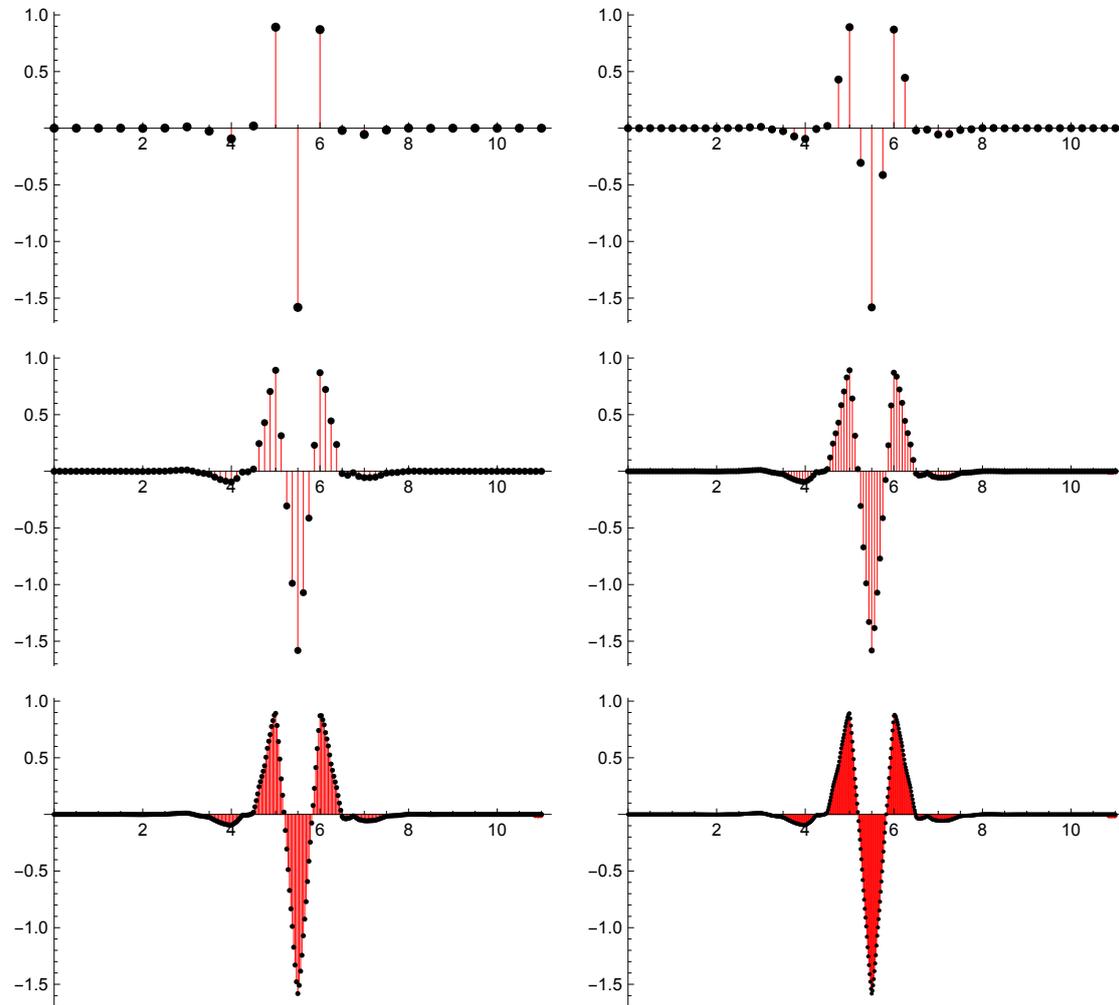
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[coif12m, t], {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[coif12m, t], {t, 1, 6}], 2]]
```



Example Coiflet-24

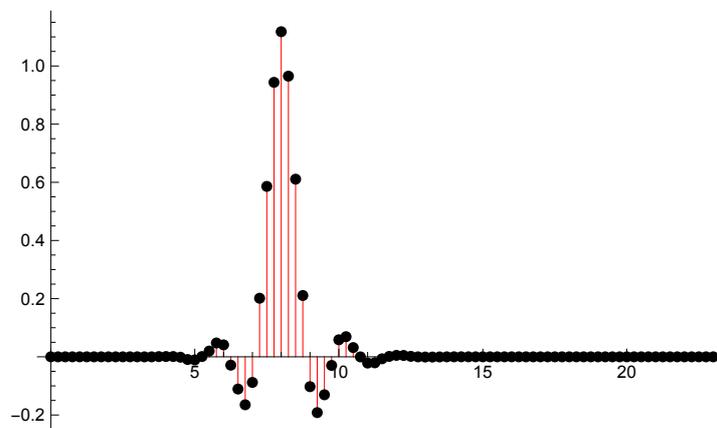
```
coif24m =
  Map[#[[1]], 2#[[2]]] &, WaveletFilterCoefficients[CoifletWavelet[4]];
Row[Table[TableForm[coif24m[[6 k - 5 ;; 6 k]], ta], {k, 1, 4}], "  "]
-8    0.00126192   -2   -0.0793053    4    0.0556273    10   -0.000833
-7    -0.00230445  -1    0.587335    5    0.0354717    11   -0.000367659
-6    -0.0103891   0     1.10625     6   -0.0215126    12   0.0000881605
-5    0.0227249    1     0.614315    7   -0.00800202   13   0.0000441657
-4    0.0377345    2    -0.0942255   8    0.00530533   14   -4.60984 × 10-6
-3    -0.114928    3    -0.136076    9    0.00179119   15   -2.52436 × 10-6

coif24m = coif24m[[All, 2]];
```

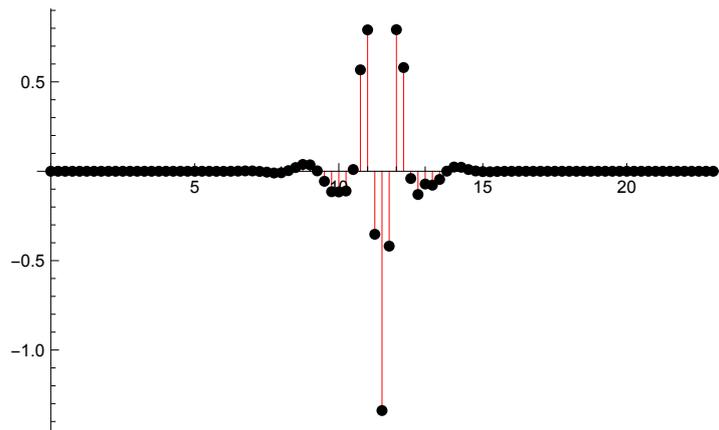
Solving the eigenvalue problem

```
eigen[coif24m]
{1.33974 × 10-9 + 0. i, 1.06411 × 10-6 + 0. i, 0.000067639 + 0. i, 0.000975503 + 0. i,
 -0.0105753 + 0. i, 0.0410692 + 0. i, -0.0880722 + 0. i, 1.11781 + 0. i,
 -0.102635 + 0. i, 0.0585417 + 0. i, -0.0214361 + 0. i, 0.00490709 + 0. i,
 -0.0000724169 + 0. i, 0.0000752093 + 0. i, -1.38855 × 10-6 + 0. i,
 -3.25612 × 10-7 + 0. i, 6.94007 × 10-9 + 0. i, 1.38763 × 10-9 + 0. i, 5.43309 × 10-12 + 0. i,
 -2.36781 × 10-14 + 0. i, -1.36065 × 10-17 + 0. i, 3.43476 × 10-23 + 0. i}
```

```
dyadicphi[coif24m, 2]
```

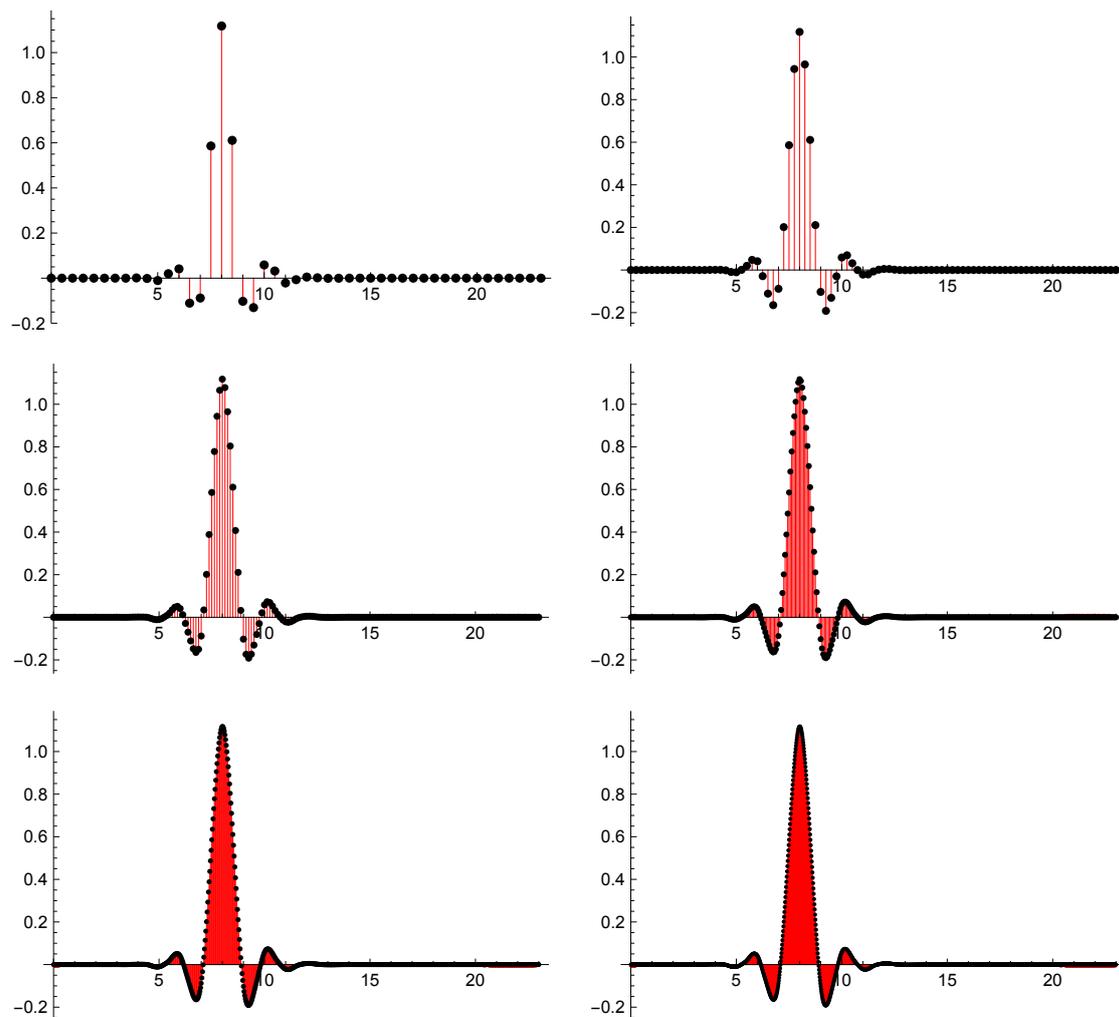


dyadicpsi[coif24m, 2]



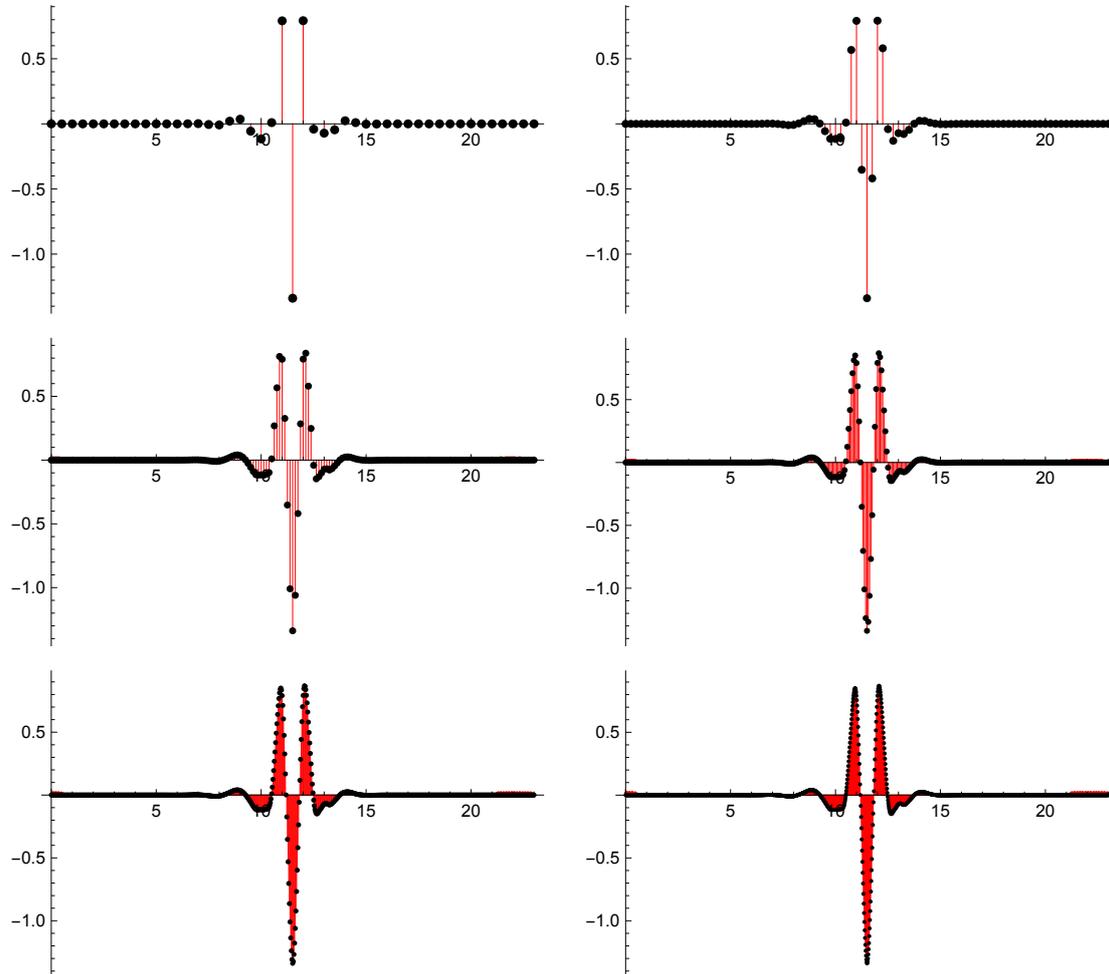
Approximating the scaling function

GraphicsGrid[Partition[Table[
dyadicphi[coif24m, t], {t, 1, 6}], 2]]



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[coif24m, t], {t, 1, 6}], 2]]
```



Example Symlet-6

```
sym6m = Map[#[[1]], 2 #[[2]]] &, WaveletFilterCoefficients[SymletWavelet[6]]];
```

```
sym6m
```

```
{ {0,  $\frac{1}{32}$ }, {1,  $\frac{3}{16}$ }, {2,  $\frac{15}{32}$ }, {3,  $\frac{5}{8}$ }, {4,  $\frac{15}{32}$ }, {5,  $\frac{3}{16}$ }, {6,  $\frac{1}{32}$ }, {7, 0} }
```

```
sym6m = sym6m[[All, 2]]
```

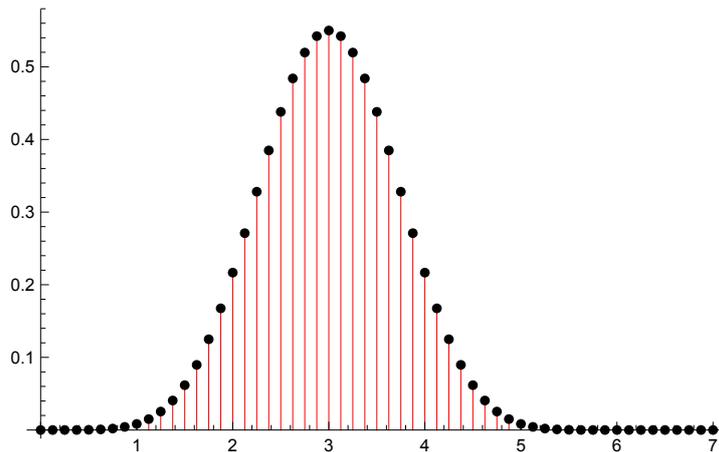
```
{  $\frac{1}{32}$ ,  $\frac{3}{16}$ ,  $\frac{15}{32}$ ,  $\frac{5}{8}$ ,  $\frac{15}{32}$ ,  $\frac{3}{16}$ ,  $\frac{1}{32}$ , 0 }
```

Solving the eigenvalue problem

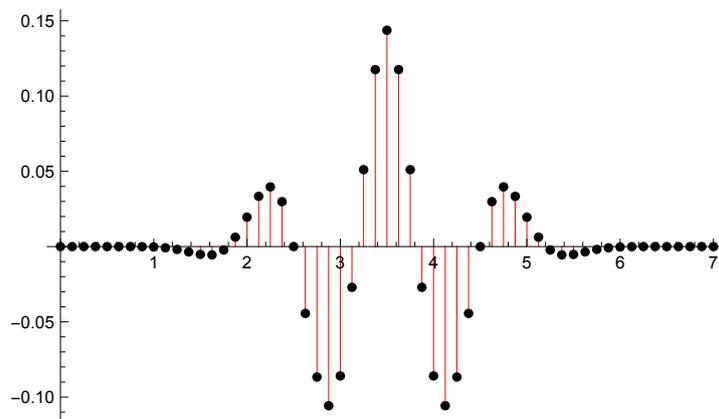
```
eigen[sym6m]
```

```
{  $\frac{1}{120}$ ,  $\frac{13}{60}$ ,  $\frac{11}{20}$ ,  $\frac{13}{60}$ ,  $\frac{1}{120}$ , 0 }
```

```
dyadicphi[sym6m, 3]
```

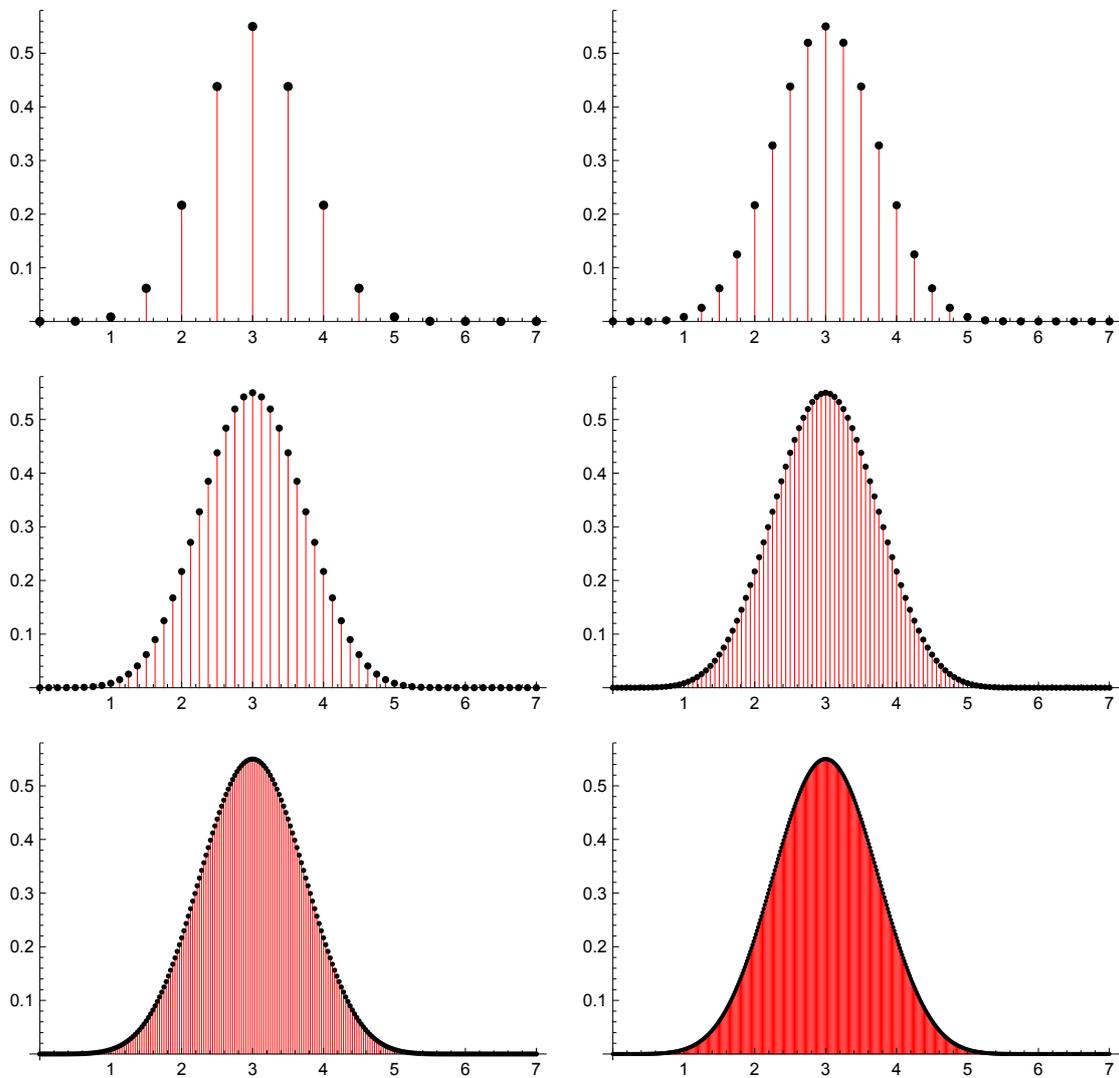


```
dyadicpsi[sym6m, 3]
```



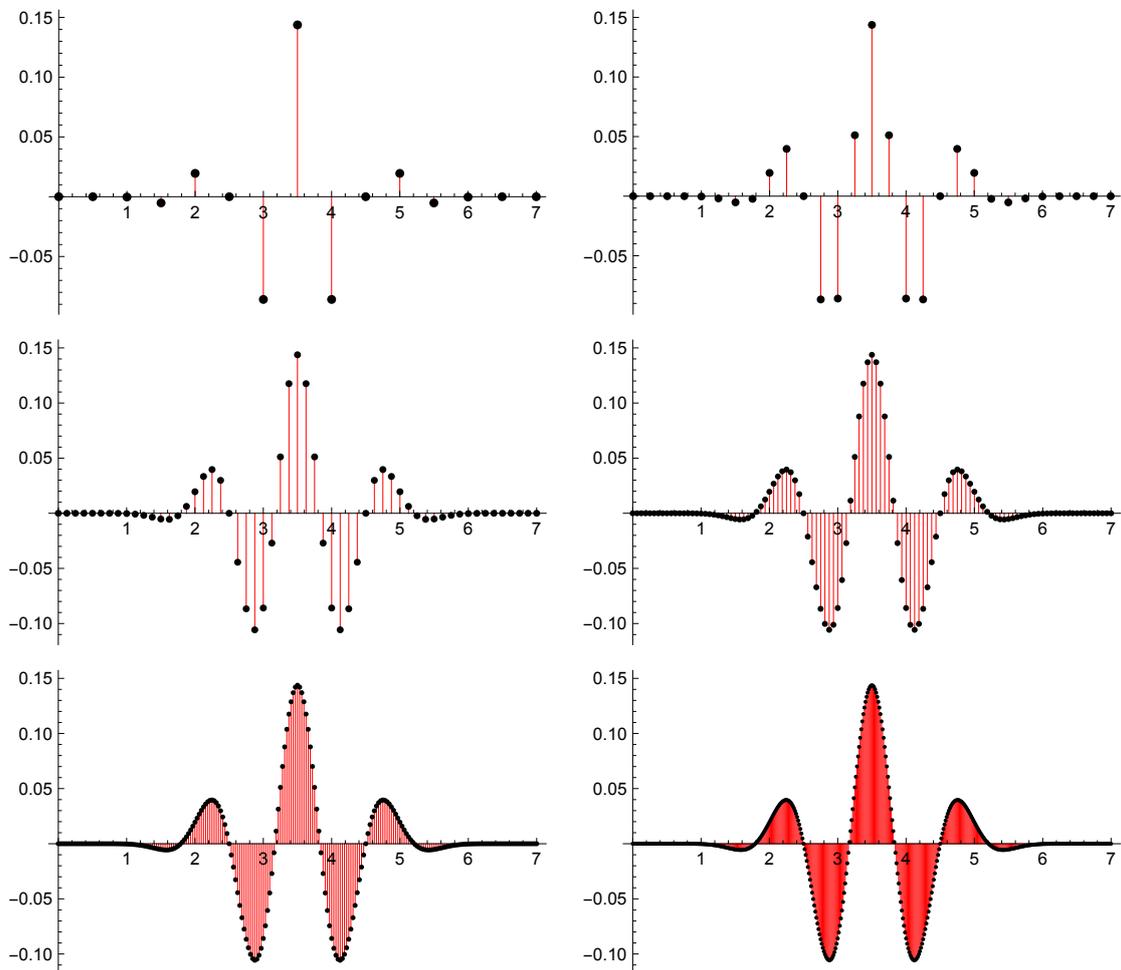
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[sym6m, t], {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[sym6m, t], {t, 1, 6}], 2]]
```



Example Symlet-12

```
sym12m = Map[#[[1]], 2#[[2]]] &, WaveletFilterCoefficients[SymletWavelet[12]]];
```

```
Row[Table[TableForm[sym12m[[2 k - 1 ;; 2 k]], ta], {k, 1, 6}], " "
```

0	$\frac{1}{2048}$	2	$\frac{33}{1024}$	4	$\frac{495}{2048}$	6	$\frac{231}{512}$	8	$\frac{495}{2048}$	10	$\frac{33}{1024}$
1	$\frac{3}{512}$	3	$\frac{55}{512}$	5	$\frac{99}{256}$	7	$\frac{99}{256}$	9	$\frac{55}{512}$	11	$\frac{3}{512}$

```
sym12m = sym12m[[All, 2]]
```

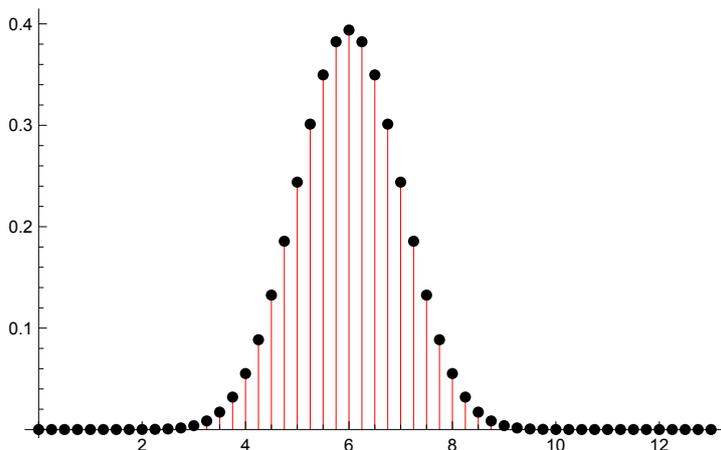
```
{ $\frac{1}{2048}$ ,  $\frac{3}{512}$ ,  $\frac{33}{1024}$ ,  $\frac{55}{512}$ ,  $\frac{495}{2048}$ ,  $\frac{99}{256}$ ,  $\frac{231}{512}$ ,  $\frac{99}{256}$ ,  $\frac{495}{2048}$ ,  $\frac{55}{512}$ ,  $\frac{33}{1024}$ ,  $\frac{3}{512}$ ,  $\frac{1}{2048}$ , 0}
```

Solving the eigenvalue problem

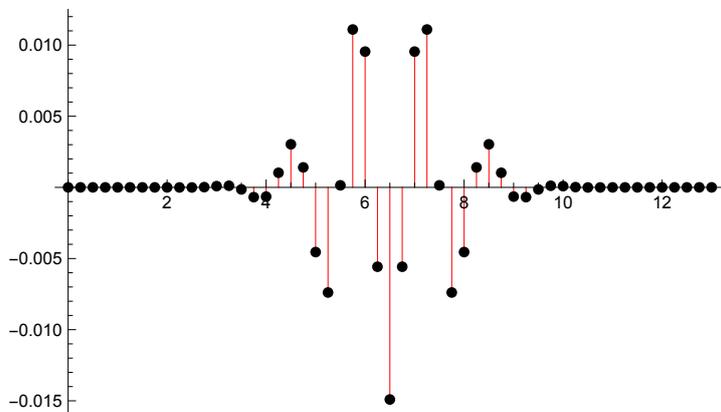
```
eigen[sym12m]
```

```
{ $\frac{1}{39916800}$ ,  $\frac{509}{9979200}$ ,  $\frac{50879}{13305600}$ ,  $\frac{1093}{19800}$ ,  $\frac{1623019}{6652800}$ ,  
 $\frac{655177}{1663200}$ ,  $\frac{1623019}{6652800}$ ,  $\frac{1093}{19800}$ ,  $\frac{50879}{13305600}$ ,  $\frac{509}{9979200}$ ,  $\frac{1}{39916800}$ , 0}
```

```
dyadicphi[sym12m, 2]
```

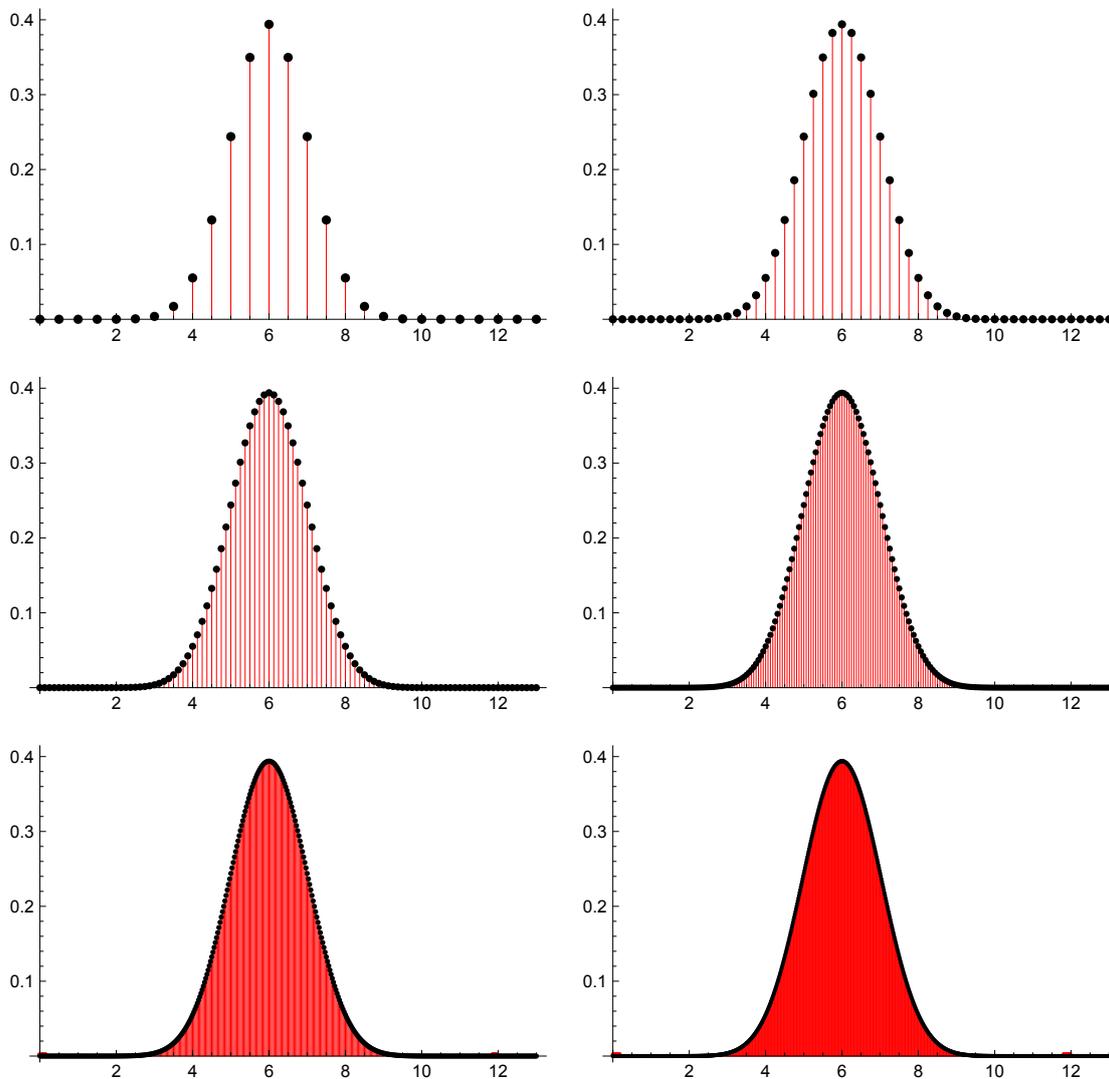


```
dyadicpsi[sym12m, 2]
```



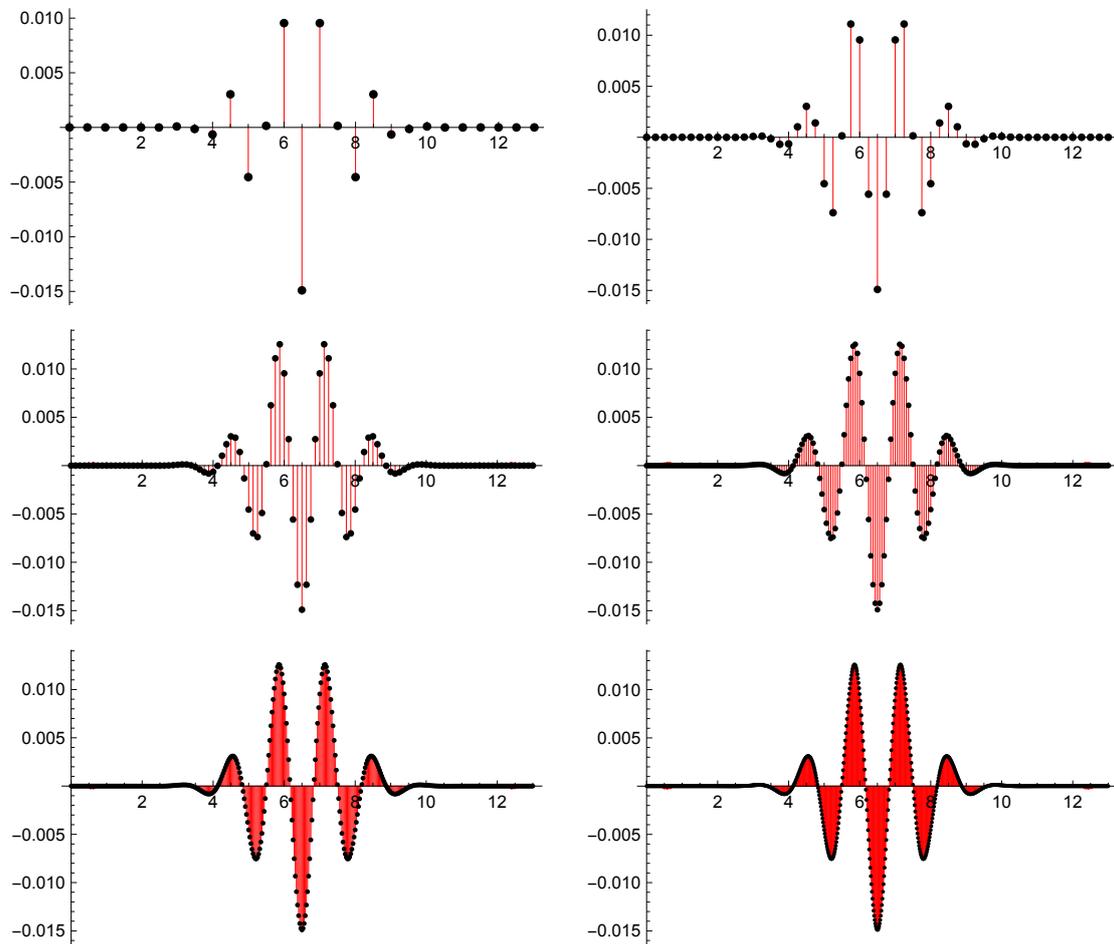
Approximating the scaling function

```
GraphicsGrid[Partition[Table[
  dyadicphi[sym12m, t] , {t, 1, 6}], 2]]
```



Approximating the wavelet function

```
GraphicsGrid[Partition[Table[
  dyadicpsi[sym12m, t], {t, 1, 6}], 2]]
```



10. Biorthogonale Filterpaare

Das bisher behandelte Schema der *orthogonalen* Wavelettransformationen mit Filtern endlicher Länge $L + 1$ basierte auf Filterpaaren

$$\begin{array}{ll} \text{Tiefpassfilter} & \mathbf{h} = (h_0, h_1, \dots, h_L), \\ \text{Hochpassfilter} & \mathbf{g} = (g_0, g_1, \dots, g_L), \end{array}$$

die eine orthogonale Transformation von Signalen (endlicher Länge) definieren, dargestellt in Matrizenform

$$W_N = \begin{bmatrix} H_N \\ G_N \end{bmatrix} \quad \text{mit} \quad W_N^{-1} = W_N^\dagger,$$

wie im Kapitel 7 ausführlich beschrieben. Die Orthogonalität $I_N = W_N W_N^\dagger$ bzw. $I_N = W_N^\dagger W_N$ stellt sich in den Matrixgleichungen

$$\begin{aligned} G_N G_N^\dagger &= I_{N/2} = H_N H_N^\dagger \\ G_N H_N^\dagger &= 0_{N/2} = H_N G_N^\dagger \\ I_N &= G_N^\dagger G_N + H_N^\dagger H_N \end{aligned}$$

dar. Die dritte dieser Gleichungen ist gerade die Rekonstruktionsaussage.

Im Formalismus der Frequenzdarstellungen sind diese Informationen in den Gleichungen

$$\begin{aligned} |H(\omega)|^2 + |H(\omega + \pi)|^2 &= 2 \\ |G(\omega)|^2 + |G(\omega + \pi)|^2 &= 2 \\ H(\omega) \overline{G(\omega)} + H(\omega + \pi) \overline{G(\omega + \pi)} &= 0 \\ H(0) = G(\pi) &= \sqrt{2} \\ H(\pi) = G(0) &= 0 \end{aligned}$$

enthalten, wobei die beiden letzten Bedingungen die Tiefpasseigenschaft von \mathbf{h} und die Hochpasseigenschaft von \mathbf{g} ausdrücken.

Ein Grund, von diesem bewährten Schema abzuweichen, liegt in den beiden folgenden Beobachtungen

- Mit *symmetrischen* Filtern (und Wavelets) erhält man oft bessere Rekonstruktionsresultate, z.B. bei der Bildkompression.
- Ausser dem Haar-Filter gibt es keine symmetrischen (Skalierungs-)Filter, mit denen man eine orthogonale Transformation bauen kann.

Der Wunsch nach symmetrischen Filtern führt zu dem allgemeineren Schema der folgenden Abschnitte.

10.1 Biorthogonale Filterpaare

Man geht nun aus von zwei Filterpaaren:

- ein Paar $(\mathbf{h}, \tilde{\mathbf{h}})$ von Tiefpassfiltern;
- ein Paar $(\mathbf{g}, \tilde{\mathbf{g}})$ von Hochpassfiltern;

Über die Länge und den Indexbereich der Filterkoeffizienten ist erst einmal nichts ausgesagt. Aber es soll sich hier immer um *endliche* Filter handeln. Insbesondere müssen \mathbf{h} und \mathbf{g} nicht gleiche Länge haben.

Die entsprechenden Transformationsmatrizen für Analyse und Synthese

$$\text{Analyse: } W_N = \begin{bmatrix} H_N \\ G_N \end{bmatrix} \qquad \text{Synthese: } \tilde{W}_N = \begin{bmatrix} \tilde{H}_N \\ \tilde{G}_N \end{bmatrix}$$

sollen natürlich invers zueinander sein, $W_N^{-1} = \tilde{W}_N^\dagger$, also:

$$W_N \tilde{W}_N^\dagger = \tilde{W}_N^\dagger W_N = I_N.$$

Dies ist genau die in Abschnitt 2.4 fomulierte Eigenschaft den *Biorthogonalität*, daher die Bezeichnung! Dies bedeutet:

$$\begin{aligned} G_N \tilde{G}_N^\dagger &= I_{N/2} = H_N \tilde{H}_N^\dagger \\ G_N \tilde{H}_N^\dagger &= 0_{N/2} = H_N \tilde{G}_N^\dagger \\ I_N &= \tilde{G}_N^\dagger G_N + \tilde{H}_N^\dagger H_N \end{aligned}$$

Im folgenden wird die Äquivalenz der drei Sichtweisen

Transformationsmatrizen \leftrightarrow Filtergleichungen \leftrightarrow Frequenzdarstellungen

$$\begin{array}{ccc} (H_N, G_N) & (\mathbf{h}, \mathbf{g}) & (H(\omega), G(\omega)) \\ (\tilde{H}_N, \tilde{G}_N) & (\tilde{\mathbf{h}}, \tilde{\mathbf{g}}) & (\tilde{H}(\omega), \tilde{G}(\omega)) \end{array}$$

ohne weiteren Kommentar verwendet. Damit gilt

- (1) $H_N \tilde{H}_N^\dagger = I_N \Leftrightarrow \sum_k \tilde{h}_k h_{k-2m} = \delta_{m,0} \Leftrightarrow \tilde{H}(\omega) \overline{H}(\omega) + \tilde{H}(\omega + \pi) \overline{H}(\omega + \pi) = 2$
- (2) $G_N \tilde{G}_N^\dagger = I_N \Leftrightarrow \sum_k \tilde{g}_k g_{k-2m} = \delta_{m,0} \Leftrightarrow \tilde{G}(\omega) \overline{G}(\omega) + \tilde{G}(\omega + \pi) \overline{G}(\omega + \pi) = 2$
- (3) $H_N \tilde{G}_N^\dagger = 0_N \Leftrightarrow \sum_k \tilde{g}_k h_{k-2m} = 0 \Leftrightarrow \tilde{H}(\omega) \overline{G}(\omega) + \tilde{H}(\omega + \pi) \overline{G}(\omega + \pi) = 0$
- (4) $G_N \tilde{H}_N^\dagger = 0_N \Leftrightarrow \sum_k \tilde{h}_k g_{k-2m} = 0 \Leftrightarrow \tilde{G}(\omega) \overline{H}(\omega) + \tilde{G}(\omega + \pi) \overline{H}(\omega + \pi) = 0$

Definition 10.1 (Biorthogonale Filterpaare).

Ein Paar $(\mathbf{h}, \tilde{\mathbf{h}})$ von (endlichen Tiefpass-)Filtern ist ein biorthogonales Filterpaar, wenn die Identität (1)

$$\tilde{H}(\omega)\overline{H(\omega)} + \tilde{H}(\omega + \pi)\overline{H(\omega + \pi)} = 2$$

erfüllt ist.

Folgende Aussage ergibt sich durch routinemässiges Nachrechnen:

Satz 10.1 Ist $(\mathbf{h}, \tilde{\mathbf{h}})$ biorthogonales Filterpaar und definiert man das Filterpaar $(\mathbf{g}, \tilde{\mathbf{g}})$ mittels

$$G(\omega) = e^{i(n\omega+b)}\overline{\tilde{H}(\omega + \pi)} \quad \tilde{G}(\omega) = e^{i(n\omega+b)}\overline{H(\omega + \pi)}$$

mit ungeradem $n \in \mathbb{Z}$ und $b \in \mathbb{R}$, so sind die Bedingungen (2), (3) und (4) erfüllt.

Diese Festlegung bedeutet auf der Ebene der Filterkoeffizienten:

$$g_k = -e^{ib}(-1)^k \tilde{h}_{n-k}, \quad \tilde{g}_k = -e^{ib}(-1)^k h_{n-k},$$

wobei man üblicherweise $b = \pi$ (weil man die Filterkoeffizienten reell haben möchte) und $n = 1$ wählt, also

$$g_k = (-1)^k \tilde{h}_{1-k}, \quad \tilde{g}_k = (-1)^k h_{1-k}.$$

Man beachte: der Filter \mathbf{h} bestimmt den Filter $\tilde{\mathbf{g}}$ – beide haben insbesondere gleiche Länge – und ebenso bestimmt der Filter $\tilde{\mathbf{h}}$ den Filter \mathbf{g} . Die Längen \mathbf{h} und von $\tilde{\mathbf{h}}$ müssen nicht übereinstimmen, gleichwohl können sie nicht ganz willkürlich gewählt werden – siehe den folgenden Satz.

■ **Beispiel 10.1** Betrachte die Filter

$$\mathbf{h} = \frac{\sqrt{2}}{4}(-2, 4, 3, -2, 1) = (h_{-2}, \dots, h_2) \quad \tilde{\mathbf{h}} = \frac{\sqrt{2}}{4}(1, 2, 1) = (\tilde{h}_{-1}, \tilde{h}_0, \tilde{h}_1)$$

Für die Frequenzdarstellungen

$$H(\omega) = \frac{\sqrt{2}}{4}(-2e^{-2i\omega} + \dots + 1e^{2i\omega}) \quad \tilde{H}(\omega) = \frac{\sqrt{2}}{4}(e^{-i\omega} + e^{i\omega})$$

rechnet man nach:

$$\begin{aligned} H(0) &= \tilde{H}(0) = \sqrt{2} \\ H(\pi) &= \tilde{H}(\pi) = 0 \\ \tilde{H}(\omega)\overline{H(\omega)} &= \frac{1}{8}(e^{-3i\omega} + 8 + 9e^{i\omega} - 2e^{3i\omega}) \\ \tilde{H}(\omega + \pi)\overline{H(\omega + \pi)} &= \frac{1}{8}(-e^{-3i\omega} + 8 - 9e^{i\omega} + 2e^{3i\omega}) \end{aligned}$$

und erhält damit auch

$$\tilde{H}(\omega)\overline{H(\omega)} + \tilde{H}(\omega + \pi)\overline{H(\omega + \pi)} = 2$$

Damit sind die notwendigen Voraussetzungen erfüllt. Die Filter \mathbf{g} und $\tilde{\mathbf{g}}$ sind dann

$$\mathbf{g} = \frac{\sqrt{2}}{4} (1, -2, 1) = (g_0, g_1, g_2) \quad \tilde{\mathbf{g}} = \frac{\sqrt{2}}{4} (-1, -2, -3, 4, 2) = (\tilde{g}_{-1}, \dots, \tilde{g}_3)$$

Die Transformationsmatrizen für Signale der Länge 8 sind:

- Analysetransformation

$$W_8 = \begin{bmatrix} H_8 \\ G_8 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & h_2 & 0 & 0 & 0 & h_{-2} & h_{-1} \\ h_{-2} & h_{-1} & h_0 & h_1 & h_2 & 0 & 0 & 0 \\ 0 & 0 & h_{-2} & h_{-1} & h_0 & h_1 & h_2 & 0 \\ h_2 & 0 & 0 & 0 & h_{-2} & h_{-1} & h_0 & h_1 \\ g_0 & g_1 & g_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & g_0 & g_1 & g_2 & 0 \\ g_2 & 0 & 0 & 0 & 0 & 0 & g_0 & g_1 \end{bmatrix}$$

- Synthesetransformation

$$\tilde{W}_8 = \begin{bmatrix} \tilde{H}_8 \\ \tilde{G}_8 \end{bmatrix} = \begin{bmatrix} \tilde{h}_0 & \tilde{h}_1 & 0 & 0 & 0 & 0 & 0 & \tilde{h}_{-1} \\ 0 & \tilde{h}_{-1} & \tilde{h}_0 & \tilde{h}_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \tilde{h}_{-1} & \tilde{h}_0 & \tilde{h}_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \tilde{h}_{-1} & \tilde{h}_0 & \tilde{h}_1 \\ \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 & 0 & 0 & 0 & \tilde{g}_{-1} \\ 0 & \tilde{g}_{-1} & \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 & 0 & 0 \\ 0 & 0 & 0 & \tilde{g}_{-1} & \tilde{g}_0 & \tilde{g}_1 & \tilde{g}_2 & \tilde{g}_3 \\ \tilde{g}_2 & \tilde{g}_3 & 0 & 0 & 0 & \tilde{g}_{-1} & \tilde{g}_0 & \tilde{g}_1 \end{bmatrix}$$

■

Satz 10.2 Ist $(\mathbf{h}, \tilde{\mathbf{h}})$ biorthogonales Filterpaar mit $\mathbf{h} = (h_\ell, \dots, h_L)$, also mit Filterlänge $N = L - \ell + 1$, und $\tilde{\mathbf{h}} = (\tilde{h}_{\tilde{\ell}}, \dots, \tilde{h}_{\tilde{L}})$, also mit Filterlänge $\tilde{N} = \tilde{L} - \tilde{\ell} + 1$, wobei $\tilde{\ell} < L < \tilde{L}$. Dann gelten folgende Aussagen:

1. Die Filterlängen N und \tilde{N} haben gleiche Parität, d.h. $N \equiv \tilde{N} \pmod{2}$.
2. Sind N und \tilde{N} beide gerade, so gilt $L \equiv \tilde{L} \pmod{2}$.
3. Sind N und \tilde{N} beide ungerade, so gilt $L \not\equiv \tilde{L} \pmod{2}$.

Definition 10.2 (Symmetrische Filter).

Ein Filter $\mathbf{h} = (h_\ell, \dots, h_L)$ ist symmetrisch, falls

- $h_k = h_{-k}$ ($k \in \mathbb{Z}$) im Falle $\ell = -L$, d.h. für ungerade Länge; oder
- $h_k = h_{1-k}$ ($k \in \mathbb{Z}$) im Falle $\ell = -L + 1$, d.h. für gerade Länge.

Satz 10.3 Ist $(\mathbf{h}, \tilde{\mathbf{h}})$ ein biorthogonales Filterpaar mit symmetrischen Filtern, wobei $L < \tilde{L}$, so schreiben sich die Orthogonalitätsbedingungen;

$$\sum_{k=p}^L h_k \tilde{h}_{k-2m} = \delta_{0,m} \quad (0 \leq m \leq \tilde{L}),$$

wobei $p = -L + 1$ (falls N gerade) oder $p = -L$ (falls N ungerade).

■ **Beispiel 10.2** Konstruktion eines symmetrischen biorthogonalen Filterpaares
 $\mathbf{h} = (h_0, h_1)$ sei ein symmetrischer Filter der Länge 2 und $\tilde{\mathbf{h}} = (\tilde{h}_{-2}, \tilde{h}_{-1}, \tilde{h}_0, \tilde{h}_1, \tilde{h}_2, \tilde{h}_3)$ sei ein symmetrischer Filter der Länge 6. Das impliziert $\tilde{h}_0 = \tilde{h}_1, \tilde{h}_{-1} = \tilde{h}_2, \tilde{h}_{-2} = \tilde{h}_3$.

Für die Fourierreihen

$$H(\omega) = h_0 + h_1 e^{i\omega}, \quad \tilde{H}(\omega) = \tilde{h}_{-2} e^{-2i\omega} + \dots + \tilde{h}_3 e^{3i\omega}$$

erhält man aus den Tiefpassbedingungen Aussagen für die Koeffizienten:

$$H(0) = 2h_0 \stackrel{!}{=} \sqrt{2} \Rightarrow h_0 = h_1 = \frac{1}{\sqrt{2}}$$

$$H(\pi) \stackrel{!}{=} 0 \quad \text{gilt!}$$

$$\tilde{H}(0) \stackrel{!}{=} \sqrt{2} \Rightarrow \tilde{h}_1 + \tilde{h}_2 + \tilde{h}_3 = \frac{1}{\sqrt{2}}$$

$$\tilde{H}(\pi) \stackrel{!}{=} 0 \Rightarrow \tilde{h}_3 - \tilde{h}_2 + \tilde{h}_1 - \tilde{h}_1 + \tilde{h}_2 - \tilde{h}_3 = 0 \quad \text{gilt!}$$

Nun kommen noch die Orthogonalitätsbedingungen hinzu:

$$h_0 \tilde{h}_0 + h_1 \tilde{h}_1 \stackrel{!}{=} 1 \Rightarrow \tilde{h}_0 = \tilde{h}_1 = \frac{1}{\sqrt{2}}$$

$$h_0 \tilde{h}_{-2} + h_1 \tilde{h}_{-1} \stackrel{!}{=} 0 \Rightarrow \tilde{h}_{-2} = -\tilde{h}_{-1} = \frac{a}{\sqrt{2}}$$

mit einem Parameter $a \neq 0$. Daraus ergibt sich

$$\mathbf{h} = \frac{1}{\sqrt{2}} (1, 1) \quad \tilde{\mathbf{h}} = \frac{1}{\sqrt{2}} (a, -a, 1, 1, -a, a) \quad \square$$

■

Um ein biorthogonales Filterpaar $(\mathbf{h}, \tilde{\mathbf{h}})$ von endlichen Tiefpassfiltern zu konstruieren, kann man folgendermaßen vorgehen. Dabei ist zu beachten, dass die Rekonstruktionseigenschaften ("Glattheit") mit zunehmender Filterlänge besser werden.

- Man wählt zunächst den (symmetrischen) Filter $\tilde{\mathbf{h}}$ so, dass genügend viele Tiefpassbedingungen $\tilde{H}^{(m)}(\pi) = 0$ ($m = 0, 1, 2, \dots$) erfüllt sind. Dieses sind *lineare* Bedingungen an die Koeffizienten.

- Man wählt die Länge des Filters \mathbf{h} , wobei die Längen von \mathbf{h} und $\tilde{\mathbf{h}}$ nicht zu unterschiedlich sein sollten, damit die beiden Synthesefilter $\tilde{\mathbf{h}}$ und $\tilde{\mathbf{g}}$ in etwa gleiche Glattheitseigenschaften haben.

Man versucht nun, das *lineare* System (1) für die unbestimmten Koeffizienten von \mathbf{h} zu lösen:

$$\tilde{H}(\omega)\overline{H}(\omega) + \tilde{H}(\omega + \pi)\overline{H}(\omega + \pi) = 2$$

Beachte: Fordert man Symmetrie, so reduziert das die Anzahl der Variablen (aber auch die Aussicht auf Lösbarkeit)!

- Falls sich das lineare System als nicht lösbar erweist, muss man eine grössere Länge für \mathbf{h} ansetzen und es erneut versuchen.

10.2 Spline-Filter

10.2.1 Symmetrische Tiefpassfilter

- Für gerades N ist

$$\cos^N(\omega/2) = \frac{1}{2^N} \sum_{k=-N/2}^{N/2} \binom{N}{N/2+k} e^{ik\omega}$$

Folglich ist

$$H(\omega) = \sqrt{2} \cos^N(\omega/2)$$

die Fourierreihe eines symmetrischen Tiefpassfilters $(h_{-N/2}, \dots, h_{N/2})$ der (ungeraden) Länge $N + 1$, wobei

$$h_k = \frac{\sqrt{2}}{2^N} \binom{N}{N/2+k} \quad -N/2 \leq k \leq N/2$$

$$h_{k-N/2} = \frac{\sqrt{2}}{2^N} \binom{N}{k} \quad 0 \leq k \leq N$$

- Für ungerades N ist

$$e^{i\omega/2} \cos^N(\omega/2) = \frac{1}{2^N} \sum_{k=-(N-1)/2}^{(N+1)/2} \binom{N}{(N-1)/2+k} e^{ik\omega}$$

Folglich ist

$$H(\omega) = \sqrt{2} e^{i\omega/2} \cos^N(\omega/2)$$

die Fourierreihe eines symmetrischen Tiefpassfilters $(h_{-(N-1)/2}, \dots, h_{(N+1)/2})$ der (geraden) Länge $N + 1$, wobei

$$h_k = \frac{\sqrt{2}}{2^N} \binom{N}{(N-1)/2+k} \quad -(N-1)/2 \leq k \leq (N+1)/2$$

$$h_{k-(N-1)/2} = \frac{\sqrt{2}}{2^N} \binom{N}{k} \quad 0 \leq k \leq N$$

10.2.2 Spline-Funktionen

Man definiert die *Spline-Funktionen* $B_N(t)$ induktiv durch

$$B_0(t) = \chi_{[-1/2, 1/2)}(t),$$

$$B_{N+1}(t) = B_0(t) * B_N(t) = \int_{-1/2}^{1/2} B_N(t-s) ds,$$

d.h., $B_N(t)$ ist die N -fache Faltungspotenz der Basisfunktion $B_0(t)$.

Eine wichtige Eigenschaft dieser Funktionen ist es, dass sie einer *Skalierungsgleichung* genügen:

$$B_N(t) = \sum_{k=0}^{N+1} \frac{1}{2^N} \binom{N+1}{k} B_N(2t + \lceil N/2 \rceil - k + 1)$$

Die Skalierungskoeffizienten sind (bis auf einen konstanten Faktor) gerade die Filterkoeffizienten der oben definierten Spline-Filter — daher die Namensgebung.

Die Abbildung zeigt die Spline-Funktionen $B_0(t), B_1(t), B_2(t), B_3(t)$.

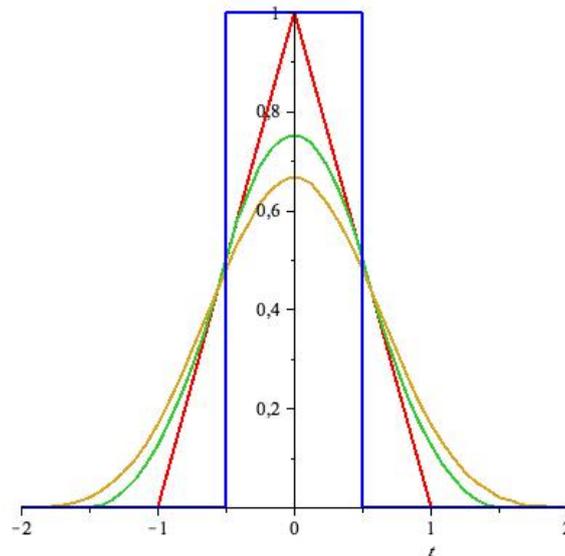


Abbildung 10.1: Spline-Funktionen $B_0(t), \dots, B_3(t)$

10.2.3 Daubechies biorthogonale Filter

Ist mittels der Fourierreihe $H(\omega) = \sqrt{2} \cos^N(\omega/2)$ ein symmetrischer Spline-Filter $\mathbf{h} = (h_{-\ell}, \dots, h_{\ell})$ ungerader Länge $N + 1 = 2\ell + 1$ (für gerades $N = 2\ell$) bzw. mit $H(\omega) = \sqrt{2} e^{i\omega/2} \cos^N(\omega/2)$ ein symmetrischer Spline-Filter $\mathbf{h} = (h_{-\ell}, \dots, h_{\ell+1})$ gerader Länge $N + 1 = 2\ell + 2$ (für ungerades $N = 2\ell + 1$) gewählt, so kann man dazu “passende” orthogonale symmetrische Filter mit Hilfe der Daubechies-Polynome

$$P_M(z) = \sum_{m=0}^M \binom{M+m}{m} z^m$$

aus Abschnitt 7 gewinnen.

Definition 10.3 (Filterdefinition mittels der Daubechies-Polynome).

\tilde{N} habe die gleiche Parität wie N .

1. Sind $N = 2\ell$ und $\tilde{N} = 2\tilde{\ell}$ beide gerade, so definiert man einen Filter $\tilde{\mathbf{h}}$ mittels der Fourierreihe

$$\tilde{H}(\omega) = \sqrt{2} \cos^{\tilde{N}}(\omega/2) P_{\ell+\tilde{\ell}-1}(\sin^2(\omega/2)).$$

2. Sind $N = 2\ell + 1$ und $\tilde{N} = 2\tilde{\ell} + 1$ beide ungerade, so definiert man einen Filter $\tilde{\mathbf{h}}$ mittels der Fourierreihe

$$\tilde{H}(\omega) = \sqrt{2} e^{i\omega/2} \cos^{\tilde{N}}(\omega/2) P_{\ell+\tilde{\ell}}(\sin^2(\omega/2)).$$

Satz 10.4 Für die mit den Daten aus der Definition bestimmten Filter $\tilde{\mathbf{h}}$ gilt:

1. Die Filter $\tilde{\mathbf{h}}$ haben die Länge $2\tilde{N} + N - 1$.
2. Die Filter $\tilde{\mathbf{h}}$ sind symmetrisch.
3. Die Filter $\tilde{\mathbf{h}}$ sind Tiefpassfilter.
4. Die Filter \mathbf{h} und $\tilde{\mathbf{h}}$ sind orthogonal.

Für den *Beweis* wird nur der Fall betrachtet, dass N und \tilde{N} beide gerade sind, also $N = 2\ell$, $\tilde{N} = 2\tilde{\ell}$. Der ungerade Fall wird ganz analog erledigt.

ad 1./2. Schreibt man die beiden Faktoren $\cos^{\tilde{N}}(\omega/2)$ und $P_{\ell+\tilde{\ell}-1}(\sin^2(\omega/2))$ als Reihen in $e^{i\omega}$ aus, so sieht man

$$\cos^{\tilde{N}}(\omega/2) = \sum_{k=-\tilde{\ell}}^{\tilde{\ell}} \alpha_k e^{ik\omega}$$

mit einer symmetrischen Koeffizientenfolge $(\alpha_{-\ell}, \dots, \alpha_{\ell})$, da die linke Seite eine gerade Funktion von ω ist. Weiter ist

$$P_{\ell+\tilde{\ell}-1}(\sin^2(\omega/2)) = \sum_{m=-\ell-\tilde{\ell}+1}^{\ell+\tilde{\ell}-1} \beta_m e^{im\omega}$$

mit einer symmetrischen Koeffizientenfolge $(\beta_{-\ell-\tilde{\ell}+1}, \dots, \beta_{\ell+\tilde{\ell}-1})$ (aus entsprechendem Grund).

Damit hat das Produkt die Form

$$\cos^{\tilde{N}}(\omega/2) \cdot P_{\ell+\tilde{\ell}-1}(\sin^2(\omega/2)) = \sum_{n=-2\tilde{\ell}-\ell+1}^{2\tilde{\ell}+\ell-1} \gamma_n e^{in\omega}$$

mit einer symmetrischen Koeffizientenfolge $(\gamma_{-2\tilde{\ell}-\ell+1}, \dots, \gamma_{2\tilde{\ell}+\ell-1})$, da das Faltingsprodukt symmetrischer Folgen wieder symmetrisch ist.

Die Länge ist also $2(2\tilde{\ell} + \ell - 1) + 1 = 2\tilde{N} + N - 1$.

ad 3. Offensichtlich ist $\tilde{H}(0) = \sqrt{2}$ und $\tilde{H}(\pi) = 0$.

ad 4. Es ist (in der Notation von Abschnitt 7.4, also mit $z = e^{i\omega}$ und $y = \sin^2(\omega/2)$):

$$\begin{aligned} H(\omega)\tilde{H}(\omega) &= 2 \cos^{N+\tilde{N}}(\omega/2) P_{\ell+\tilde{\ell}-1}(\sin^2(\omega/2)) \\ &= 2(1-y)^{\ell+\tilde{\ell}} P_{\ell+\tilde{\ell}-1}(y) = 2\hat{P}_{N+\tilde{N}-1}(z) = 2\hat{P}_{N+\tilde{N}-1}(e^{i\omega}). \end{aligned}$$

Also ist

$$H(\omega)\tilde{H}(\omega) + H(\omega + \pi)\tilde{H}(\omega + \pi) = 2(\hat{P}_{N+\tilde{N}-1}(z) + \hat{P}_{N+\tilde{N}-1}(-z)) = 2$$

wegen der im Abschnitt 7.4 vermerkten Eigenschaft der Daubechies-Polynome:

$$\hat{P}_{2M-1}(z) + \hat{P}_{2M-1}(-z) = 1.$$

(Auf die komplexe Konjugation kann hier natürlich verzichtet werden, da die Filter reell sind.)

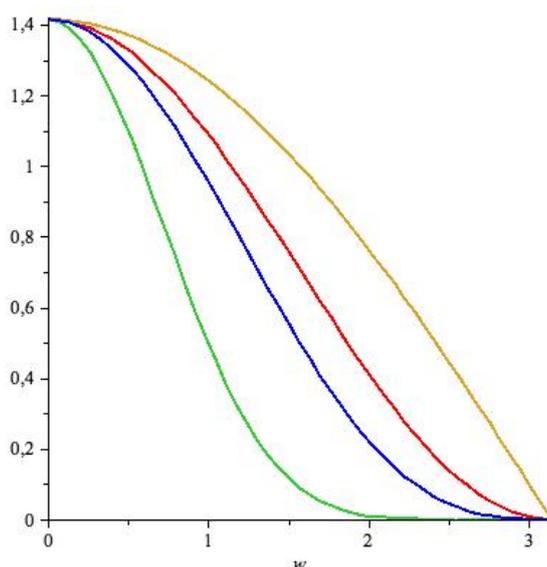


Abbildung 10.2: Frequenzdarstellung der Bsplinefilter der Längen 2,3,4,9

10.3 Cohen-Daubechies-Feauveau-Filter

10.3.1 Nochmals Daubechies-Polynome

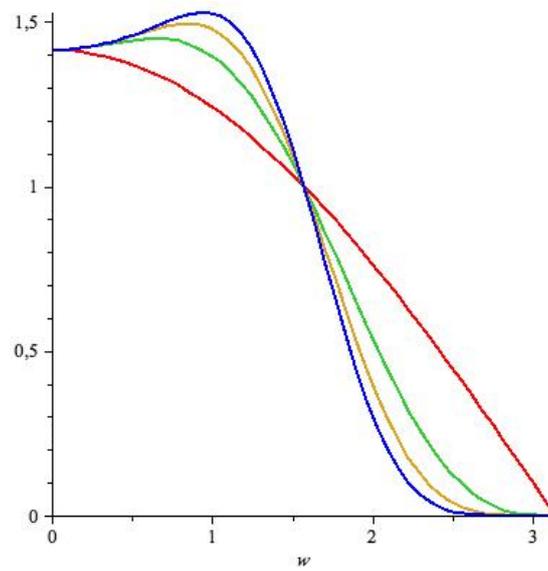
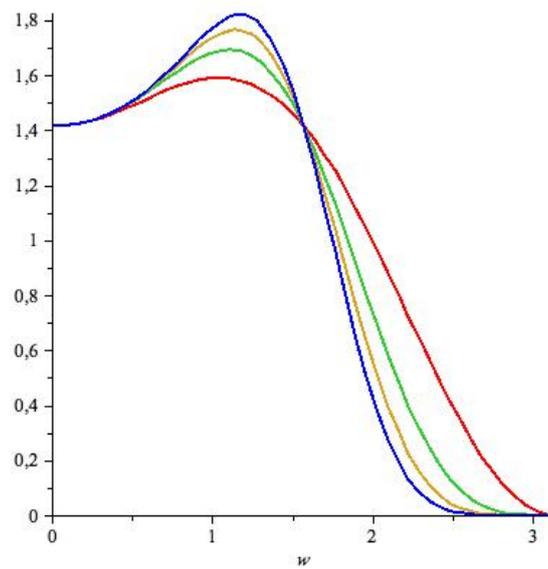
Die Daubechies-Polynome

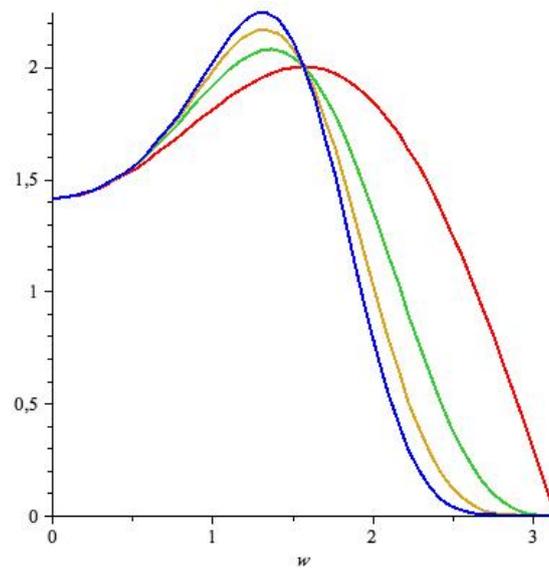
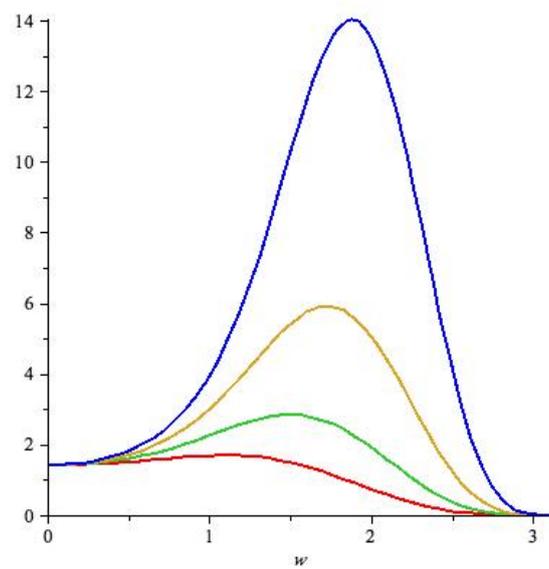
$$P_M(z) = \sum_{m=0}^M \binom{M+m}{m} z^m$$

genügen der fundamentalen Gleichung

$$(1-z)^{M+1}P_M(z) + z^{M+1}P_M(1-z) = 1.$$

Die beiden Polynome $(1-z)^{M+1}$ und z^{M+1} sind teilerfremd (d.h., sie haben keine gemeinsamen Nullstellen, was ja offensichtlich ist). Liest man diese Gleichung als

Abbildung 10.3: Bsplinefilterpartner $K_{1,1}, K_{3,1}, K_{5,1}, K_{7,1}$ Abbildung 10.4: Bsplinefilterpartner $K_{2,2}, K_{2,4}, K_{2,6}, K_{2,8}$

Abbildung 10.5: Bsplinefilterpartner $K_{1,3}, K_{3,3}, K_{5,3}, K_{7,3}$ Abbildung 10.6: Bsplinefilterpartner $K_{4,2}, K_{4,4}, K_{4,6}, K_{4,8}$

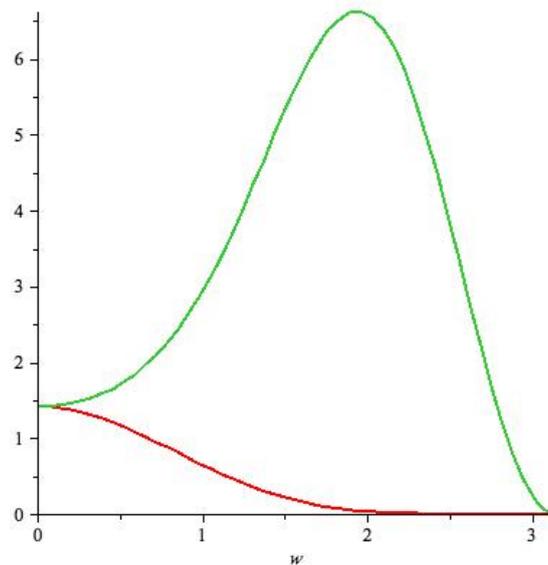


Abbildung 10.7: (7,9)-Bsplinefilterpaar

eine Bezout-Beziehung im Polynomring $\mathbb{Q}[z]$, so zeigt sich, dass $q_1(z) = P_M(z)$ und $q_2(z) = P_M(1 - z)$ die eindeutig bestimmten Polynome $q_1(z)$ und $q_2(z)$ vom Grad $\leq M$ sind, für die eine Bezout-Beziehung

$$(1 - z)^{M+1} q_1(z) + z^{M+1} q_2(z) = 1$$

gilt. Dies sind aber überhaupt (ohne Gradbeschränkung!) die einzigen Lösungen dieser Gleichung! Denn zunächst muss für ein solches Lösungspaar $(q_1(z), q_2(z))$ die Beziehung $q_2(z) = q_1(1 - z)$ gelten. (Die Bezout-Beziehung nochmal hinschreiben, aber mit z durch $1 - z$ ersetzt, dann beide Bezoutbeziehungen subtrahieren).

Hat man aber

$$(1 - z)^{M+1} q(z) + z^{M+1} q(1 - z) = 1,$$

so muss

$$q(z) = P_M(z) + a(z) z^{M+1}, \quad q(1 - z) = P_M(1 - z) - a(z)(1 - z)^{M+1},$$

gelten für ein Polynom $a(z)$. Das geht aber ersichtlich nur für das Nullpolynom.

Schreibt man die Bezout-Beziehung in der Form

$$P_M(z) = (1 - z)^{-M-1} - \left(\frac{z}{1 - z} \right)^{M+1} \cdot P_M(1 - z),$$

so erhält man aus der Reihenentwicklung der rechten Seite gerade die explizite Darstellung, denn es ist (Newton-Reihe)

$$(1 - z)^{-M-1} = \sum_{m \geq 0} \binom{M + m}{m} z^m,$$

die linke Seite der Gleichung ist ein Polynom vom Grad $\leq M$ und der zweite Term auf der rechten Seite hat nur Beiträge zu Graden $> M$.

10.3.2 Symmetrische Filter ungerader Länge

Ist $\mathbf{h} = (h_{-L}, \dots, h_L)$ ein symmetrischer Filter der Länge $2L + 1$, so ist seine Fourierreihe $H(\omega) = \sum_{k=-L}^L h_k e^{ik\omega}$ eine gerade Funktion:

$$H(\omega) = h_0 + 2 \sum_{k=1}^L h_k \cos(k\omega).$$

Da man für $k \in \mathbb{Z}$ den Term $\cos(k\omega)$ als ein Polynom vom Grad k in $\cos(\omega)$ schreiben kann, ist $H(\omega)$ ein Polynom vom Grad L in $\cos(\omega)$.

Verlangt man nun noch als Tiefpassbedingung, dass $H(0) = \sqrt{2}$ ist und $H(\pi) = H'(\pi) = \dots = H^{(\ell)}(\pi) = 0$, aber $H^{(\ell+1)}(\pi) \neq 0$ so muss sich $H(\omega)$ schreiben lassen in der Form

$$H(\omega) = \sqrt{2} (1 + \cos(\omega))^\ell q(\cos(\omega)),$$

wobei $q(z)$ ein Polynom vom Grad $L - \ell$ ist, für das $q(\cos(\pi)) = q(-1) \neq 0$ ist. Aus $H(0) = \sqrt{2}$ folgt $q(1) = 2^{-\ell}$. Ersetzt man nun noch $1 + \cos(\omega)$ durch $2 \cos^2(\omega/2)$, so erhält man eine Darstellung von $H(\omega)$ in der Form

$$H(\omega) = \sqrt{2} \cos^{2\ell}(\omega/2) p(\cos(\omega)),$$

wobei $p(z)$ ein Polynom vom Grad $L - \ell$ ist mit $p(1) = 1$ und $p(-1) \neq 0$.

Satz 10.5 Sind \mathbf{h} und $\tilde{\mathbf{h}}$ symmetrische Filter ungerader Länge, deren Fourierreihen

$$H(\omega) = \sqrt{2} \cos^{2\ell}(\omega/2) p(\cos(\omega)), \quad \tilde{H}(\omega) = \sqrt{2} \cos^{2\tilde{\ell}}(\omega/2) \tilde{p}(\cos(\omega)),$$

die Orthogonalitätsbedingung

$$H(\omega) \tilde{H}(\omega) + H(\omega + \pi) \tilde{H}(\omega + \pi) = 2$$

erfüllen, so gilt mit $K = \ell + \tilde{\ell}$

$$p(\cos(\omega)) \cdot \tilde{p}(\cos(\omega)) = P_{K-1}(\sin^2(\omega/2)).$$

Zum *Beweis* setzt man in die Orthogonalitätsbedingung ein und erhält

$$\cos^{2K}(\omega/2) p(\cos(\omega)) \tilde{p}(\cos(\omega)) + \sin^{2K}(\omega/2) p(-\cos(\omega)) \tilde{p}(-\cos(\omega)) = 2.$$

Mit $P(z) = p(z) \tilde{p}(z)$ ist $P(\cos(\omega))$ auch ein Polynom in $y = \sin^2(\omega/2)$, also etwa $P(\cos(\omega)) = \hat{P}(y)$ und die Orthogonalität schreibt sich somit als

$$(1 - y)^K \hat{P}(y) + y^K \hat{P}(1 - y) = 1$$

10.3.3 Das Cohen-Daubechies-Feauveau-(7,9)-Filterpaar

Das bekannteste biorthogonale Filterpaar, das aus der Konstruktion des vorigen Abschnitts gewonnen werden kann, ist ein Paar von (Tiefpass-)Filtern der Längen 7 und 9. Dieses Filterpaar wird für die (verlustbehaftete, *lossy*) Datenkompression im Standard JPEG2000 verwendet.

Die Koeffizienten des CDF-Tiefpass-Filters der Länge 7 sind

-3	-0.0645388826
-2	-0.0406894175
-1	0.4180922731
0	0.7884856164
1	0.4180922731
2	-0.0406894175
3	-0.0645388826

Die Koeffizienten des CDF-Tiefpass-Filters der Länge 9 sind

-4	0.0378284555
-3	-0.0238494650
-2	-0.1106244044
-1	0.3774028555
0	0.8526986788
1	0.3774028555
2	-0.1106244044
3	-0.0238494650
4	0.0378284555

Die Details der Konstruktion kann man auf dem anschliessend wiedergegebenen **Mathematica** notebook nachvollziehen.

Ein anderer Zugang zu diesem Filterpaar wird im Rahmen der Lifting-Methode im Kapitel 11 beschrieben.

r

The CDF-7/9 filter pair

Construction

Starting with the Daubechies polynomial

```
Daub[N_, z_] := Sum[Binomial[N + k, k] z^k, {k, 0, N}]
```

```
Daub[3, z]
```

```
1 + 4 z + 10 z^2 + 20 z^3
```

Determining the complex roots of this polynomial

```
roots = NSolve[Daub[3, z] == 0, z]
```

```
{{z -> -0.342384}, {z -> -0.078808 - 0.373931 i}, {z -> -0.078808 + 0.373931 i}}
```

There is one real root and two complex conjugate roots. So the Daubechies polynomial can be factored into two polynomials with real coefficients: a linear polynomial $p(z)$ and a quadratic polynomial $q(z)$. The parameter a (which is the leading coefficient of the polynomial $p(z)$) will be fixed later.

```
p[z_] = a (z - (z /. roots[[1]]))
```

```
a (0.342384 + z)
```

```
q[z_] = (20/a) (z - (z /. roots[[2]])) (z - (z /. roots[[3]]))
```

```
 $\frac{1}{a} 20 ((0.078808 - 0.373931 i) + z) ((0.078808 + 0.373931 i) + z)$ 
```

```
Expand[%]
```

```
 $\frac{2.9207 + 0. i}{a} + \frac{(3.15232 + 0. i) z}{a} + \frac{20 z^2}{a}$ 
```

```
Chop[%]
```

```
 $\frac{2.9207}{a} + \frac{3.15232 z}{a} + \frac{20 z^2}{a}$ 
```

The frequency representation of the H-filter that goes with the factor $p(z)$

```
H[w_] = Sqrt[2] Cos[w/2]^4 p[Sin[w/2]^2]
```

```
 $\sqrt{2} a \text{Cos}\left[\frac{w}{2}\right]^4 \left(0.342384 + \text{Sin}\left[\frac{w}{2}\right]^2\right)$ 
```

The frequency representation of the K-filter that goes with the factor $q(z)$

```
K[w_] = Chop[Expand[Sqrt[2] Cos[w/2]^4 q[Sin[w/2]^2]]]
```

```
 $\frac{4.13049 \text{Cos}\left[\frac{w}{2}\right]^4}{a} + \frac{1}{a} 4.45805 \text{Cos}\left[\frac{w}{2}\right]^4 \text{Sin}\left[\frac{w}{2}\right]^2 + \frac{1}{a} 20 \sqrt{2} \text{Cos}\left[\frac{w}{2}\right]^4 \text{Sin}\left[\frac{w}{2}\right]^4$ 
```

Evaluating $H(0)$ (which should be $\sqrt{2}$)

```
H[0]
```

```
0.484204 a
```

```
const = Solve[{H[0] == Sqrt[2]}, {a}]
{{a -> 2.9207}}
```

The Fourier series if the H-filter after fixing the constant

```
H[w_] = H[w] /. const[[1]]
4.13049 Cos[w/2]^4 (0.342384 + Sin[w/2]^2)
```

The Fourier series of the K-filter after fixing the constant

```
K[w_] = K[w] /. const[[1]]
1.41421 Cos[w/2]^4 + 1.52637 Cos[w/2]^4 Sin[w/2]^2 + 9.68408 Cos[w/2]^4 Sin[w/2]^4
```

Checking that all is correct

```
K[0]
1.41421
```

Checking orthogonality conditions for the CDF-7/9 filter pair

```
H[w] Conjugate[K[w]] + H[w + Pi] Conjugate[K[w + Pi]]
4.13049 Cos[w/2]^4 (Conjugate[1.52637 Cos[w/2]^4 Sin[w/2]^2 + 9.68408 Cos[w/2]^4 Sin[w/2]^4] +
  1.41421 Cos[Conjugate[w]/2]^4) (0.342384 + Sin[w/2]^2) + 4.13049 Cos[(pi+w)/2]^4
  (Conjugate[1.52637 Cos[(pi+w)/2]^4 Sin[(pi+w)/2]^2 + 9.68408 Cos[(pi+w)/2]^4 Sin[(pi+w)/2]^4] +
  1.41421 Cos[1/2 (pi + Conjugate[w])]^4) (0.342384 + Sin[(pi+w)/2]^2)
```

```
Assuming[w ∈ Reals, Simplify[%]]
```

```
4.13049 Cos[w/2]^4 (0.342384 + Sin[w/2]^2)
  (1.41421 Cos[w/2]^4 + (0.605255 + 0.0953979 Csc[w/2]^2) Sin[w]^4) +
  4.13049 (0.342384 + Cos[w/2]^2) Sin[w/2]^4
  (1.41421 Sin[w/2]^4 + (0.605255 + 0.0953979 Sec[w/2]^2) Sin[w]^4)
```

```
FullSimplify[Expand[%]]
```

```
2. + 3.33067 × 10-16 Cos[2 w] + 2.77556 × 10-17 Cos[3 w] -
  2.08167 × 10-17 Cos[4 w] - 3.46945 × 10-18 Cos[5 w]
```

```
Chop[%]
```

```
2.
```

Properties of the CDF-7/9 filter pair

```
TrigToExp[H[w]]
```

```
0.788486 + 0.418092 e-i w + 0.418092 ei w -
  0.0406894 e-2 i w - 0.0406894 e2 i w - 0.0645389 e-3 i w - 0.0645389 e3 i w
```

```
hcoeffs = CoefficientList[Expand[Exp[3 I ω %], E^{I ω}]
{-0.0645389, -0.0406894, 0.418092, 0.788486, 0.418092, -0.0406894, -0.0645389}
```

The filter coefficients of the H-filter of length 7

```
hfilter = Table[{k - 4, hcoeffs[[k]]}, {k, 1, 7}];
TableForm[hfilter, TableAlignments -> "."]
```

-3	-0.0645389
-2	-0.0406894
-1	0.418092
0	0.788486
1	0.418092
2	-0.0406894
3	-0.0645389

```
TrigToExp[K[ω]]
```

```
0.852699 + 0.377403 e^{-i ω} + 0.377403 e^{i ω} - 0.110624 e^{-2 i ω} - 0.110624 e^{2 i ω} -
0.0238495 e^{-3 i ω} - 0.0238495 e^{3 i ω} + 0.0378285 e^{-4 i ω} + 0.0378285 e^{4 i ω}
```

```
kcoeffs = Re[CoefficientList[Expand[Exp[4 I ω %], E^{I ω}]]
```

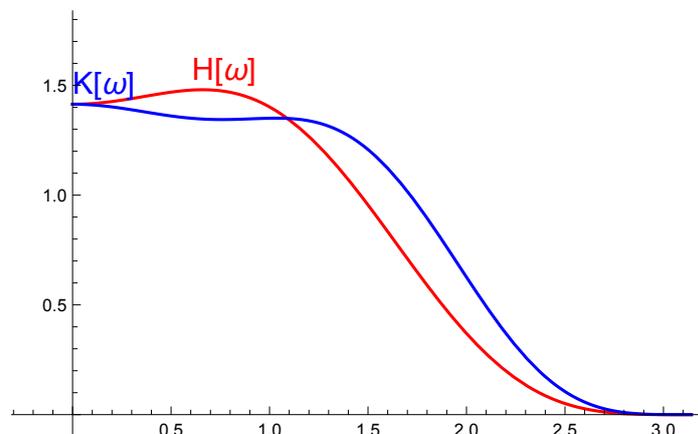
```
{0.0378285, -0.0238495, -0.110624, 0.377403,
0.852699, 0.377403, -0.110624, -0.0238495, 0.0378285}
```

The filter coefficients of the K-filter of length 9

```
kfilter = Table[{k - 5, kcoeffs[[k]]}, {k, 1, 9}];
TableForm[kfilter, TableAlignments -> "."]
```

-4	0.0378285
-3	-0.0238495
-2	-0.110624
-1	0.377403
0	0.852699
1	0.377403
2	-0.110624
3	-0.0238495
4	0.0378285

```
Plot[{H[ω], K[ω]}, {ω, 0, Pi}, PlotStyle -> {Red, Blue}, PlotLabels ->
Placed[{Style["H[ω]", Red, 16], Style["K[ω]", Blue, 16]}, {Above, Above}]]
```



Procedures for the cascade algorithm

```

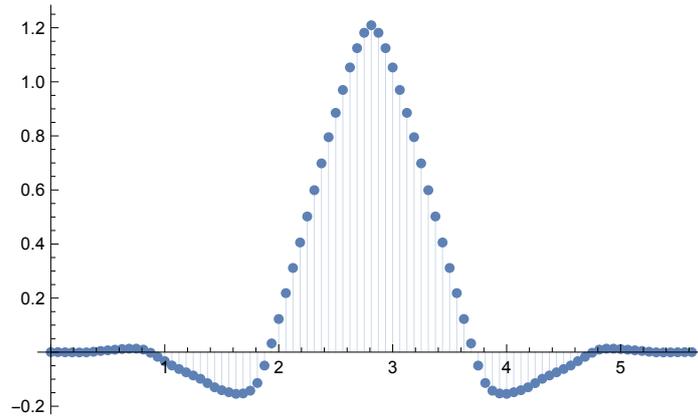
cascadephidis[h_, n_] :=
Module[{degrees, min, max, sum, hmodif, hpol, hpols, tbl, clist, z},
  (*Approximation of the scaling function belonging
  to the filter h by using the cascade algorithm*)
  degrees = Map[First[#] &, h];
  min = Min[degrees];
  max = Max[degrees];
  sum = Last[Total[h]];
  hmodif = Map[{First[#], Last[#] / sum} &, h];
  hpol := Apply[Plus, Map[Last[#] z^First[#] &, hmodif]];
  hpols := Expand[Product[hpol /. z -> z^(2^t), {t, 0, n-1}]];
  clist = 2^n CoefficientList[z^(-min (2^n - 1)) hpols, z];
  tbl = Table[{(k - 1) 2^(-n), clist[[k]]}, {k, 1, Length[clist]}];
  ListPlot[tbl, PlotRange -> All, Filling -> Axis, PlotStyle -> PointSize[0.015]]
]

cascadepsidis[h_, n_] :=
Module[{degrees, min, max, sum, hmodif, gpol, hpol, hpols, clist, tbl, z},
  (*Approximation of the wavelet function belonging
  to the filter h by using the cascade algorithm*)
  degrees = Map[First[#] &, h];
  min = Min[degrees];
  max = Max[degrees];
  sum = Last[Total[h]];
  hmodif = Map[{First[#], Last[#] / sum} &, h];
  hpol := Apply[Plus, Map[Last[#] z^First[#] &, hmodif]];
  gpol := hpol /. (z -> -1 / z);
  hpols :=
  Expand[Product[hpol /. z -> z^(2^t), {t, 0, n-2}] (gpol /. z -> z^(2^(n-1)))];
  clist = 2^n CoefficientList[z^(-min (2^(n-1) - 1) + 2^(n-1) max) hpols, z];
  tbl = Table[{(k - 1) 2^(-n), clist[[k]]}, {k, 1, Length[clist]}];
  ListPlot[tbl, PlotRange -> All, Filling -> Axis, PlotStyle -> PointSize[0.015]]
]

```

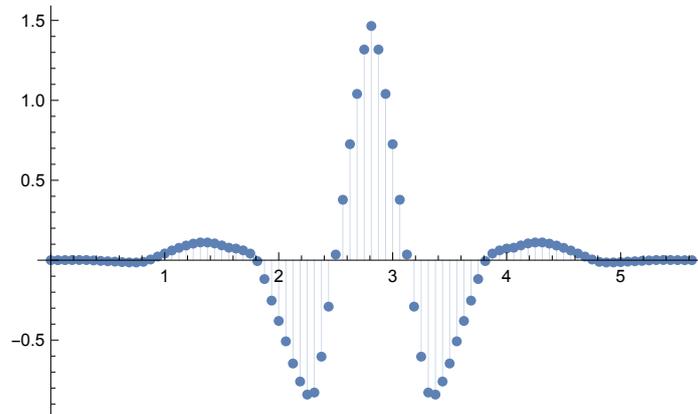
The scaling function belonging to the H-filter of length 7

`cascadephidis[hfilter, 4]`



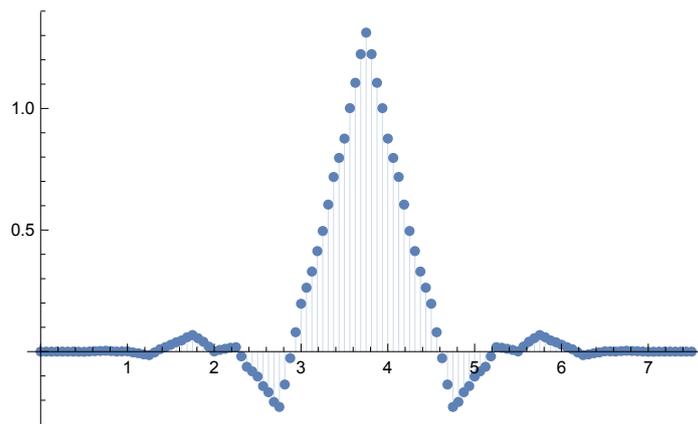
The wavelet function belonging to the H-filter of length 7

`cascadepsidis[hfilter, 4]`



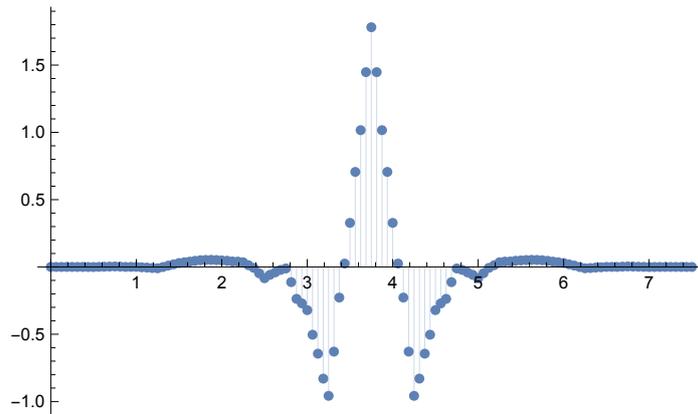
The scaling function belonging to the K-filter of length 9

`cascadephidis[kfilter, 4]`



The wavelet function belonging to the K-filter of length 9

`cascadepsidis[kfilter, 4]`



Low-pass conditions for the H-filter

$\{H'[\omega], H'[\omega] / \omega \rightarrow \text{Pi}\}$

$$\{4.13049 \cos\left[\frac{\omega}{2}\right]^5 \sin\left[\frac{\omega}{2}\right] - 8.26098 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right] \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right), 0.\}$$

$\{H''[\omega], H''[\omega] / \omega \rightarrow \text{Pi}\}$

$$\{2.06524 \cos\left[\frac{\omega}{2}\right]^6 - 18.5872 \cos\left[\frac{\omega}{2}\right]^4 \sin\left[\frac{\omega}{2}\right]^2 - 4.13049 \cos\left[\frac{\omega}{2}\right]^4 \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right) + 12.3915 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^2 \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right), 0.\}$$

$\{H'''[\omega], H'''[\omega] / \omega \rightarrow \text{Pi}\}$

$$\{-28.9134 \cos\left[\frac{\omega}{2}\right]^5 \sin\left[\frac{\omega}{2}\right] + 49.5659 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right]^3 + 20.6524 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right] \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right) - 12.3915 \cos\left[\frac{\omega}{2}\right] \sin\left[\frac{\omega}{2}\right]^3 \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right), 0.\}$$

$\{H''''[\omega], H''''[\omega] / \omega \rightarrow \text{Pi}\}$

$$\{-14.4567 \cos\left[\frac{\omega}{2}\right]^6 + 167.285 \cos\left[\frac{\omega}{2}\right]^4 \sin\left[\frac{\omega}{2}\right]^2 - 86.7403 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^4 + 10.3262 \cos\left[\frac{\omega}{2}\right]^4 \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right) - 49.5659 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^2 \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right) + 6.19573 \sin\left[\frac{\omega}{2}\right]^4 \left(0.342384 + \sin\left[\frac{\omega}{2}\right]^2\right), 8.31705\}$$

So the H-filter of length 7 has 4 vanishing moments

Low-pass conditions for the K-filter

$\{K'[\omega], K'[\omega] / . \omega \rightarrow \text{Pi}\}$

$$\{-2.82843 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right] + 1.52637 \cos\left[\frac{\omega}{2}\right]^5 \sin\left[\frac{\omega}{2}\right] - 3.05273 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right]^3 + 19.3682 \cos\left[\frac{\omega}{2}\right]^5 \sin\left[\frac{\omega}{2}\right]^3 - 19.3682 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right]^5, 0.\}$$

$\{K''[\omega], K''[\omega] / . \omega \rightarrow \text{Pi}\}$

$$\{-1.41421 \cos\left[\frac{\omega}{2}\right]^4 + 0.763183 \cos\left[\frac{\omega}{2}\right]^6 + 4.24264 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^2 - 8.39501 \cos\left[\frac{\omega}{2}\right]^4 \sin\left[\frac{\omega}{2}\right]^2 + 29.0523 \cos\left[\frac{\omega}{2}\right]^6 \sin\left[\frac{\omega}{2}\right]^2 + 4.5791 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^4 - 96.8408 \cos\left[\frac{\omega}{2}\right]^4 \sin\left[\frac{\omega}{2}\right]^4 + 29.0523 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^6, 0.\}$$

$\{K'''[\omega], K'''[\omega] / . \omega \rightarrow \text{Pi}\}$

$$\{7.07107 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right] - 10.6846 \cos\left[\frac{\omega}{2}\right]^5 \sin\left[\frac{\omega}{2}\right] + 29.0523 \cos\left[\frac{\omega}{2}\right]^7 \sin\left[\frac{\omega}{2}\right] - 4.24264 \cos\left[\frac{\omega}{2}\right] \sin\left[\frac{\omega}{2}\right]^3 + 25.9482 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right]^3 - 280.838 \cos\left[\frac{\omega}{2}\right]^5 \sin\left[\frac{\omega}{2}\right]^3 - 4.5791 \cos\left[\frac{\omega}{2}\right] \sin\left[\frac{\omega}{2}\right]^5 + 280.838 \cos\left[\frac{\omega}{2}\right]^3 \sin\left[\frac{\omega}{2}\right]^5 - 29.0523 \cos\left[\frac{\omega}{2}\right] \sin\left[\frac{\omega}{2}\right]^7, 0.\}$$

$\{K''''[\omega], K''''[\omega] / . \omega \rightarrow \text{Pi}\}$

$$\{3.53553 \cos\left[\frac{\omega}{2}\right]^4 - 5.34228 \cos\left[\frac{\omega}{2}\right]^6 + 14.5261 \cos\left[\frac{\omega}{2}\right]^8 - 16.9706 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^2 + 65.6337 \cos\left[\frac{\omega}{2}\right]^4 \sin\left[\frac{\omega}{2}\right]^2 - 522.941 \cos\left[\frac{\omega}{2}\right]^6 \sin\left[\frac{\omega}{2}\right]^2 + 2.12132 \sin\left[\frac{\omega}{2}\right]^4 - 50.3701 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^4 + 1404.19 \cos\left[\frac{\omega}{2}\right]^4 \sin\left[\frac{\omega}{2}\right]^4 + 2.28955 \sin\left[\frac{\omega}{2}\right]^6 - 522.941 \cos\left[\frac{\omega}{2}\right]^2 \sin\left[\frac{\omega}{2}\right]^6 + 14.5261 \sin\left[\frac{\omega}{2}\right]^8, 18.937\}$$

So the K-filter of length 9 has 4 vanishing moments

11. Lifting

Unter “Lifting” versteht man eine spezielle Methode, orthogonale und biorthogonale Filterpaare zu konstruieren und zu implementieren. Von dieser Technik soll dieser Abschnitt einen Eindruck vermittelt werden. Es beginnt mit dem elementarsten aller Beispiele:

11.1 Faktorisierung der Haar-Transformation

Die Haar-Wavelettransformation benutzt die beiden Filter

$$\begin{array}{ll} \text{Tiefpassfilter} & \mathbf{h} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \end{bmatrix} \\ \text{Hochpassfilter} & \mathbf{g} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \end{bmatrix} \end{array}$$

und die daraus resultierende orthogonale Transformationsmatrix

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

Als eine Operation auf Signalen geschrieben:

$$\mathbf{a} = (a[n])_{n \in \mathbb{Z}} \mapsto \begin{cases} \tilde{\mathbf{a}} = (\tilde{a}[n])_{n \in \mathbb{Z}} \\ \tilde{\mathbf{d}} = (\tilde{d}[n])_{n \in \mathbb{Z}} \end{cases}$$

mit

$$\tilde{a}[n] = \frac{1}{\sqrt{2}} (a[2n] + a[2n + 1]), \quad \tilde{d}[n] = \frac{1}{\sqrt{2}} (a[2n] - a[2n + 1]).$$

Im Formalismus der z -Transformierten ausgedrückt: es ist, indem man Koeffizienten mit geradem bzw. ungeradem Index in separaten Reihen zusammenfasst

$$a(z) = \sum_{n \in \mathbb{Z}} a[n] z^n = a_{\text{even}}(z^2) + z \cdot a_{\text{odd}}(z^2),$$

wobei

$$a_{\text{even}}(z) = \sum_{n \in \mathbb{Z}} a[2n] z^n = \left. \frac{a(z) + a(-z)}{2} \right|_{z^2 \leftarrow z},$$

$$a_{\text{odd}}(z) = \sum_{n \in \mathbb{Z}} a[2n+1] z^n = \left. \frac{a(z) - a(-z)}{2z} \right|_{z^2 \leftarrow z}.$$

Dann gilt

$$\tilde{a}(z) = \sum_{n \in \mathbb{Z}} \tilde{a}[n] z^n = \frac{1}{\sqrt{2}} (a_{\text{even}}(z) + a_{\text{odd}}(z)),$$

$$\tilde{d}(z) = \sum_{n \in \mathbb{Z}} \tilde{d}[n] z^n = \frac{1}{\sqrt{2}} (a_{\text{even}}(z) - a_{\text{odd}}(z)).$$

Das schreibt sich nun als Matrixgleichung, bei der Potenzreihen als Koeffizienten auftreten:

$$\begin{bmatrix} \tilde{a}(z) \\ \tilde{d}(z) \end{bmatrix} = H \cdot \begin{bmatrix} a_{\text{even}}(z) \\ a_{\text{odd}}(z) \end{bmatrix}.$$

Dieser allereinfachste Fall ist insofern untypisch, als die Koeffizienten der Matrix H Konstante sind. Das wird im allgemeinen nicht mehr so sein.

Wichtig ist nun, dass sich die durch die Matrix H vermittelte Transformation faktorisieren lässt:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}$$

Diese Form der Zerlegung in ein Produkt von sehr speziellen Matrizen (Dreiecksmatrizen mit Einsen in der Diagonalen) lässt sich suggestiv als ein *straight-line-Programm* für die parallel auszuführende Transformation $(a[2n], a[2n+1]) \mapsto (\tilde{a}[n], \tilde{d}[n])$ ($n \in \mathbb{Z}$) schreiben:

$$\begin{aligned} x &\leftarrow a[2n] \\ y &\leftarrow a[2n+1] \\ y &\leftarrow y - x \\ x &\leftarrow x + \frac{1}{2}y \\ \tilde{a}[n] &\leftarrow \sqrt{2}x \\ \tilde{d}[n] &\leftarrow -\frac{1}{\sqrt{2}}y \end{aligned}$$

In Abbildung 11.1 ist das schematisch dargestellt. Aus der Produktzerlegung von H

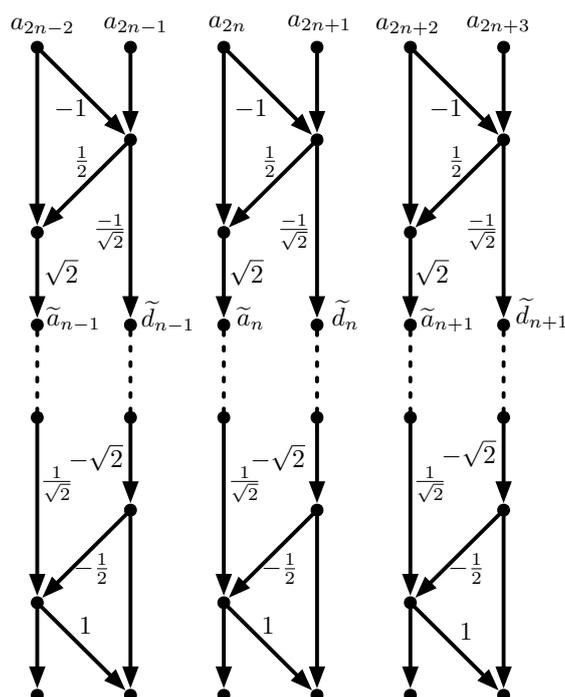


Abbildung 11.1: Lifting-Darstellung der Haar-Transformation

in besonders einfache Matrizen erhält man sofort eine entsprechende Zerlegung für die *inverse Transformation*:

$$H^{-1} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 1 & \frac{1}{2} \\ 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\frac{1}{\sqrt{2}} \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \sqrt{2} \end{bmatrix}$$

Das mag hier künstlich wirken, da ja $H = H^{-1}$ ist. Aber in der allgemeinen Situation ist das ein immenser Vorteil, da man aus der Lifting-Zerlegung sofort sieht, wie die inverse Transformation zu implementieren ist.

11.2 Faktorisierung der Daubechies-4-Transformation

Als weitere Illustration soll hier das Beispiel der Matrixzerlegung und der entsprechenden Lifting-Darstellung der Daubechies-D4-Transformation gegeben werden.

$$\begin{bmatrix} \frac{\sqrt{3}-1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}+1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & -z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{\sqrt{3}}{4} & -\frac{\sqrt{3}-2}{4}z^{-1} \end{bmatrix} \begin{bmatrix} 1 & \sqrt{3} \\ 0 & 1 \end{bmatrix}$$

$$\alpha = \sqrt{3}, \quad \beta = -\frac{\sqrt{3}}{4}, \quad \gamma = -\frac{\sqrt{3}-2}{4}, \quad \delta = -1,$$

$$\mu = \frac{\sqrt{3}-1}{\sqrt{2}}, \quad \nu = \frac{\sqrt{3}+1}{\sqrt{2}}$$

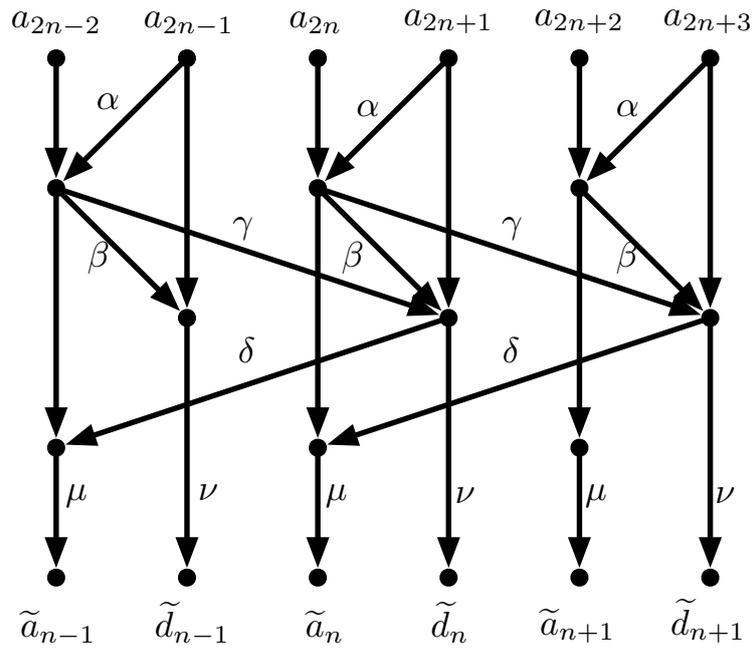


Abbildung 11.2: Lifting-Darstellung der D4-Transformation

$$\begin{aligned}\tilde{a}[n] &= \mu [(1 + \gamma\delta) a_{2n} + (\alpha + \alpha\gamma\delta) a[2n + 1] \\ &\quad + \delta\beta a[2n + 2] + (\alpha\beta\delta + \delta)a[2n + 3]] \\ \tilde{d}[n] &= \nu [(1 + \alpha\beta) a[2n + 1] + \beta a[2n] + \alpha\gamma a[2n - 1] + \gamma a[2n - 2]]\end{aligned}$$

11.3 Die Polyphasenmatrix

Vor der allgemeineren Betrachtung des Lifting ist eine kurze Bemerkung über das Verhältnis von Faltung und Downsampling nötig. Mit dem Zeichen \downarrow für das Downsampling (um den Faktor 2) ist

$$\begin{aligned}a(z) &= a_{\text{even}}(z^2) + z \cdot a_{\text{odd}}(z^2) \\ a_{\text{even}}(z) &= \downarrow a(z) = \left. \frac{a(z) + a(-z)}{2} \right|_{z^2 \leftarrow z}, \\ a_{\text{odd}}(z) &= \downarrow z^{-1} \cdot a(z) = \left. \frac{a(z) - a(-z)}{2z} \right|_{z^2 \leftarrow z}.\end{aligned}$$

Für das Downsampling einer Faltung $\mathbf{a} \star \mathbf{b}$ von zwei Signalen \mathbf{a} und \mathbf{b} gilt, im Formalismus der z -Transformierten geschrieben:

$$\downarrow (a(z) \cdot b(z)) = a_{\text{even}}(z) \cdot b_{\text{even}}(z) + z \cdot a_{\text{odd}}(z) \cdot b_{\text{odd}}(z),$$

was man schnell nachrechnet.

In der Situation einer Filterbank mit Filtern \mathbf{h}, \mathbf{g} und anschliessendem Downsampling gilt also (wenn \ddot{h} und \ddot{g} die zeitlich invertierten Filter bezeichnen):

$$\mathbf{a} = (a[n])_{n \in \mathbb{Z}} \mapsto \begin{cases} \tilde{\mathbf{a}} = (\tilde{a}[n])_{n \in \mathbb{Z}} = a_{\text{even}}(z) \cdot \ddot{h}_{\text{even}}(z) + z \cdot a_{\text{odd}}(z) \cdot \ddot{h}_{\text{odd}}(z) \\ \tilde{\mathbf{d}} = (\tilde{d}[n])_{n \in \mathbb{Z}} = a_{\text{even}}(z) \cdot \ddot{g}_{\text{even}}(z) + z \cdot a_{\text{odd}}(z) \cdot \ddot{g}_{\text{odd}}(z) \end{cases}$$

Somit lässt sich diese Transformation darstellen als

$$\begin{bmatrix} a_{\text{even}}(z) \\ a_{\text{odd}}(z) \end{bmatrix} \mapsto \begin{bmatrix} \tilde{a}(z) \\ \tilde{d}(z) \end{bmatrix} = \begin{bmatrix} \ddot{h}_{\text{even}}(z) & z \cdot \ddot{h}_{\text{odd}}(z) \\ \ddot{g}_{\text{even}}(z) & z \cdot \ddot{g}_{\text{odd}}(z) \end{bmatrix} \begin{bmatrix} a_{\text{even}}(z) \\ a_{\text{odd}}(z) \end{bmatrix}$$

Berücksichtigt man nun noch

$$\ddot{h}_{\text{even}}(z) = h_{\text{even}}(z^{-1}), \quad z \cdot \ddot{h}_{\text{odd}}(z) = h_{\text{odd}}(z^{-1})$$

so schreibt sich die Transformationsmatrix als

$$P(z^{-1})^{\mathfrak{t}} = \begin{bmatrix} h_{\text{even}}(z^{-1}) & h_{\text{odd}}(z^{-1}) \\ g_{\text{even}}(z^{-1}) & g_{\text{odd}}(z^{-1}) \end{bmatrix}$$

und die Matrix

$$P(z) = \begin{bmatrix} h_{\text{even}}(z) & g_{\text{even}}(z) \\ h_{\text{odd}}(z) & g_{\text{odd}}(z) \end{bmatrix}$$

bezeichnet man als die *Polyphasenmatrix* zum Filterpaar (\mathbf{h}, \mathbf{g})

Die folgende Aussage zeigt, wie sich die Orthogonalität eines Filterpaares (\mathbf{h}, \mathbf{g}) mittels der Polyphasenmatrix ausdrückt:

Satz 11.1 Das Filterpaar (\mathbf{h}, \mathbf{g}) ist genau dann orthogonal, wenn

$$P(z^2)^{-1} = P(z^{-2})^{\mathfrak{t}}.$$

Der Beweis ist nicht schwierig. Es ist

$$P(z^2) = \begin{bmatrix} h_{\text{even}}(z^2) & g_{\text{even}}(z^2) \\ h_{\text{odd}}(z^2) & g_{\text{odd}}(z^2) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 \\ z^{-1} & -z^{-1} \end{bmatrix} \begin{bmatrix} h(z) & g(z) \\ h(-z) & g(-z) \end{bmatrix}$$

und somit

$$P(z^{-2})^{\mathfrak{t}} \cdot P(z^2) = \frac{1}{4} \begin{bmatrix} h(z^{-1}) & h(-z^{-1}) \\ g(z^{-1}) & g(-z^{-1}) \end{bmatrix} \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \begin{bmatrix} 1 & 1 \\ z^{-1} & -z^{-1} \end{bmatrix} \begin{bmatrix} h(z) & g(z) \\ h(-z) & g(-z) \end{bmatrix}.$$

Wegen

$$\begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \begin{bmatrix} 1 & 1 \\ z^{-1} & -z^{-1} \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

ergibt sich

$$P(z^{-2})^{\mathfrak{t}} \cdot P(z^2) = \frac{1}{2} \begin{bmatrix} h(z^{-1})h(z) + h(-z^{-1})h(-z) & h(z^{-1})g(z) + h(-z^{-1})g(-z) \\ g(z^{-1})h(z) + g(-z^{-1})h(-z) & g(z^{-1})g(z) + g(-z^{-1})g(-z) \end{bmatrix}.$$

Jetzt muss man sich nur an die Orthogonalitätsaussagen aus Kapitel 8 erinnern. \square

Ganz entsprechend kann man biorthogonale Filterpaare charakterisieren. Zur Erinnerung: ein biorthogonales Filterpaar $\mathbf{h}, \tilde{\mathbf{h}}$ ist durch die Bedingung

$$\tilde{H}(\omega) \cdot \overline{H(\omega)} + \tilde{H}(\omega + \pi) \cdot \overline{H(\omega + \pi)} = 2$$

für die Frequenzdarstellungen gekennzeichnet: Die Filter $\mathbf{g}, \tilde{\mathbf{g}}$ sind durch

$$G(\omega) = e^{i\omega} \overline{\tilde{H}(\omega + \pi)}, \quad \tilde{G}(\omega) = e^{i\omega} \overline{G(\omega + \pi)}$$

bestimmt, was auf Koeffizientenebene

$$g_k = (-1)^k \tilde{h}_{1-k}, \quad \tilde{g}_k = (-1)^k h_{1-k}$$

bedeutet. Die charakteristische Bedingung schreibt sich als Gleichung für die z -Transformierten als

$$\tilde{h}(z) h(z^{-1}) + z g(z^{-1}) z^{-1} \tilde{g}(z) = 2$$

und es ist ausserdem

$$\tilde{h}(z) h(-z^{-1}) + g(-z^{-1}) \tilde{g}(z) = 0$$

Zusammengefasst formuliert sich Biorthogonalität also als

$$\begin{bmatrix} h(z^{-1}) & g(z^{-1}) \\ h(-z^{-1}) & g(-z^{-1}) \end{bmatrix} \begin{bmatrix} \tilde{h}(z) & \tilde{h}(-z) \\ \tilde{g}(z) & \tilde{g}(-z) \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Mit

$$P(z) = \begin{bmatrix} h_{\text{even}}(z) & g_{\text{even}}(z) \\ h_{\text{odd}}(z) & g_{\text{odd}}(z) \end{bmatrix}, \quad \tilde{P}(z) = \begin{bmatrix} \tilde{h}_{\text{even}}(z) & \tilde{g}_{\text{even}}(z) \\ \tilde{h}_{\text{odd}}(z) & \tilde{g}_{\text{odd}}(z) \end{bmatrix}$$

ergibt sich nun:

$$\begin{aligned} \tilde{P}(z^{-2})^t \cdot P(z^2) &= \frac{1}{4} \begin{bmatrix} \tilde{h}(z^{-1}) & \tilde{h}(-z^{-1}) \\ \tilde{g}(z^{-1}) & \tilde{g}(-z^{-1}) \end{bmatrix} \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix} \begin{bmatrix} 1 & 1 \\ z^{-1} & -z^{-1} \end{bmatrix} \begin{bmatrix} h(z) & g(z) \\ h(-z) & g(-z) \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \tilde{h}(z^{-1}) & \tilde{h}(-z^{-1}) \\ \tilde{g}(z^{-1}) & \tilde{g}(-z^{-1}) \end{bmatrix} \begin{bmatrix} h(z) & g(z) \\ h(-z) & g(-z) \end{bmatrix} \end{aligned}$$

und somit

Satz 11.2 Das Filterpaar $(\mathbf{h}, \tilde{\mathbf{h}})$ definiert genau dann (zusammen mit \mathbf{g} und $\tilde{\mathbf{g}}$) eine biorthogonale Transformation, wenn

$$\tilde{P}(z^2)^{-1} = P(z^{-2})^t.$$

Was bedeutet nun ‘‘Lifting’’? Es geht darum, die Polyphasenmatrix $P(z)$ eines Filterpaares (\mathbf{h}, \mathbf{g}) als Produkt von sehr einfachen Matrizen zu schreiben:

$$\begin{aligned} P(z) &= \begin{bmatrix} h_{\text{even}}(z) & g_{\text{even}}(z) \\ h_{\text{odd}}(z) & g_{\text{odd}}(z) \end{bmatrix} \\ &= \begin{bmatrix} 1 & q_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_2(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & q_3(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_4(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ q_k(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \end{aligned}$$

Dabei kann k ohne Einschränkung als gerade angenommen werden. Die $q_i(z)$ werden, wenn \mathbf{h} und \mathbf{g} endliche Filter sind (nur der Fall soll uns hier interessieren) Polynome in z in einem erweiterten Sinne sein: das sind endliche Summen $\sum_{j=s}^t \alpha_j z^j$, wobei positive und negative Potenzen von z auftreten können, d.h. es ist $s \leq t$, aber diese Summationsgrenzen können positiv oder auch negativ sein. Man spricht in Anlehnung an den funktionentheoretischen Begriff der Laurent-Reihen von *Laurent-Polynomen*.

In grafischer Darstellung kann man mittels der Produktdarstellung der Polyphasenmatrix $P(z)$ die durch \mathbf{h} und \mathbf{g} definierte Filterbanktransformation so darstellen:

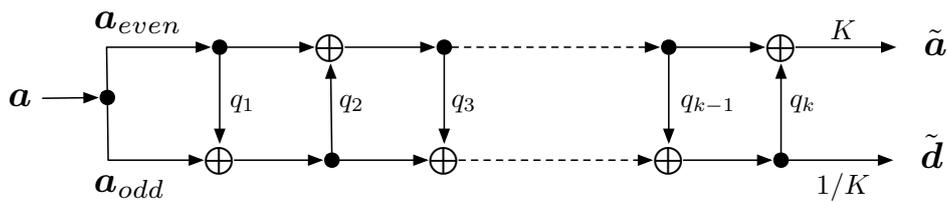


Abbildung 11.3: Liftinganalyse

Ein Bonus dieser Liftingdarstellung ist, dass sich die Darstellung der Umkehrtransformation unmittelbar ergibt:

$$P(z)^{-1} = \begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -q_k(z) & 1 \end{bmatrix} \cdots \begin{bmatrix} 1 & 0 \\ -q_4(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -q_3(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -q_2(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -q_1(z) \\ 0 & 1 \end{bmatrix}$$

was sich grafisch entsprechend veranschaulichen lässt:

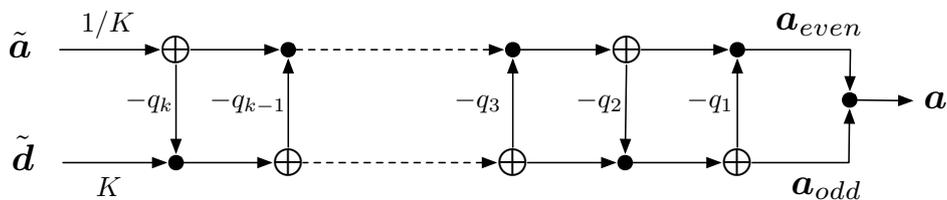


Abbildung 11.4: Liftingssynthese

11.4 Euklidische Division und Lifting

Wie kommt man nun zu einer Lifting-Zerlegung der Polyphasenmatrix? Die vielleicht überraschende Antwort: der *euklidische Algorithmus* für Polynome leistet genau dies! Die erforderlichen Fakten sollen kurz wiederholt werden. Grundlegend ist:

Satz 11.3 Zu Polynomen $a(X), b(X)$ mit $b(X) \neq 0$ (mit Koeffizienten in einem Körper) gibt es eindeutig bestimmte Polynome $q(X), r(X)$ mit

$$a(X) = b(X) \cdot q(X) + r(X), \quad \text{wobei } r(X) = 0 \text{ oder } \deg r(X) < \deg b(X).$$

Der *euklidische Algorithmus* sollte bekannt sein. Mittels iterierter Division mit Rest bestimmt man den grössten gemeinsamen Teiler zweier Polynome. Das Rechen-schema sieht so aus:

$$\begin{array}{l}
 \text{input } r_0 = r_0(X) = a(X), r_1 = r_1(X) = b(X) \\
 r_0 = r_1 \cdot q_1 + r_2 \\
 r_1 = r_2 \cdot q_2 + r_3 \\
 r_2 = r_3 \cdot q_3 + r_4 \\
 \vdots \\
 r_{k-2} = r_{k-1} \cdot q_{k-1} + r_k \quad (r_k \neq 0) \\
 r_{k-1} = r_k \cdot q_k + 0 \\
 \text{output } r_k(X) = \text{ggT}(a(X), b(X))
 \end{array}$$

Das kann man auch übersichtlich in Matrixform schreiben:

$$\begin{array}{l}
 \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -q_1 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} \\
 \begin{bmatrix} r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -q_2 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \\
 \vdots \\
 \begin{bmatrix} r_{k-1} \\ r_k \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -q_{k-1} \end{bmatrix} \begin{bmatrix} r_{k-2} \\ r_{k-1} \end{bmatrix} \\
 \begin{bmatrix} r_k \\ 0 \end{bmatrix} = \begin{bmatrix} q_k & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_{k-1} \\ r_k \end{bmatrix}
 \end{array}
 \qquad
 \begin{array}{l}
 \begin{bmatrix} r_0 \\ r_1 \end{bmatrix} = \begin{bmatrix} q_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \\
 \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} = \begin{bmatrix} q_2 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_2 \\ r_3 \end{bmatrix} \\
 \vdots \\
 \begin{bmatrix} r_{k-2} \\ r_{k-1} \end{bmatrix} = \begin{bmatrix} q_{k-1} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_{k-1} \\ r_k \end{bmatrix} \\
 \begin{bmatrix} r_{k-1} \\ r_k \end{bmatrix} = \begin{bmatrix} q_k & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_k \\ 0 \end{bmatrix}
 \end{array}$$

Zusammengefasst:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} q_1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_2 & 1 \\ 1 & 0 \end{bmatrix} \cdots \begin{bmatrix} q_{k-1} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} q_k & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r_k \\ 0 \end{bmatrix}$$

Eine kleine Modifikation ist nötig, um diese Produktdarstellung in eine solche umzuformen, wie sie für die Polyphasenmatrix benötigt wird:

$$\begin{bmatrix} q & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q & 1 \end{bmatrix} = \begin{bmatrix} 1 & q \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Die Lifting-Form des euklidischen Algorithmus sieht also folgendermassen aus:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 1 & q_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_2 & 1 \end{bmatrix} \begin{bmatrix} 1 & q_3 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_4 & 1 \end{bmatrix} \cdots \begin{bmatrix} r_k \\ 0 \end{bmatrix}$$

Bei der Faktorisierung der Polyphasenmatrix hat man es allerdings nicht nur mit Polynomen zu tun, sondern allgemeiner mit *Laurent-Polynomen*:

Definition 11.1 (Laurent-Polynom).

Ein Laurent-Polynom ist ein Ausdruck

$$a(X) = \sum_{i=k}^K a_i X^i \quad \text{mit } k, K \in \mathbb{Z}, k \leq K.$$

Falls $a_k \neq 0 \neq a_K$, so ist $\deg a(X) = K - k$ der Grad des Laurent-Polynoms

Für die Division von Laurent-Polynomen gilt eine Aussage wie in Satz 11.3 für Polynome festgehalten wurde:

Satz 11.4 Sind $a(X) = \sum_{i=k}^K a_i X^i$ und $b(X) = \sum_{j=\ell}^L b_j X^j$ Laurent-Polynome mit $\deg a = K - k \geq \deg b = L - \ell$, so gibt es Laurent-Polynome $q(X)$ und $r(X)$ mit

$$a = b \cdot q + r$$

und $r = 0$ oder $\deg r < \deg b$.

Diese sind aber nicht mehr eindeutig bestimmt!

■ **Beispiel 11.1** 1. Es sei $\deg a = \deg b$.

Dann geht

$$r(X) = a(X) - b(X) \cdot \frac{a_k}{b_\ell} X^{k-\ell}$$

und

$$r(X) = a(X) - b(X) \cdot \frac{a_K}{b_L} X^{k-\ell}$$

2. Es sei $\deg a = \deg b + 1$.

Dann betrachte man mit noch zu bestimmenden α, β :

$$\begin{aligned} r(X) &= a(X) - b(X) \cdot X^{k-\ell} \cdot (\alpha X + \beta) \\ &= (a_k - \beta b_\ell) X^k + (a_{k+1} - (\alpha b_\ell + \beta b_{\ell+1})) X^{k+1} + \dots \\ &\quad \dots + (a_{K-1} - (\alpha b_{L-1} + \beta b_L)) X^{K-1} + (a_K - \alpha b_L) X^K \end{aligned}$$

Es gibt die drei Möglichkeiten:

(a) Koeffizienten von X^k und von X^{k+1} verschwinden:

$$a_k = \beta b_\ell, a_{k+1} = \alpha b_\ell + \beta b_{\ell+1};$$

(b) Koeffizienten von X^k und von X^K verschwinden:

$$a_k = \beta b_\ell, \alpha_K = \alpha b_L;$$

(c) Koeffizienten von X^K und von X^{K-1} verschwinden:

$$a_K = \alpha b_L, a_{K-1} = \alpha b_{L-1} + \beta b_L,$$

aus denen sich α und β bestimmen.

Auf die genaueren Details, wie man bei der Situation der Lifting-Zerlegung der Polyphasenmatrix vorgeht, soll hier nicht weiter eingegangen werden. Wichtig ist das Prinzip. ■

Theorem 11.1 — Konstruktion der Lifting-Zerlegung.

Ist (h, g) ein orthogonales Filterpaar, so liefert die Anwendung des euklidischen Algorithmus auf die Polynome $h_{even}(z), h_{odd}(z)$ eine Lifting-Zerlegung der Polyphasenmatrix

$$\begin{aligned}
 P(z) &= \begin{bmatrix} h_{even}(z) & g_{even}(z) \\ h_{odd}(z) & g_{odd}(z) \end{bmatrix} \\
 &= \begin{bmatrix} 1 & q_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_2(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & q_3(z) \\ 0 & 1 \end{bmatrix} \dots \begin{bmatrix} 1 & q_{k-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_k(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}
 \end{aligned}$$

(Hierbei wurde der Fall angenommen, dass k gerade ist und die Input Polynome $h_{even}(z)$ und $h_{odd}(z)$ teilerfremd sind. Also $K = \text{ggT}(h_{even}, h_{odd})$.)

Es fällt auf, dass die Polynome $g_{even}(z)$ und $g_{odd}(z)$ bei der Berechnung der Zerlegung überhaupt keine Rolle spielen! Die Quotientenfolge hängt nur von $h_{even}(z)$ und $h_{odd}(z)$ ab. Das hat man so zu verstehen: Aus der Lifting-Zerlegung ergibt sich automatisch, was $g_{even}(z)$ und $g_{odd}(z)$ sein müssen, wenn man die Quotientenfolge kennt!

11.5 Die Wavelettransformationen in JPEG2000

Der Standard JPEG2000 für die Codierung, Speicherung, Kompression und Übertragung von Bildern verwendet zwei Wavelettransformationen: das im Abschnitt 9.3.3 behandelte biorthogonale symmetrische Cohen-Daubechies-Feauveau-(7,9)-Filterpaar für *verlustbehaftete* Kompression und das biorthogonale symmetrische LeGall-Filterpaar für *verlustfreie* Kompression.

11.5.1 Lifting-Schema für das Cohen-Daubechies-Feauveau-(7,9)-Wavelet

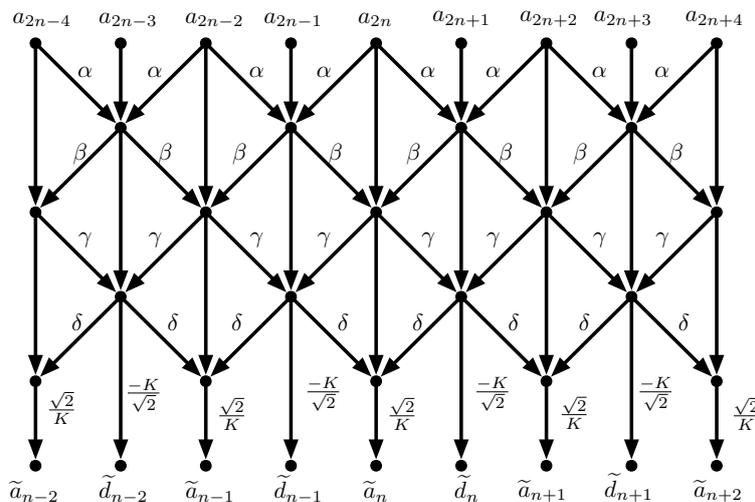


Abbildung 11.5: Lifting-Schema für das CDF-(7,9)-Filterpaar

Numerische Werte der Koeffizienten $\alpha, \beta, \gamma, \delta, K$:

$$\begin{aligned}\alpha &= -1.586134342 \\ \beta &= -0.052980118 \\ \gamma &= 0.882911075 \\ \delta &= 0.443506852 \\ K &= 1.230174105\end{aligned}$$

Koeffizienten des h -Filters, ausgedrückt durch die Koeffizienten $\alpha, \beta, \gamma, \delta, K$:

$$\begin{aligned}h_0 &= \frac{\sqrt{2}}{K} [6\alpha\beta\gamma\delta + 2\alpha\beta + 2\gamma\delta + 2\alpha\delta + 1] \\ h_1 &= \frac{\sqrt{2}}{K} [3\beta\gamma\delta + \beta + \delta] \\ h_2 &= \frac{\sqrt{2}}{K} [4\alpha\beta\gamma\delta + \gamma\delta + \alpha\delta + \alpha\beta] \\ h_3 &= \frac{\sqrt{2}}{K} [\beta\gamma\delta] \\ h_4 &= \frac{\sqrt{2}}{K} [\alpha\beta\gamma\delta] \\ g_1 &= \frac{-K}{\sqrt{2}} [1 + 2\beta\gamma] \\ g_2 &= \frac{-K}{\sqrt{2}} [3\alpha\beta\gamma + \alpha + \gamma] \\ g_3 &= \frac{-K}{\sqrt{2}} [\beta\gamma] \\ g_4 &= \frac{-K}{\sqrt{2}} [\alpha\beta\gamma]\end{aligned}$$

11.5.2 Liftingschema für das biorthogonale LeGall-(5,3)-Wavelet

Die Filter sind

$$\begin{aligned}\mathbf{h} &= \frac{1}{8} [-1 \quad 2 \quad 6 \quad 2 \quad -1]_{-2..2} \\ \tilde{\mathbf{h}} &= \frac{1}{2} [1 \quad 2 \quad 1]_{-1..1} \\ \mathbf{g} &= \frac{1}{2} [-1 \quad 2 \quad -1]_{0..2} \\ \tilde{\mathbf{g}} &= \frac{1}{8} [-1 \quad -2 \quad 6 \quad -2 \quad -1]_{-1..3}\end{aligned}$$

Dabei ist also $\tilde{\mathbf{h}}$ ein Spline-Filter. Als Lifting dargestellt (mit $\alpha = -\frac{1}{2}$ und $\beta = \frac{1}{4}$):

$$\begin{aligned}\tilde{d}[n] &= a[2n+1] + \alpha (a[2n] + a[2n+2]) \\ \tilde{a}[n] &= a[2n] + \beta (\tilde{d}[n-1] + \tilde{d}[n]) \\ a[2n] &= \tilde{a}[n] - \beta (\tilde{d}[n-1] + \tilde{d}[n]) \\ a[2n+1] &= \tilde{d}[n] - \alpha (a[2n] + a[2n+2])\end{aligned}$$

Das Interessante an diesem Filterpaar ist, dass man damit eine umkehrbare *integer-to-integer* Transformation definieren kann – das ist der Grund für die Verlustfreiheit! Dafür transformiert man:

$$\begin{aligned}d[n]^* &= a[2n+1] - \left[\frac{1}{2} (a[2n] + a[2n+2]) \right] \\ a[n]^* &= a[2n] + \left[\frac{1}{4} (d[n-1]^* + d[n]^*) \right]\end{aligned}$$

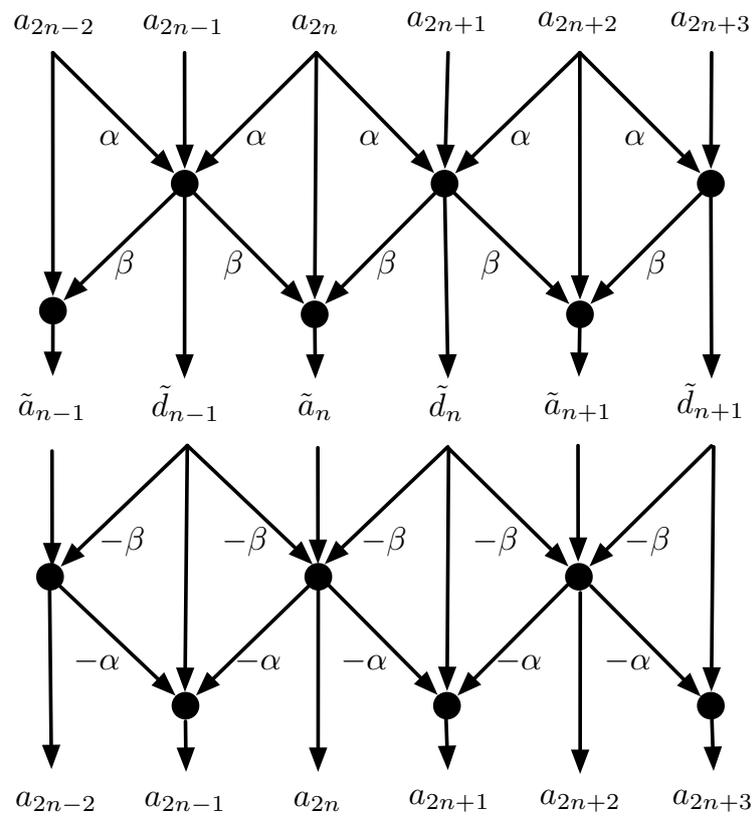


Abbildung 11.6: Lifting-Schema der LeGall-Transformation

und prüft nach, dass die Rücktransformation durch

$$\begin{aligned} a[2n] &= a[n]^* - \left\lfloor \frac{1}{4} (d[n-1]^* + d[n]^*) + \frac{1}{2} \right\rfloor \\ a[2n+1] &= d[n]^* + \left\lfloor \frac{1}{2} (a[2n] + a[2n+2]) \right\rfloor \end{aligned}$$

korrekt beschrieben ist. Der Grund hierfür ist:

$$\left\lfloor \frac{a+b}{2} \right\rfloor = \begin{cases} \frac{a+b}{2} & \text{falls } a+b \text{ gerade} \\ \frac{a+b}{2} - \frac{1}{2} & \text{falls } a+b \text{ ungerade} \end{cases}$$

12. Denoising using wavelets

This chapter contains the slides of the lectures dealing with denoising of signals and using the discrete wavelet transform images. For illustrations see the material in the web page of the lecture.

- Threshold functions
- Wavelet shrinking
- The VISUSHRINK method
- The SURE method

- ▶ Threshold functions $s_\lambda(t)$
 - ▶ are used to suppress parts of a signal with very low amplitudes and (usually) high frequencies (“noise”)
 - ▶ examples are

$$\begin{aligned}
 \{\text{Hard}, \lambda\} & \begin{cases} 0 & |x| \leq \lambda \\ x & |x| > \lambda \end{cases} \\
 \{\text{Soft}, \lambda\} & \begin{cases} 0 & |x| \leq \lambda \\ \text{sgn}(x)(|x| - \lambda) & |x| > \lambda \end{cases} \\
 \{\text{PiecewiseGarrote}, \lambda\} & \begin{cases} 0 & |x| \leq \lambda \\ x - \frac{\lambda^2}{x} & |x| > \lambda \end{cases} \\
 \{\text{SmoothGarrote}, \lambda, n\} & \frac{x^{2n+1}}{x^{2n} + \lambda^{2n}} \\
 \{\text{Hyperbola}, \lambda\} & \begin{cases} 0 & |x| \leq \lambda \\ \text{sgn}(x)\sqrt{x^2 - \lambda^2} & |x| > \lambda \end{cases}
 \end{aligned}$$

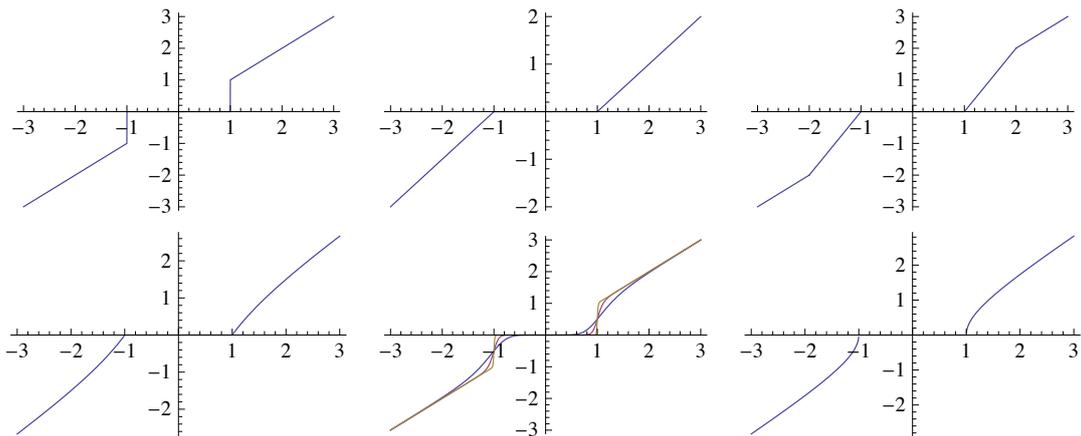


Figure: Examples of threshold functions $s_\lambda(t)$

▶ Setting and strategy

- ▶ “true” signal : $\mathbf{v} = (v_1, v_2, \dots, v_N)$
- ▶ noise vector: $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N)$
- ▶ noised signal $\mathbf{y} = (y_1, y_2, \dots, y_N) = \mathbf{v} + \boldsymbol{\varepsilon}$
- ▶ wavelet filtering (orthogonal transform!)

$$\mathbf{y} \xrightarrow{WT} \mathbf{z} = (\mathbf{a}, \mathbf{d}) = (H\mathbf{y}, G\mathbf{y})$$

- ▶ applying thresholding with $s_\lambda(t)$ to the high-pass component

$$\mathbf{z} \mapsto \hat{\mathbf{z}} = (\mathbf{a}, \hat{\mathbf{d}}) \quad \text{with} \quad \hat{\mathbf{d}} = s_\lambda(\mathbf{d})$$

- ▶ inverse wavelet transform

$$\hat{\mathbf{z}} \xrightarrow{WT^{-1}} \hat{\mathbf{v}} = H^\dagger \mathbf{a} + G^\dagger \hat{\mathbf{d}}$$

▶ Heuristic considerations

- ▶ Noise modelled as *Gaussian white noise* with noise level σ :
 $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N)$ generated by independent and identically $\mathcal{N}(0, \sigma^2)$ -distributed random variables
- ▶ For a vector $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_N)$ of independent, $\mathcal{N}(0, \sigma^2)$ -distributed random variables ε_i and an orthogonal $(N \times N)$ matrix U the components γ_i of the vector

$$\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_N) = U\boldsymbol{\varepsilon},$$

are again independent $\mathcal{N}(0, \sigma^2)$ -distributed random variables

- ▶ Due to orthogonality of the wavelet transform the transformed noise term $WT(\boldsymbol{\varepsilon})$ in

$$\mathbf{y} = \mathbf{v} + \boldsymbol{\varepsilon} \mapsto WT(\mathbf{y}) = WT(\mathbf{v}) + WT(\boldsymbol{\varepsilon})$$

is still characterized by being white noise with noise level σ

- ▶ Heuristic considerations (contd.)
 - ▶ In wavelet transformations most energy goes into the approximation (low-pass) component \mathbf{a}
 - ▶ Noise of high frequency goes into the detail (high-pass) component \mathbf{d}
 - ▶ \implies the detail component mainly (but not exclusively) consists of noise (detail coefficients $\lesssim \sigma$) — that is where to attack!
 - ▶ The problem: the noise level σ is not known and has to be estimated from the denoised data themselves
 - ▶ How to choose λ (depending on the estimate for σ) ?
 - ▶ Measure of quality for denoising: *mean squared error* (MSE)

$$E [\|\mathbf{v} - \hat{\mathbf{v}}\|^2] = E \left[\sum_{1 \leq j \leq N} (v_j - \hat{v}_j)^2 \right]$$

- ▶ $\mathbf{v} = (v_1, \dots, v_N) \in \mathbb{R}^N$
- ▶ $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_N)$ white noise with variance σ^2 , $\mathbf{y} = \mathbf{v} + \boldsymbol{\varepsilon}$
- ▶ $\hat{\mathbf{v}} = (\hat{v}_1, \dots, \hat{v}_N)$ estimate for \mathbf{v} with $A \subseteq \{1, 2, \dots, N\}$ and

$$\hat{v}_j = \begin{cases} y_j & \text{if } j \in A \\ 0 & \text{if } j \notin A \end{cases}$$

- ▶ In this case the MSE equals

$$E [\|\mathbf{v} - \hat{\mathbf{v}}\|^2] = E \left[\sum_{1 \leq j \leq N} (v_j - \hat{v}_j)^2 \right] = \sum_{j \in A} E [\varepsilon_j^2] + \sum_{j \notin A} E [v_j^2]$$

and this is minimized by setting $j \in A \iff v_j^2 > \sigma^2$

- ▶ so that the ideal MSE is

$$E [\|\mathbf{v} - \hat{\mathbf{v}}\|^2] = \sum_{1 \leq j \leq N} \min(v_j^2, \sigma^2)$$

▶ VISUSHRINK :

- ▶ is Wavelet shrinkage with

$$\lambda = \lambda^{\text{univ}} = \sigma \cdot \sqrt{2 \ln N} \quad (\text{"universal tolerance"})$$

- ▶ Theorem [DONOHO-JOHNSTONE, 1995]

For $\mathbf{v} \in \mathbb{R}^N$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \dots, \varepsilon_N)$ white noise with noise level σ ,

for $\hat{\mathbf{v}} = s_\lambda(\mathbf{v} + \boldsymbol{\varepsilon})$ (soft threshold) one gets

$$E [\|\mathbf{v} - \hat{\mathbf{v}}\|^2] \leq (2 \ln N + 1) \cdot \left(\sigma^2 + \sum_{1 \leq j \leq N} \min(v_j^2, \sigma^2) \right)$$

- ▶ σ will be estimated (on the first high-pass component!) using the *mean absolute deviation* (MAD):

- ▶ $\mathbf{w} = (w_1, \dots, w_N)$
- ▶ $\tilde{\mathbf{w}} = \text{Median of } \mathbf{w}$
- ▶ $\mathbf{v} = (|w_1 - \tilde{w}|, \dots, |w_N - \tilde{w}|)$
- ▶ $\text{MAD}(\mathbf{w}) = \text{Median of } \mathbf{v} = \tilde{v}$

- ▶ Theorem [HAMPEL, 1974]

$$\text{MAD}(\mathbf{w}) \approx 0.6745 \cdot \sigma$$

- ▶ SURE method (STEINS unbiased risk estimator, 1981)

- ▶ Goal: choice of the λ parameter for soft-shrinking methods

$$s_\lambda(x) = \begin{cases} x - \lambda & \text{if } x > \lambda \\ 0 & \text{if } |x| \leq \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases}$$

- ▶ C.M. STEIN, Estimation of the mean of a multivariate normal distribution, *Ann. Stat.* 1981.

D. DONOHO, I. JOHNSTONE, Adapting to unknown smoothness via wavelet shrinkage, *J. Amer. Stat. Assoc.* 1995.

P. VAN FLEET, *Discrete Wavelet Transformations*, Wiley, 2008 (ch. 9).

► Lemma:

For a $\mathcal{N}(0, \sigma^2)$ -distributed random variable ε , any $z \in \mathbb{R}$ and any piecewise differentiable function $g : \mathbb{R} \rightarrow \mathbb{R}$ one has

$$\mathbb{E}[\varepsilon \cdot g(z + \varepsilon)] = \sigma^2 \cdot \mathbb{E}[g'(z + \varepsilon)]$$

► This follows from partial integration:

$$\begin{aligned} \mathbb{E}[\varepsilon \cdot g(z + \varepsilon)] &= \frac{1}{\sqrt{2\pi\sigma^2}} \int x \cdot g(z + x) \cdot e^{-\frac{x^2}{2\sigma^2}} dx \\ &= \frac{-1}{\sqrt{2\pi\sigma^2}} \int \sigma^2 \cdot g(y) \cdot \frac{d}{dy} e^{-\frac{(y-z)^2}{2\sigma^2}} dy \\ &= \sigma^2 \cdot \frac{1}{\sqrt{2\pi\sigma^2}} \int \frac{d}{dy} g(y) \cdot e^{-\frac{(y-z)^2}{2\sigma^2}} dy \\ &= \sigma^2 \cdot \mathbb{E}[g'(z + \varepsilon)] \end{aligned}$$

► Consequence:

With z, ε, g as in the Lemma, one gets for the MSE of the estimate $\hat{z} = w + g(w)$ of the random variable $w = z + \varepsilon$:

$$\begin{aligned} \mathbb{E}[(\hat{z} - z)^2] &= \mathbb{E}[(\varepsilon + g(z + \varepsilon))^2] \\ &= \mathbb{E}[\varepsilon^2 + 2\varepsilon \cdot g(z + \varepsilon) + g(z + \varepsilon)^2] \\ &= \mathbb{E}[\sigma^2 + 2\sigma^2 \cdot g'(w) + g(w)^2] \end{aligned}$$

► Special case: soft-shrinking with threshold value λ

► The function is

$$g(z) = \begin{cases} -z & \text{if } |z| < \lambda \\ -\lambda \operatorname{sgn}(z) & \text{if } |z| \geq \lambda \end{cases}$$

and thus

$$\frac{d}{dz} g(z) = \begin{cases} -1 & \text{if } |z| < \lambda \\ 0 & \text{if } |z| \geq \lambda \end{cases}$$

► Therefore

$$\sigma^2 + 2\sigma^2 g'(w) + g(w)^2 = \begin{cases} w^2 - \sigma^2 & \text{if } |w| < \lambda \\ \sigma^2 + \lambda^2 & \text{if } |w| \geq \lambda \end{cases}$$

► General situation:

- $\mathbf{z} = (z_1, z_2, \dots, z_N) \in \mathbb{R}^N$
- $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N)$ vector of $\mathcal{N}(0, \sigma_k^2)$ -distributed random variables
(noise, not necessarily independent and identically distributed)
 $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_N)$
- $\mathbf{w} = \mathbf{z} + \boldsymbol{\varepsilon}$ noised vector
- $\mathbf{g} = (g_1, g_2, \dots, g_N)$ with $g_k : \mathbb{R}^N \rightarrow \mathbb{R}$ correcting functions
- $\hat{\mathbf{z}} = \mathbf{w} + \mathbf{g}(\mathbf{w})$ estimate for \mathbf{z}

► Theorem (STEIN)

With the notions just introduced, the MSE can be written as

$$\begin{aligned} \mathbb{E} [\|\hat{\mathbf{z}} - \mathbf{z}\|^2] &= \mathbb{E} \left[\sum_{1 \leq j \leq N} \left(\sigma_j^2 + 2\sigma_j^2 \frac{\partial}{\partial w_j} g_j(\mathbf{w}) + g_j(\mathbf{w})^2 \right) \right] \\ &= \|\boldsymbol{\sigma}\|^2 + 2 \sum_{1 \leq j \leq N} \sigma_j^2 \mathbb{E} \left[\frac{\partial}{\partial w_j} g_j(\mathbf{w}) \right] + \mathbb{E} [\|\mathbf{g}(\mathbf{w})\|^2] \end{aligned}$$

► Special case: soft-shrinking with threshold value λ

$$\begin{aligned} \mathbb{E} [\|\hat{\mathbf{z}} - \mathbf{z}\|^2] &= \mathbb{E} \left[\sum_{1 \leq j \leq N} \left(w_j^2 - \sigma_j^2 + (2\sigma_j^2 - w_j^2 + \lambda^2) \chi_{|w_j| \geq \lambda} \right) \right] \\ &= \mathbb{E} [\|\mathbf{w} - \boldsymbol{\sigma}\|^2] + \mathbb{E} \left[\sum_{1 \leq j \leq N} (2\sigma_j^2 - w_j^2 + \lambda^2) \chi_{|w_j| \geq \lambda} \right] \end{aligned}$$

► The left summand is independent of λ .

Minimizing the MSE can be achieved by choosing λ depending on the sample vector \mathbf{w} so that the integrand in the second summand is minimal!

$$f(\lambda) = \sum_{1 \leq j \leq N} (2\sigma_j^2 - w_j^2 + \lambda^2) \chi_{|w_j| \geq \lambda}$$

- ▶ Instead of dealing with $f(\lambda)$ it is more convenient to consider

$$\tilde{f}(\lambda) = \sum_{1 \leq j \leq N} (2\sigma_j^2 - w_j^2 + \lambda^2) \chi_{|w_j| > \lambda}$$

which changes nothing as far as the expectation is concerned

- ▶ Assume that the components of the vector $\mathbf{w} = (w_1, \dots, w_N)$ are ordered by increasing absolute value

$$|w_0| = 0 \leq |w_1| \leq |w_2| \leq \dots \leq |w_N|$$

- ▶ The function $\tilde{f}(\lambda)$ is continuous from the right for $\lambda \in \mathbb{R}_+$, and if $|w_j| < |w_{j+1}|$ holds, then $\tilde{f}(\lambda)$ is strictly increasing on the half-open interval $[|w_j|, |w_{j+1}|)$, so that it takes its minimum at $|w_j|$:

$$\tilde{f}(|w_j|) = (N - j)w_j^2 + \sum_{j+1 \leq k \leq N} (2\sigma_k^2 - w_k^2)$$

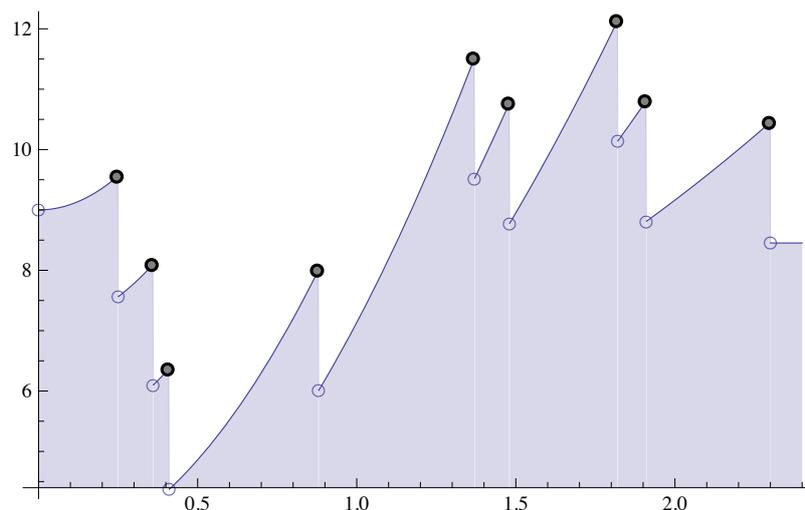


Figure: Example for the computation of the minimum value of $f(\lambda)$ for the sequence $\{0.25, 0.36, 0.41, 0.88, 1.37, 1.48, 1.82, 1.91, 2.3\}$

- ▶ From this one gets

$$\min_{\lambda \in \mathbb{R}_+} \tilde{f}(\lambda) = \min_{0 \leq j \leq N} \tilde{f}(|w_j|) \quad \lambda^{sure} := \operatorname{argmin}_{|w_j|} \tilde{f}(|w_j|)$$

- ▶ The definition of \tilde{f} yields a (downward) recursion

$$\tilde{f}(|w_j|) = \tilde{f}(|w_{j+1}|) + 2\sigma_{j+1}^2 + (N - j)(w_j^2 - w_{j+1}^2)$$

which starting from

$$\tilde{f}(w_N) = 0$$

gives a fast computation of the minimum!

- ▶ The usual assumption $\sigma_1 = \dots = \sigma_N$ somewhat simplifies the formulas and the computations

SURE in the context of the wavelet transform

- ▶ Consider a one-level WT

$$\mathbf{z} \mapsto \mathbf{y} = \mathbf{z} + \varepsilon \xrightarrow{WT} (\mathbf{a}, \mathbf{d}) \mapsto (\mathbf{a}, \hat{\mathbf{d}}) \xrightarrow{WT^{-1}} \hat{\mathbf{z}}$$

where

$$\mathbf{d} = G(\mathbf{z} + \varepsilon) = G\mathbf{z} + G\varepsilon \xrightarrow{s_\lambda} \hat{\mathbf{d}}$$

Note: the high-pass component $G\varepsilon$ has the same noise as ε

- ▶ σ must be estimated beforehand (as in the VISUSHRINK method)
- ▶ One has $\lambda^{sure} \leq \lambda^{univ}$. Recommendation: If \mathbf{y} is sparse, it is better to use λ^{univ} instead of λ^{sure} , based on the criterion

$$\frac{1}{N} \sum_{1 \leq j \leq n} (y_j^2 - \sigma^2) \leq \frac{3}{2\sqrt{N}} \log_2(N)$$

(DONOHOE, JOHNSTONE)



13. DWT and the SPIHT algorithm

13.1 Introduction

13.1.1 What this chapter is about

This chapter gives an outline and examples on a method for image compression, called SPIHT (= *set partitioning in hierarchical trees*), that is prototypical for the application of wavelet transform methods in this field. It combines two ideas: The first one, *progressive transmission* is frequently found in signal transmission and data compression methods: digital data (like pixel values of an image) are organized in *bit-planes*, thus giving preference in the transmission process to high significant bits over less significant bits. The second idea, *embedded trees*, makes use of dependencies that exist in wavelet transformed data over several layers of resolution (sub-bands) so that pixels that relate to the same position in the original image are organized in quaternary trees

13.1.2 References for this chapter

1. Methods for data compression, including wavelet methods for image compression, are treated in an encyclopedic way in [SM10]:

D. Salomon, G. Motta, *Handbook of Data Compression*, Springer-Verlag London, 2010.

The book has 1360 pages, of which more than 120 are devoted to wavelet methods (Chap. 8). The SPIHT algorithm is the subject of Section 8.14 (12 pages).

2. A shorter presentation of both wavelet methods in general and SPIHT in particular is contained in Chapter 4 and Section 4.8 of [Sal02]:

D. Salomon, *A Guide to Data Compression Methods*, Springer-Verlag New York, 2002.

3. Another recent source is [Say12]:

K. Sayood, *Introduction to Data Compression*, 4-th ed., Elsevier, 2012.

Chapter 12 introduces Fourier techniques, Chap. 13 deals with Transform Coding, Chap. 14 with Sub-band coding (including filter banks and all that), Chap. 15 gives a concise treatment of wavelets, and Chap. 16 is devoted to Wavelet-based Image Compression. SPIHT is presented in Sec. 16.4., the JPEG2000 standard is discussed in detail in Sec. 16.5.

4. A wavelet book that gives special attention to data compression is [RB98]:

R.M. Rao, A.S. Bopardikar, *Wavelet Transforms, Introduction to Theory and Applications*, Springer, Addison-Wesley, 1998.

Chapter 5 is on Wavelet Transform and Data Compression, Sec. 5.3.2 presents the SPIHT algorithm, including an example that is reproduced in this notes.

5. It is always worthwhile to look into [Mal09]:

S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd. ed., Elsevier, 2009.

This book is mathematically more demanding than the others mentioned before. Chap. 10 is devoted to Compression. Sec. 10.4.2 gives a compact presentation of embedded coding and SPIHT.

6. The SPIHT algorithm is due to [SP96]:

A. Said, W.A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, *IEEE Trans. Circ. Syst. Video Technol.* 6(3): 243–250, 1996.

Due to its success the authors have established a web page cipr.rpi.edu where you can find lots of material (descriptions, demos, code, applications, ...) and you can also download the mentioned article (which is very readable!). The list of references shows that there have been many approaches to the use of wavelets in data compression. SPIHT undeniably belongs to the most efficient ones.

13.2 The scheme of wavelet transform image coding

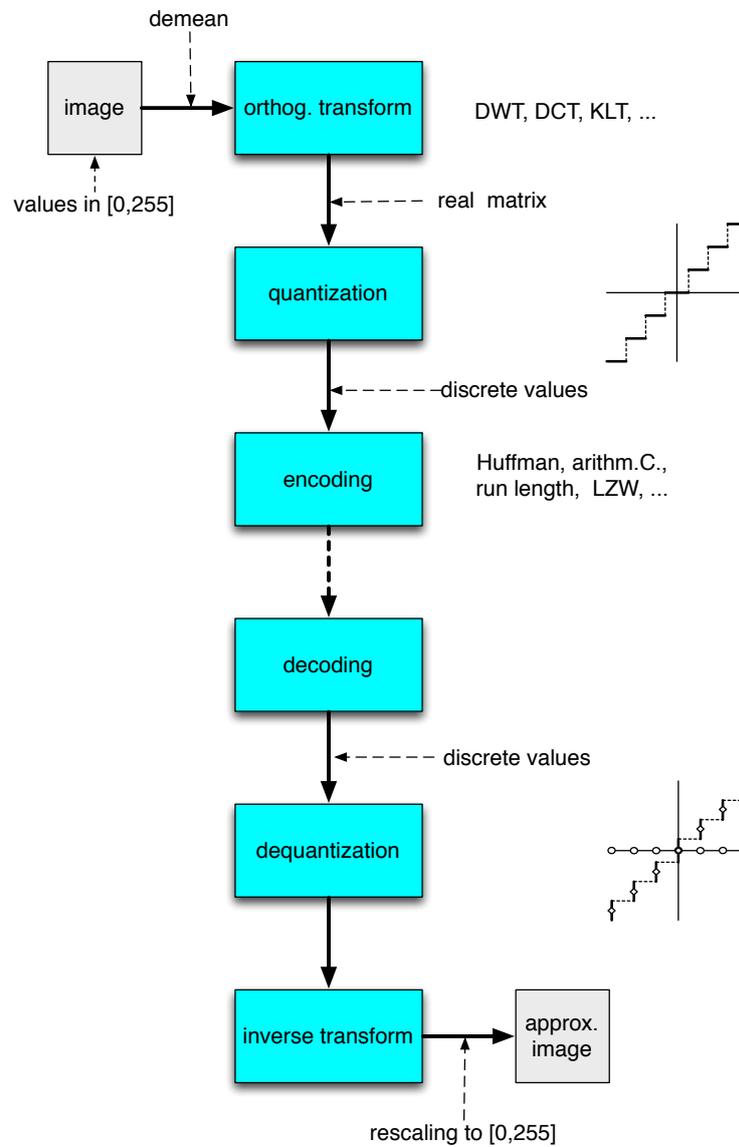


Abbildung 13.1: Structure of a wavelet transform image codec

13.3 Progressive transmission

13.3.1 Heuristics for progressive transmission

Assume that $\mathbf{p} = \{p_{i,j}\}_{(i,j) \in I}$ is an image, which is transform-coded by applying an orthogonal transform \mathbf{T} to it, which yields the data $\mathbf{c} = \{c_{i,j}\}_{(i,j) \in I}$. An approximation $\hat{\mathbf{c}} = \{\hat{c}_{i,j}\}_{(i,j) \in I}$ (e.g., the k most significant bits of each coefficient) is obtained by transforming and rearranging using a procedure \mathbf{S} and coefficient-wise back-transforming using \mathbf{T}^{-1} , which gives an approximation $\hat{\mathbf{p}} = \{\hat{p}_{i,j}\}_{(i,j) \in I}$ of the original input \mathbf{p} . Schematically

$$\mathbf{p} \xrightarrow{\mathbf{T}} \mathbf{c} \xrightarrow{\mathbf{S}} \hat{\mathbf{c}} \xrightarrow{\mathbf{T}^{-1}} \hat{\mathbf{p}}$$

Then, by orthogonality of T ,

$$\|\mathbf{p} - \hat{\mathbf{p}}\|^2 = \sum_{(i,j) \in I} |p_{i,j} - \hat{p}_{i,j}|^2 = \sum_{(i,j) \in I} |c_{i,j} - \hat{c}_{i,j}|^2 = \|\mathbf{c} - \hat{\mathbf{c}}\|^2$$

If $N = \#I$ is the total number of coefficients, then

$$D_{MSE} = \frac{1}{N} \|\mathbf{p} - \hat{\mathbf{p}}\|^2 = \frac{1}{N} \|\mathbf{c} - \hat{\mathbf{c}}\|^2$$

is the mean-squared error of the approximation.

Now assume that for a subset $J \subseteq I$ one has

$$\hat{c}_{i,j} = \begin{cases} c_{i,j} & \text{if } (i,j) \in J \\ 0 & \text{if } (i,j) \notin J \end{cases}$$

Then

$$D_{MSE} = \frac{1}{N} \sum_{i,j \notin J} |c_{i,j}|^2$$

Thus, when considering all subsets J of I of the same cardinality N_0 , the one(s) which yield(s) the minimum D_{MSE} is obtained by sorting the coefficients $c_{i,j}$ ($(i,j) \in I$) according to their (decreasing) absolute value and taking for J the initial N_0 positions of the sorted sequence.

In practice, what one does is:

1. order the elements $c_{i,j}$ according to the bitlength $\lfloor \log_2 |c_{i,j}| \rfloor + 1$ of their absolute values
2. transmit positions and signs of the elements of maximal bitlength r
3. for $k = r - 1, r - 2, \dots$
 - (a) transmit positions and signs of the elements of bitlength k
 - (b) transmit the k -th bit of all elements of bitlength $> k$
(i.e., transmitted in earlier rounds)

13.3.2 Progressive transmission, the procedures

procedure *scan*(L,C,N,LIP,LSP)

(* (L,C): a list $(C(\xi))_{\xi \in L}$ of integers

N: significance level

LIP, LSP: lists that are updated during the execution*)

for ξ in LIP **do**

if $|C(\xi)| \geq N$ **then**

output($\xi, \text{sgn}(C(\xi))$)

remove(LIP, ξ)

append(LSP, ξ)

end if

end for

return (LIP,LSP)

procedure *progtrans*(L,C)

(* (L,C): a list $(C(\xi))_{\xi \in L}$ of integers *)

(*initialization*)

$r \leftarrow \lfloor \log_2 \max_{\xi \in L} |C(\xi)| \rfloor$

output(r)

$N \leftarrow 2^r$

LIP \leftarrow L

LSP \leftarrow empty list

(*sorting*)

(LIP,LSP) \leftarrow *scan*(L,C,N,LIP,LSP)

oldLSP \leftarrow LSP

repeat

$r \leftarrow r - 1, N \leftarrow N/2$

(*sorting*)

(LIP,LSP) \leftarrow *scan*(L,C,N,LIP,LSP)

(*refinement*)

for $\xi \in$ oldLSP **do**

output(r -th bit of $|C(\xi)|$)

end for

oldLSP \leftarrow LSP

until “as needed”

13.3.3 An example for progressive transmission

Output of *progtrans*(3, -12, 4, 9, -5, 0, -1, -8, 8, 0)

round	position	sign	bit values
1	{2, 4, 8, 9}	{1, 0, 1, 0}	{}
2	{3, 5}	{0, 1}	{1, 0, 0, 0}
3	{1}	{0}	{0, 0, 0, 0, 0, 0}
4	{7}	{1}	{0, 1, 0, 0, 0, 1, 1}

Array representation of progressive transmission

pos	sgn	bit ₁	bit ₂	bit ₃	bit ₄
2	1	1	1	0	0
4	0	1	0	0	1
8	1	1	0	0	0
9	0	1	0	0	0
3	0	0	1	0	0
5	1	0	1	0	1
1	0	0	0	1	1
7	1	0	0	0	1
6	0	0	0	0	0
10	0	0	0	0	0

Successive approximations

	1	2	3	4	5	6	7	8	9	10	error
1 bit	0	-11	0	11	0	0	0	-11	11	0	8.60233
2 bit	0	-13	5	9	-5	0	0	-9	9	0	3.74166
3 bit	2	-12	4	8	-4	0	0	-8	8	0	2.00000
4 bit	3	-12	4	9	-5	0	-1	-8	8	0	0

13.4 Zig-zag ordering

Progressive transmission is a sequential procedure, so when applying it to image data, one has to serialize the 2-dimensional data somehow. There are many ways to do that. A particular convenient method, that is adapted to work well with DWT, is the so-called *zig-zag-ordering* of the elements of $\mathbb{N} \times \mathbb{N}$. It can be described inductively as follows, using the abbreviation

$$\square_n = \{0, \dots, 2^n - 1\} \times \{0, \dots, 2^n - 1\}$$

for denoting initial 2D-segments of $\mathbb{N} \times \mathbb{N}$. Furthermore, for $0 \leq i, j \leq 1$ let

$$\square_n^{i,j} = \{2^{i \cdot n} + 0, \dots, 2^{i \cdot n} + 2^n - 1\} \times \{2^{j \cdot n} + 0, \dots, 2^{j \cdot n} + 2^n - 1\},$$

so that schematically

$$\square_{n+1} = \begin{bmatrix} \square_n^{0,0} & \square_n^{0,1} \\ \square_n^{1,0} & \square_n^{1,1} \end{bmatrix}$$

The zig-zag-ordering will be defined on $\mathbb{N} \times \mathbb{N}$ by iteratively defining it on \square_n ($n = 1, 2, 3, \dots$), where \square_n is identified with the upper-left part $\square_{n+1}^{0,0}$ of \square_{n+1} . This way one has an order-compatible inclusion chain

$$\square_1 \subset \square_2 \subset \square_3 \dots \nearrow \mathbb{N} \times \mathbb{N}$$

and it suffices to inductively define the zig-zag ordering on each \square_n .

1. Order the elements of $\square_1 = \{0, 1\} \times \{0, 1\}$ in *Z*-form

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 0 \rightarrow & 1 \\ & & \swarrow \\ 1 & 2 \rightarrow & 3 \end{array}$$

2. If the ordering has been defined for \square_n for some $n \geq 1$, Then the ordering on \square_{n+1} is given by

- (a) The ordering on each of $\square_{n+1}^{i,j}$ ($0 \leq i, j \leq 1$) is “the same” as the ordering of \square_n
- (b) The ordering among the elements of different $\square_{n+1}^{i,j}$ is again given by zig-zag, as symbolized by

$$\square_{n+1}^{0,0} < \square_{n+1}^{0,1} < \square_{n+1}^{1,0} < \square_{n+1}^{1,1}$$

So the zig-zag ordering of \square_2 is

$$\begin{array}{cccc} 0 & 1 & 4 & 5 \\ 2 & 3 & 6 & 7 \\ 8 & 9 & 12 & 13 \\ 10 & 11 & 14 & 15 \end{array}$$

The function $m2l : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, where $m2l(i, j)$ is the ordinal number of (i, j) in the zig-zag ordering, and its inverse $l2m = \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$, where $l2m(k)$ gives the position of the element with ordinal number k in $\mathbb{N} \times \mathbb{N}$, can easily be described in terms of the binary expansions of the numbers involved.

13.5 Embedded trees in DWT-data

13.5.1 The 1D-case

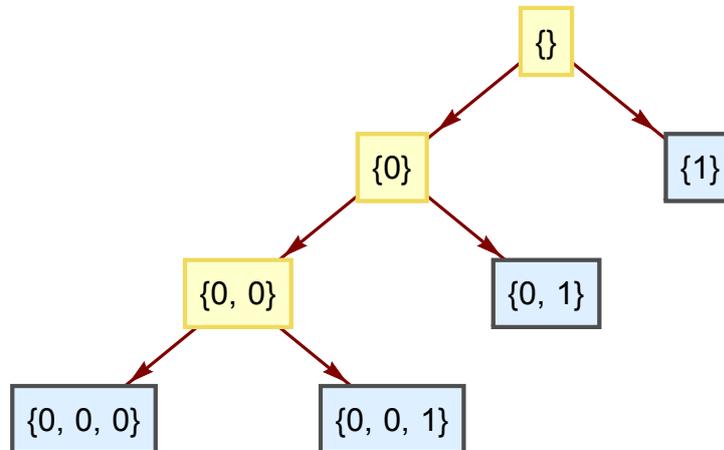


Abbildung 13.2: The decomposition scheme of a 3-level 1D-DWT (analysis)

The various parts of the wavelet decomposition tree are henceforth named *sub-bands* and denoted as indicated as sub-bands 0^k or $0^k 1$.

The embedded trees in the case of a data vector of length 32 for a 3-level transform.

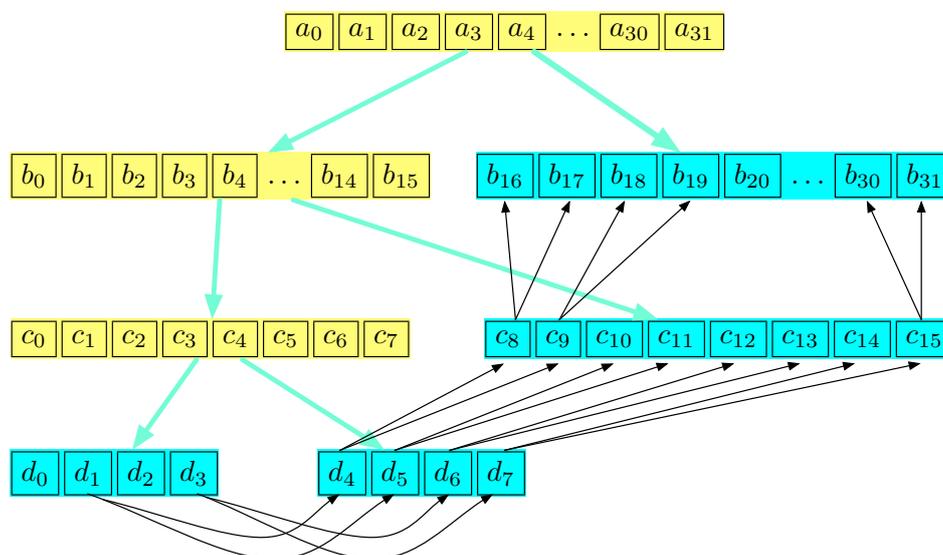


Abbildung 13.3: Embedded binary trees in context

Positions joined by black arrows refer to same position in the original data, hence the wavelet coefficients should be related. E.g., if d_5 is small, then most probably its descendants c_{10} , c_{11} and also b_{20}, \dots, b_{23} will be small too. In the case of smooth signal data there should be strongly decreasing tendency as the resolution increases. In particular: if a wavelet coefficient is 0, the most probably the coefficients of all its descendants will be 0 too. This phenomenon can be used profitably in tasks like data compression and progressive transmission.

The embedded trees graph as drawn by Mathematica. Positions joined by black arrows refer to same position in the original data

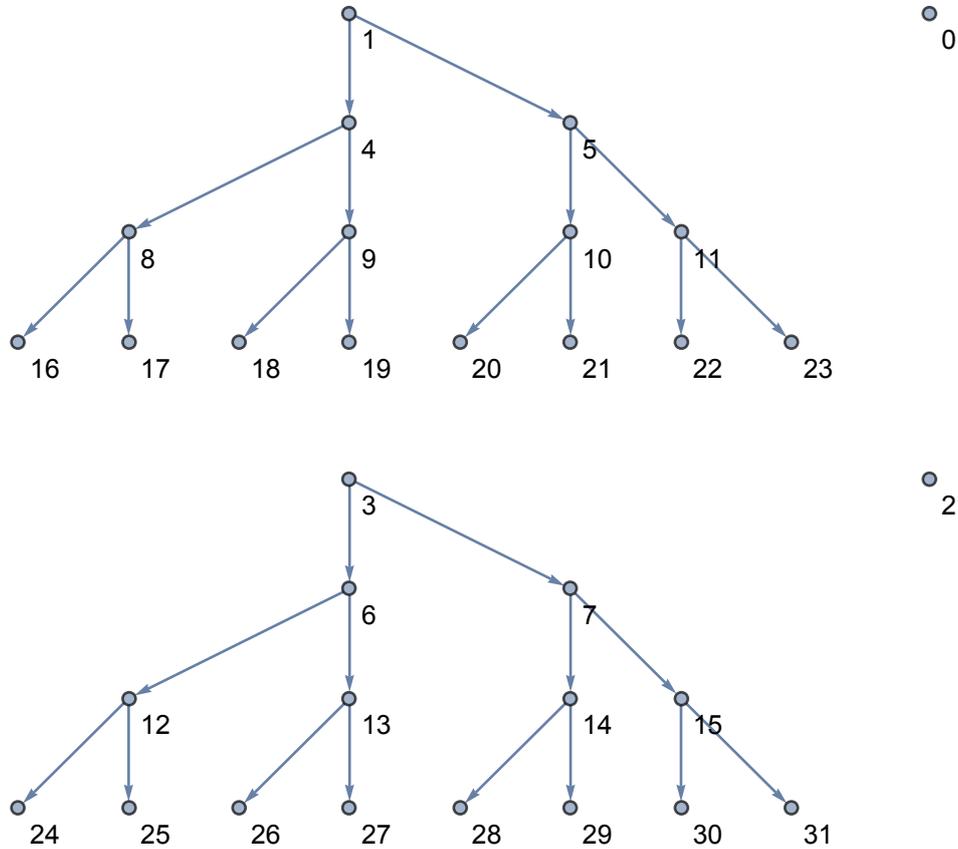


Abbildung 13.4: Embedded binary trees

This forest can be described linearly by giving a list of predecessor labels (i.e., reversing the arrows, with the convention that each root label is its own predecessor):

0, 0, 0, 0, 1, 1, 3, 3, 4, 4, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 10, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15

This list is just the bottom row in

0	1	2	3	4	5	6	...	31	32
↓	↓	↓	↓	↓	↓	↓	...	↓	↓
0	0	0	0	1	1	3	...	15	15

The general rule for constructing these embedded trees for a data vector of length N that represents the data from an ℓ -level DWT is:

- Index the data from 0 to $N - 1$
- Set $a = N/2^\ell$: this is the length of the 0^ℓ and of the $0^{\ell-1}1$ components (sub-bands) – this number must be an even integer
- All even indexed positions in the 0^ℓ sub-band will be isolated vertices
- All odd indexed positions $2k + 1$ in the 0^ℓ sub-band have successors $a + 2k$ and $a + 2k + 1$ in the $0^{\ell-1}1$ sub-band ($0 \leq k < \lfloor a/2 \rfloor$)
- Each position k for the transformed data which does not belong to the 0^ℓ sub-band or to the 1-sub-band has $2k$ and $2k + 1$ as its successors.

This procedure gives $a/2$ complete binary trees of depth ℓ .

13.5.2 The 2D-case (*spatial orientation trees*)

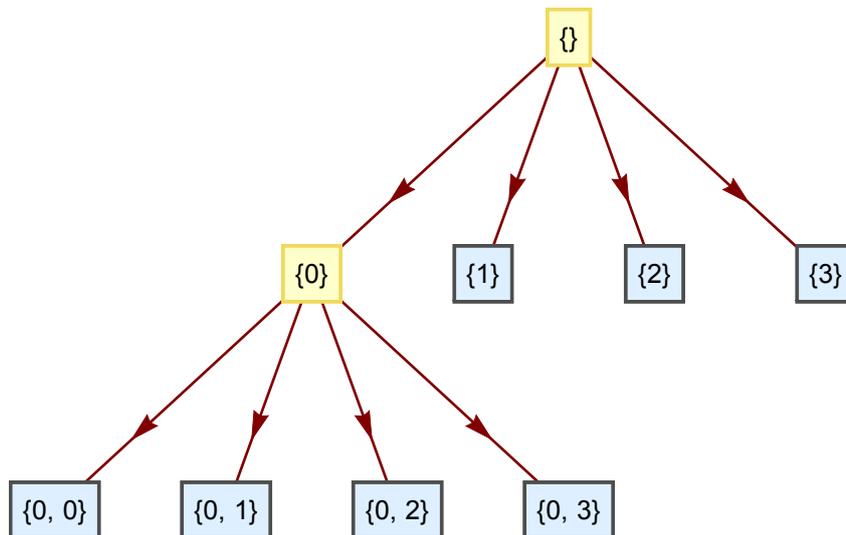


Abbildung 13.5: Decomposition scheme for a 2-level 2D-DWT (analysis)

Construction of the embedded quaternary trees

- The image is assumed to be square of size $2^n \times 2^n$.
Numbering of rows and columns starts at 0.
Let ℓ denote the number of levels of a 2D-DWT, then the four sub-bands on the lowest level 0^ℓ and $0^{\ell-1}1, 0^{\ell-1}2, 0^{\ell-1}3$ have size $a \times a$ with $a = 2^{n-\ell}$. a is assumed to be even, i.e., $\ell < n$.
 - Partition the 0^ℓ -sub-band into 2×2 squares. The upper left position of each square becomes an isolated vertex. These are the positions (i, j) with $0 \leq i, j < a$ and with both i and j even.
- (A) All other positions in the 0^ℓ sub-band become roots of quaternary trees of height ℓ . These roots have their four successors in the sub-bands $0^{\ell-1}1, 0^{\ell-1}2,$

$0^{\ell-1}3$. Precisely:

$$(i, j) \longrightarrow \begin{cases} (i, a + j - 1) + \varepsilon & \text{if } i \text{ even and } j \text{ odd} \\ (i + a - 1, j) + \varepsilon & \text{if } i \text{ odd and } j \text{ even} \\ (i + a - 1, j + a - 1) + \varepsilon & \text{if } i \text{ and } j \text{ odd} \end{cases}$$

where $\varepsilon \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$.

- (B) From then on, each vertex in a sub-band (other than the sub-bands 1, 2, 3, where positions are “leaf” positions, i.e., positions without successors) has four successors in one of the sub-bands of the next higher frequency “in the same relative position”, as indicated in the graphical sketch. Precisely

$$(i, j) \longrightarrow (2i, 2j) + \varepsilon$$

An example with $n = 4, \ell = 2, a = 4$ is sketched in the figure.

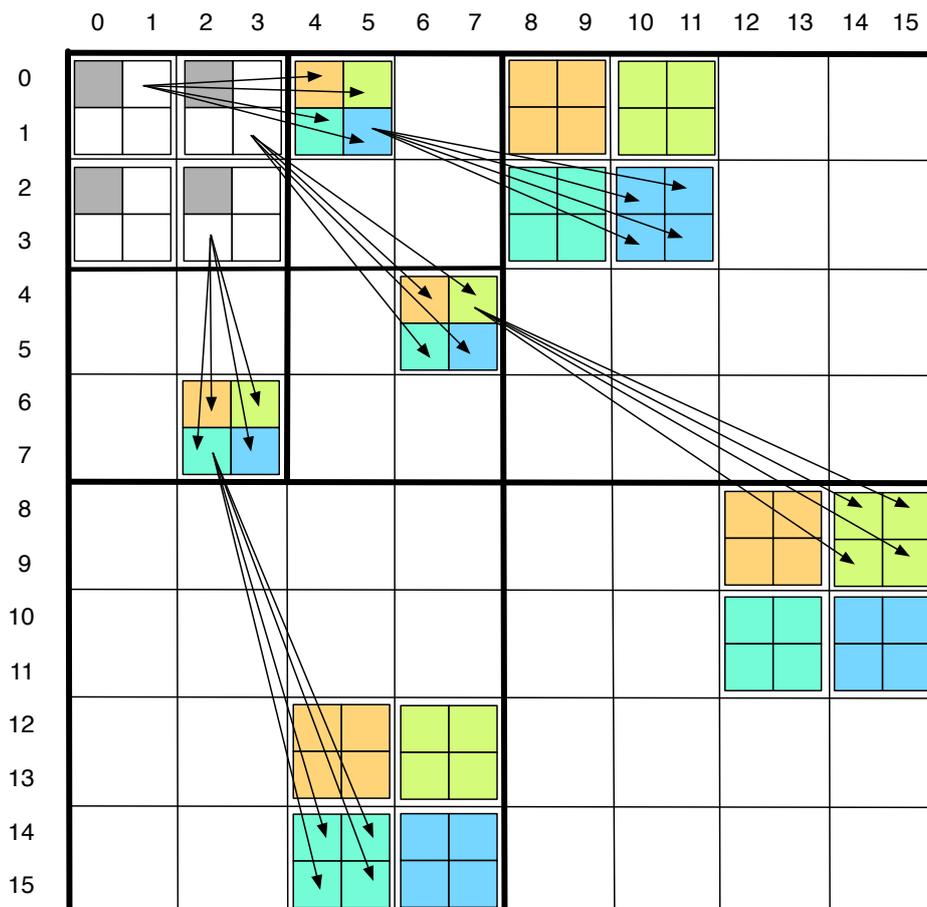


Abbildung 13.6: Embedded spatial orientation trees

The exemplary arrows drawn in this figure are

$$\begin{aligned}
 (0, 1) &\longrightarrow (0, 4) + \varepsilon = \{(0, 4), (0, 5), (1, 4), (1, 5)\} && \text{by (A)} \\
 (1, 3) &\longrightarrow (4, 6) + \varepsilon = \{(4, 6), (4, 7), (5, 5), (5, 7)\} && \text{by (A)} \\
 (1, 5) &\longrightarrow (2, 10) + \varepsilon = \{(2, 10), (2, 11), (3, 10), (3, 11)\} && \text{by (B)} \\
 (3, 2) &\longrightarrow (6, 2) + \varepsilon = \{(6, 2), (6, 3), (7, 2), (7, 3)\} && \text{by (A)} \\
 (4, 7) &\longrightarrow (8, 14) + \varepsilon = \{(8, 14), (8, 15), (9, 14), (9, 15)\} && \text{by (B)} \\
 (7, 2) &\longrightarrow (14, 4) + \varepsilon = \{(14, 4), (14, 5), (15, 4), (15, 5)\} && \text{by (B)}
 \end{aligned}$$

Besides the 4 isolated vertices $(0, 0), (0, 2), (2, 0), (2, 2)$ (shaded in the figure) there are 12 quaternary trees of height 2, rooted in the remaining 12 vertices of the 0^2 -sub-band.

A convenient way to linearly encode the structure of these embedded trees is by referring to the zigzag-ordering.

As an example: in an (8×8) -array the positions are numbered by the zigzag-ordering as shown here:

$$\begin{pmatrix}
 0 & 1 & 4 & 5 & 16 & 17 & 20 & 21 \\
 2 & 3 & 6 & 7 & 18 & 19 & 22 & 23 \\
 8 & 9 & 12 & 13 & 24 & 25 & 28 & 29 \\
 10 & 11 & 14 & 15 & 26 & 27 & 30 & 31 \\
 32 & 33 & 36 & 37 & 48 & 49 & 52 & 53 \\
 34 & 35 & 38 & 39 & 50 & 51 & 54 & 55 \\
 40 & 41 & 44 & 45 & 56 & 57 & 60 & 61 \\
 42 & 43 & 46 & 47 & 58 & 59 & 62 & 63
 \end{pmatrix}$$

The embedded trees relevant for a 2-level DWT can be displayed in a matrix that for each position gives the position of its tree-predecessor vertex

$$\begin{pmatrix}
 0 & 1 & 1 & 1 & 4 & 4 & 5 & 5 \\
 2 & 3 & 1 & 1 & 4 & 4 & 5 & 5 \\
 2 & 2 & 3 & 3 & 6 & 6 & 7 & 7 \\
 2 & 2 & 3 & 3 & 6 & 6 & 7 & 7 \\
 8 & 8 & 9 & 9 & 12 & 12 & 13 & 13 \\
 8 & 8 & 9 & 9 & 12 & 12 & 13 & 13 \\
 10 & 10 & 11 & 11 & 14 & 14 & 15 & 15 \\
 10 & 10 & 11 & 11 & 14 & 14 & 15 & 15
 \end{pmatrix}$$

and the zigzag-ordering would give a 1-dimensional representation of this information

$$0, 1, 2, 3, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3, \dots, 15, 15, 15, 15$$

A nice and useful feature of the zigzag-ordering is apparent: positions that are offsprings of the same position w.r.t. the embedded quaternary trees are neighbors in the matrix and the linear version of the predecessor information.

13.6 The SPIHT method

13.6.1 General remarks

- The procedures *scantree* and *spiht* operate on forests T of rooted trees. T is specified by a list of root vertices and a function which to every vertex lists its immediate successors. The vertices carry an integer valuation given by a function C . C -values are thought of as binary strings of some common length r together with an extra sign bit.
- The purpose of *spiht* is to compute a signed k -bit approximation of C for some value $k \leq r$ that may or may not be fixed in advance. *scantree* is used to compute a $(k + 1)$ -bit approximation from a k -bit approximation (for $k = 1, 2, \dots$) using information accumulated during the computation of the latter. It uses the idea of *progressive transmission* together with a systematic exploration of the tree structures of T in view of the mapping C .
- Running *spiht* produces a bit stream as output. This stream contains significance and sign information about (T, C) which encode the branchings of *spiht* and its procedure *scantree* when exploring T . This stream contains all the information for exactly simulating the action of *spiht* by just knowing the tree (or forest) structure T and reading the stream, but not knowing C . If one considers *spiht* as an *encoder*, then replacing all *output-commands* in *spiht* and *scantree* by *input-commands* (thus reading elements of the bit stream precisely where in *spiht* they were produced) gives the view of a *decoder* who wants to compute C to a certain degree of accuracy with as little information as possible. This is where the aspect of *data compression* comes into play.
- The *spiht* algorithm is given below in an abstract form, i.e., the structure of T occurs only implicitly by referring to a successor map which is assumed to be known, but not further specified. The main application of *spiht* is with images, notably their wavelet transforms with the structure of *embedded trees* made explicit. In this situation T is a forest consisting of a certain number of isolated points and a set of quaternary trees (all of the same height). The typical properties of coefficient value arrays produced with DWT are exploited profitably when using these embedded trees in the *spiht* algorithm.

13.6.2 Data for and verbal description of the SPIHT-algorithm

- For each vertex ξ of T one defines
 - $\mathcal{O}(\xi)$: the (ordered) set of *offsprings* (immediate successors) of ξ
 - $\mathcal{D}(\xi)$: the set of *descendants* of ξ (i.e., offsprings and their descendants)
 - $\mathcal{L}(\xi) = \mathcal{D}(\xi) \setminus \mathcal{O}(\xi)$: the set of grandchildren of ξ and their descendants
- Given a significance level N , the *significance map* σ_N is defined as

$$\sigma_N : \xi \mapsto \begin{cases} 1 & \text{if } |C(\xi)| \geq N \\ 0 & \text{if } |C(\xi)| < N \end{cases}$$

- Using σ_N one defines *significance* (w.r.t the significance level N) for sets $\mathcal{D}(\xi)$ and $\mathcal{L}(\xi)$ by mappings δ_N and λ_N :

$$\delta_N : \xi \mapsto \begin{cases} 1 & \text{if } \sigma_N(\eta) = 1 \text{ for at least one } \eta \in \mathcal{D}(\xi) \\ 0 & \text{otherwise} \end{cases}$$

$$\lambda_N : \xi \mapsto \begin{cases} 1 & \text{if } \sigma_N(\eta) = 1 \text{ for at least one } \eta \in \mathcal{L}(\xi) \\ 0 & \text{otherwise} \end{cases}$$

- The functions σ_N , δ_N and λ_N can be efficiently computed: for σ_N this is clear, for δ_N and λ_N this can be done inductively in a bottom-up manner starting at the leaves of T :

$$\begin{aligned} \delta_N(\xi) &= \lambda_N(\xi) = 0 \text{ if } \xi \text{ is a leaf of } T, \text{ otherwise} \\ \lambda_N(\xi) &= \max_{\eta \in \mathcal{O}(\xi)} \delta_N(\eta) \\ \delta_N(\xi) &= \max\{ \lambda_N(\xi), \max_{\eta \in \mathcal{O}(\xi)} \sigma_N(\eta) \} \end{aligned}$$

- The lists LIS (*list of insignificant sets*), LIP (*list of insignificant points*), and LSP (*list of significant points*) are used for bookkeeping during the execution of *scantree* and *spiht*
 - $\xi \in \text{LSP}$ means that vertex ξ has been inspected by the algorithm and has been classified as significant (w.r.t. the current significance level, i.e., $\sigma_N(\xi) = 1$)
 - $\xi \in \text{LIP}$ means that vertex ξ has been inspected by the algorithm and has been classified as insignificant (w.r.t. the current significance level, i.e., $\sigma_N(\xi) = 0$)
 - entries in LIS are either of type (ξ, A) or of type (ξ, B) , where
 - * (ξ, A) represents the set $\mathcal{D}(\xi)$
 - * (ξ, B) represents the set $\mathcal{L}(\xi)$
 - The current entries of LIS, LIP and LSP during execution of *scantree* constitute a partition of the vertex set of T into disjoint subsets:
 - * entries $\xi \in \text{LIP}$ and $\xi \in \text{LSP}$ represent singleton sets $\{\xi\}$
 - * an entry $(\xi, A) \in \text{LIS}$ represents the set $\mathcal{D}(\xi)$
 - * an entry $(\xi, B) \in \text{LIS}$ represents the set $\mathcal{L}(\xi)$
 - This partition and the three lists are updated (the partition is refined) during the execution of *scantree*:

- * If the top element of LIS is (ξ, A) and if the set $\mathcal{D}(\xi)$ is significant, i.e., $\delta_N(\xi)=1$, then
 - the offsprings $\eta \in \mathcal{O}(\xi)$ are examined for significance
 - the set $\mathcal{D}(\xi)$ is split into

$$\bigsqcup_{\eta \in \mathcal{O}(\xi)} \{\eta\} \uplus \mathcal{L}(\xi)$$

- a new entry (ξ, B) added to LIS provided $\mathcal{L}(\xi) \neq 0$
- * If the top element of LIS is (ξ, B) and if the set $\mathcal{L}(\xi)$ is significant, i.e., $\lambda_N(\xi)=1$, then
 - the set $\mathcal{L}(\xi)$ is replaced by

$$\bigsqcup_{\eta \in \mathcal{O}(\xi)} \mathcal{D}(\eta),$$

- entries (η, A) are added to LIS for all offsprings $\eta \in \mathcal{O}(\xi)$
 - Elements (ξ, B) or (ξ, A) thus added to LIS have to be examined in the same execution round of scantree.
 - At the beginning of each execution of *scantree* the current elements of the list LIP are tested for significance w.r.t. to the actual significance level N , which is supposed to be lower than the level used in the previous execution of *scantree*. In case vertex is found N -significant, it will be moved from LIP to LSP.
 - During the execution of *scantree* information about the significance of vertices (and the corresponding sign in the case of vertices found significant) and \mathcal{D} - and \mathcal{L} -sets is encoded in the bit stream *output*.
 - Algorithm *spih*t iterates the sorting procedure *scantree* together with a refinement procedure in the spirit of *progressive transmission*.
 - *spith* is initialized by computing the maximal number r of bits needed to represent the absolute values of the $C(\xi)$, the significance level for the first run of *scantree* is set to $N = 2^r$, so that there are indeed entries $C(\xi)$ with $2^r \leq |C(\xi)| < 2^{r+1}$ and no entries with $2^{r+1} \leq |C(\xi)|$ in T . LIS and LIP are initialized using the root vertices of T , LSP is initially empty.
 - The first call of *scantree* examines T with $N = 2^r$ as significance level. T is explored only to the extent needed for discovering all N -significant vertices of (T, C) . After this first sorting phase there is no refinement phase necessary since LSP was empty before calling *scantree*.
 - The exploration of (T, C) is then done by iteratively by
 - * *sorting*: calling *scantree* with the significance level halved at every iteration
 - * *refinement*: enhancing the bit information of previous added elements of LSP in a *progressive transmission* way
- Each iteration uses the information accumulated in the lists LIS, LIP, and LSP, and updates these lists.
- Execution of *spith* can be stopped after every *sorting-refinement* round. The output stream produced up to this point is sufficient to reconstruct the respective k -bit approximation of C .

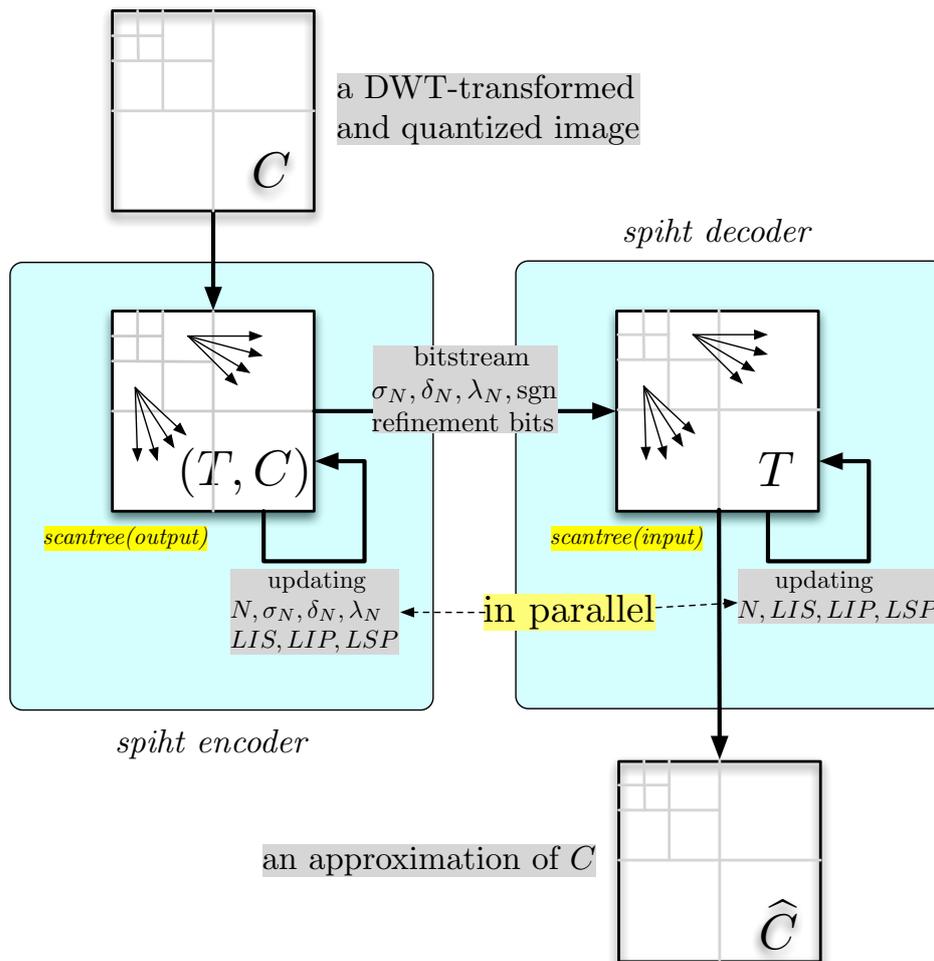


Abbildung 13.7: The SPIHT encoding-decoding scheme

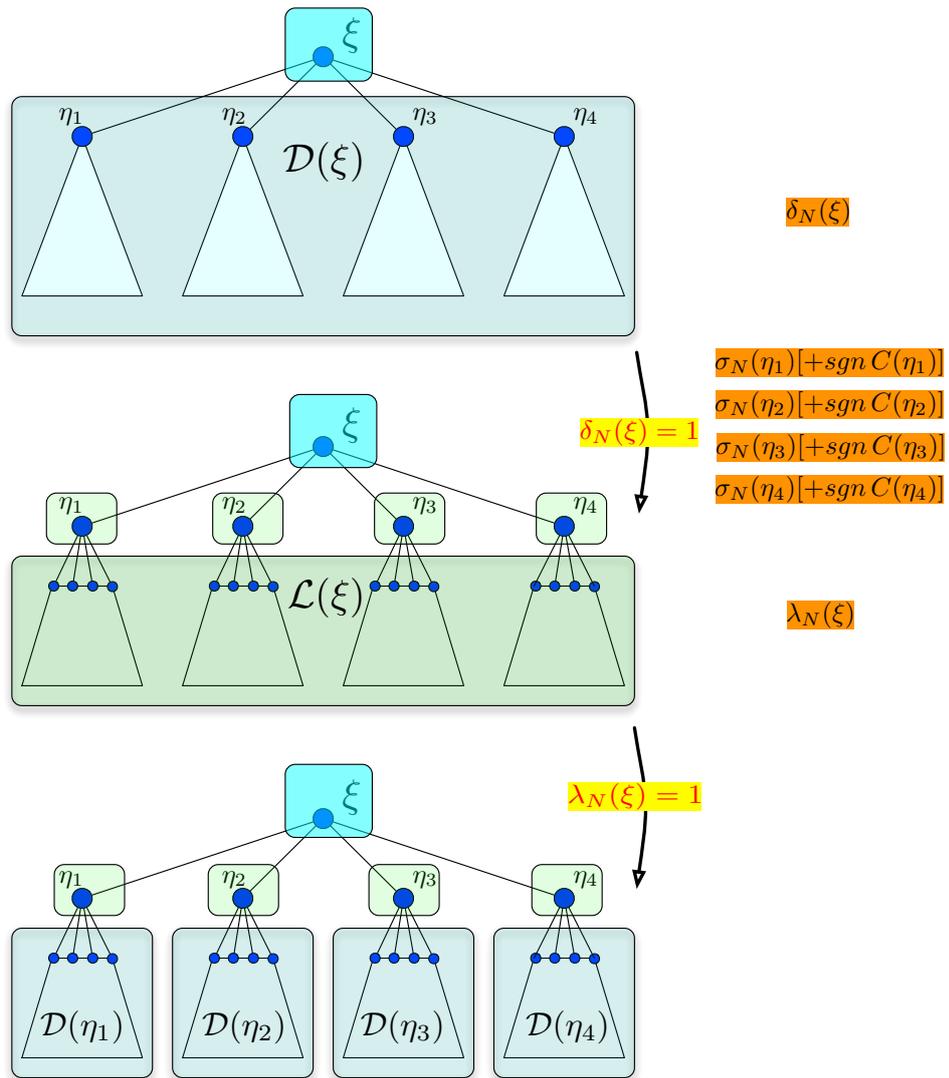


Abbildung 13.8: The set partitioning strategy of SPIHT

13.6.3 The procedures for SPIHT

procedure *scantree*(T,C,N,LIS,LIP,LSP)

(* T: a forest of rooted trees

C: an integer valuation of the vertices of T

N: significance level

LIS, LIP, LSP: lists that are updated during the execution*)

```

for  $\xi$  in LIP do
  output( $\sigma_N(\xi)$ )
  if  $\sigma_N(\xi) = 1$  then
    output( $\text{sgn}(C(\xi))$ )
    remove(LIP, $\xi$ )
    append(LSP, $\xi$ )
  end if
end for
for  $\xi$  in LIS do
  if type( $\xi$ ,A) then
    output( $\delta_N(\xi)$ )
    if  $\delta_N(\xi) = 1$  then
      for  $\eta$  in  $\mathcal{O}(\xi)$  do
        output( $\sigma_N(\eta)$ )
        if  $\sigma_N(\eta) = 1$  then
          output( $\text{sgn}(C(\eta))$ )
          append(LSP, $\eta$ )
        else
          append(LIP, $\eta$ )
        end if
      end for
      remove(LIS, $\xi$ -A)
      if  $\mathcal{L}(\xi) \neq \emptyset$  then
        append(LIS, $\xi$ -B)
      end if
    end if
  end if
  if type( $\xi$ ,B) then
    output( $\lambda_N(\xi)$ )
    if  $\lambda_N(\xi) = 1$  then
      remove(LIS, $\xi$ -B)
      for  $\eta \in \mathcal{O}(\xi)$  do
        append(LIS, $\eta$ -A)
      end for
    end if
  end if
end for
return (LIS,LIP,LSP)

```

```

procedure spiht(T,C)
(* T: a forest of rooted trees
C: an integer valuation of the vertices of T *)
(*initialization*)
 $r \leftarrow \lfloor \log_2 \max_{\xi \in T} |C(\xi)| \rfloor$ 
 $output(r)$ 
 $N \leftarrow 2^r$ 
  LIS  $\leftarrow$  list of all  $\xi$ -A, where  $\xi$  is the root of a  $T$ -subtree with  $\mathcal{D}(\xi) \neq \emptyset$ 
  LIP  $\leftarrow$  list of all  $\xi$ , which are roots of subtrees of  $T$ ,
    including isolated vertices
  LSP  $\leftarrow$  empty list
(*sorting*)
(LIS,LIP,LSP)  $\leftarrow scantree(T,C,N,LIS,LIP,LSP)$ 
oldLSP  $\leftarrow$  LSP
repeat
   $r \leftarrow r - 1, N \leftarrow N/2$ 
  (*sorting*)
  (LIS,LIP,LSP)  $\leftarrow scantree(T,C,N,LIS,LIP,LSP)$ 
  (*refinement*)
  for  $\xi \in oldLSP$  do
     $output(r\text{-th bit of } |C(\xi)|)$ 
  end for
  oldLSP  $\leftarrow$  LSP
until "as needed"

```

13.7 Examples

13.7.1 Example 1

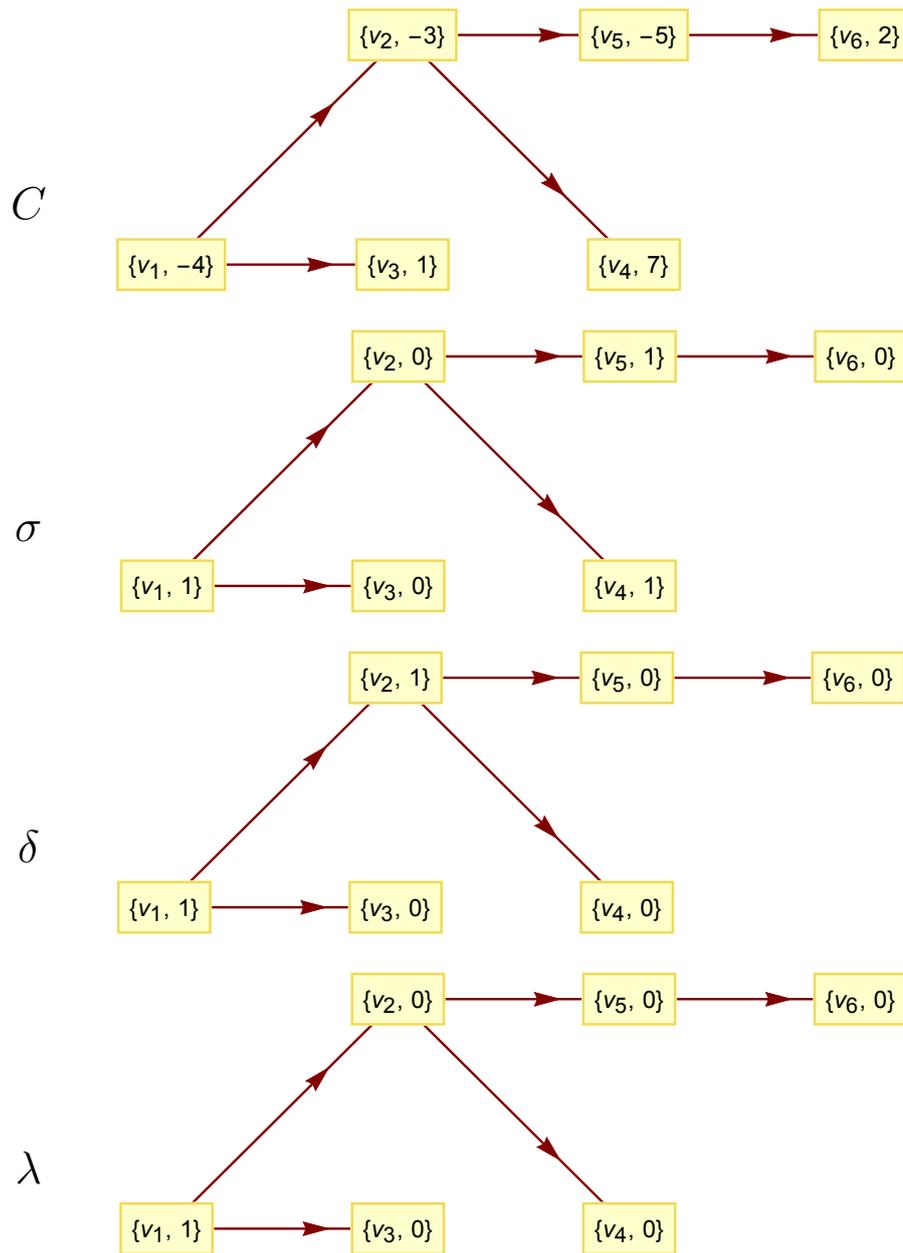
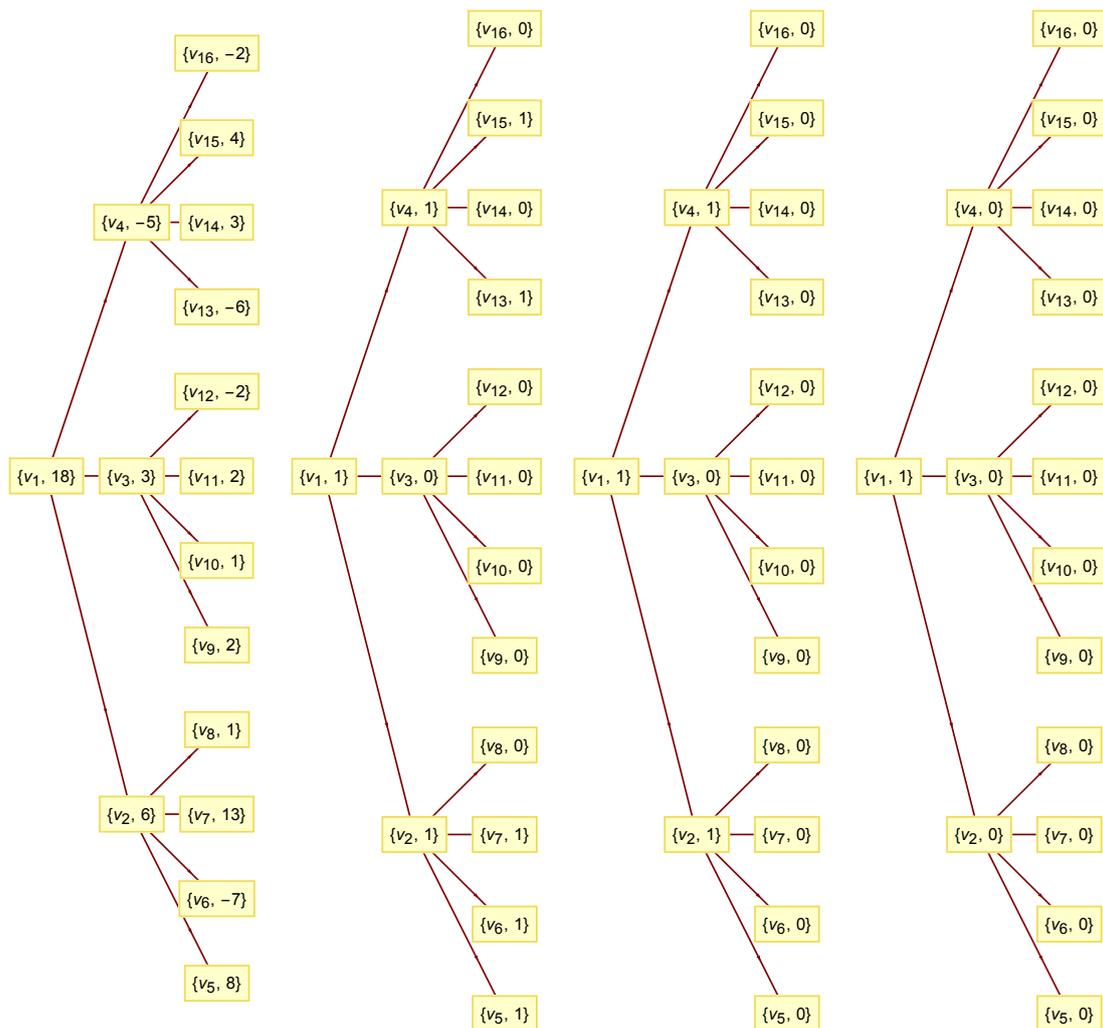


Abbildung 13.9: A tree T , its valuation C and the σ, δ, λ mappings of (T, C) for significance level $N = 4$

Execution of *scantree*[T, C, 4, {{1, A}}, {1}, {},]

out: $\sigma[1] = 1$
 out: *sign*[-4]
 LSP= {1} LIP= {}
 LIS: {{1, A}}
 out: $\delta[1] = 1$
 out: $\sigma[2] = 0$
 LSP= {1} LIP= {2}
 out: $\sigma[3] = 0$
 LSP= {1} LIP= {2, 3}
 LIS: {{1, B}}
 out: $\lambda[1] = 1$
 LIS: {{2, A}, {3, A}}
 out: $\delta[2] = 1$
 out: $\sigma[4] = 1$
 out: *sign*[7]
 LSP= {1, 4} LIP= {2, 3}
 out: $\sigma[5] = 1$
 out: *sign*[-5]
 LSP= {1, 4, 5} LIP= {2, 3}
 LIS: {{3, A}, {2, B}}
 out: $\delta[3] = 0$
 LIS: {{2, B}}
 out: $\lambda[2] = 0$
 new LIS: {{3, A}, {2, B}}
 new LIP: {2, 3}
 new LSP: {1, 4, 5}
 output: {1, *sign*[-4], 1, 0, 0, 1, 1, 1, *sign*[7], 1, *sign*[-5], 0, 0}

13.7.2 Example 2

Abbildung 13.10: $T, C, \sigma, \delta, \lambda$ for the second example

```
scantree[T2, C2, 16, {{1,A}}, {1}, {}, 1]
```

```
out: \[Sigma][1]= 1
out: sign[18]
LSP= {1} LIP= {}
LIS: {{1,A}}
out: \[Delta][1]= 0
```

```
new LIS: {{1,A}}
new LIP: {}
new LSP: {1}
output: {1,sign[18],0}
```

```
scantree[T2, C2, 8, {{1,A}}, {}, {1}, 1]
```

```
LIS: {{1,A}}
out: \[Delta][1]= 1
out: \[Sigma][2]= 0
LSP= {1} LIP= {2}
out: \[Sigma][3]= 0
LSP= {1} LIP= {2,3}
out: \[Sigma][4]= 0
LSP= {1} LIP= {2,3,4}
LIS: {{1,B}}
out: \[Lambda][1]= 1
LIS: {{2,A},{3,A},{4,A}}
out: \[Delta][2]= 1
out: \[Sigma][5]= 1
out: sign[8]
LSP= {1,5} LIP= {2,3,4}
out: \[Sigma][6]= 0
LSP= {1,5} LIP= {2,3,4,6}
out: \[Sigma][7]= 1
out: sign[13]
LSP= {1,5,7} LIP= {2,3,4,6}
out: \[Sigma][8]= 0
LSP= {1,5,7} LIP= {2,3,4,6,8}
LIS: {{3,A},{4,A}}
out: \[Delta][3]= 0
LIS: {{4,A}}
out: \[Delta][4]= 0
```

```
new LIS: {{3,A},{4,A}}
new LIP: {2,3,4,6,8}
new LSP: {1,5,7}
output: {1,0,0,0,1,1,1,sign[8],0,1,sign[13],0,0,0}
```

```
scantree[T2, C2, 4, {{3,A},{4,A}}, {2,3,4,6,8}, {1,5,7}, 1]
```

```

out: \[Sigma][2]= 1
out: sign[6]
LSP= {1,5,7,2} LIP= {}
out: \[Sigma][3]= 0
LSP= {1,5,7,2} LIP= {3}
out: \[Sigma][4]= 1
out: sign[-5]
LSP= {1,5,7,2,4} LIP= {3}
out: \[Sigma][6]= 1
out: sign[-7]
LSP= {1,5,7,2,4,6} LIP= {3}
out: \[Sigma][8]= 0
LSP= {1,5,7,2,4,6} LIP= {3,8}
LIS: {{3,A},{4,A}}
out: \[Delta][3]= 0
LIS: {{4,A}}
out: \[Delta][4]= 1
out: \[Sigma][13]= 1
out: sign[-6]
LSP= {1,5,7,2,4,6,13} LIP= {3,8}
out: \[Sigma][14]= 0
LSP= {1,5,7,2,4,6,13} LIP= {3,8,14}
out: \[Sigma][15]= 1
out: sign[4]
LSP= {1,5,7,2,4,6,13,15} LIP= {3,8,14}
out: \[Sigma][16]= 0
LSP= {1,5,7,2,4,6,13,15} LIP= {3,8,14,16}

new LIS: {{3,A}}
new LIP: {3,8,14,16}
new LSP: {1,5,7,2,4,6,13,15}
output: {1,sign[6],0,1,sign[-5],1,sign[-7],0,0,1,1,sign[-6],0,1,sign[4],0}

```

```
scantree[T2, C2, 2, {{3,A}}, {3,8,14,16},{1,5,7,2,4,6,13,15},1]
```

```

out: \[Sigma][3]= 1
out: sign[3]
LSP= {1,5,7,2,4,6,13,15,3} LIP= {}
out: \[Sigma][8]= 0
LSP= {1,5,7,2,4,6,13,15,3} LIP= {8}
out: \[Sigma][14]= 1
out: sign[3]
LSP= {1,5,7,2,4,6,13,15,3,14} LIP= {8}
out: \[Sigma][16]= 1
out: sign[-2]
LSP= {1,5,7,2,4,6,13,15,3,14,16} LIP= {8}
LIS: {{3,A}}
out: \[Delta][3]= 1
out: \[Sigma][9]= 1
out: sign[2]
LSP= {1,5,7,2,4,6,13,15,3,14,16,9} LIP= {8}
out: \[Sigma][10]= 0
LSP= {1,5,7,2,4,6,13,15,3,14,16,9} LIP= {8,10}
out: \[Sigma][11]= 1
out: sign[2]
LSP= {1,5,7,2,4,6,13,15,3,14,16,9,11} LIP= {8,10}
out: \[Sigma][12]= 1
out: sign[-2]
LSP= {1,5,7,2,4,6,13,15,3,14,16,9,11,12} LIP= {8,10}

new LIS: {}
new LIP: {8,10}
new LSP: {1,5,7,2,4,6,13,15,3,14,16,9,11,12}
output: {1,sign[3],0,1,sign[3],1,sign[-2],1,1,sign[2],0,1,sign[2],1,sign[-2]}

```

```
scantree[T2, C2, 1, {}, {8,10}, {1,5,7,2,4,6,13,15,3,14,16,9,11,12}, 1]
```

```

out: \[Sigma][8]= 1
out: sign[1]
LSP= {1,5,7,2,4,6,13,15,3,14,16,9,11,12,8} LIP= {}
out: \[Sigma][10]= 1
out: sign[1]
LSP= {1,5,7,2,4,6,13,15,3,14,16,9,11,12,8,10} LIP= {}
new LIS: {}
new LIP: {}
new LSP: {1,5,7,2,4,6,13,15,3,14,16,9,11,12,8,10}
output: {1,sign[1],1,sign[1]}

```

13.7.3 Example 3

Execution of the SPIHT-algorithm on an example in
Rao, Bopardikar: Wavelet Transforms, Addison-Wesley, 1998.

$$\begin{bmatrix} 62 & 34 & 18 & 17 & -4 & 1 & -2 & 6 \\ -31 & 24 & -15 & 14 & -11 & 0 & 4 & -1 \\ 42 & 29 & -35 & 10 & 29 & 10 & 6 & 9 \\ -12 & 15 & -9 & 15 & -1 & 9 & 5 & 13 \\ 4 & 45 & 13 & -1 & 26 & -21 & 3 & 1 \\ 3 & 0 & -2 & 21 & -1 & 0 & 7 & 9 \\ 0 & 13 & 4 & 5 & 4 & 5 & 6 & 0 \\ -1 & 7 & -11 & 3 & 0 & 8 & 2 & 7 \end{bmatrix}$$

- 2-level DWT-decomposition
- bound for coefficient size: $\lceil \log_2(62) \rceil = 5 \Rightarrow N = 2^5 = 32$

LIS	LIP	LSP
(0,1)-A	(0,0)	
(1,0)-A	(0,1)	
(1,1)-A	(1,0)	
	(1,1)	

Abbildung 13.11: Initializing the lists

LIS	LIP	LSP	output
(0,1)-A		(0,0) →	1,0
(1,0)-A		(0,1) →	1,0
(1,1)-A	(1,0) →		0
	(1,1) →		0

Abbildung 13.12: Lists after the first N -significance test for LIP

LIS	LIP	LSP	output
(0,1)-A		(0,0)	
(1,0)-A		(0,1)	
(1,1)-A	(1,0)		
	(1,1)		
(1,0)-B			0
(1,1)-B			1
		(2,0)	1,0
	(2,1)		0
	(3,0)		0
	(3,1)		0
			1
		(2,2)	1,1
	(2,3)		0
	(3,2)		0
	(3,3)		0

Abbildung 13.13: Lists after the first scanning of LIS-entries of type A

LIS	LIP	LSP	output
(0,1)-A		(0,0)	
		(0,1)	
	(1,0)		
	(1,1)		
(1,0)-B		(2,0)	
(1,1)-B	(2,1)		
(2,0)-A	(3,0)		
(2,1)-A	(3,1)		
(3,0)-A		(2,2)	
(3,1)-A	(2,3)		
	(3,2)		
	(3,3)		
			1
			0

Abbildung 13.14: Lists after the first scanning of LIS-entries of type B

LIS	LIP	LSP	output
(0,1)-A		(0,0)	
(1,1)-B		(0,1)	
	(1,0)		
(2,0)-A	(1,1)		
(2,1)-A		(2,0)	
(3,0)-A	(2,1)		
(3,1)-A	(3,0)		
	(3,1)		
		(2,2)	
	(2,3)		
	(3,2)		
	(3,3)		
			1
	(4,0)		0
		(4,1)	1,0
	(5,0)		0
	(5,1)		0
			0
			0
			0

Abbildung 13.15: Lists after finishing the first *scantree*-execution

- The first execution of *scantree* is finished.
- After this round there is no *refinement* done!
- The significance threshold for the next round is set to $N = 16$.

LIS	LIP	LSP	output
(0,1)-A	(1,0)	(0,0)	
(1,1)-B	(1,1)	(0,1)	
(2,1)-A	(2,1)	(2,0)	
(3,0)-A	(3,0)	(2,2)	
(3,1)-A	(3,1)	(4,1)	
	(2,3)	(1,0)	1,1
	(3,2)	(1,1)	1,0
	(3,3)	(2,1)	1,0
	(4,0)		0
	(5,0)		0
	(5,1)		0
			0
			0
			0
			0
			0
			0

Abbildung 13.16: Lists after the significance test for LIP in the second round

LIS	LIP	LSP	output
	(3,0)	(0,0)	
	(3,1)	(0,1)	
	(2,3)	(2,0)	
	(3,2)	(2,2)	
	(3,3)	(4,1)	
	(4,0)	(1,0)	
	(5,0)	(1,1)	
	(5,1)	(2,1)	
(0,1)-A			1
(1,1)-B		(0,2)	1,0
(2,1)-A		(0,3)	1,0
(3,0)-A	(1,2)		0
(3,1)-A	(1,3)		0
(0,1)-B			1
			1
	(4,2)		0
	(4,3)		0
	(5,2)		0
		(5,3)	1,0
			0
			0
			1
(2,2)-A			1
(2,3)-A		(4,4)	1,0
(3,2)-A		(4,5)	1,1
(3,3)-A	(5,4)		0
(0,2)-A	(5,5)		0
(0,3)-A			0
(1,2)-A			0
(1,3)-A			0
			0
			0
			0
		(2,4)	1,0
	(2,5)		0
	(3,4)		0
	(3,5)		0
			0

Abbildung 13.17: Lists after the second round of *scantree*

This is followed by *refinement*: output of the 4-th bits of the LSP-entries from the first round: 1, 0, 1, 0, 1.

Procedures for the SPIHT algorithm

Usage files

Old versions

Basic

`tosignedbinary`, `bitseq2num`, `convertlist`, `approx`, `NNorm`

■ `tosignedbinary[x,n]`

? `tosignedbinary`

Converts an integer *int* with $-2^n < int < 2^n$ into signed binary representation of length $n+1$ (1 sign bit + n binary digits).
Uses: --
Original file: decode.nb

In[2]:=

```
tosignedbinary[int_, n_] :=  
Module[{s},  
  If[Abs[int] >= 2^n, Throw["input too big"]];  
  s = If[int >= 0, 0, 1];  
  Flatten[{s, IntegerDigits[int, 2, n]}]  
]
```

■ `bitseq2num[bsq,k]`

? `bitseq2num`

Computes a nonnegative integer from the first k entries of the binary sequence *bseq*. The length of *bseq* is taken into account.
Uses: --
Original file: decode.nb

```
bseq2num[bsq_, k_] :=  
Module[{n, p},  
  n = Length[bsq];  
  If[k > n, Throw["k too big"]];  
  If[bsq == Table[0, {n}], Return[0]];  
  {p} = FirstPosition[bsq, 1];  
  If[k < p, Return[0]];  
  If[k == p && p < n,  
    Return[  
      (2^(n - p) - (-1)^(bsq[[p]])*2^(n - p - 1) - 1)];  
  If[k == p && p == n, Return[bsq[[p]]];  
  If[k > p && k < n,  
    Return[  
      bitseq2num[bsq, k - 1] - (-1)^(bsq[[k]])*2^(n - k - 1)];  
  If[k > p && k == n,  
    Return[  
      bitseq2num[bsq, n - 1] + bsq[[n]]]  
    ]
```

■ `convertlist[list]`

? convertlist

Converts a *list* of integers into their signed binary representations.
 The required binary length is computed and given as output before the list.
 Uses: **tosignedbinary**
 Original file: decode.nb

```
In[4]:= convertlist[list_] :=
Module[{lmax},
  lmax = Max[Abs[list]];
  n = IntegerLength[lmax, 2];
  {n, Map[tosignedbinary[#, n] &, list]}
]
```

■ `approx[array,order]`

? approx

array: a 2D-array of integers
order: approximation order intended
 Computes k-bit approximation of the array entries
 Uses: **tosignedbinary**, **bitseq2num**
 Original file: lena.nb

```
In[5]:= approx[array_, order_] :=
Module[{maxorder, bps, dim, sgn, max, bpss, sgns},
  dim = Dimensions[array];
  max = Max[Abs[array]];
  maxorder = Floor[Log[2, max]] + 1;
  Print["number of bits needed: ", maxorder];
  If[order > maxorder, Print["order bigger than needed"]];
  bps = Map[tosignedbinary[#, maxorder] &, array, {2}];
  bpss = Map[Rest[#] &, bps, {2}];
  sgns = Map[(-1)^First[#] &, bps, {2}];
  sgns*Map[bitseq2num[#, Min[order, maxorder]] &, bpss, {2}]
]
```

■ `NNorm[matrix]`

? NNorm

Computes the 2-norm of a *matrix*
 Uses: --
 Original file: lena.nb

```
In[6]:= NNorm[matrix_] := N[Norm[Flatten[matrix], 2]]
```

Progressive transmission

reorder, transmit, protrans, recons

■ `reorder[list]`

? reorder

list: a list of integers
 Applies **convertlist** to *list* and then orders the entries according to their binary integer length. The original position of an entry in the input list is prepended to the signed binary representation.
 Uses: **convertlist**
 Original file: decode.nb

```
In[7]:= reorder[list_] :=
Module[{cvl, in, out},
  cvl = convertlist[list];
  in = Table[
    Prepend[cvl[[2, i]], i], {i, 1, Length[cvl[[2]]]};
  out = {};
  For[i = 1, i <= cvl[[1]], i++,
    out = Join[out, Select[in, #[[i + 2]] == 1 &]];
    in = Select[in, #[[i + 2]] == 0 &]];
  Join[out, in]
]
```

■ `transmit[list]`

? transmit

Creates from *list* the same data as **reorder**, but arranges these in one table.
 Uses: **convertlist**
 Original file: progscan.nb

```
transmit[list_] :=
Module[{len, cvl, n, in, out, newout, sgnout, posout, bitout},
  len = Length[list];
  cvl = convertlist[list];
  n = cvl[[1]];
  in = Table[
    Append[cvl[[2, i]], i], {i, 1, len};
  out = {};
  For[i = 1, i <= cvl[[1]], i++,
    newout = Select[in, #[[i + 1]] == 1 &];
    in = Select[in, #[[i + 1]] == 0 &];
    sgnout = newout[[All, 1]];
    posout = newout[[All, n + 2]];
    bitout = If[i == 1, {}, out[[All, i + 1]]];
    out = Join[out, newout];
    Print["round ", i]; Print[{{posout, sgnout, bitout}}];
  ];
  Join[out, in] // MatrixForm
]
```

 ■ `progtrans[list,K]`

? progtrans

Computes the same data as `transmit`, but restricts output to the first K bits.

Uses: `convertlist`

Original file: progtrans

In[9]=

```
progtrans[list_, K_] :=
Module[{len, cvl, n, in, out, newout, sgnout, posout, bitout},
  len = Length[list];
  cvl = convertlist[list];
  n = cvl[[1]];
  If[K > n, Print["order too big"]];
  in = Table[
    Prepend[cvl[[2, i]], i], {i, 1, len}];
  out = {};
  For[j = 1, j <= Min[n, K], j++,
    newout = Select[in, #[[j + 2]] == 1 &];
    in = Select[in, #[[j + 2]] == 0 &];
    sgnout = newout[[All, 2]];
    posout = newout[[All, 1]];
    bitout = If[j == 1, {}, out[[All, j + 2]]];
    out = Join[out, newout];
    Print["round ", j];
    Print[{posout, sgnout, bitout}];
  ];
  Print["remaining ", in // MatrixForm];
  Join[out, in] // MatrixForm
]
```

 ■ `recons[reo,k]`

? recons

Assumes that `reo` has been constructed using procedure `reorder` and that k is not bigger than the binary length of the integers encoded. Reconstructs a k -bit decimal approximation of these integers and outputs them sorted according to their original positions.

Uses: `bitseq2num`

Original file: decode.nb

In[10]=

```
recons[reo_, k_] :=
Module[{rows, cols, bd, out},
  {rows, cols} = Dimensions[reo];
  bd = cols - 2;
  If[k < 1 || k > bd, Throw["k false"]];
  out = Map[{-#[[1]], (-1)^#[[2]]*
    bitseq2num[#[[3 ;; cols]], k]} &, reo];
  SortBy[out, First]
]
```

Zigzag ordering and embedded trees

L2M, M2L, embtrees

■ L2M[int]

? L2M

Computes the coordinates of *int* w.r.t. the zigzag ordering.
 Uses: --
 Original file: zigzag.nb

```
L2M[int_] :=
Module[{id, len, row, col},
  id = IntegerDigits[int, 2];
  len = Length[id];
  If[OddQ[len], id = Prepend[id, 0]; len++];
  row = Table[id[[2 k - 1]], {k, 1, len/2}];
  col = Table[id[[2 k]], {k, 1, len/2}];
  {FromDigits[row, 2], FromDigits[col, 2]}
]
```

■ M2L[row,col]

? M2L

Computes the ordinal of the *(row,col)*-entry of N^2 w.r.t. the zigzag ordering.
 Uses: --
 Original file: zigzag.nb

In[11]:=

```
M2L[row_, col_] :=
Module[{rbin, cbin, diff},
  rbin = IntegerDigits[row, 2];
  cbin = IntegerDigits[col, 2];
  diff = Length[rbin] - Length[cbin];
  If[diff > 0, cbin = Join[Table[0, diff], cbin],
    rbin = Join[Table[0, -diff], rbin]
  ];
  FromDigits[Flatten[Table[{rbin[[k]], cbin[[k]]},
    {k, 1, Length[rbin]}], 2]
]
```

- `embtrees[imgdata,order]`

? *embtrees*

imgdata: a square matrix of order 2^k
order: number of levels of a DWT
 Computes a linear representation of the 2D-data *imgdata*, where the zigzag-ordering is used. Output *T* is the encoding of the forest of quadrees, output *C* is the corresponding data vector.
 Uses: **M2L**, **L2M**
 Original file: zigzag

In[12]=

```
embtrees[imgdata_, order_] :=
Module[{r, c, rlog, a, i, j, tab, T, C},
  {r, c} = Dimensions[imgdata];
  If[! r == c, Throw["image not square"]];
  rlog = Log2[r];
  If[! IntegerQ[rlog], Throw["size is not power of 2"]];
  If[order >= rlog, Throw["order too large"]];
  a = 2^(rlog - order);
  tab = Table[Quotient[M2L[i, j], 4],
    {i, 0, r - 1}, {j, 0, c - 1}];
  For[i = 0, i < a, i++,
    For[j = 0, j < a, j++,
      tab[[i + 1, j + 1]] = M2L[i, j];
      If[EvenQ[i] && OddQ[j],
        tab[[i + 1 ;; i + 2, a + j ;; a + j + 1]]
          = {{M2L[i, j], M2L[i, j]}, {M2L[i, j], M2L[i, j]}}];
      If[OddQ[i] && EvenQ[j],
        tab[[a + i ;; a + i + 1, j + 1 ;; j + 2]]
          = {{M2L[i, j], M2L[i, j]}, {M2L[i, j], M2L[i, j]}}];
      If[OddQ[i] && OddQ[j],
        tab[[a + i ;; a + i + 1, a + j ;; a + j + 1]]
          = {{M2L[i, j], M2L[i, j]}, {M2L[i, j], M2L[i, j]}}];
    ]];
  T = Table[tab[[L2M[k][[1]] + 1, L2M[k][[2]] + 1]],
    {k, 0, r^2 - 1}];
  T = Map[# + 1 &, T];
  C = Table[imgdata[[L2M[k][[1]] + 1, L2M[k][[2]] + 1]], {k, 0, r^2 - 1};
  {T, C}
]
```

Exploring and scanning trees/forests

maketree, scantree (old), sigma, scantree (new), spiht

■ maketree[T, flag]

? maketree

T: a linear encoded forest
 Use *flag=1* is used for non/verbose output
 Computes *roots*, *leaves* and successor map ω for the SPIHT algorithm,
 and also the set of *proper edges* and the vertex valuation *label* for
 purposes of display.
 Uses: --
 Original file: spiht.nb

In[13]=

```
maketree[T_, flag_] :=
Module[{n, treetest, vertices, edges, roots, leaves, properedges,  $\omega$ , graph, labels},
  n = Length[T];
  vertices = Range[n];
  If[! Complement[T, vertices] == {},
    Throw["input error"]];
  Tk_ := T[[k]];
  treetest = ReplacePart[Table[Ti <= i, {i, 1, n}], 0 -> And];
  If[! treetest, Throw["T not a tree"]];
  Print["treetest finished"]; edges = Table[{Tj, j}, {j, 1, n}];
  roots = Flatten[Map[DeleteDuplicates[#] &, Select[edges, #[[1]] == #[[2]] &]]];
  properedges =
    Select[edges, #[[1]] < #[[2]] &];
  Print["properedges finished"];
   $\omega$  = Table[Map[#[[2]] &, Select[properedges, #[[1]] == k &]], {k, 1, n}];
   $\omega$ k_ :=  $\omega$ [[k]];
  Print["omega finished"]; leaves = Select[vertices,  $\omega$ # == {} &];
  graph = Map[#[[1]] -> #[[2]] &, properedges];
  labels = Table[j -> "v"j, {j, 1, n}];
  If[flag == 1,
    Print["vertices: ", vertices];
    Print["roots: ", roots];
    Print["leaves: ", leaves]
  ];
  {vertices, roots, leaves,  $\omega$ , graph, labels}
]
```

■ sigmap[T,C,N,flag]
? sigmap

T: a linear encoded tree or forest
C: a vertex valuation
N: a significance level
 Use *flag=1* for verbose output
 Computes roots, leaves, the successor map ω and the σ , δ and λ valuations
 Uses: **maketree**
 Original file: spiht.nb

In[14]:=

```

sigmap[T_, C_, N_, flag_] :=
Module[{n, vertices, roots, leaves,  $\omega$ ,  $\sigma$ ,  $\delta$ ,  $\lambda$ , graph, labels},
  n = Length[T];
  vertices = Range[n];
  {roots, leaves,  $\omega$ } = maketree[T, flag][[2 ;; 4]];
  If[!(n == Length[C]), Throw["input error"]];
  Tk_ := T[[k]];
  Ck_ := C[[k]];
   $\omega$ k_ :=  $\omega$ [[k]];
  For[j = n, j > 0, j--,
    If[Abs[Cj] ≥ N,  $\sigma_j = 1$ ,  $\sigma_j = 0$ ];
    If[ $\omega_j == \{\}$ ,  $\delta_j = 0$ ;
       $\lambda_j = 0$ ; Continue[]];
     $\lambda_j = \text{Max}[\text{Map}[\delta_{\#} \&, \omega_j]]$ ;
     $\delta_j = \text{Max}[\text{Join}[\text{Map}[\sigma_{\#} \&, \omega_j], \{\lambda_j\}]]$ 
  ];
   $\sigma = \text{Table}[\sigma_j, \{j, 1, n\}]$ ;
   $\delta = \text{Table}[\delta_j, \{j, 1, n\}]$ ;
   $\lambda = \text{Table}[\lambda_j, \{j, 1, n\}]$ ;
  {roots, leaves,  $\omega$ ,  $\sigma$ ,  $\delta$ ,  $\lambda$ }
]

```

■ scantree
? scantree

T: a linear encoded tree or forest
C: a vertex valuation
N: a significance level
LIS,LIP,LSP: list required for SPIHT
 Use *flag=1* for verbose output
 Computes updated lists *LIS*, *LIP*, and *LSP* and binary output for
 one round of the SPIHT algorithm
 Uses: **sigmap**
 Original file: spiht.nb

In[15]=

```

scantree[T_, C_, N_, LIS_, LIP_, LSP_, flag_] :=
Module[{roots, leaves,  $\omega$ ,  $\sigma$ ,  $\delta$ ,  $\lambda$ , out, tmpLIS,
  newLIS, oldLIP, newLIP, oldLSP, newLSP, p, t, succ, elem, z, j, k},
  {roots, leaves,  $\omega$ ,  $\sigma$ ,  $\delta$ ,  $\lambda$ }
  = sigmap[T, C, N, flag] [[1 ;; 6]];
  Ck_ := C[[k]];
   $\omega_k$  :=  $\omega$ [[k]];
   $\sigma_k$  :=  $\sigma$ [[k]];
   $\delta_k$  :=  $\delta$ [[k]];
   $\lambda_k$  :=  $\lambda$ [[k]];
  {tmpLIS, newLIS} = {LIS, {}};
  {oldLIP, newLIP} = {LIP, {}};
  newLSP = LSP;
  If[flag == 1,
    Print["N: ", N];
    Print["LIS: ", tmpLIS];
    Print["new LIS: ", newLIS];
    Print["LIP: ", oldLIP];
    Print["LSP: ", newLSP]
  ];
  out = {};
  For[j = 1, j ≤ Length[oldLIP], j++,
    elem = oldLIP[[j]];
    out = Append[out,  $\sigma_{elem}$ ];
    If[flag == 1,
      Print["out: ", " $\sigma$ "[elem], "= ",  $\sigma_{elem}$ ]
    ];
    If[ $\sigma_{elem}$  == 1,
      out = Append[out, sign[Celem]];
      If[flag == 1,
        Print["out: ", sign[Celem]]
      ];
      newLSP = Append[newLSP, elem],
      newLIP = Append[newLIP, elem]
    ];
  ];
  If[flag == 1,
    Print["LSP: ", newLSP];
    Print["LIP: ", Join[oldLIP, newLIP]]
  ];
  While[! tmpLIS == {},
    If[flag == 1,
      Print["LIS: ", tmpLIS]];
    {p, t} = First[tmpLIS];
    tmpLIS = Rest[tmpLIS];
    If[t == "A",
      out = Append[out,  $\delta_p$ ];
      If[flag == 1,

```

```

    Print["out: ", " $\delta$ "[p], "= ",  $\delta_p$ ]
];
If[ $\delta_p == 1$ ,
  For[k = 1, k ≤ Length[ $\omega_p$ ], k++,
    z =  $\omega_p$ [ [k] ];
    out = Append[out,  $\sigma_z$ ];
    If[flag == 1,
      Print["out: ", " $\sigma$ "[z], "= ",  $\sigma_z$ ]
    ];
    If[ $\sigma_z == 1$ ,
      out = Append[out, sign[Cz]];
      If[flag == 1,
        Print["out: ", sign[Cz]]
      ];
      newLSP = Append[newLSP, z],
      newLIP = Append[newLIP, z]
    ];
    If[flag == 1,
      Print["LSP: ", newLSP];
      Print["LIP: ", Join[oldLIP, newLIP]];
    ];
  ];
  ssucc = Flatten[Map[ $\omega_H$  &,  $\omega_p$ ]];
  If[! ssucc == {},
    tmpLIS = Append[tmpLIS, {p, "B"}];
    If[flag == 1,
      Print["LIS: ", tmpLIS];
    ];
  ];
];
If[ $\delta_p == 0$ ,
  newLIS = Append[newLIS, {p, t}];
  If[flag == 1,
    Print["newLIS: ", newLIS];
  ];
];
If[t == "B",
  out = Append[out,  $\lambda_p$ ];
  If[flag == 1,
    Print["out: ", " $\lambda$ "[p], "= ",  $\lambda_p$ ]
  ];
];
If[ $\lambda_p == 1$ ,
  For[k = 1, k ≤ Length[ $\omega_p$ ], k++,
    elem =  $\omega_p$ [ [k] ];
    tmpLIS = Append[tmpLIS, {elem, "A"}];
  ];
  If[flag == 1,
    Print["LIS: ", tmpLIS]
  ];
];
];
If[ $\lambda_p == 0$ ,

```

```

    newLIS = Append[newLIS, {p, t}];
    If[flag == 1,
      Print["newLIS: ", newLIS]
    ];
  ];
];
];
If[flag == 1,
  Print["new LIS: ", newLIS];
  Print["new LIP: ", newLIP];
  Print["new LSP: ", newLSP];
  Print["output: ", out];
  {newLIS, newLIP, newLSP, out}
]

```

■ spiht[T,C,K,flag]

? spiht

T: a linear encoded tree or forest
C: a vertex valuation
K: number of rounds for scantree to execute
 Use *flag=1* for verbose output
 Executes the SPIHT algorithm
 Uses: **maketree**, **scantree**
 Original file: spiht.nb

In[16]=

```

spiht[T_, C_, K_, flag_] :=
Module[{n, r, j, vertices, roots,
  properroots, leaves,
  NN, LIS, LIP, LSP, oldLSP, sortout, refout, total},
  n = Length[T];
  If[!(n == Length[C]), Throw["input error"]];
  vertices = Range[n];
  {roots, leaves, ω} = maketree[T, flag][[1 ;; 3]];
  properroots = Complement[roots, leaves];
  Tk_ := T[[k]];
  Ck_ := C[[k]];
  ωk_ := ω[[k]];
  r = Floor[Log[2, Max[Abs[C]]]];
  NN = 2^r;
  LIS = Map[{#, "A"} &, properroots];
  LIP = roots;
  LSP = {};
  If[flag == 1, Print[{LIS, LIP, LSP}]]
  ];
  {LIS, LIP, LSP, sortout} = scantree[T, C, NN, LIS, LIP, LSP, flag];
  If[flag == 1, Print[{LIS, LIP, LSP}]]
  ];
  Print["round: 1", " NN: ", NN];

```

```
Print["sorting: ", sortout];
total = {{sortout, {}}};
oldLSP = LSP;
For[j = 1, j < K, j++,
  r = r - 1;
  NN = NN / 2;
  Print["round: ", j + 1, " NN: ", NN];
  {LIS, LIP, LSP, sortout} = scantree[T, C, NN, LIS, LIP, LSP, flag];
  Print["sorting: ", sortout];
  refout = Map["bit"[Abs[C#], r] &, oldLSP];
  Print["refinement: ", refout];
  total = Append[total, {sortout, refout}];
  oldLSP = LSP;
  If[flag == 1, Print[{LIS, LIP, LSP}]]
];
Map[Length[#] &, total, {2}]
]
```

Using matrices for ω in 2D-spiht

ω 2D, sigmap2D, scantree2D, init2D, spiht2D

■ ω 2D[n,m,i,j]

? ω 2D

n: Log2 of matrix size
m: levels for DWT
(i,j): position in the image matrix
 Computes the successor nodes for embedded trees
 Uses: --
 Original file: spiht2D

```
In[17]:=  $\omega$ 2D[n_, m_, i_, j_] :=
Module[{a},
  If[n <= m, Throw["not feasible"]];
  a = 2^(n - m);
   $\epsilon$  = {{0, 0}, {0, 1}, {1, 0}, {1, 1}};
  If[i >= 2^(n - 1) || j >= 2^(n - 1), Return[{}]];
  If[0 <= i < a && 0 <= j < a,
    If[EvenQ[i] && EvenQ[j],
      Return[{}]];
    If[EvenQ[i] && OddQ[j],
      Return[Map[{i, j + a - 1} + # &,  $\epsilon$ ]];
    If[OddQ[i] && EvenQ[j],
      Return[Map[{i + a - 1, j} + # &,  $\epsilon$ ]];
    If[OddQ[i] && OddQ[j],
      Return[Map[{i + a - 1, j + a - 1} + # &,  $\epsilon$ ]];
  ];
  Return[Map[{2 i, 2 j} + # &,  $\epsilon$ ]]
]
```

■ sigmap2D

? sigmap2D

M: a square image matrix with sidelength $n=2^r$
m: order of envisaged DWT
NN: significance level
 Computes functions σ, δ, λ as matrices
 Uses: **L2M**, **ω 2D**
 Original file: spiht-2D.nd

```
In[18]:= sigmap2D[M_, m_, NN_] :=
Module[{r, c, k, n, a, s, i, j,  $\Sigma$ ,  $\Delta$ ,  $\Lambda$ , succ},
  {r, c} = Dimensions[M];
  If[! r == c, Throw["matrix is not square"]];
  n = Log2[r];
  If[! IntegerQ[n], Throw["size is not power of 2"]];
  If[m >= n, Throw["order too large"]];
  a = 2^(n - m);
   $\Sigma$  = Map[If[Abs[#] >= NN, 1, 0] &, M, {2}];
   $\Delta$  = Table[0, {i, 1, 2^n}, {j, 1, 2^n}];
   $\Lambda$  = Table[0, {i, 1, 2^n}, {j, 1, 2^n}];
  For[k = n - 1, k >= 1, k--,
```

```

For[s = 2^(2 k) - 1, s >= 2^(2 k - 2), s--,
  {i, j} = L2M[s];
  (*
  Print[{i,j}];
  *)
  succ =  $\omega$ 2D[n, m, i, j];
  If[succ == {},
     $\Delta$ [[i + 1, j + 1]] = 0;  $\Delta$ [[i + 1, j + 1]] = 0;
    Continue[]];
   $\Delta$ [[i + 1, j + 1]] =
  Max[Map[ $\Delta$ [[#[[1]] + 1, #[[2]] + 1]] &, succ]];
   $\Delta$ [[i + 1, j + 1]] =
  Max[Join[ $\Delta$ [[i + 1, j + 1]], Map[ $\Sigma$ [[#[[1]] + 1, #[[2]] + 1]] &, succ]]]
  ]
];
{ $\Sigma$ ,  $\Delta$ ,  $\Lambda$ }
]

```

■ scantree2D

? scantree2D

M: a square image matrix with sidelength $n=2^r$
m: order of envisaged DWT
NN: significance level
LIS, LIP, LSP: lists for the spiht algorithm
 Use *flag=1* for verbose output
 Computes one round of the spiht algorithm,
 updates *LIS, LIP* and *LSP* and produces binary output
 Uses: **sigmap2D, ω 2D, M2L**
 Original file: spiht-2D.nb

```

scantree2D[M_, m_, NN_, LIS_, LIP_, LSP_, flag_] := Module[ $\{\Sigma, \Delta, \Lambda, out, tmpLIS,$ 
newLIS, oldLIP, newLIP, oldLSP, newLSP, p, p1, p2, t, elem, z1, z2, j, k, a1, a2,
b, r, c, n, a},
  {r, c} = Dimensions[M];
  If[! r == c, Throw["matrix is not square"]];
  n = Log2[r];
  If[! IntegerQ[n], Throw["size is not power of 2"]];
  If[m >= n, Throw["order too large"]];
  a = 2^(n - m);
  { $\Sigma, \Delta, \Lambda$ } = sigmap2D[M, m, NN];
  {tmpLIS, newLIS} = {LIS, {}};
  {oldLIP, newLIP} = {LIP, {}};
  newLSP = LSP;
  out = {};
  For[j = 1, j <= Length[oldLIP], j++,
    {a1, a2} = oldLIP[[j]];
    out = Append[out,  $\Sigma$ [[a1 + 1, a2 + 1]]];
    If[ $\Sigma$ [[a1 + 1, a2 + 1]] == 1,
      out = Append[out, sign[M[[a1 + 1, a2 + 1]]]];
      newLSP = Append[newLSP, {a1, a2}],
      newLIP = Append[newLIP, {a1, a2}]];
  ];
  While[! tmpLIS == {},
    If[flag == 1, Print["LIS: ", tmpLIS]];
    {p, t} = First[tmpLIS];
    {p1, p2} = p;
  ];

```

```

tmpLIS = Rest[tmpLIS];
If[t == "A",
  out = Append[out, Δ[[p1 + 1, p2 + 1]]];
  If[Δ[[p1 + 1, p2 + 1]] == 1,
    For[k = 1, k <= 4, k++,
      {z1, z2} = ω2D[n, m, p1, p2][[k]];
      out = Append[out, Σ[[z1 + 1, z2 + 1]]];
      If[Σ[[z1 + 1, z2 + 1]] == 1,
        out = Append[out, sign[M[[z1 + 1, z2 + 1]]]];
        newLSP = Append[newLSP, {z1, z2}],
        newLIP = Append[newLIP, {z1, z2}]
      ]];
  If[M2L[p1, p2] < 2^(2 n - 4),
    tmpLIS = Append[tmpLIS, {p, "B"}]
  ];
  If[Δ[[p1 + 1, p2 + 1]] == 0,
    newLIS = Append[newLIS, {p, t}]
  ];
If[t == "B",
  out = Append[out, Δ[[p1 + 1, p2 + 1]]];
  If[Δ[[p1 + 1, p2 + 1]] == 1,
    For[k = 1, k <= 4, k++,
      elem = ω2D[n, m, p1, p2][[k]];
      tmpLIS = Append[tmpLIS, {elem, "A"}]
    ];
  If[Δ[[p1 + 1, p2 + 1]] == 0,
    newLIS = Append[newLIS, {p, t}]
  ];
];
If[flag == 1,
  Print["new LIS: ", newLIS];
  Print["new LIP: ", newLIP];
  Print["new LSP: ", newLSP];
  Print["output: ", out];
  {newLIS, newLIP, newLSP, out}
]

```

■ `init2D[M,m]`

? `init2D`

M: a square image matrix with sidelength $n=2^r$
m: order of envisaged DWT
 Initializes data for the SPIHT algorithm
 Uses: --
 Original file: spiht2D.nb

In[20]=

```

init2D[M_, m_] := Module[{r, c, n, a, roots, properroots},
  {r, c} = Dimensions[M];
  If[! r == c, Throw["matrix is not square"]];
  n = Log2[r];
  If[! IntegerQ[n], Throw["size is not power of 2"]];
  If[m >= n, Throw["order too large"]];
  a = 2^(n - m);
  roots = Table[L2M[k], {k, 0, a^2 - 1}];
  properroots =
    Select[roots, OddQ[#[[1]]] || OddQ[#[[2]]] &];
  {roots, properroots}
]

```

 ■ spiht2D[M,m,K,flag]

? spiht2D

M: a square image matrix with sidelength $n=2^r$
m: order of envisaged DWT
K: number of rounds of scantree2D to execute
 Use *flag=1* for verbose output
 Executes the SPIHT algorithm on 2D data
 Uses: **init2D**, **scantree2D**
 Original file: spiht-2D.nb

```

spiht2D[M_, m_, K_, flag_] :=
Module[
  {r, j, roots, properroots, leaves, NN, LIS, LIP, LSP, oldLSP, sortout, refout,
total},
  {roots, properroots} = init2D[M, m];
  r = Floor[Log[2, Max[Abs[M]]]];
  NN = 2^r;
  LIS = Map[{"#", "A"} &, properroots];
  LIP = roots;
  LSP = {};
  If[flag == 1, Print[{LIS, LIP, LSP}]];
  {LIS, LIP, LSP, sortout}
    = scantree2D[M, m, NN, LIS, LIP, LSP, flag];
  If[flag == 1, Print[{LIS, LIP, LSP}]];
  Print["round: 1", " NN: ", NN];
  Print["sorting: ", sortout];
  total = {{sortout, {}}};
  oldLSP = LSP;
  For[k = 1, k < K, k++,
    r = r - 1;
    NN = NN/2;
    Print["round: ", k + 1, " NN: ", NN];
    {LIS, LIP, LSP, sortout}
      = scantree2D[M, m, NN, LIS, LIP, LSP, flag];
    Print["sorting: ", sortout];
    refout =
      Map[ "bit"[Abs[M[[#[[1]] + 1, #[[2]] + 1]]], r] &, oldLSP];
    Print["refinement: ", refout];
    total = Append[total, {sortout, refout}];
    oldLSP = LSP;
    If[flag == 1, Print[{LIS, LIP, LSP}]]
  ];
  Map[Length[#] &, total, {2}]
]

```


14. CWT and edge detection

This chapter contains the slides of the lectures dealing with the continuous wavelet transform (CWT) and its application to edge detection in images. For illustrations see the material in the web page of the lecture.

- 1D edge detection
 - The dangers of taking derivatives numerically
 - Continuous wavelet transform
 - Wavelet functions and derivatives
 - CWT, scaling and wavelet identities
 - The *à-trous* algorithm
- 2D edge detection
 - Canny's method for edge detection
 - 2D separable DWT and gradients
 - Gradient computation using scaling and wavelet identities
- References for CWT: [GM84], [Cal64], [JMR01], [KL16], [Hol95], [Bla98], [Mal09]
- References for edge detection: [Can86], [MZ92], [Mal09]

The following Mathematica notebooks are included

- Edge detection using the Sobel operators
- The *à trous algorithm* illustrated
- Edge detection using wavelets

- ▶ Initial event:
A. GROSSMANN and J. MORLET,
Decompositions of Hardy functions into square integrable
wavelets of constant shape, *SIAM J. Math. Analysis*, 1984
(Analysis of seismic signals)
 - ▶ ... but there were precursors ... e.g.
A. P. CALDÉRON,
Intermediate Spaces and Interpolation, the Complex Method,
Studia Mathematica, 1964
 - ▶ see:
S. JAFFARD, Y. MEYER, R. RYAN,
Wavelets, Tools for Science and Technology, SIAM 2001,
in particular: Chap. 2: Wavelets from a Historical Perspective
-
- ▶ J. CANNY,
A computational approach to edge detection, *IEEE Trans
Patt. Recog. and Mach. Intell.* 6, 961–1005, 1986.
 - ▶ S. G. MALLAT and S. ZHONG,
Characterization of signals from multiscale edges, *IEEE Trans
Patt. Recog. and Mach. Intell.*, 14, 710–732, 1992.

- ▶ When dealing with a discretized version of a function $f(t)$

$$\dots f(t_0), f(t_0 + h), f(t_0 + 2h), \dots$$

(step size h for sampling) one may take the difference quotient as a numerical approximation of the derivative

$$f'(t_0) = \frac{d}{dt}f(t_0) \approx \frac{f(t_0 + h) - f(t_0)}{h}$$

- ▶ From the numerical point of view this is a highly dangerous method (in particular if it is used iteratively) if the step size h is small
 - ▶ As a rule: first apply a *smoothing* operator to the data before taking differences
 - ▶ See the notebook *ramp-en.pdf* for illustration
-
- ▶ See handout *cwt-en.pdf* for the details and more information
 - ▶ See notebook *CWT-15-en.pdf* for illustrations of the continuous wavelet transform
 - ▶ Relevant notebooks for illustration of edge detection
 - ▶ *ramp-en.pdf*
 - ▶ *atrous-poly-en.pdf*
 - ▶ *sobel-en.pdf*
 - ▶ *circlestest.pdf*
 - ▶ *wedges-en.pdf*

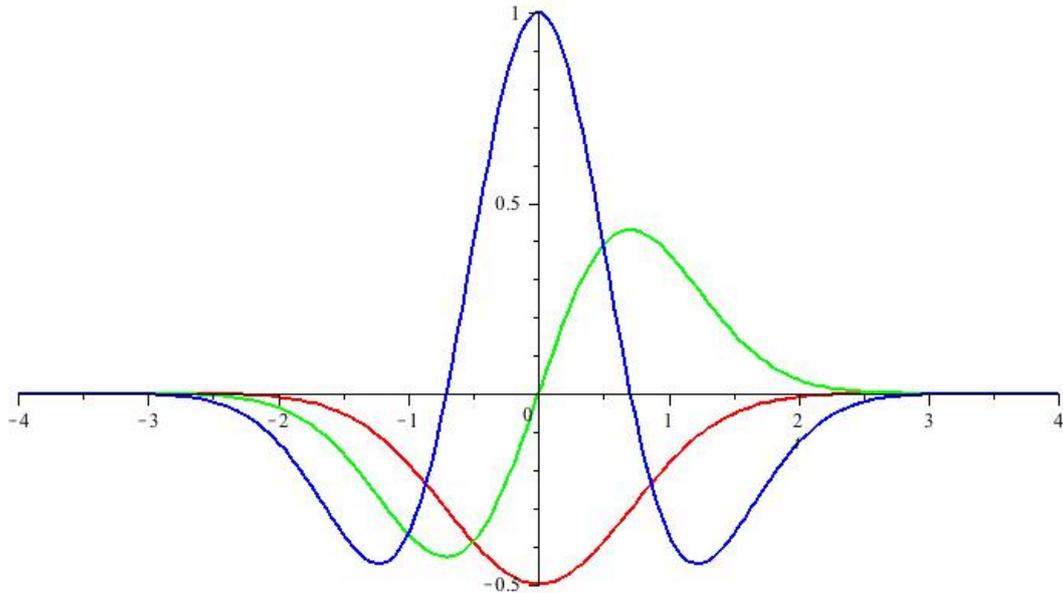


Figure: mexican-hat wavelet as second derivative of a Gaussian

- ▶ Generally: wavelet transforms consist in both
 - ▶ *smoothing* operations (approximation, low-pass filtering)
 - ▶ *differencing* operations (detail, high-pass filtering)
- ▶ $\psi(t)$ a “suitable” wavelet function, e.g.,
 - (1) $\psi(t)$ is continuous and has vanishing zero-th moment

$$\widehat{\psi}(0) = \int_{\mathbb{R}} \psi(t) dt = 0$$

- (2) $\psi(t)$ decays rapidly as $t \rightarrow \pm\infty$

$$|\psi(t)| \leq A e^{-B|t|} \quad (t \in \mathbb{R}) \quad \text{for some constants } A, B > 0$$

- ▶ normalization (for convenience) $\|\psi\|^2 = 1$
- ▶ Localization (mean and variance)

$$\mu = \int_{\mathbb{R}} t |\psi(t)|^2 dt \quad \sigma^2 = \int_{\mathbb{R}} (t - \mu)^2 |\psi(t)|^2 dt$$

- ▶ continuous scaling (dilation and translation) of $\psi(t)$

$$\psi_{s,a}(t) = \frac{1}{\sqrt{|s|}} \psi\left(\frac{t-a}{s}\right) \quad (s, a \in \mathbb{R})$$

(this notation differs from the one used in the DWT context)

- ▶ Localization

$$\begin{aligned} \mu_{s,a} &= \int t |\psi_{s,a}(t)|^2 dt = \dots = s\mu + a \\ \sigma_{s,a}^2 &= \int (t - \mu_{s,a})^2 |\psi_{s,a}(t)|^2 dt = \dots = s^2 \sigma^2 \end{aligned}$$

- ▶ Fourier transform

$$\widehat{\psi}_{s,a}(\lambda) = \sqrt{s} e^{-2\pi i a \lambda} \widehat{\psi}(s\lambda)$$

- ▶ Definition of the continuous wavelet transform (CWT)

$$\begin{aligned} f(t) &\longmapsto f^\psi(s, a) = \langle f | \psi_{s,a} \rangle \\ &= \int_{\mathbb{R}} f(t) \overline{\psi_{s,a}(t)} dt \end{aligned}$$

- ▶ Note

$$\|f - \psi_{s,a}\|^2 = \|f\|^2 + \|\psi\|^2 - 2 \Re [f^\psi(s, a)]$$

- ▶ Intuitively: $f^\psi(s, a)$ correlates the behavior of $f(t)$ with that of $\psi(t)$ in the vicinity of $a \in \mathbb{R}$ (if $\mu = 0$) in resolution (scaling) $s \in \mathbb{R}$
- ▶ The CWT transform data $\{f^\psi(s, a)\}_{s,a \in \mathbb{R}}$ are highly redundant!

▶ CALDÉRON's reconstruction formula

$f(t)$ can be reconstructed from its CWT transform data $\{f^\psi(s, a)\}_{s, a \in \mathbb{R}}$ under suitable conditions

$$f(t) = \frac{1}{C_\psi} \int_{s \in \mathbb{R}} \int_{a \in \mathbb{R}} f^\psi(s, a) \psi_{s, a}(t) da \frac{ds}{s^2}$$

▶ Here the number

$$C_\psi = \int_{\lambda \in \mathbb{R}} \frac{|\widehat{\psi}(\lambda)|^2}{|\lambda|} d\lambda$$

must be finite and > 0

▶ This holds if conditions (1) and (2) for $\psi(t)$ are satisfied

▶ The HAAR wavelet function $\psi_{haar}(t)$ can be regarded as a derivative

$$\psi_{haar}(t) = \frac{d}{dt} \Delta(t) \quad \text{where} \quad \Delta(t) = \begin{cases} t & 0 \leq t \leq 1/2 \\ 1 - t & 1/2 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

$\Delta(t)$ is a smoothing function

▶ The mexican-hat wavelet function $\psi_{mex}(t)$ is a derivative

$$\psi_{mex}(t) = \frac{d}{dt} (t e^{-t^2}) = -\frac{1}{2} \frac{d^2}{dt^2} e^{-t^2}$$

but $t e^{-t^2}$ is not really a smoothing function

- ▶ Take $\psi(t)$ as the derivative of a smoothing function $\theta(t)$

$$\psi(t) = \frac{d}{dt} \theta(t)$$

and define the scaled (dilated) and reversed version of $\theta(t)$ as

$$\overleftarrow{\theta}_s(t) = \frac{1}{\sqrt{s}} \theta\left(-\frac{t}{s}\right)$$

- ▶ Then one has (simple exercise in differentiating under the integral)

$$f^\psi(s, a) = -s \frac{d}{da} (f \star \overleftarrow{\theta}_s)(a)$$

- ▶ Note: the convolution $f \star \overleftarrow{\theta}_s$ is a $\overleftarrow{\theta}_s$ -smoothed version of f
- ▶ Interpretation: Edges in the graph of $f(t)$ (high absolute values of the derivative) can be recognized by absolutely large values of the wavelet coefficients $f^\psi(s, a)$ over many scales (s values)

- ▶ Assume that for the wavelet function $\psi(t)$ one has a scaling function $\phi(t)$ (as in the MRA situation)
- ▶ Scaling and wavelet identities are

$$\phi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} h_k \phi(2t - k) \quad \psi(t) = \sqrt{2} \sum_{k \in \mathbb{Z}} g_k \phi(2t - k)$$

- ▶ Consider dyadic scaling for $\phi(t)$ and $\psi(t)$, i.e.,

$$\phi_{2^m, a}(t) = \frac{1}{2^{m/2}} \phi\left(\frac{t - a}{2^m}\right) \quad \psi_{2^m, a}(t) = \frac{1}{2^{m/2}} \psi\left(\frac{t - a}{2^m}\right)$$

- ▶ Scaling and wavelet identities turn into

$$\phi_{2^{m+1}, a}(t) = \sum_{k \in \mathbb{Z}} h_k \phi_{2^m, a+k 2^m}(t) \quad \psi_{2^{m+1}, a}(t) = \sum_{k \in \mathbb{Z}} g_k \phi_{2^m, a+k 2^m}(t)$$

- Approximation and detail coefficients of a function $f(t)$, for dyadic scaling and integer translation
 $(s, a) = (2^m, n) \quad (m, n \in \mathbb{Z})$

$$a_{m,n} = \langle f | \phi_{2^m,n} \rangle \quad d_{m,n} = \langle f | \psi_{2^m,n} \rangle$$

- Recursion formulas for approximation and wavelet coefficients

$$a_{m+1,n} = \sum_{k \in \mathbb{Z}} h_k a_{m,n+k 2^m} \quad (n \in \mathbb{Z})$$

$$d_{m+1,n} = \sum_{k \in \mathbb{Z}} g_k a_{m,n+k 2^m} \quad (n \in \mathbb{Z})$$

- Written as filtering operations

$$\mathbf{a}^{(m+1)} = (a_{m+1,n})_{n \in \mathbb{Z}} = \overleftarrow{[(\uparrow_2)^m \mathbf{h}]} \star \mathbf{a}^{(m)}$$

$$\mathbf{d}^{(m+1)} = (d_{m+1,n})_{n \in \mathbb{Z}} = \overleftarrow{[(\uparrow_2)^m \mathbf{g}]} \star \mathbf{a}^{(m)}$$

- Here $(\uparrow_2)^m \mathbf{h}$ is the filter constructed from \mathbf{h} by using m -fold upsampling with factor 2 (“spreading”)

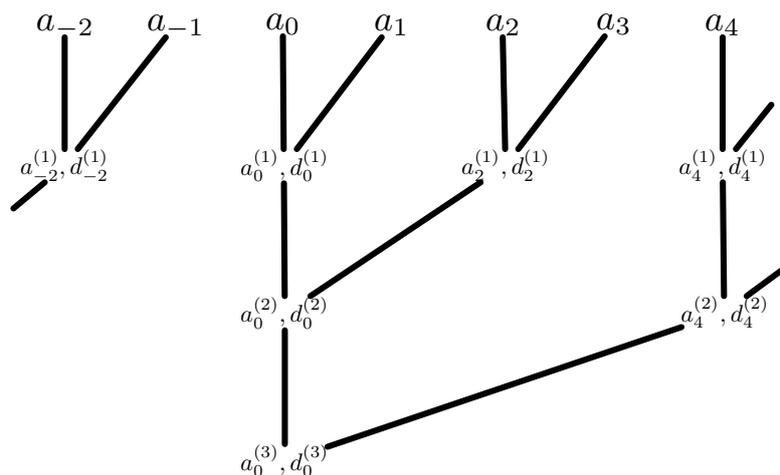


Figure: Scheme of the Haar transform

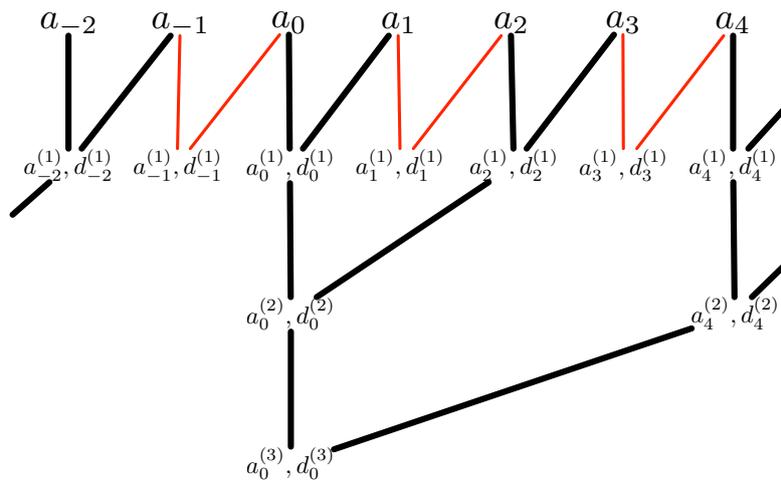


Figure: \grave{a} -trous scheme (one level) for the Haar transform

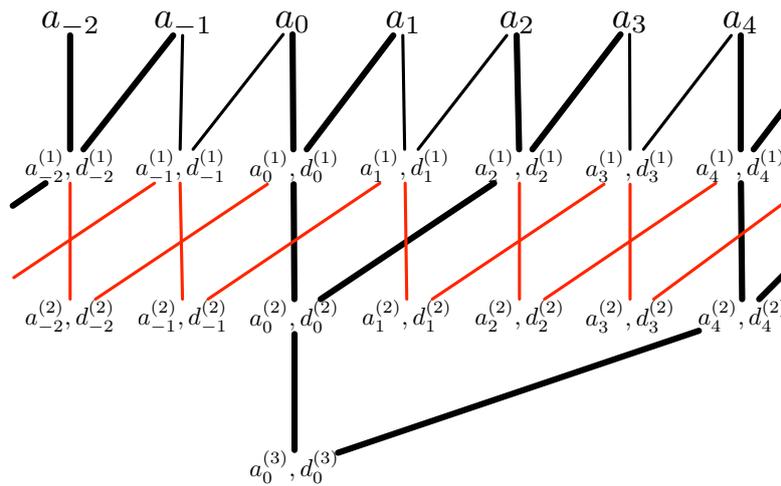


Figure: \grave{a} -trous scheme (two levels) for the Haar transform

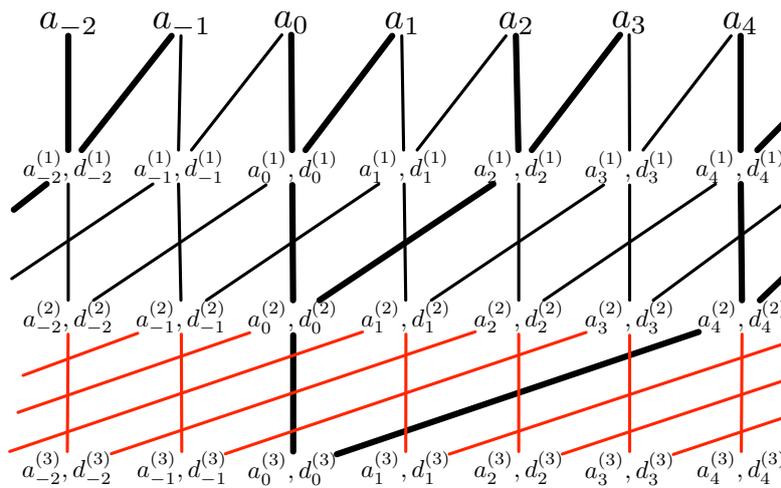


Figure: \grave{a} -trous scheme (three levels) for the Haar transform

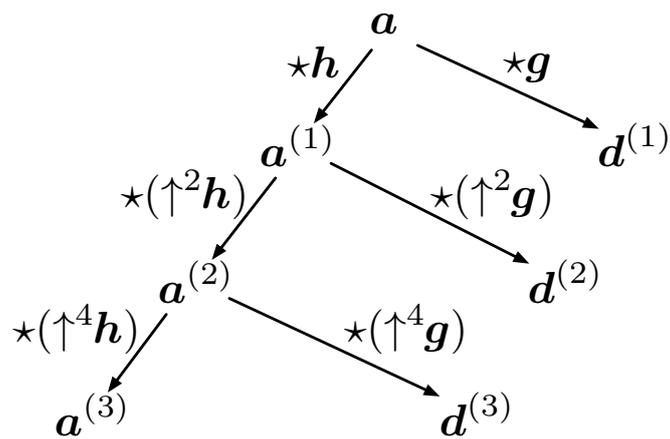


Figure: \grave{a} -trous scheme (three levels)

high-pass filter: \mathbf{g} low-pass filter: \mathbf{h} signal: $\mathbf{a} = (a_n)_{n \in \mathbb{Z}}$
 filtered signals: $\mathbf{a}^{(k)} = (a_n^{(k)})_{n \in \mathbb{Z}}$ (approximation),
 $\mathbf{d}^{(k)} = (d_n^{(k)})_{n \in \mathbb{Z}}$ (detail)

Consider a function (an "image") $f(x, y) \in \mathcal{L}^2(\mathbb{R}^2)$.

$\partial_x f, \partial_y f$: partial derivatives of f ,

$$\nabla f(x_0, y_0) = (\partial_x f(x_0, y_0), \partial_y f(x_0, y_0)) \quad \text{gradient of } f$$

► CANNY's edge definition

$(x_0, y_0) \in \mathbb{R}^2$ is an *edge vertex* of $f(x, y)$ if the function

$$(x, y) \mapsto |\nabla f(x, y)| = \sqrt{(\partial_x f(x, y))^2 + (\partial_y f(x, y))^2}$$

has a local maximum in (x_0, y_0) when running through this point in the direction $(\nabla f)(x_0, y_0)$ of steepest ascent/descent, formally:

$$|\nabla f [(x_0, y_0) + \varepsilon \cdot \nabla f(x_0, y_0)]| \leq |\nabla f(x_0, y_0)| \quad \text{for } \varepsilon \approx 0$$

► Discretization and approximation of the gradient

► Discretized function

$$A = [a_{p,q}]_{\substack{1 \leq p \leq m \\ 1 \leq q \leq n}} \quad \text{where } a_{p,q} = f(\xi_{p,q})$$

► Approximation of the gradient

$$D^x = [d_{p,q}^x]_{\substack{1 \leq p \leq m \\ 1 \leq q \leq n}} \quad \text{where } d_{p,q}^x \approx (\partial_x f)(\xi_{p,q})$$

$$D^y = [d_{p,q}^y]_{\substack{1 \leq p \leq m \\ 1 \leq q \leq n}} \quad \text{where } d_{p,q}^y \approx (\partial_y f)(\xi_{p,q})$$

computed using

- $A \mapsto D^x$: "derivation" in x -direction, smoothing in y -direction
- $A \mapsto D^y$: "derivation" in y -direction, smoothing in x -direction

- ▶ Example: The SOBEL operators

$$S_h : A \mapsto D^x = A \star \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$S_v : A \mapsto D^y = A \star \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

- ▶ Written as Kronecker products:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

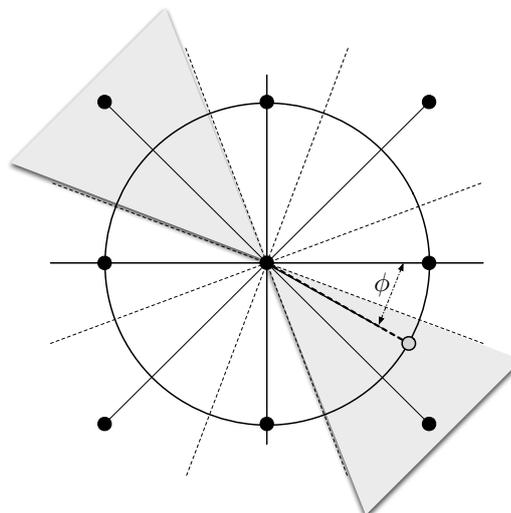
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

- ▶ From rectangular coordinates to polar coordinates

$$\langle x, y \rangle \mapsto \langle r, \phi \rangle = [\sqrt{x^2 + y^2}, \arctan(y/x)]$$

where $-\pi < \arctan(y/x) \leq \pi$

- ▶ Discretizing the directions



- ▶ The 8 neighbors $\langle \tilde{p}, \tilde{q} \rangle$ of a vertex $\langle p, q \rangle$ with integer coordinates are

$$\langle \tilde{p}, \tilde{q} \rangle = \left\{ \begin{array}{l} \langle p \pm 1, q \pm 1 \rangle \\ \langle p \pm 1, 0 \rangle \\ \langle 0, q \pm 1 \rangle \end{array} \right\} = \langle p, q \rangle + \delta$$

$$\text{where } \delta \in \Delta = \{ \langle \pm 1, \pm 1 \rangle, \langle \pm 1, 0 \rangle, \langle 0, \pm 1 \rangle \}$$

- ▶ They define 8 conic sectors rooted in $\langle p, q \rangle$ and of 45° angular width, symmetric w.r.t. to the respective straight lines joining $\langle p, q \rangle \longleftrightarrow \langle \tilde{p}, \tilde{q} \rangle$
- ▶ For an integer vertex $\langle p, q \rangle$ any direction $[1, \phi]$ defines a unique sector containing the straight line with direction $[1, \phi]$ rooted in $\langle p, q \rangle$, hence a neighbor $\langle \tilde{p}, \tilde{q} \rangle$ and the vector

$$\delta_{p,q}(\phi) = \langle \tilde{p}, \tilde{q} \rangle - \langle p, q \rangle = \langle \tilde{p} - p, \tilde{q} - q \rangle \in \Delta$$

- ▶ From the gradient matrices

$$D^x = [d_{p,q}^x] \quad D^y = [d_{p,q}^y]$$

one obtains

- ▶ the matrix of absolute length of the gradient vectors

$$R = [r_{p,q}] = \left[\sqrt{(d_{p,q}^x)^2 + (d_{p,q}^y)^2} \right]$$

- ▶ the matrix of discretized gradient directions

$$S = [\delta_{p,q}(\phi)] \quad \text{where } \phi = \arctan(d_{p,q}^y/d_{p,q}^x)$$

- ▶ Given f, A, D^x, D^y, R, S one defines

- ▶ $\langle p, q \rangle$ is an *edge-candidate* if

$$r_{p,q} = \max\{ r_{p,q}, r_{p,q} \pm \delta_{p,q}(\phi) \}$$

- ▶ $\langle p, q \rangle$ is a *level- λ edge vertex* (for $\lambda \in [0, 1]$) if

$$r_{p,q} = \max\{ r_{p,q}, r_{p,q} \pm \delta_{p,q}(\phi) \} \quad \text{and} \quad r_{p,q} \geq \lambda \cdot \max_{p',q'} r_{p',q'}$$

$$\text{(or } r_{p,q} \geq \lambda \cdot \text{aver}_{p',q'} r_{p',q'} \text{)}$$

- ▶ Two-level method with $0 < \lambda_{low} < \lambda_{high} \leq 1$:

- ▶ $\langle p, q \rangle$ is a *strong edge vertex* if it is a level- λ_{high} edge vertex
- ▶ $\langle p, q \rangle$ is a *weak edge vertex* if it is a level- λ_{low} edge vertex, but not a strong one
- ▶ Weak edge vertices are iteratively turned into strong edge vertices if they are neighbors of strong edge vertices

- ▶ For a 1D wavelet function $\psi(x)$ let

$$\Psi(x, y) = \psi(x) \psi(y)$$

be the 2D separable wavelet function constructed from it

- ▶ The 2D CWT of a function $f(x, y)$ is defined as

$$f^\Psi(a, b, s) = \frac{1}{s} \iint_{x,y \in \mathbb{R} \times \mathbb{R}} f(x, y) \Psi\left(\frac{x-a}{s}, \frac{y-b}{s}\right) dx dy$$

- ▶ As in the 1D case, let $\psi(x) = \frac{d}{dx} \theta(x)$ be the derivative of a “smoothing function” $\theta(x)$
- ▶ The 2D separable smoothing function constructed from $\theta(x)$ is

$$\Theta(x, y) = \theta(x) \theta(y)$$

- ▶ The 2D partial wavelet functions are

$$\Psi^x(x, y) = \psi(x) \theta(y) = \partial_x \Theta(x, y)$$

$$\Psi^y(x, y) = \theta(x) \psi(y) = \partial_y \Theta(x, y)$$

- ▶ The 2D partial continuous wavelet transform (CWT) is defined as

$$f^{\Psi^x}(a, b, s) = \frac{1}{s} \iint f(x, y) \Psi^x\left(\frac{x-a}{s}, \frac{y-b}{s}\right) dx dy$$

$$f^{\Psi^y}(a, b, s) = \frac{1}{s} \iint f(x, y) \Psi^y\left(\frac{x-a}{s}, \frac{y-b}{s}\right) dx dy$$

- ▶ Defining a smoothed version of f using an s -scaling of Θ

$$f^\Theta(a, b, s) = \iint_{x, y \in \mathbb{R} \times \mathbb{R}} f(x, y) \Theta\left(\frac{x-a}{s}, \frac{y-b}{s}\right) dx dy$$

one has

$$\begin{bmatrix} -f^{\Psi^x}(a, b, s) \\ -f^{\Psi^y}(a, b, s) \end{bmatrix} = \begin{bmatrix} \partial_a f^\Theta(a, b, s) \\ \partial_b f^\Theta(a, b, s) \end{bmatrix} = \nabla f^\Theta(a, b, s)$$

- ▶ This shows how to compute gradient values of f^Θ using the 2D CWT

- ▶ Assume that 1D scaling, wavelet functions are described by

$$\phi(x) = \sqrt{2} \sum_k h_k \phi(2x - k) \quad \psi(x) = \sqrt{2} \sum_k g_k \phi(2x - k)$$

- ▶ and that the smoothing function $\theta(x)$ also satisfies a scaling identity

$$\theta(x/2) = \sqrt{2} \sum_\ell r_\ell \theta(x - \ell/2)$$

- ▶ A scaling function $\Phi^x(x, y)$ for the wavelet function $\Psi^x(x, y) = \psi(x) \theta(y)$ can be defined as

$$\Phi^x(x, y) = \phi(x) \theta(y/2)$$

which then satisfies a 2D scaling identity

$$\Phi^x(x, y) = 2 \sum_{k,\ell} h_k r_\ell \Phi^x(2x - k, 2y - \ell)$$

- ▶ The 2D wavelet identity for $\Psi^x(x, y)$ is simply

$$\Psi^x(x, y) = \sqrt{2} \sum_k g_k \Phi^x(2x - k, 2y) = 2 \sum_{k,\ell} g_k \epsilon_\ell \Phi^x(2x - k, 2y - \ell)$$

with $\epsilon_\ell = \frac{1}{\sqrt{2}} \delta_{\ell,0}$. Similarly for $\Phi^y(x, y)$ and $\Psi^y(x, y)$

- ▶ Example: The HAAR wavelet function $\psi_{haar}(t)$ is the derivative of the smoothing function $\theta(t) = \Delta(t)$:

$$\psi_{haar}(t) = \frac{d}{dt} \Delta(t) \quad \text{where} \quad \Delta(t) = \begin{cases} t & 0 \leq t \leq 1/2 \\ 1 - t & 1/2 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

- ▶ The function $\Delta(t)$ satisfies

$$\Delta(x) + 2 \Delta(x - 1/2) + \Delta(x - 1) = 2\Delta(x/2)$$

- ▶ which can be written as a scaling equation

$$\Delta(x) = \frac{1}{2} (\Delta(2x) + 2 \Delta(2x - 1) + \Delta(2x - 2))$$

- ▶ so that

$$\mathbf{r} = \frac{1}{2\sqrt{2}} \langle 1, 2, 1 \rangle$$

is a B-spline filter

- Approximation and detail coefficients are defined as usual

$$a_{m;k,\ell}^x = \langle f | \Phi_{2^m,k,\ell}^x \rangle = \iint f(x,y) \frac{1}{2^m} \Phi^x\left(\frac{x-k}{2^m}, \frac{y-\ell}{2^m}\right) dx dy$$

$$d_{m;k,\ell}^x = \langle f | \Psi_{2^m,k,\ell}^x \rangle = \iint f(x,y) \frac{1}{2^m} \Psi^x\left(\frac{x-k}{2^m}, \frac{y-\ell}{2^m}\right) dx dy$$

and analogously for $a_{m;k,\ell}^y$ and $d_{m;k,\ell}^y$

- Recursion formulas for the approximation coefficients

$$a_{m+1;p,q}^x = \sum_{k,\ell} h_k r_\ell a_{m;p+k2^m,q+\ell2^m}^x$$

$$a_{m+1;p,q}^y = \sum_{k,\ell} r_k h_\ell a_{m;p+k2^m,q+\ell2^m}^y$$

- Formulas for the detail coefficients

$$d_{m+1;p,q}^x = \sum_{k,\ell} g_k \epsilon_\ell a_{m;p+k2^m,q+\ell2^m}^x = \frac{1}{\sqrt{2}} \sum_k g_k a_{m;p+k2^m,q}^x$$

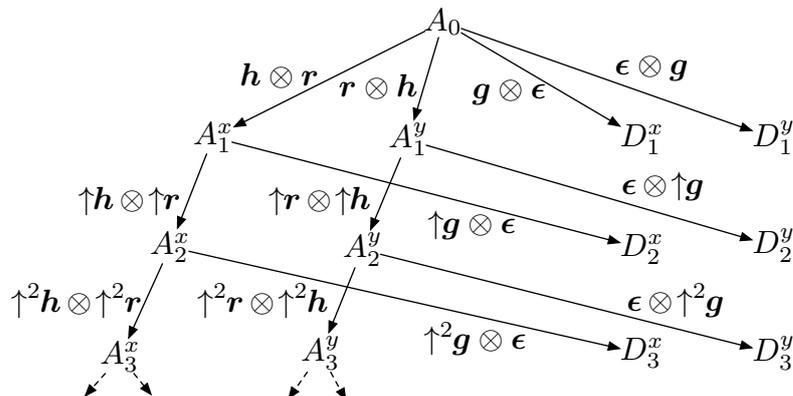
$$d_{m+1;p,q}^y = \sum_{k,\ell} \epsilon_k g_\ell a_{m;p+k2^m,q+\ell2^m}^y = \frac{1}{\sqrt{2}} \sum_k g_\ell a_{m;p,q+\ell2^m}^y$$

- Computational scheme (*à trous* algorithm)

$$A_m^x = \left[f^{\Phi^x}(2^m; p, q) \right]_{p,q} \quad A_m^y = \left[f^{\Phi^y}(2^m; p, q) \right]_{p,q}$$

$$D_m^x = \left[f^{\Psi^x}(2^m; p, q) \right]_{p,q} \quad D_m^y = \left[f^{\Psi^y}(2^m; p, q) \right]_{p,q}$$

where $A_0 = A_0^x = A_0^y = [f(p, q)]_{p,q}$



Edge detection using the Sobel operators

The Sobel operators

```
In[13]:= IA = ImageAdjust;
```

Defining the horizontal and vertical Sobel operators

```
In[14]:= sobelh = KroneckerProduct[{1, 2, 1}, {-1, 0, 1}]; sobelh // MatrixForm
```

Out[14]//MatrixForm=

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

```
In[15]:= sobelv = KroneckerProduct[{-1, 0, 1}, {1, 2, 1}]; sobelv // MatrixForm
```

Out[15]//MatrixForm=

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Applying the Sobel operators separately

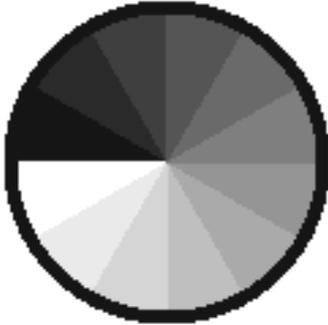
```
In[16]:= sobel[A_]:=  
Module[{h,v},  
AA=ArrayPad[A,{1,1},"Extrapolated"];  
h=ListConvolve[sobelh,AA];  
v=ListConvolve[sobelv,AA];  
{h,v}  
]
```

A test image

```
In[17]:= circle7 = Import["~/LEHRE/Wavelets-All/WTBV-15/CWT-Edges/circle7.m"];
```

```
In[18]:= img = Image[circle7, ImageSize → Small]
```

Out[18]=

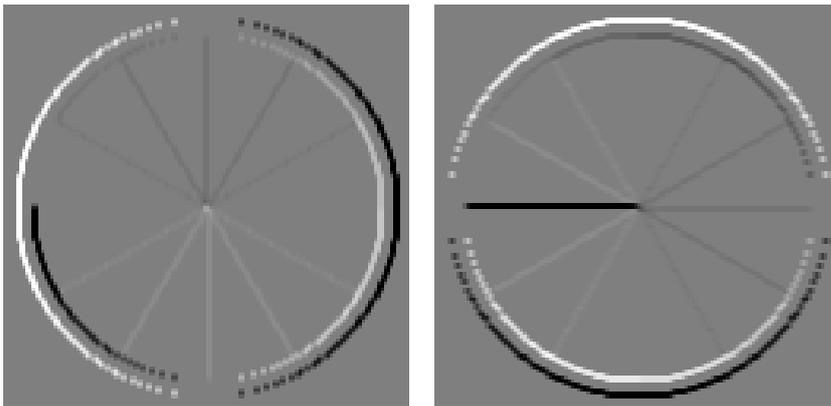


```
In[19]:= {Dx, Dy} = sobel[circle7];
```

Applying the horizontal and the vertical Sobel filters separately

```
In[20]:= GraphicsRow[{IA[Image[Dx]], IA[Image[Dy]]}, ImageSize → Scaled[0.7]]
```

Out[20]=



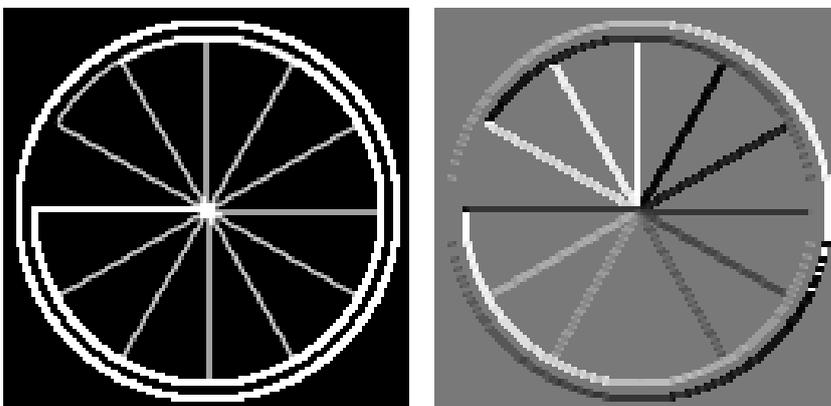
The size and direction of the approximate gradients obtained from the previous data

```
In[21]:= W = Dx + I * Dy;
```

```
In[22]:= {R, S} = {Abs[W], Arg[W]};
```

```
In[23]:= GraphicsRow[{IA[Image[R], {1, 0.7}], IA[Image[S]]}, ImageSize → Scaled[0.7]]
```

Out[23]=



Encoding the gradient

Hue colors

```
In[24]:= Graphics[Table[{Hue[s], EdgeForm[Gray], Rectangle[{20 s, 0}]}, {s, 0, 1, 1/20}]]
```



```
In[25]:= Graphics[Table[{Hue[s], EdgeForm[Gray], Rectangle[{10 s, 0}]}, {s, 0, 2, 1/10}]]
```



Visualizing the gradient

```
In[26]:= grd[f_, x_, y_] := Sqrt[Total[(∇_{x,y} f)^2]]
```

```
In[27]:= f[x_, y_] = Sin[x^2 * y]
```

Out[27]= $\text{Sin}[x^2 y]$

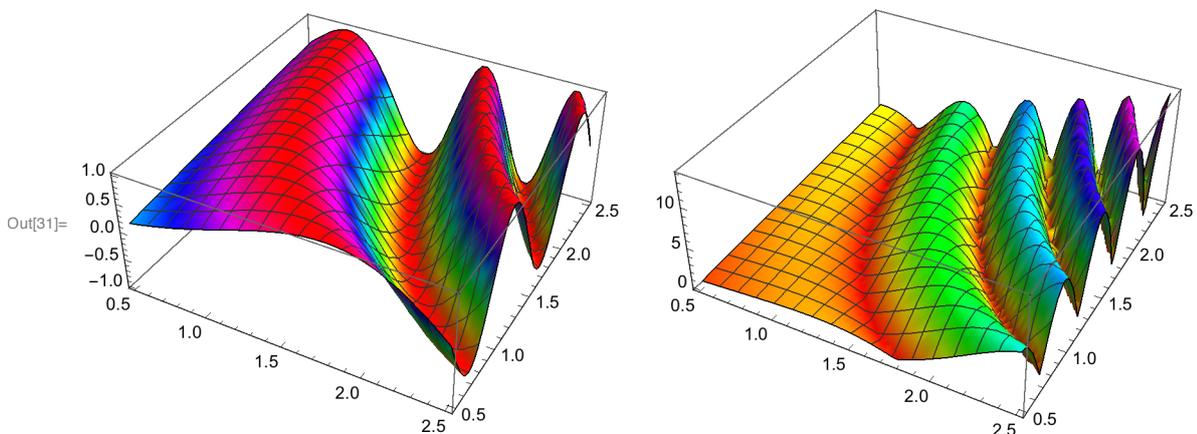
```
In[28]:= g[x_, y_] = grd[f[x, y], x, y]
```

Out[28]= $\sqrt{x^4 \text{Cos}[x^2 y]^2 + 4 x^2 y^2 \text{Cos}[x^2 y]^2}$

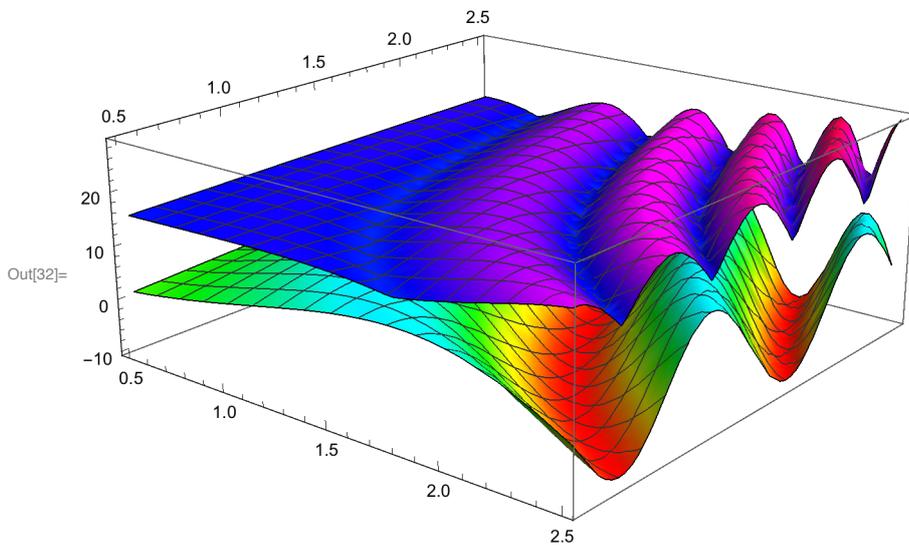
```
In[29]:= pf = Plot3D[f[x, y], {x, 0.5, 2.5}, {y, 0.5, 2.5},
  ColorFunction -> Function[{x, y, z}, Hue[z]], ColorFunctionScaling -> True];
```

```
In[30]:= pg = Plot3D[g[x, y], {x, 0.5, 2.5}, {y, 0.5, 2.5},
  ColorFunction -> Function[{x, y, z}, Hue[z]], ColorFunctionScaling -> True];
```

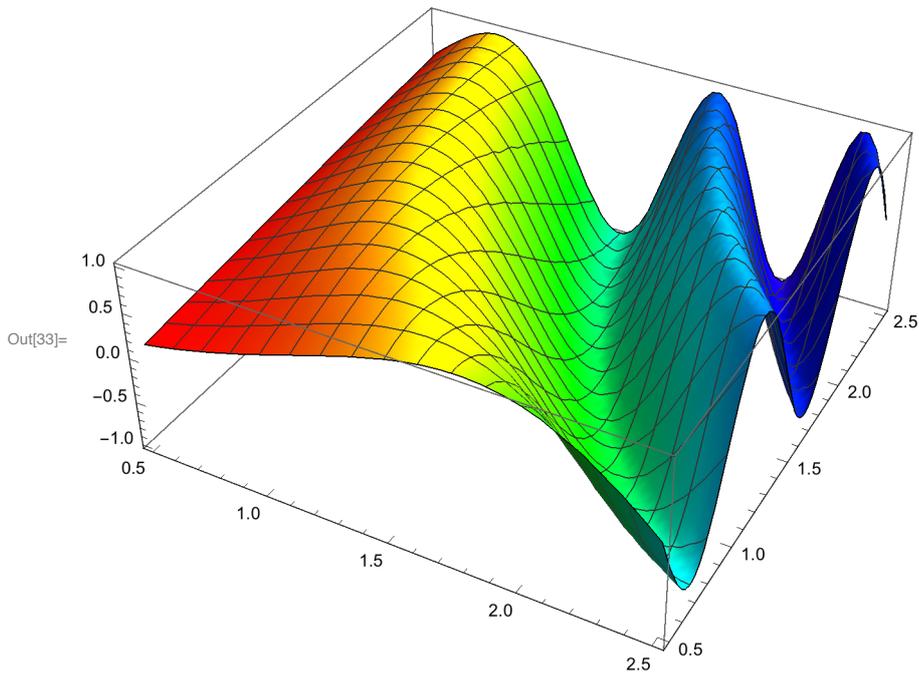
```
In[31]:= GraphicsRow[{pf, pg}, ImageSize -> Large]
```



```
In[32]:= Plot3D[{10 f[x, y], g[x, y] + 15}, {x, 0.5, 2.5}, {y, 0.5, 2.5},
  ColorFunction -> Function[{x, y, z}, Hue[z]],
  ColorFunctionScaling -> True, ImageSize -> Scaled[0.7]]
```



```
In[33]:= Plot3D[f[x, y], {x, 0.5, 2.5}, {y, 0.5, 2.5},
  ColorFunction -> Function[{x, y, z}, Hue[g[x, y] / 2]],
  ColorFunctionScaling -> True, ImageSize -> Scaled[0.7]]
```



Discretizing directions

```
In[34]:= ddir2vec[a_]:=Module[{aa},
aa=Mod[a,2 Pi,-Pi];
Which[
Abs[aa]≤Pi/8,{1,0},
Pi/8<aa≤3 Pi/8,{1,1},
3 Pi/8<aa≤5 Pi/8,{0,1},
5 Pi/8<aa≤7 Pi/8,{-1,1},
-7 Pi/8<aa<-5 Pi/8,{-1,-1},
-5 Pi/8<aa≤-3 Pi/8,{0,-1},
-3 Pi/8 <aa ≤-Pi/8,{1,-1},
Abs[aa]≥ 7 Pi/8,{-1,0},
True,{0,0}]
]
```

Direction vectors for angles which are multiples of 36 degrees

```
In[35]:= Transpose[Table[{k * Pi / 5, ddir2vec[k * Pi / 5] // MatrixForm}, {k, 0, 9}]] //
MatrixForm
```

```
Out[35]//MatrixForm=

$$\begin{pmatrix} 0 & \frac{\pi}{5} & \frac{2\pi}{5} & \frac{3\pi}{5} & \frac{4\pi}{5} & \pi & \frac{6\pi}{5} & \frac{7\pi}{5} & \frac{8\pi}{5} & \frac{9\pi}{5} \\ \begin{pmatrix} 1 \\ 0 \end{pmatrix} & \begin{pmatrix} 1 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} 0 \\ 1 \end{pmatrix} & \begin{pmatrix} -1 \\ 1 \end{pmatrix} & \begin{pmatrix} -1 \\ 0 \end{pmatrix} & \begin{pmatrix} -1 \\ -1 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} & \begin{pmatrix} 0 \\ -1 \end{pmatrix} & \begin{pmatrix} 1 \\ -1 \end{pmatrix} \end{pmatrix}$$

```

```
In[36]:= dirlist = {{1, 0}, {1, 1}, {0, 1}, {-1, 1}, {-1, 0}, {-1, -1}, {0, -1}, {1, -1}}
```

```
Out[36]= {{1, 0}, {1, 1}, {0, 1}, {-1, 1}, {-1, 0}, {-1, -1}, {0, -1}, {1, -1}}
```

```
In[37]:= color =
<|Join[Table[dirlist[[i]] → Hue[(i - 1) / 8], {i, 1, 8}], {{0, 0} → White}]|>
```

```
Out[37]= <|{1, 0} → ■, {1, 1} → ■, {0, 1} → ■, {-1, 1} → ■,
{-1, 0} → ■, {-1, -1} → ■, {0, -1} → ■, {1, -1} → ■, {0, 0} → □|>
```

```

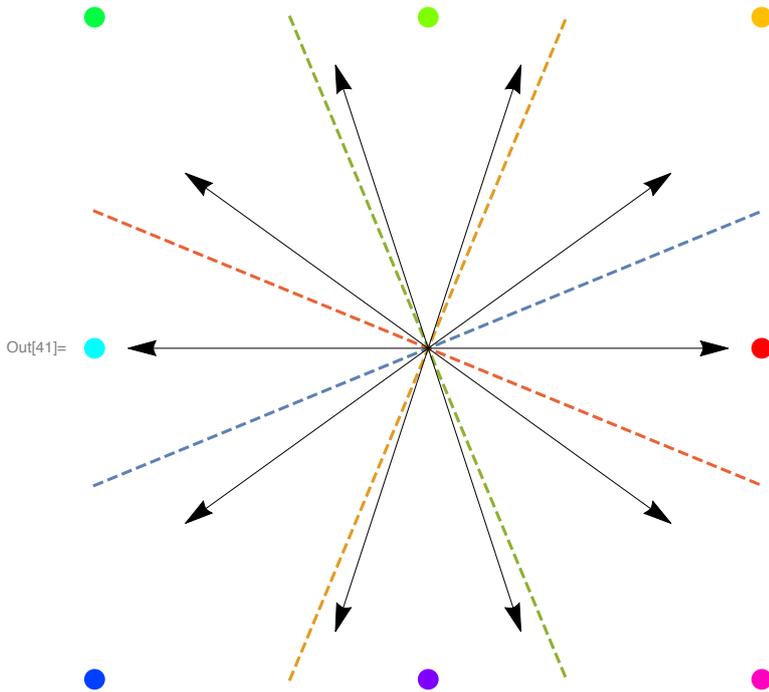
In[38]:= p0 = Graphics[
  Map[{PointSize[0.03], color[#], Point[#]}
    &, dirlist] ];

In[39]:= p1 = Plot[
  {x * Tan[Pi/8], x * Tan[3 Pi/8], x * Tan[5 Pi/8], x * Tan[7 Pi/8]}, {x, -1, 1},
  AspectRatio -> 1, PlotRange -> {-1, 1}, PlotStyle -> Dashed];

In[40]:= p2 =
  Graphics[Table[Arrow[{0, 0}, 0.9 * {Cos[k * Pi/5], Sin[k * Pi/5]}], {k, 0, 9}]];

In[41]:= Show[{p0, p1, p2}, Background -> White]

```



Computing absolute values and discretized directions of the gradients of an array (discretized image) using the Sobel operators

```

In[42]:= ggrad[A_]:=
Module[{AA,val,dir,m,n,h,v,hh,vv,dv},
  {m,n}=Dimensions[A];
  AA=ArrayPad[A,{1,1},"Extrapolated"];
  h=ListConvolve[sobelh,AA];
  v=ListConvolve[sobelv,AA];
  val=Table[0,{x,1,m},{y,1,n}];
  dir=Table[0,{x,1,m},{y,1,n}];
  Do[
    {hh,vv}={h[[x,y]],v[[x,y]]};
    val[[x,y]]=N[Sqrt[hh^2+vv^2]];
    dir[[x,y]]=
    If[
      hh==0&&vv==0,
      {0,0},
      dv=-ddir2vec[N[ArcTan[hh,vv]]];
      {dv[[1]],dv[[2]]},
      {x,1,m},{y,1,n}];
    {val,dir}
  ]

```

Plotting the gradients of an array as a vector field

```

In[43]:= ggradplot[A_]:=
Module[{AA,val,dir,p1,p2,p3,q,m,n,h,v,hh,vv,dv},
{m,n}=Dimensions[A];
AA=ArrayPad[A,{1,1},"Extrapolated"];
h=ListConvolve[sobelh,AA];
v=ListConvolve[sobelv,AA];
p1=ListVectorPlot[
Table[Table[{h[[x,y]],v[[x,y]]},{x,1,m}},{y,1,n}],VectorPoints→All];
val=Table[Table[0,{y,1,n}},{x,1,m}];
dir=Table[Table[0,{y,1,n}},{x,1,m}];
Do[
{hh,vv}={h[[x,y]],v[[x,y]]};
val[[x,y]]=N[Sqrt[hh^2+vv^2]];
dir[[x,y]]=
If[
hh==0&&vv==0,
{0,0},
dv=ddir2vec[N[ArcTan[hh,vv]]];
- { dv[[1]],dv[[2]]},
{x,1,m},{y,1,n}];
p2=ArrayPlot[val];
q=3*val*dir;
p3 =ListVectorPlot[ Transpose[q],VectorPoints→All];
p3
]

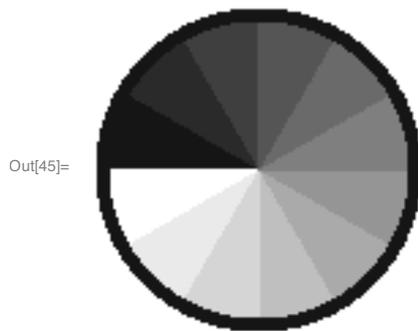
```

The gradient encoding

In[44]:= `color`

Out[44]:= `<| {1, 0} → ■, {1, 1} → ■, {0, 1} → ■, {-1, 1} → ■,
 {-1, 0} → ■, {-1, -1} → ■, {0, -1} → ■, {1, -1} → ■, {0, 0} → □ |>`

In[45]:= `img`

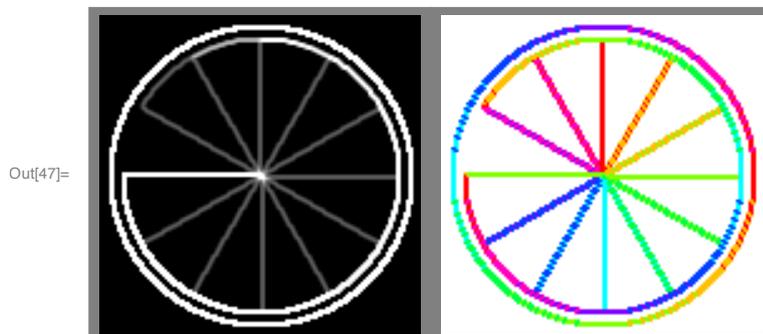


In[46]:= `imggrad = ggrad[circle7];`

The gradient given as (value,direction)-pairs with discretized directions

In[47]:= `GraphicsRow[`

`{Image[imggrad[[1]]], Image[(imggrad[[2]] /. color)]}, Background → Gray]`



Examples for gradients

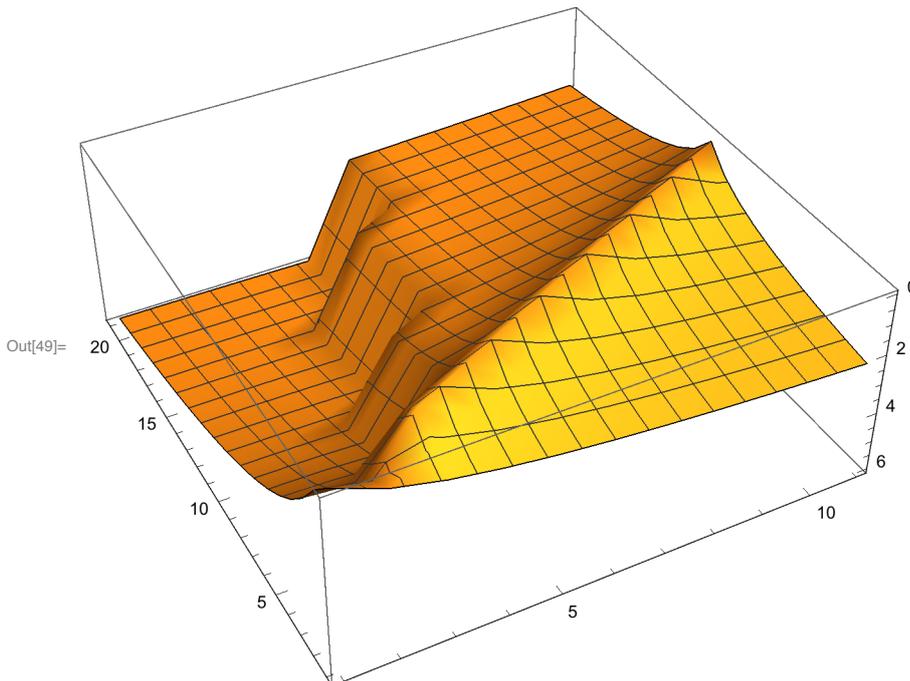
First example

```
In[48]:= A=Table[
  Piecewise[
    {{Log[1+Abs[x^2+y^2]],x^2<=y},
    {Log[1+Abs[x-y]],x^2>y}},
    {x,0,10},{y,0,20}]; A//MatrixForm
```

Out[48]//MatrixForm=

0	Log[2]	Log[5]	Log[10]	Log[17]	Log[26]	Log[37]	Log[50]	Log[65]	Log[80]
Log[2]	Log[3]	Log[6]	Log[11]	Log[18]	Log[27]	Log[38]	Log[51]	Log[66]	Log[81]
Log[3]	Log[2]	0	Log[2]	Log[21]	Log[30]	Log[41]	Log[54]	Log[69]	Log[84]
Log[4]	Log[3]	Log[2]	0	Log[2]	Log[3]	Log[4]	Log[5]	Log[6]	Log[9]
Log[5]	Log[4]	Log[3]	Log[2]	0	Log[2]	Log[3]	Log[4]	Log[5]	Log[6]
Log[6]	Log[5]	Log[4]	Log[3]	Log[2]	0	Log[2]	Log[3]	Log[4]	Log[5]
Log[7]	Log[6]	Log[5]	Log[4]	Log[3]	Log[2]	0	Log[2]	Log[3]	Log[4]
Log[8]	Log[7]	Log[6]	Log[5]	Log[4]	Log[3]	Log[2]	0	Log[2]	Log[3]
Log[9]	Log[8]	Log[7]	Log[6]	Log[5]	Log[4]	Log[3]	Log[2]	0	Log[2]
Log[10]	Log[9]	Log[8]	Log[7]	Log[6]	Log[5]	Log[4]	Log[3]	Log[2]	0
Log[11]	Log[10]	Log[9]	Log[8]	Log[7]	Log[6]	Log[5]	Log[4]	Log[3]	Log[2]

```
In[49]:= ListPlot3D[A, ImageSize -> Scaled[0.7]]
```



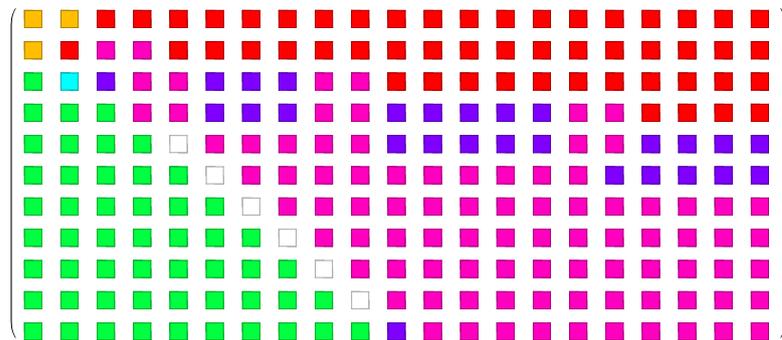
In[50]:= `ggrad[A][[1]] // MatrixForm`

Out[50]//MatrixForm=

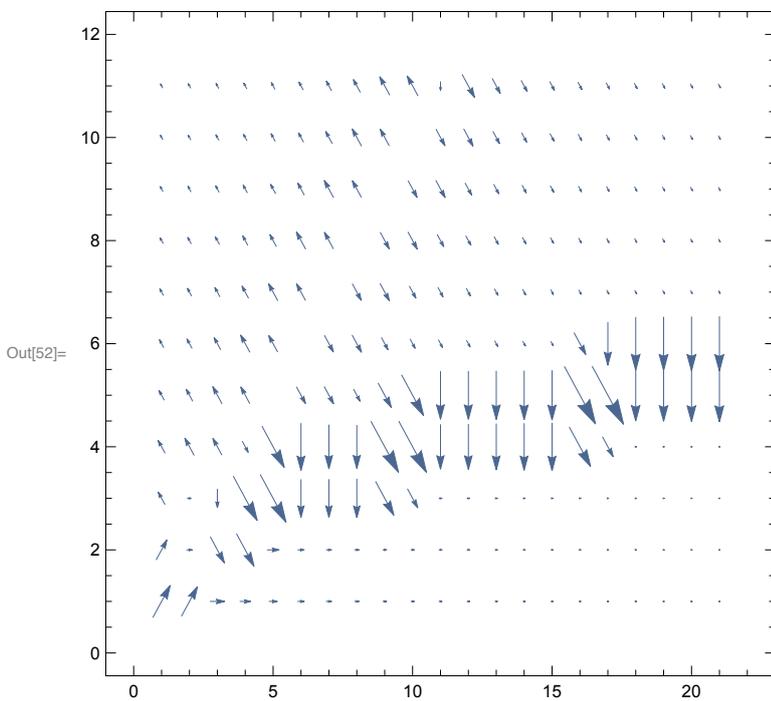
7.84207	7.26777	6.66637	4.97011	3.85393	3.12709	2.62509	2.25977	1.98272	1.76
4.91314	2.75581	6.40468	7.94479	5.55832	3.00179	2.54943	2.21088	1.94942	1.74
3.09985	1.83706	4.5995	10.823	11.6516	9.26919	9.3163	9.50643	8.24755	4.69
3.15805	4.23661	4.0876	2.75193	10.0723	11.5159	10.7892	10.6062	12.4537	12.5
2.4472	2.99851	4.23661	4.0876	0.	4.0876	4.23661	2.99851	5.97247	9.95
2.00332	2.34509	2.99851	4.23661	4.0876	0.	4.0876	4.23661	2.99851	2.34
1.69778	1.93195	2.34509	2.99851	4.23661	4.0876	0.	4.0876	4.23661	2.99
1.47398	1.64494	1.93195	2.34509	2.99851	4.23661	4.0876	0.	4.0876	4.23
1.30274	1.43324	1.64494	1.93195	2.34509	2.99851	4.23661	4.0876	0.	4.08
1.16738	1.27035	1.43324	1.64494	1.93195	2.34509	2.99851	4.23661	4.0876	0.
1.07831	1.16738	1.30274	1.47398	1.69778	2.00332	2.4472	3.15805	4.53088	4.91

In[51]:= `ggrad[A][[2]] /. color // MatrixForm`

Out[51]//MatrixForm=



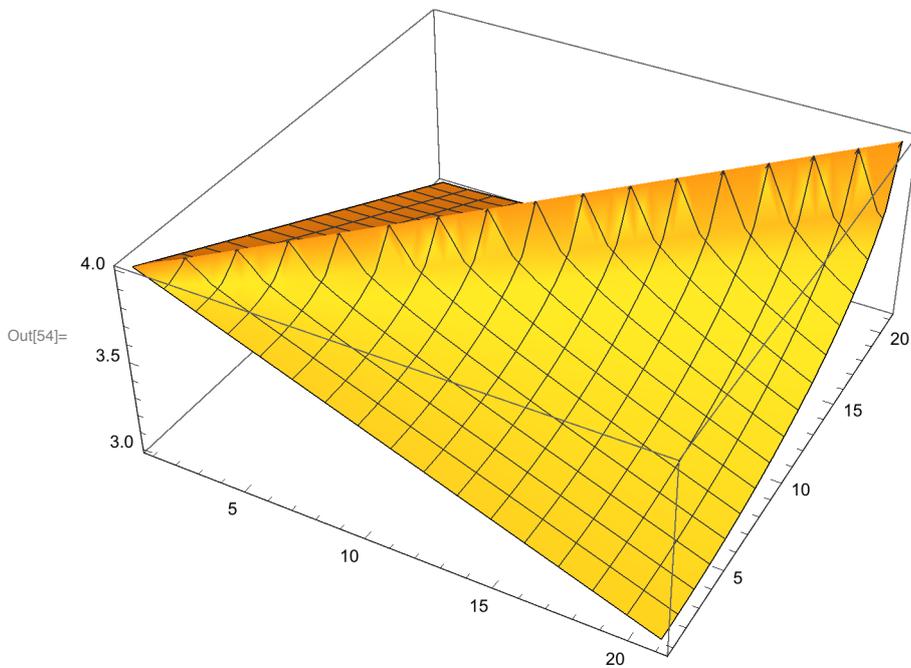
In[52]:= `Show[ggradplot[A]]`



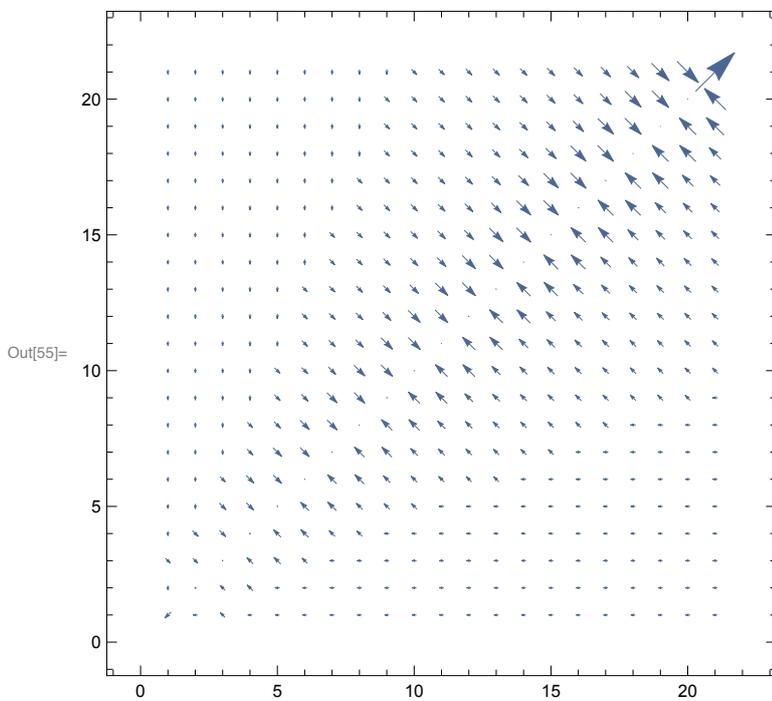
An example image with a sharp edge

```
In[53]:= fvals1 = Table[  
    4 - Sqrt[Abs[x^2 - y^2]], {x, 0, 1, 1/20}, {y, 0, 1, 1/20}];
```

```
In[54]:= ListPlot3D[fvals1, ImageSize -> Scaled[0.7]]
```



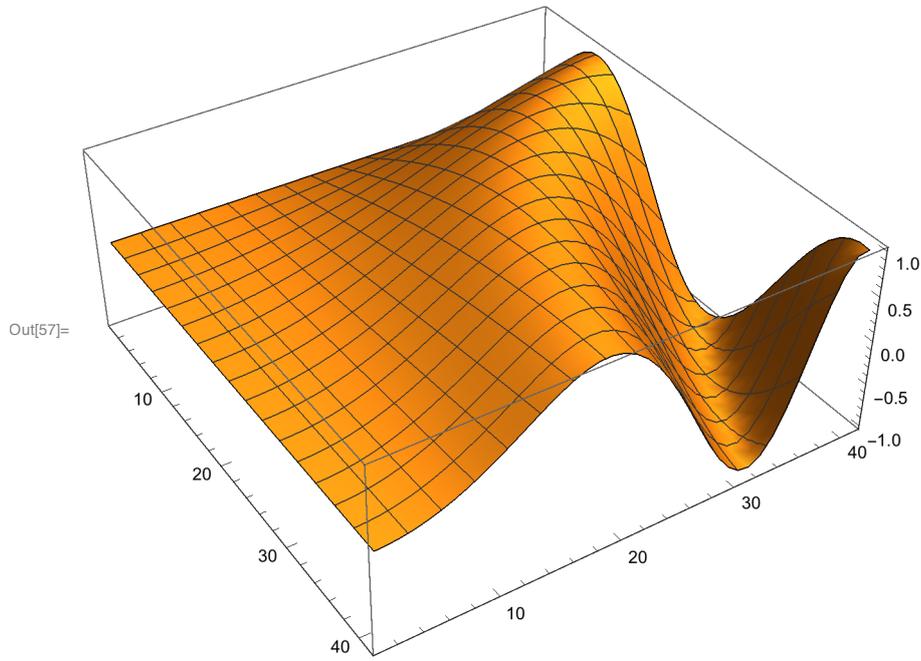
```
In[55]:= ggradplot[fvals1]
```



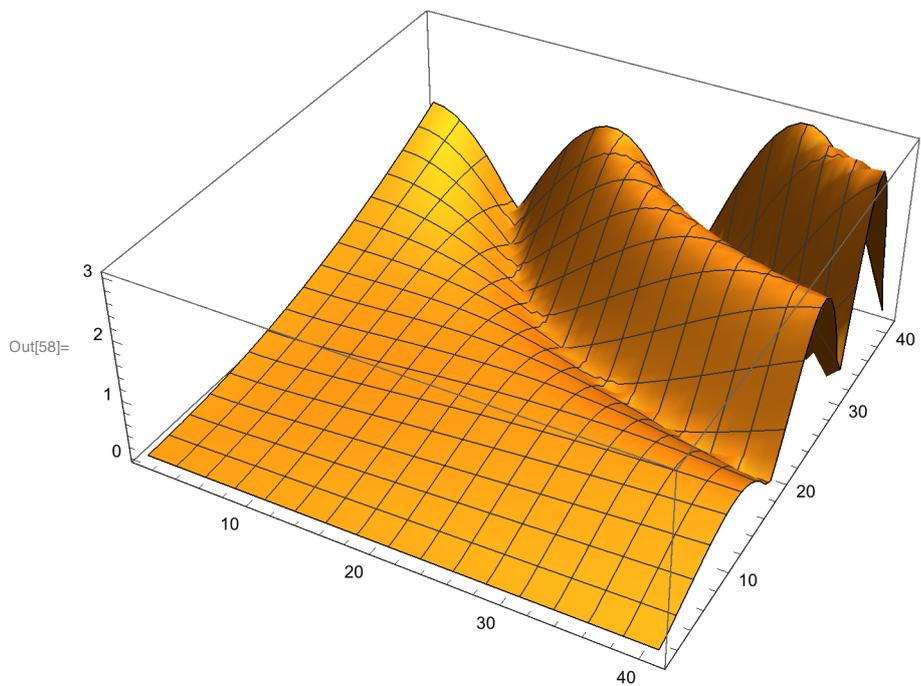
Another example image

```
In[56]:= fvals2 = Table[  
    Sin[x^2 * y], {x, 0, 2, 1/20}, {y, 0, 2, 1/20}];
```

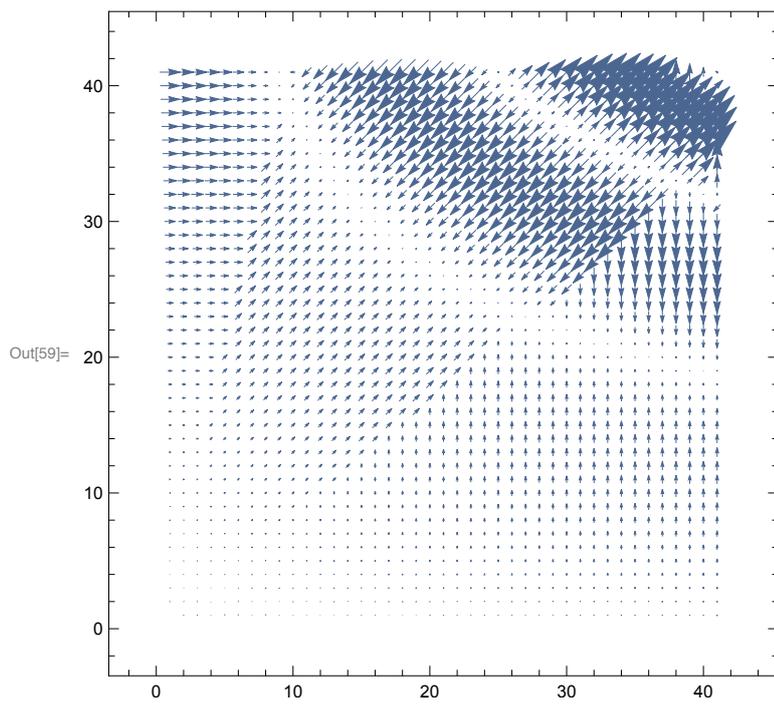
```
In[57]:= ListPlot3D[fvals2, ImageSize → Scaled[0.7]]
```



```
In[58]:= ListPlot3D[ggrad[fvals2][[1]], ImageSize → Scaled[0.7]]
```



```
In[59]:= ggradplot[fvals2]
```



Canny's algorithm

Identifying edge vertices of level λ in an array

```

In[60]:= edges[A_, level_] :=
Module[{m,n,val,dir,max,VVal,p,x1,y1,x2,y2,EC,v,v1,v2,x,y},
  {m,n}=Dimensions[A];
  {val,dir}=ggrad[A];
  max=Max[val];
  VVal=ArrayPad[val,{1,1},"Extrapolated"];
  EC=Table[0,{x,1,m},{y,1,n}];
  Do[
    {p={x+1,y+1};
    {x1,y1}=p+dir[[x,y]];
    {x2,y2}=p-dir[[x,y]];
    v=VVal[[x,y]];
    v1=VVal[[x1,y1]];
    v2=VVal[[x2,y2]];
    If[
      v>Max[v1,v2]&&v>level max,
      EC[[x,y]]=v,
      EC[[x,y]]=0}],
    {x,1,m},{y,1,n}];
  EC
]

```

Canny's algorithm using two levels and a prescribed number of iterations

```

In[61]:= canny[A_,low_,high_,iter_]:=
Module[{k,m,n,max,ALow,AAlow,Ahigh,AAhigh,Diff},
{m,n}=Dimensions[A];
max=Max[A];
ALow=Map[If[#<low max,0,1]&,A,{2}];
AAlow=ArrayPad[ALow,{1,1},"Extrapolated"];
Ahigh =Map[If[#<high max,0,1]&,A,{2}];
AAhigh=ArrayPad[Ahigh,{1,1},"Extrapolated"];
Print["strong edge vertices: ",Total[Flatten[Ahigh]]];
Print["weak edge vertices: ",Total[Flatten[ALow]]];
Print["further edge vertices: "];
For[k=1,k<iter,k++,
Diff=AAlow-AAhigh;
Do[
If[
Diff[[x,y]]==1,
Diff[[x,y]]=
Max[Take[AAhigh,{x-1,x+1},{y-1,y+1}]]
],
{x,2,m+1},{y,2,n+1}
];
Print["round ",k," : ",Total[Flatten[Diff]]];
AAhigh=AAhigh+Diff
];
Take[AAhigh,{2,m+1},{2,n+1}]
]

```

In[62]=

```

cannydemo[A_,low_,high_,iter_] :=
Module[{k,m,n,max,ALow,AAlow,Ahigh,AAhigh,Diff,out},
  {m,n}=Dimensions[A];
  max=Max[A];
  ALow=Map[ If[#<low max,0,1]&,A,{2}];
  AAlow=ArrayPad[ALow,{1,1} ];
  Ahigh =Map[ If[#<high max,0,1]&,A,{2}];
  AAhigh=ArrayPad[Ahigh,{1,1} ];
  Print["strong edge vertices: ",Total[Flatten[Ahigh]]];
  Print["weak edge vertices: ",Total[Flatten[ALow]]];
  out={Ahigh};
  Print["further edge vertices: "];
  For[k=1,k<=iter,k++,
  Diff=AAlow-AAhigh;
  Do[
  If[
  Diff[[x,y]]==1,
  Diff[[x,y]]=
  Max[Take[AAhigh,{x-1,x+1},{y-1,y+1}]]
  ],
  {x,2,m+1},{y,2,n+1}
  ];
  Print["round ",k," : ",Total[Flatten[Diff]]];
  AAhigh=AAhigh+Diff;
  AppendTo[out,Take[AAhigh,{2,m+1},{2,n+1}]]
  ];
  out
]

```

Examples for the Canny algorithm

First example

```
In[63]:= img1 = Import["~/LEHRE/Wavelets-All/WTBV-10/CWT/camera.jpg"]
```

```
Out[63]=
```



```
In[64]:= img1 = ColorConvert[img1, "Grayscale"]
```

```
Out[64]=
```



```
In[65]:= ImageDimensions[img1]
```

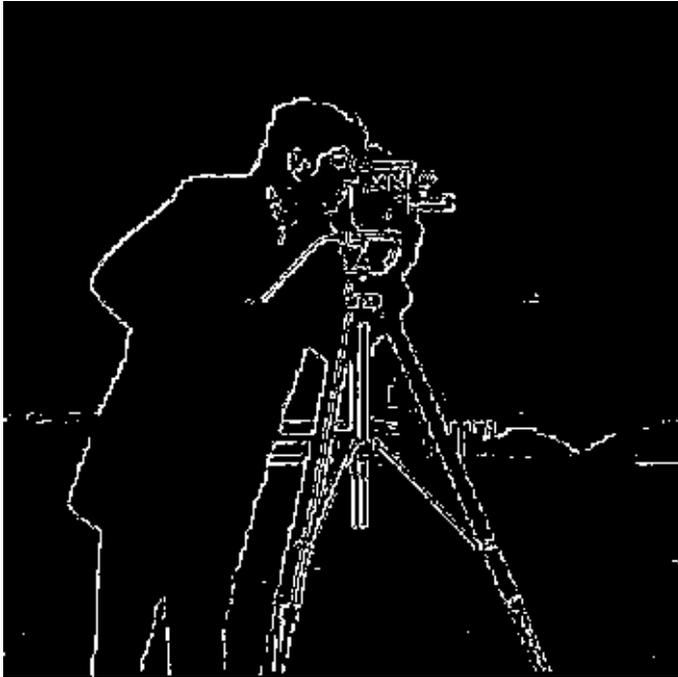
```
Out[65]= {353, 352}
```

```
In[66]:= data1 = ImageData[img1];
```

```
In[67]:= edge1 = edges[data1, 0.15];
```

```
In[68]:= Image[edge1]
```

```
Out[68]=
```



```
In[69]:= cout1 = cannydemo[edge1, 0.2, 0.5, 10];
```

```
strong edge vertices: 2434
```

```
weak edge vertices: 4492
```

```
further edge vertices:
```

```
round 1 : 1101
```

```
round 2 : 229
```

```
round 3 : 112
```

```
round 4 : 62
```

```
round 5 : 39
```

```
round 6 : 18
```

```
round 7 : 10
```

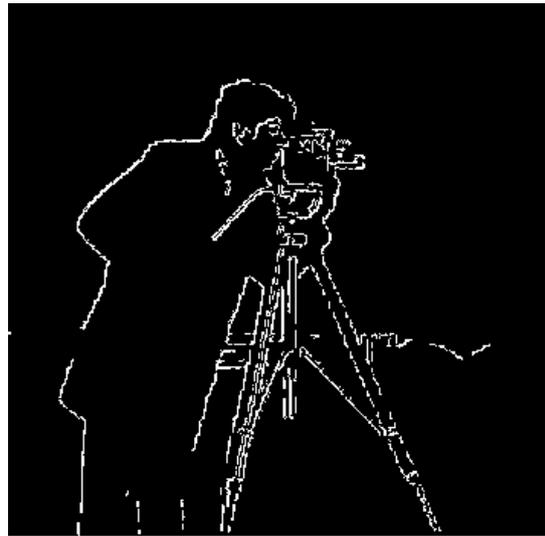
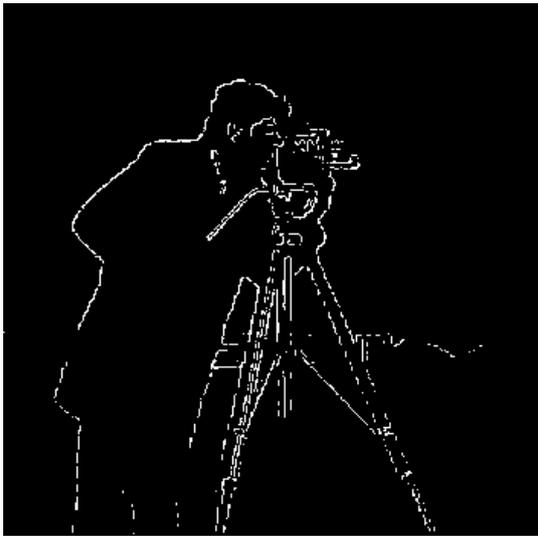
```
round 8 : 6
```

```
round 9 : 5
```

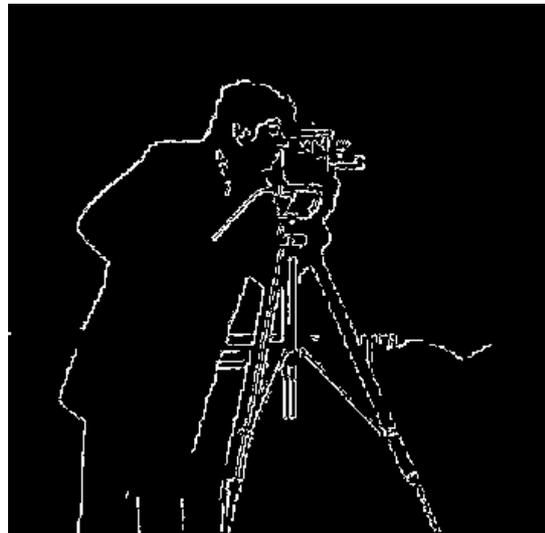
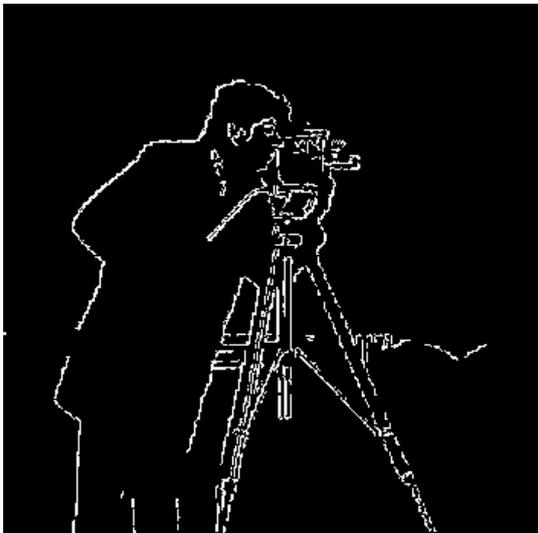
```
round 10 : 5
```

```
In[70]:= Animate[Image[cout1[[k]]], {k, 1, Length[cout1], 1}, AnimationRunning → False]
```

```
In[71]= GraphicsGrid[{{Image[cout1[[1]]], Image[cout1[[3]]]},  
  {Image[cout1[[5]]], Image[cout1[[10]]]}}, ImageSize -> Large]
```

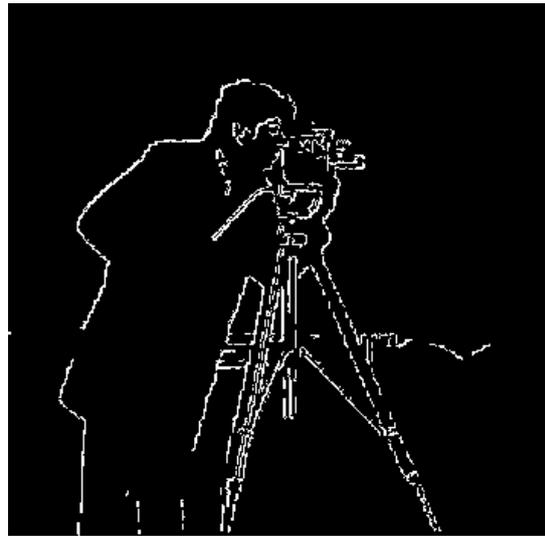
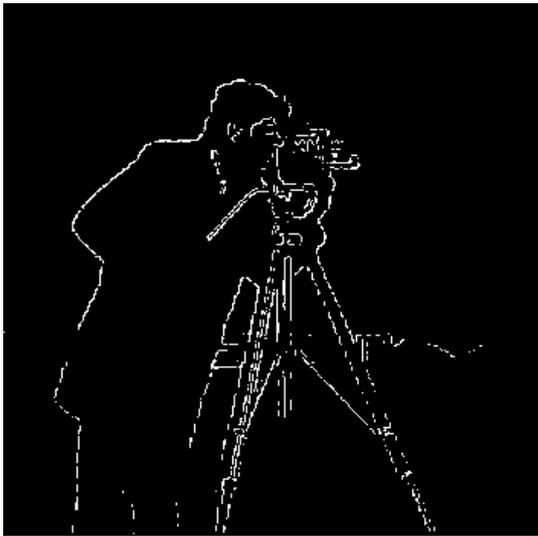


Out[71]=

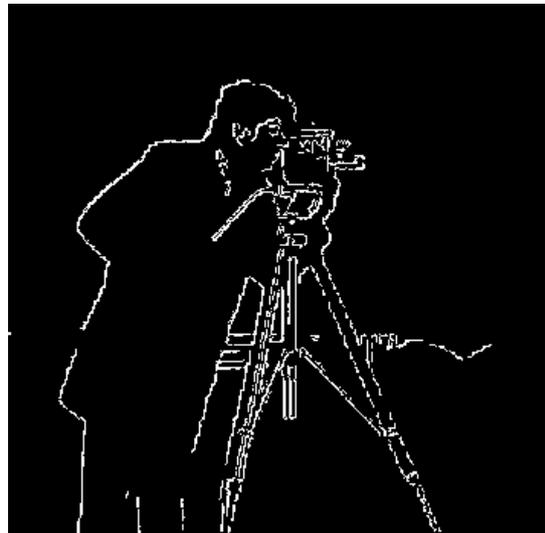
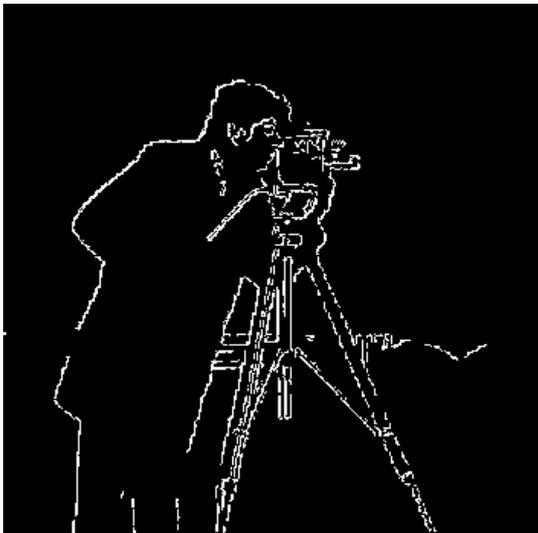


```
In[72]:= cout2 = cannydemo[edge1, 0.3, 0.5, 10];  
strong edge vertices: 2434  
weak edge vertices: 4492  
further edge vertices:  
round 1 : 1101  
round 2 : 229  
round 3 : 112  
round 4 : 62  
round 5 : 39  
round 6 : 18  
round 7 : 10  
round 8 : 6  
round 9 : 5  
round 10 : 5  
  
In[73]:= Animate[Image[cout2[[k]]], {k, 1, Length[cout2], 1}, AnimationRunning -> False]
```

```
In[74]= GraphicsGrid[{{Image[cout2[[1]]], Image[cout2[[3]]]},  
  {Image[cout2[[5]]], Image[cout2[[10]]]}}, ImageSize -> Large]
```



Out[74]=



```
In[75]:= cout3 = cannydemo[edge1, 0.15, 0.7, 10];
```

```
strong edge vertices: 684
```

```
weak edge vertices: 4492
```

```
further edge vertices:
```

```
round 1 : 823
```

```
round 2 : 367
```

```
round 3 : 193
```

```
round 4 : 122
```

```
round 5 : 97
```

```
round 6 : 60
```

```
round 7 : 48
```

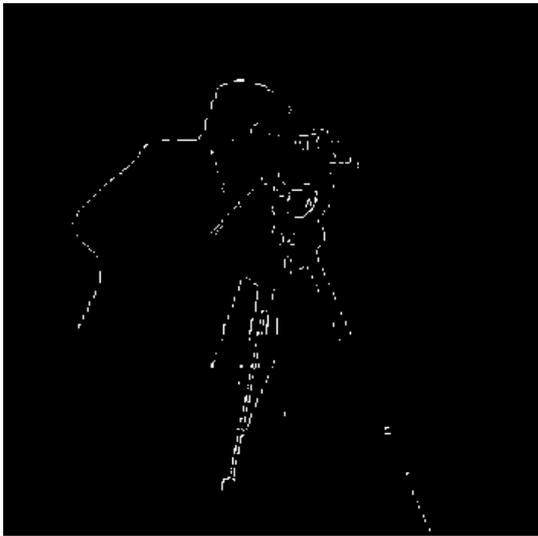
```
round 8 : 41
```

```
round 9 : 41
```

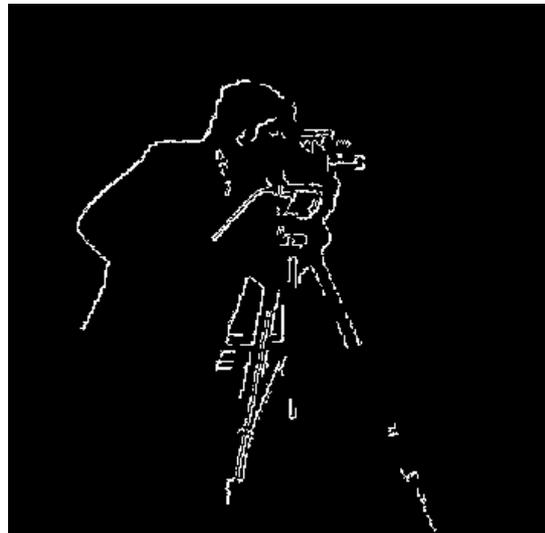
```
round 10 : 33
```

```
In[76]:= Animate[Image[cout3[[k]]], {k, 1, Length[cout3], 1}, AnimationRunning → False]
```

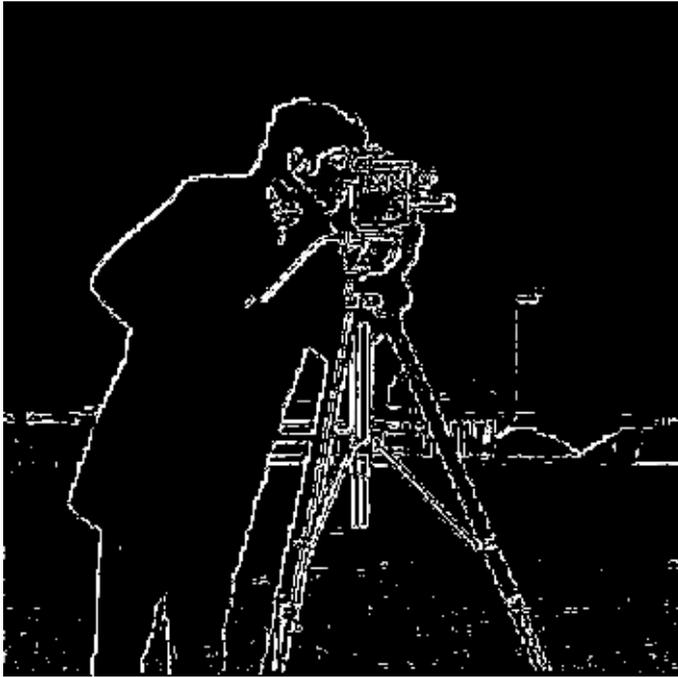
```
In[77]= GraphicsGrid[{{Image[cout3[[1]]], Image[cout3[[3]]]},  
  {Image[cout3[[5]]], Image[cout3[[10]]]}}, ImageSize -> Large]
```



Out[77]=



```
In[78]:= edge1a = edges[data1, 0.1]; Image[edge1a]
```



```
In[79]:= cout4 = cannydemo[edge1a, 0.1, 0.5, 10];
```

```
strong edge vertices: 2434
```

```
weak edge vertices: 6236
```

```
further edge vertices:
```

```
round 1 : 1453
```

```
round 2 : 409
```

```
round 3 : 203
```

```
round 4 : 120
```

```
round 5 : 85
```

```
round 6 : 56
```

```
round 7 : 35
```

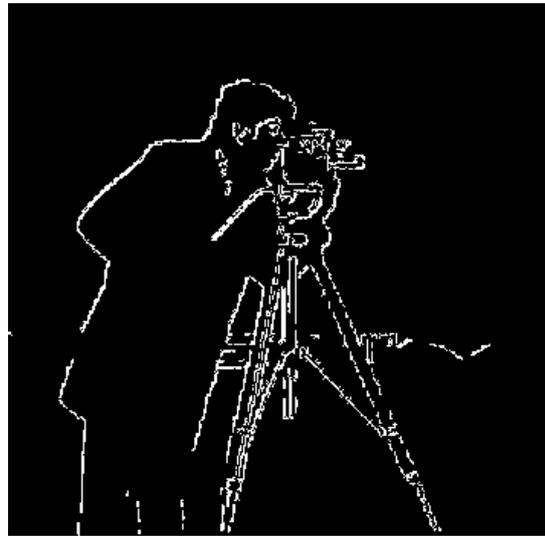
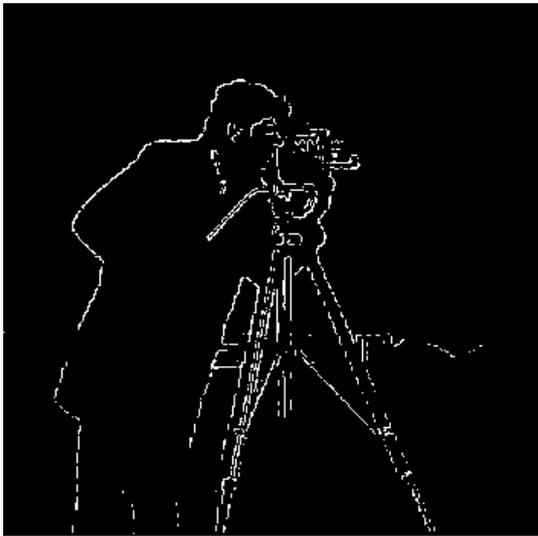
```
round 8 : 27
```

```
round 9 : 28
```

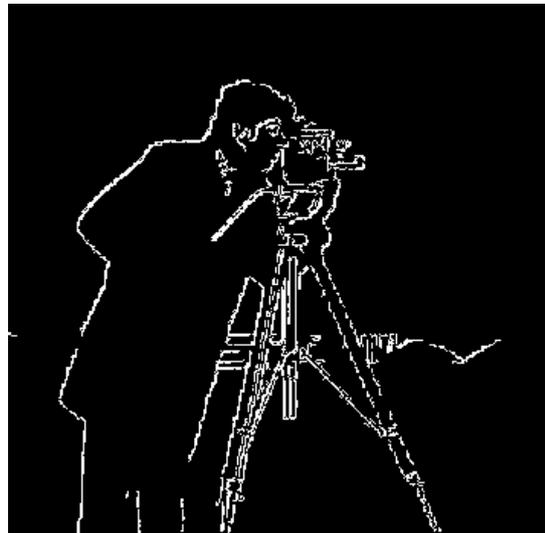
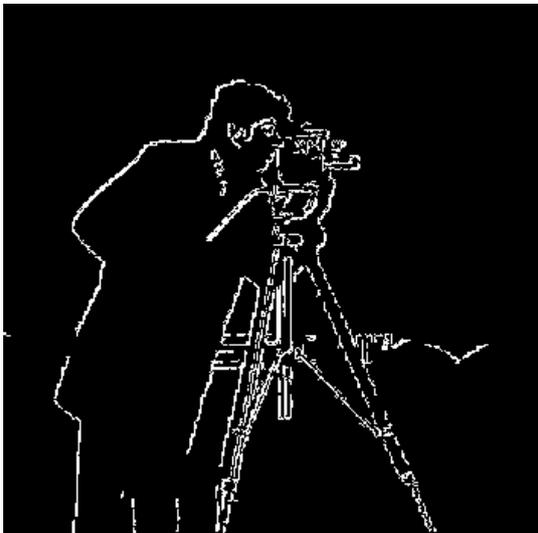
```
round 10 : 25
```

```
In[80]:= Animate[Image[cout4[[k]]], {k, 1, Length[cout4], 1}, AnimationRunning → False]
```

```
In[81]= GraphicsGrid[{{Image[cout4[[1]]], Image[cout4[[3]]]},  
  {Image[cout4[[5]]], Image[cout4[[10]]]}}, ImageSize -> Large]
```



Out[81]=



```
In[82]:= edge1b = edges[data1, 0.07]; Image[edge1b]
```

Out[82]=



```
In[83]:= cout5 = cannydemo[edge1b, 0.07, 0.2, 10];
```

strong edge vertices: 6615

weak edge vertices: 8461

further edge vertices:

round 1 : 887

round 2 : 128

round 3 : 53

round 4 : 19

round 5 : 9

round 6 : 5

round 7 : 6

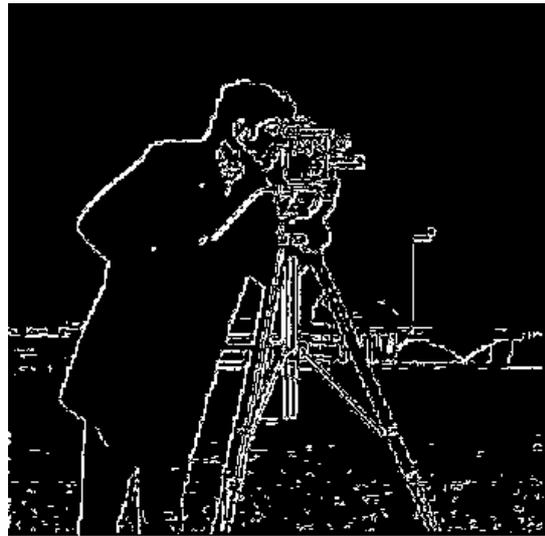
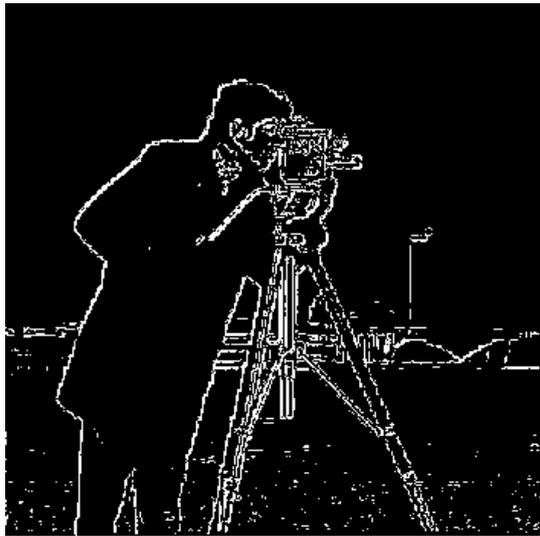
round 8 : 2

round 9 : 2

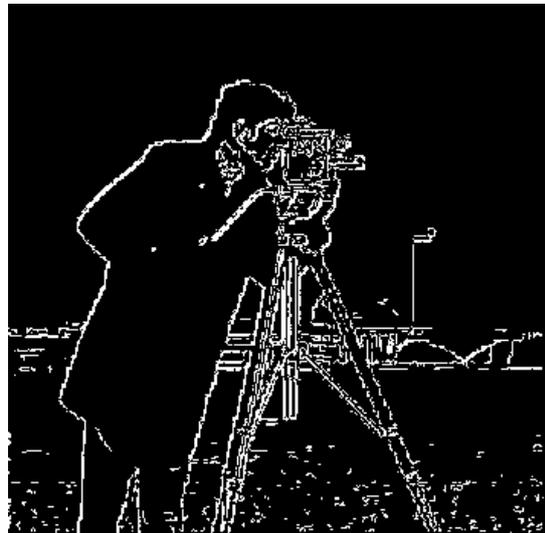
round 10 : 1

```
In[84]:= Animate[Image[cout5[[k]]], {k, 1, Length[cout5], 1}, AnimationRunning -> False]
```

```
In[85]= GraphicsGrid[{{Image[cout5[[1]]], Image[cout5[[3]]]},  
  {Image[cout5[[5]]], Image[cout5[[10]]]}}, ImageSize -> Large]
```



Out[85]=



In[86]=

Second example

```
In[87]:= img2 = Import["~/Lehre/Wavelets-All/WTBV-10/CWT/maira.jpg"]
```



```
In[88]:= img2 = ColorConvert[img2, "Grayscale"]
```



```
In[89]:= data2 = ImageData[img2][[301 ;; 700, 301 ;; 700]];
```

```
In[90]:= Dimensions[data2]
```

```
Out[90]= {400, 400}
```

```
In[91]:= Image[data2]
```



```
In[92]:= edge2 = edges[data2, 0.25];
```

```
In[93]:= cout6 = cannydemo[edge2, 0.25, 0.4, 20];
```

strong edge vertices: 5444

weak edge vertices: 13217

further edge vertices:

round 1 : 2801

round 2 : 232

round 3 : 126

round 4 : 80

round 5 : 47

round 6 : 42

round 7 : 31

round 8 : 25

round 9 : 25

round 10 : 21

round 11 : 23

round 12 : 15

round 13 : 16

round 14 : 14

round 15 : 14

round 16 : 15

round 17 : 16

round 18 : 13

round 19 : 13

round 20 : 13

In[94]:= `Animate[Image[cout6[[k]]], {k, 1, Length[cout6], 1}, AnimationRunning → False]`

```
In[95]= GraphicsGrid[{{Image[cout6[[1]]], Image[cout6[[5]]]},  
  {Image[cout6[[10]]], Image[cout6[[20]]]}}, ImageSize -> Large]
```

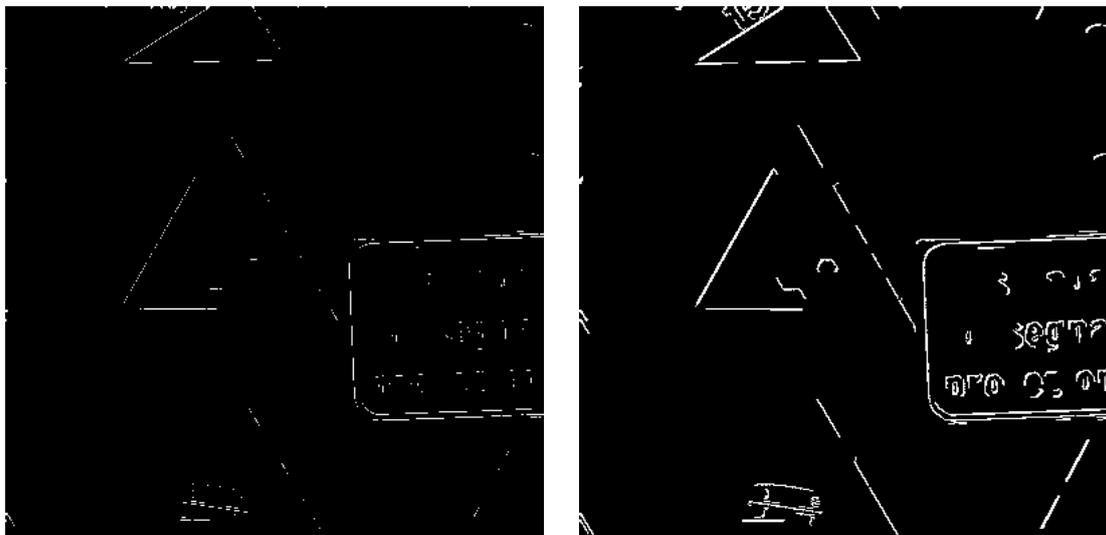


Out[95]=

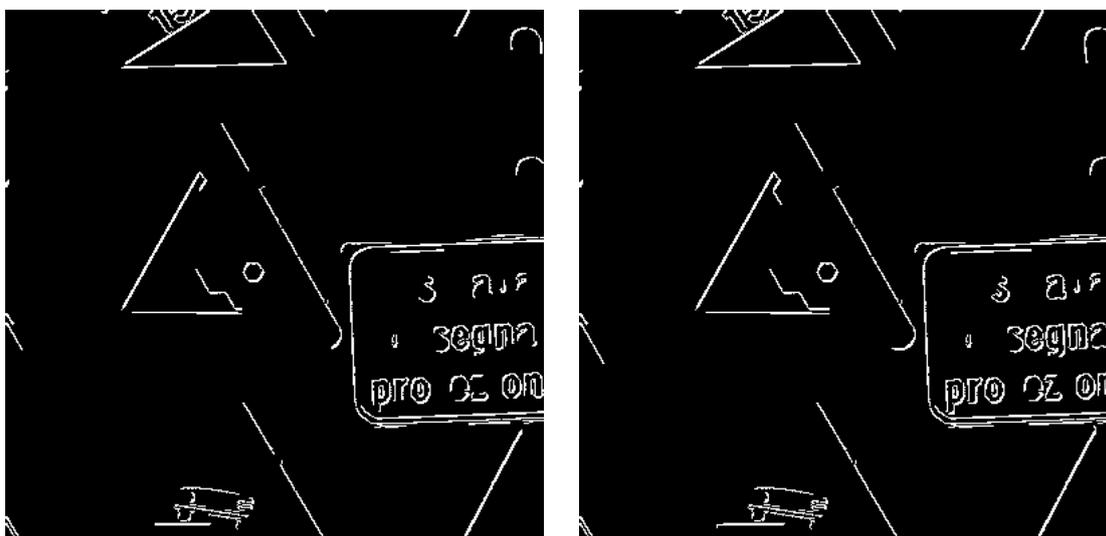


```
In[96]:= cout7 = cannydemo[edge2, 0.2, 0.5, 30];  
strong edge vertices: 880  
weak edge vertices: 15576  
further edge vertices:  
round 1 : 1132  
round 2 : 421  
round 3 : 311  
round 4 : 259  
round 5 : 215  
round 6 : 191  
round 7 : 148  
round 8 : 144  
round 9 : 148  
round 10 : 133  
round 11 : 137  
round 12 : 122  
round 13 : 112  
round 14 : 102  
round 15 : 93  
round 16 : 89  
round 17 : 74  
round 18 : 73  
round 19 : 57  
round 20 : 54  
round 21 : 52  
round 22 : 47  
round 23 : 40  
round 24 : 38  
round 25 : 38  
round 26 : 35  
round 27 : 35  
round 28 : 31  
round 29 : 30  
round 30 : 27  
  
In[97]:= Animate[Image[cout7[[k]]], {k, 1, Length[cout7], 1}, AnimationRunning → False]
```

```
In[98]= GraphicsGrid[{{Image[cout7[[1]]], Image[cout7[[10]]]},
  {Image[cout7[[20]]], Image[cout7[[30]]]}}, ImageSize -> Large]
```



Out[98]=



Third example: smoothed version of the second example

```
In[99]:= img3 = GaussianFilter[img2, 3]
```

Out[99]=



```
In[100]:= data3 = ImageData[img3][[301 ;; 700, 301 ;; 700]];
```

```
In[101]:= Image[data3]
```

```
Out[101]=
```



```
In[102]:= edge3 = edges[data3, 0.25];
```

```
In[103]:= cout8 = cannydemo[edge3, 0.25, 0.6, 50];
```

```
strong edge vertices: 7637
```

```
weak edge vertices: 22882
```

```
further edge vertices:
```

```
round 1 : 4389
```

```
round 2 : 780
```

```
round 3 : 312
```

```
round 4 : 245
```

```
round 5 : 170
```

```
round 6 : 134
```

```
round 7 : 117
```

```
round 8 : 90
```

```
round 9 : 71
```

```
round 10 : 55
```

```
round 11 : 50
```

```
round 12 : 45
```

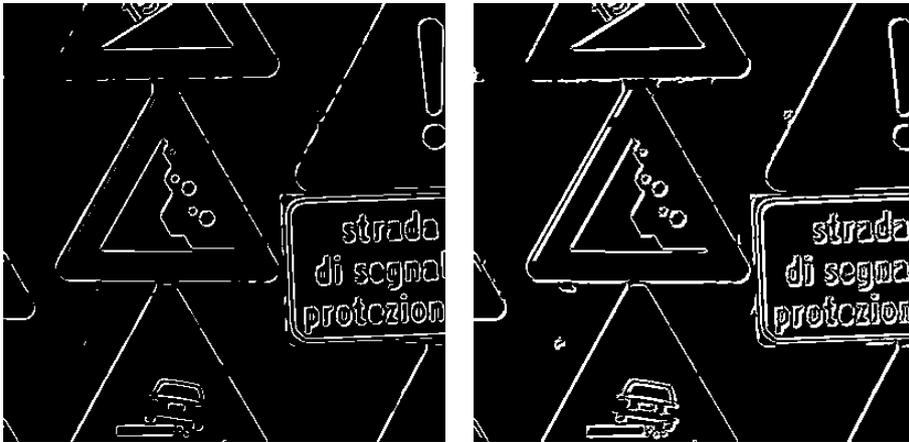
```
round 13 : 40
```

```
round 14 : 28
```

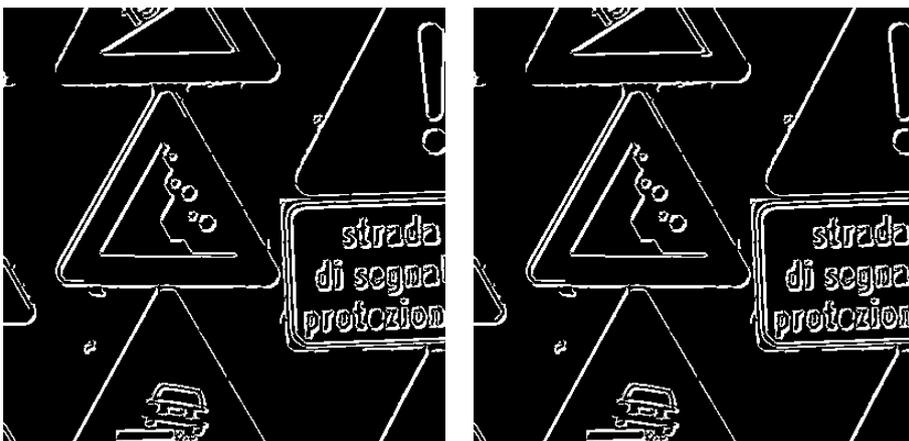
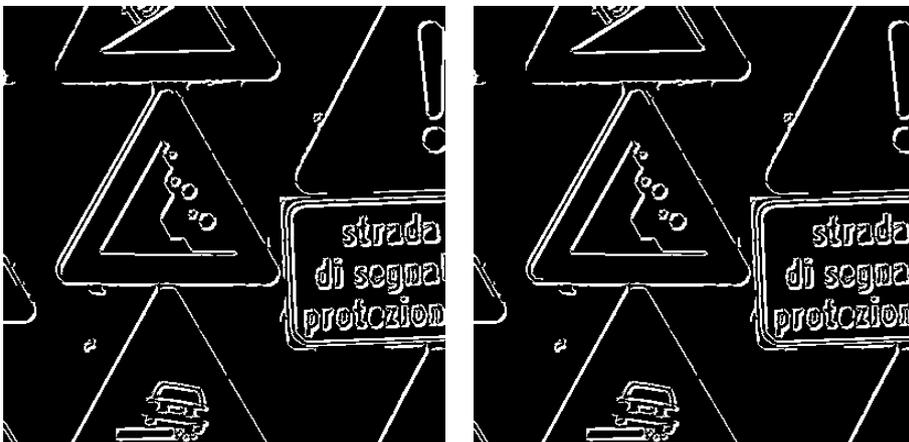
```
round 15 : 19
round 16 : 20
round 17 : 19
round 18 : 16
round 19 : 21
round 20 : 18
round 21 : 17
round 22 : 16
round 23 : 17
round 24 : 16
round 25 : 15
round 26 : 16
round 27 : 16
round 28 : 16
round 29 : 18
round 30 : 16
round 31 : 15
round 32 : 14
round 33 : 12
round 34 : 13
round 35 : 11
round 36 : 13
round 37 : 13
round 38 : 12
round 39 : 11
round 40 : 10
round 41 : 11
round 42 : 10
round 43 : 12
round 44 : 11
round 45 : 9
round 46 : 10
round 47 : 9
round 48 : 11
round 49 : 9
round 50 : 10
```

```
In[104]:= Animate[Image[cout8[[k]]], {k, 1, Length[cout8], 1}, AnimationRunning → False]
```

```
In[105]= GraphicsGrid[{{Image[cout8[[1]]], Image[cout8[[10]]]},
  {Image[cout8[[20]]], Image[cout8[[30]]]},
  {Image[cout8[[40]]], Image[cout8[[50]]]}, ImageSize -> Large]
```



```
Out[105]=
```



Fourth example

```
In[106]:= img4 = Import["~/Lehre/Wavelets-All/WTBV-10/CWT/zebras.jpg"]
```

Out[106]=



```
In[107]:= img4=ColorConvert[img4,"GrayScale"]
```

Out[107]=



```
In[108]:= img4g=GaussianFilter[img4,3]
```

```
Out[108]=
```



```
In[109]:= data4g=ImageData[img4g];
```

```
In[110]:= edge4=edges[data4g,0.2];
```

```
In[111]:= cout9=cannydemo[edge4,0.2,0.5,30];
```

strong edge vertices: 5518

weak edge vertices: 17175

further edge vertices:

round 1 : 2501

round 2 : 883

round 3 : 385

round 4 : 318

round 5 : 245

round 6 : 205

round 7 : 188

round 8 : 176

round 9 : 165

round 10 : 147

round 11 : 135

round 12 : 121

round 13 : 109

round 14 : 81

round 15 : 77

round 16 : 64

round 17 : 62

round 18 : 56

round 19 : 47

round 20 : 51

round 21 : 49

round 22 : 46

round 23 : 41

round 24 : 32

round 25 : 38

round 26 : 40

round 27 : 43

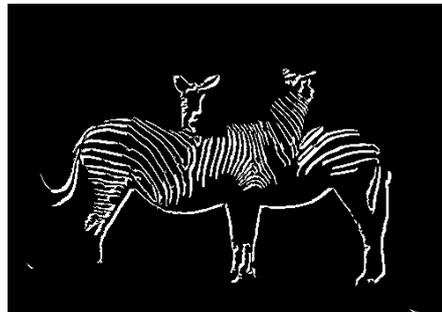
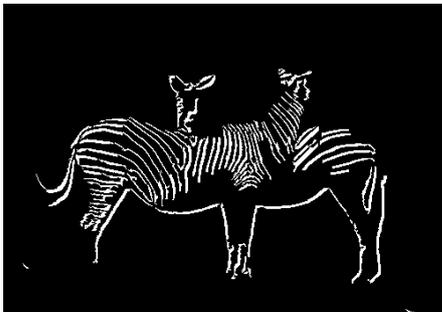
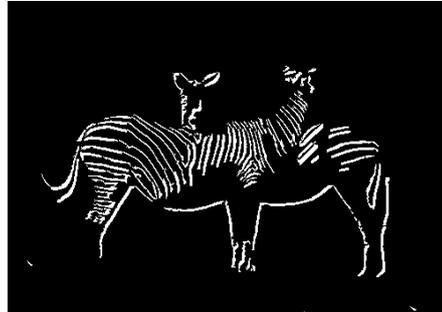
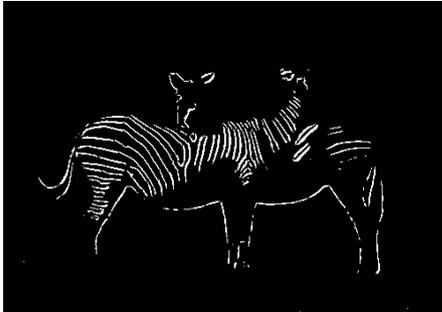
round 28 : 30

round 29 : 30

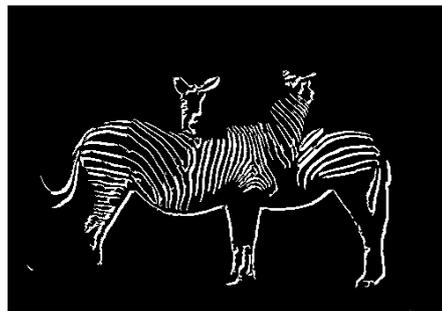
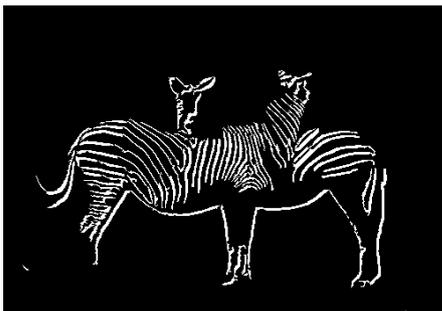
round 30 : 30

In[112]:= Animate[Image[cout9[[k]]],{k,1,Length[cout9],1},AnimationRunning->False]

```
In[113]= GraphicsGrid[{{Image[cout9[[1]]], Image[cout9[[5]]]},  
  {Image[cout9[[10]]], Image[cout9[[15]]]},  
  {Image[cout9[[20]]], Image[cout9[[30]]]}, ImageSize -> Large]
```



Out[113]=



The à-trous algorithm illustrated

Important remark

Signals and filters are represented as list of *(position,value)*-pairs:

```
{ { k, a[k] }, { k + 1, a[k + 1] }, { k + 2, a[k + 2] }, ..., { m, a[m] } }
```

All the signals and filters shown here are finite and can thus be represented using Laurent polynomials.

Filtering operations (i.e. convolutions) are then nothing but multiplications of L-polynomials.

Functions

```
In[9]:= sig2pol[sig_,z_]:=
(*
Converting signals into L-polynomials
*)
Module[{len},
len=Last[sig][[1]]-First[sig][[1]]+1;
Sum[sig[[k,2]] z^sig[[k,1]],{k,1,len}]
]
```

```
In[10]:= pol2sig[pol_,z_]:=Module[{low,high},
(*
Converting L-polynomials into signals
*)
low=Exponent[pol,z,Min];
high=Exponent[pol,z,Max];
Table[{k,Coefficient[pol,z,k]
},{k,low,high}]
]
```

```
In[11]:= convolve[sig_,fil_]:=Module[{sigpol,filpol,outpol,z},
(*
Convolution of the signal sig with the filter fil,
using polynomial multiplication
*)
sigpol=sig2pol[sig,z];
filpol= sig2pol[fil,z];
outpol=sigpol filpol;
pol2sig[outpol,z]
]
```

```

In[12]:= spread[sig_,m_]:=
  (*
  Spreading a signal (or filter) by a factor 2^m
  which corresponds to m-fold upsampling by a factor 2
  *)
  Module[{sigpol,output,z},
    sigpol=sig2pol[sig,z];
    output=sigpol/(z->z^(2^m));
    pol2sig[output,z]
  ]

```

```

In[13]:= atrous[sig_,gfil_,hfil_,m_,t_]:=
  (*
  Applying the à-trous algorithm over m levels to a signal sig
  using filters gfil (high-pass filter) and hfil (low-pass filter);
  output repeats every row t times (for better visibility);
  All the computations are made using polynomial representations
  of signals and filters
  *)
  Module[{left,right,sigpol,gpol,hpol,current,A,out,z},
    left=First[sig][[1]];
    right=Last[sig][[1]];
    sigpol=sig2pol[sig,z];
    gpol=sig2pol[gfil,z];
    hpol=sig2pol[hfil,z];
    current=sigpol*hpol;
    A[0]=sigpol*gpol;
    out[0]=Table[Coefficient[A[0],z,n],{n,left,right}];
    For[k=1,k<=m,k++,
      A[k]= current*(z*gpol)/(z->z^(2^k));
      out[k]=Table[Coefficient[A[k],z,n],{n,left,right}];
      current=current*(hpol/(z->z^(2^k)));
      Table[out[Quotient[s,t]},{s,0,t(m+1)-1}]
    ]

```

```

In[14]:= show[sig_]:=Transpose[sig]//MatrixForm

```

```

In[15]:= isl=ImageSize->Large;

```

```

In[16]:= opt[a_]:= {Filling->Axis,PlotRange->All,PlotStyle->{PointSize[a],Black}};

```

Signals and polynomials

first signal

```
In[17]:= signal1 = {{2, a}, {3, b}, {4, c}};
          {show[signal1], polynom1 = sig2pol[signal1, z]}
```

```
Out[17]= { ( 2 3 4 ) , a z^2 + b z^3 + c z^4 }
          ( a b c )
```

second signal

```
In[18]:= signal2 = {{-1, d}, {0, e}};
          {show[signal2], polynom2 = sig2pol[signal2, z]}
```

```
Out[19]= { ( -1 0 ) , e + d/z }
          ( d e )
```

convolution of both signals

```
In[20]:= signal12 = convolve[signal1, signal2]; show[signal12]
```

Out[20]/MatrixForm=

```
( 1 2 3 4 )
( a d b d + a e c d + b e c e )
```

product of the polynomials

```
In[21]:= polynom1 * polynom2
```

```
Out[21]= ( e + d/z ) ( a z^2 + b z^3 + c z^4 )
```

```
In[22]:= Collect[polynom1 * polynom2, z]
```

```
Out[22]= a d z + ( b d + a e ) z^2 + ( c d + b e ) z^3 + c e z^4
```

spreading the signals

```
In[23]:= sp1 = spread[signal1, 2]; show[sp1]
```

Out[23]/MatrixForm=

```
( 8 9 10 11 12 13 14 15 16 )
( a 0 0 0 b 0 0 0 c )
```

```
In[24]:= sp2 = spread[signal2, 1]; show[sp2]
```

Out[24]/MatrixForm=

```
( -2 -1 0 )
( d 0 e )
```

Filtering example

low-pass filter (Haar)

In[25]:= $H = \left\{ \left\{ -1, 1/\sqrt{2} \right\}, \left\{ 0, 1/\sqrt{2} \right\} \right\}$

Out[25]:= $\left\{ \left\{ -1, \frac{1}{\sqrt{2}} \right\}, \left\{ 0, \frac{1}{\sqrt{2}} \right\} \right\}$

high-pass filter (Haar)

In[26]:= $G = \left\{ \left\{ -1, -1/\sqrt{2} \right\}, \left\{ 0, 1/\sqrt{2} \right\} \right\}$

Out[26]:= $\left\{ \left\{ -1, -\frac{1}{\sqrt{2}} \right\}, \left\{ 0, \frac{1}{\sqrt{2}} \right\} \right\}$

signal A

In[27]:= $sA = \text{Table}[\{k, k\}, \{k, 1, 5\}]; \text{show}[sA]$

Out[27]/MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix}$$

signal B

In[28]:= $sB = \text{Table}[\{k, -k + 6\}, \{k, 6, 10\}]; \text{show}[sB]$

Out[28]/MatrixForm=

$$\begin{pmatrix} 6 & 7 & 8 & 9 & 10 \\ 0 & -1 & -2 & -3 & -4 \end{pmatrix}$$

signal C

In[29]:= $sC = \text{Join}[sA, sB]; \text{show}[sC]$

Out[29]/MatrixForm=

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 3 & 4 & 5 & 0 & -1 & -2 & -3 & -4 \end{pmatrix}$$

low-pass filtering the signal C

In[30]:= $CH = \text{convolve}[sC, H]; \text{show}[CH]$

Out[30]/MatrixForm=

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} + \sqrt{2} & \frac{3}{\sqrt{2}} + \sqrt{2} & \frac{3}{\sqrt{2}} + 2\sqrt{2} & \frac{5}{\sqrt{2}} + 2\sqrt{2} & \frac{5}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} - \sqrt{2} & -\frac{3}{\sqrt{2}} - \sqrt{2} & -\frac{3}{\sqrt{2}} - \end{pmatrix}$$

high-pass filtering the signal C

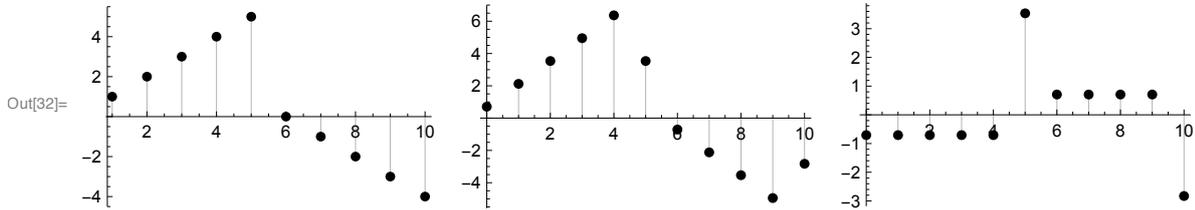
In[31]:= $CG = \text{convolve}[sC, G]; \text{show}[CG]$

Out[31]/MatrixForm=

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} - \sqrt{2} & -\frac{3}{\sqrt{2}} + \sqrt{2} & \frac{3}{\sqrt{2}} - 2\sqrt{2} & -\frac{5}{\sqrt{2}} + 2\sqrt{2} & \frac{5}{\sqrt{2}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} + \sqrt{2} & \frac{3}{\sqrt{2}} - \sqrt{2} & -\frac{3}{\sqrt{2}} + \end{pmatrix}$$

displaying the results

```
In[32]:= GraphicsRow[{
  ListPlot[sC, opt[0.03]], ListPlot[CH, opt[0.03]], ListPlot[CG, opt[0.03]], isl]
```

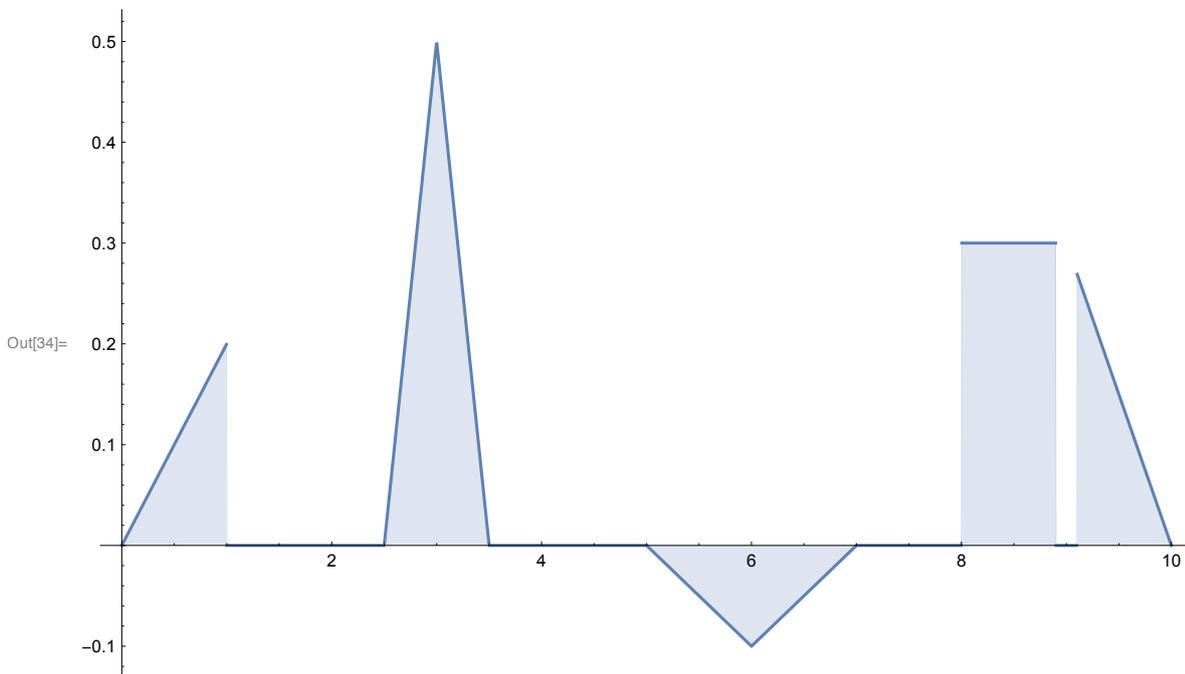


ID à-trous algorithm (example)

defining a piecewise signal

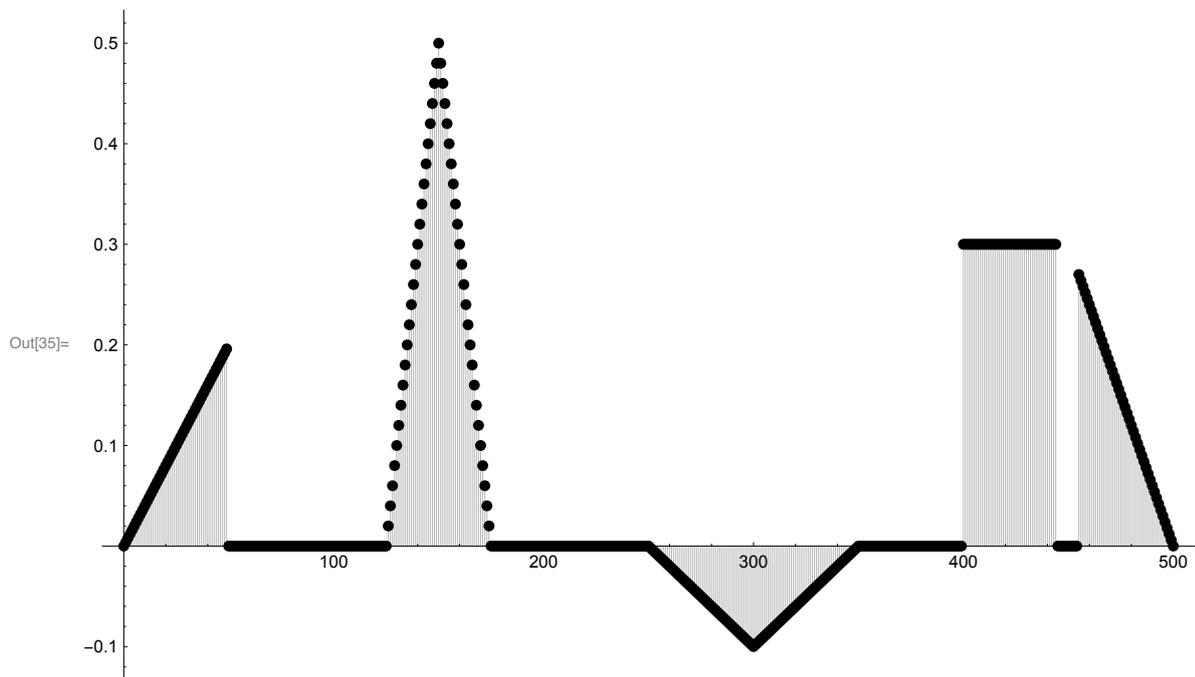
```
In[33]:= f[x_] := Piecewise[
  {{.2 * x, x < 1},
  {.5 - Abs[3 - x], 2.5 ≤ x && x < 3.5}, {- .1 * (1 - Abs[x - 6]), 5 ≤ x && x < 7},
  {.3, 8 ≤ x && x < 8.9},
  {0.3 (10 - x), 9.1 ≤ x ≤ 10}}]
```

```
In[34]:= Plot[f[x], {x, 0, 10}, PlotRange → All, Filling → Axis, ImageSize → Large]
```



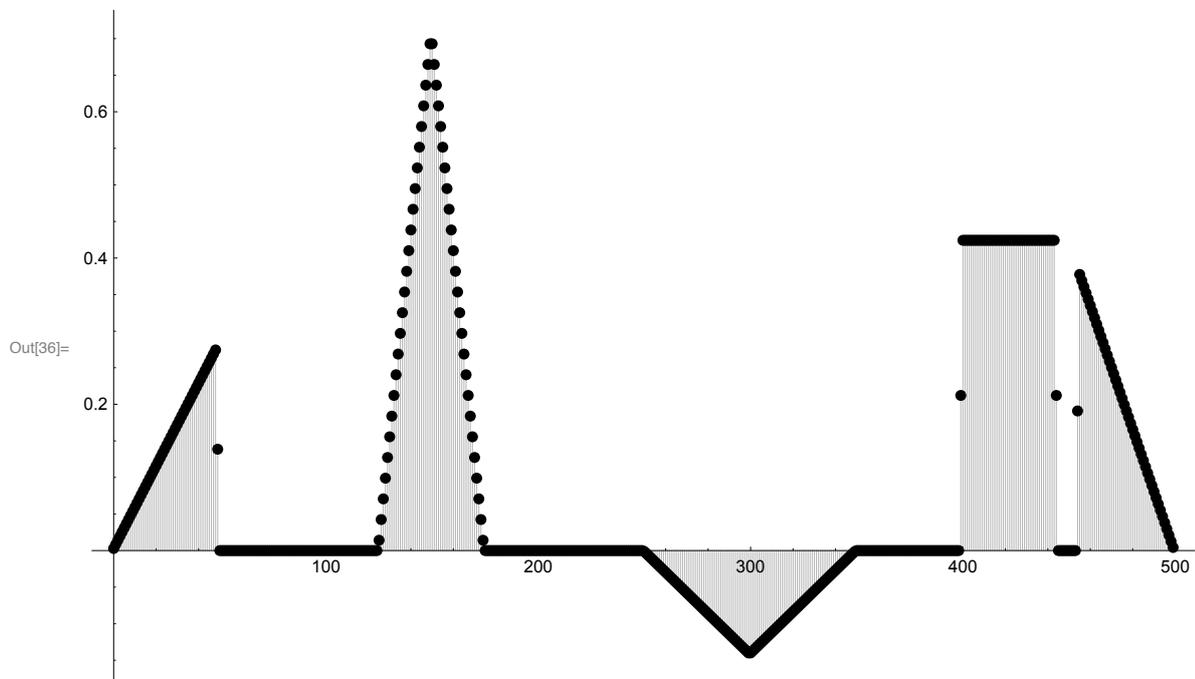
discretizing the signal

```
In[35]:= A = Table[{x, f[x/50]}, {x, 0, 500}]; a = ListPlot[A, opt[0.01], isl]
```



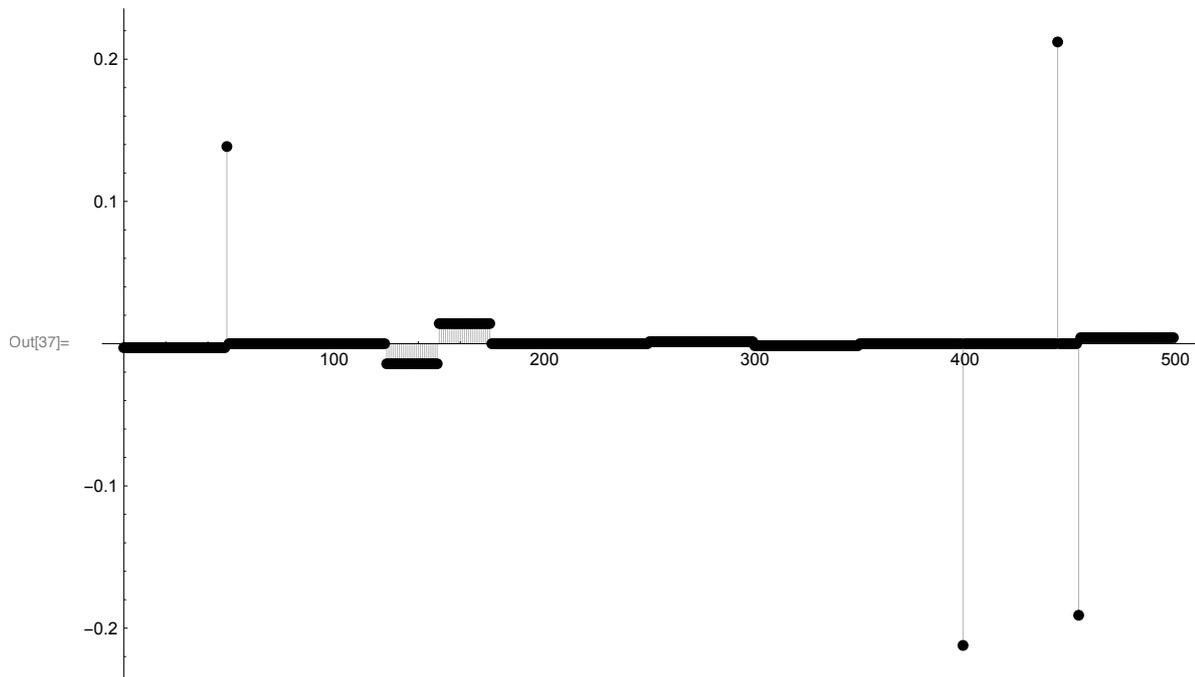
low-pass filtering of the signal

```
In[36]:= AH = convolve[A, H]; ah = ListPlot[AH, opt[0.01], isl]
```



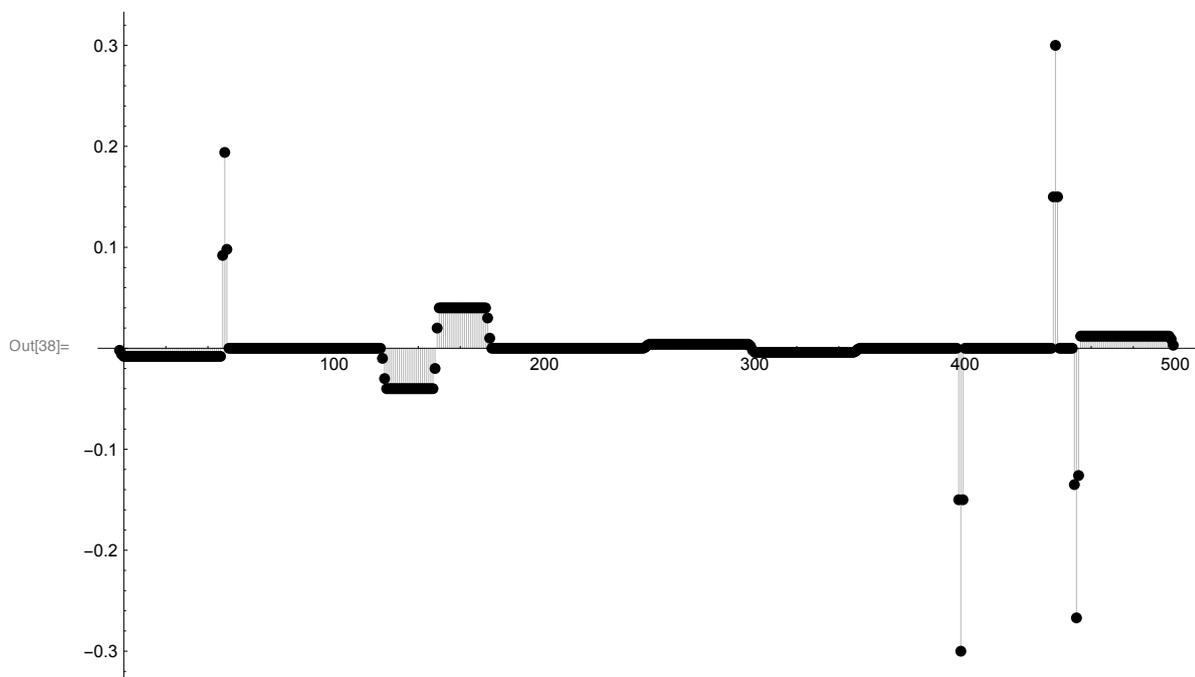
high-pass filtering of the signal

```
In[37]:= AG = convolve[A, G]; ag = ListPlot[AG, opt[0.01], isl]
```



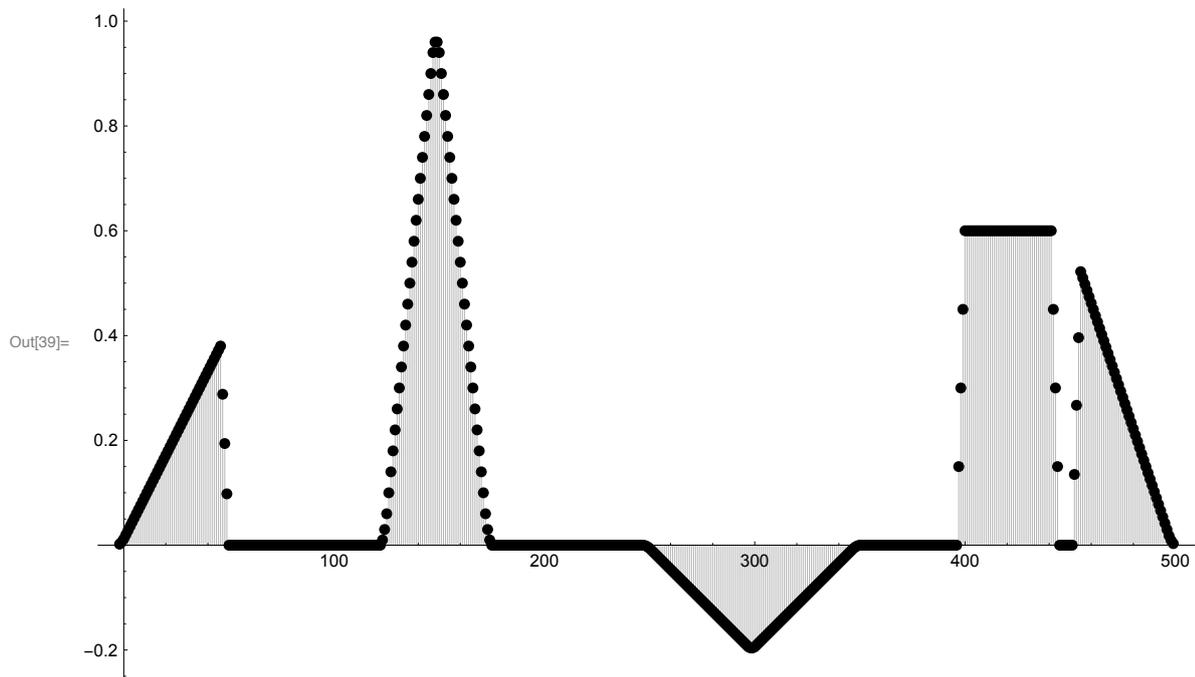
high-pass filtering the low-pass filtered signal

```
In[38]:= AHG = convolve[AH, spread[G, 1]]; ahg = ListPlot[AHG, opt[0.01], isl]
```



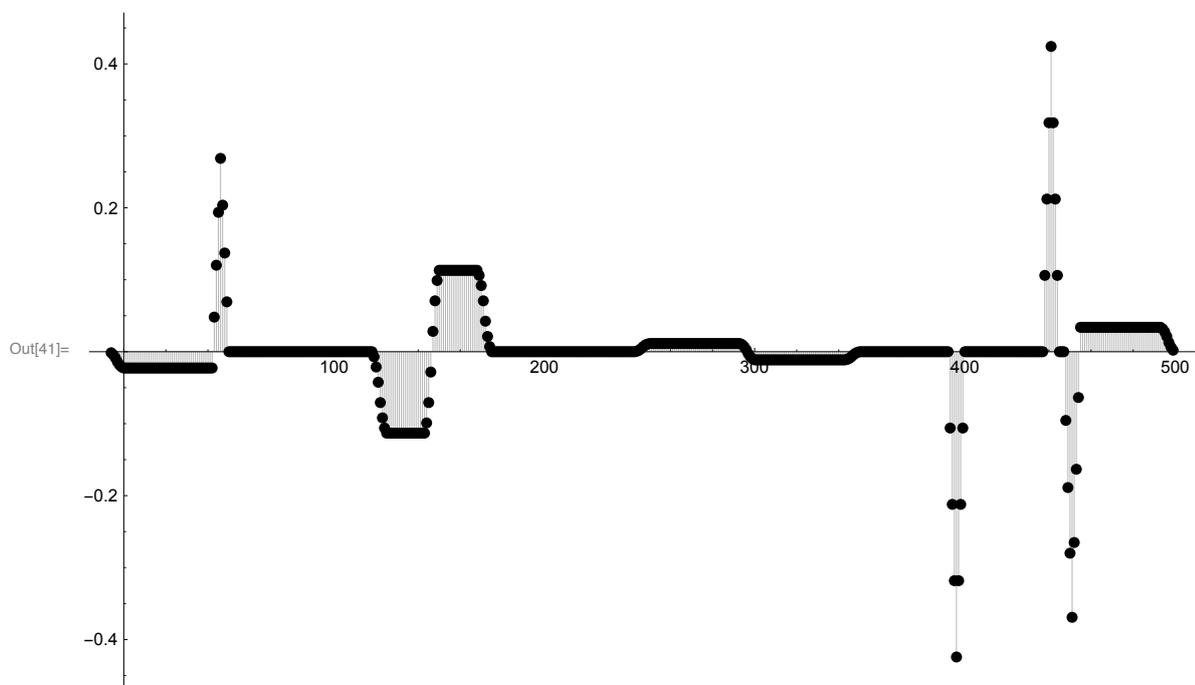
low-pass filtering the low-pass filtered signal

```
In[39]:= AHH = convolve[AH, spread[H, 1]]; ahh = ListPlot[AHH, opt[0.01], isl]
```

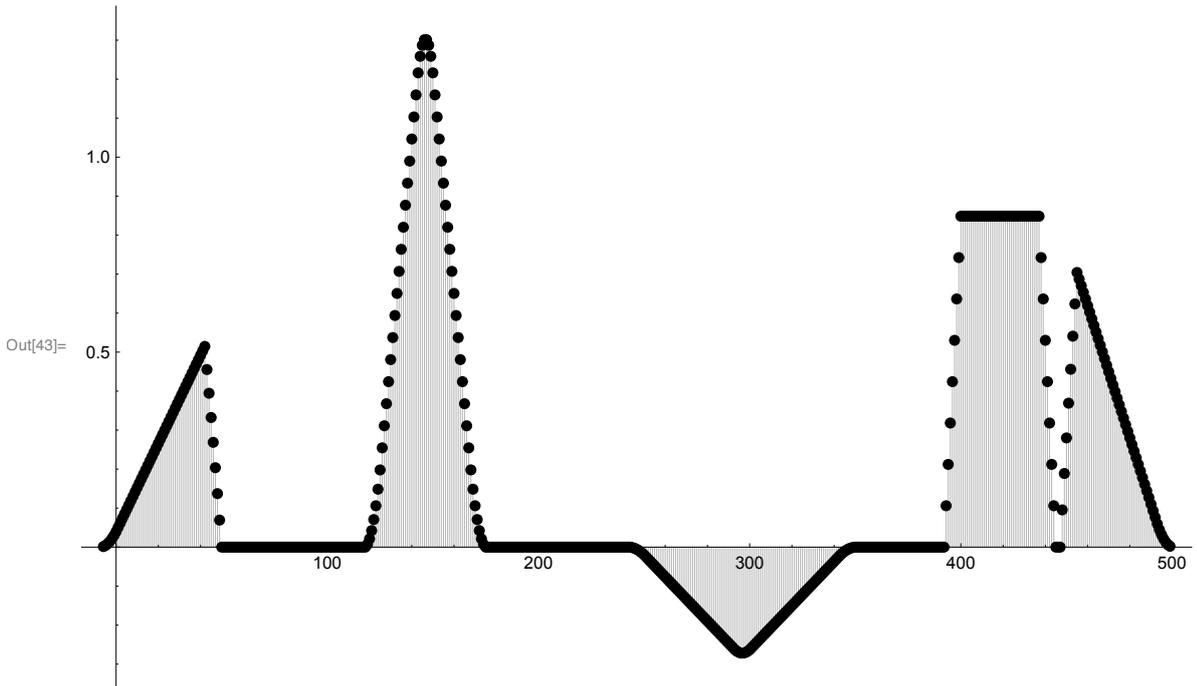


and more ...

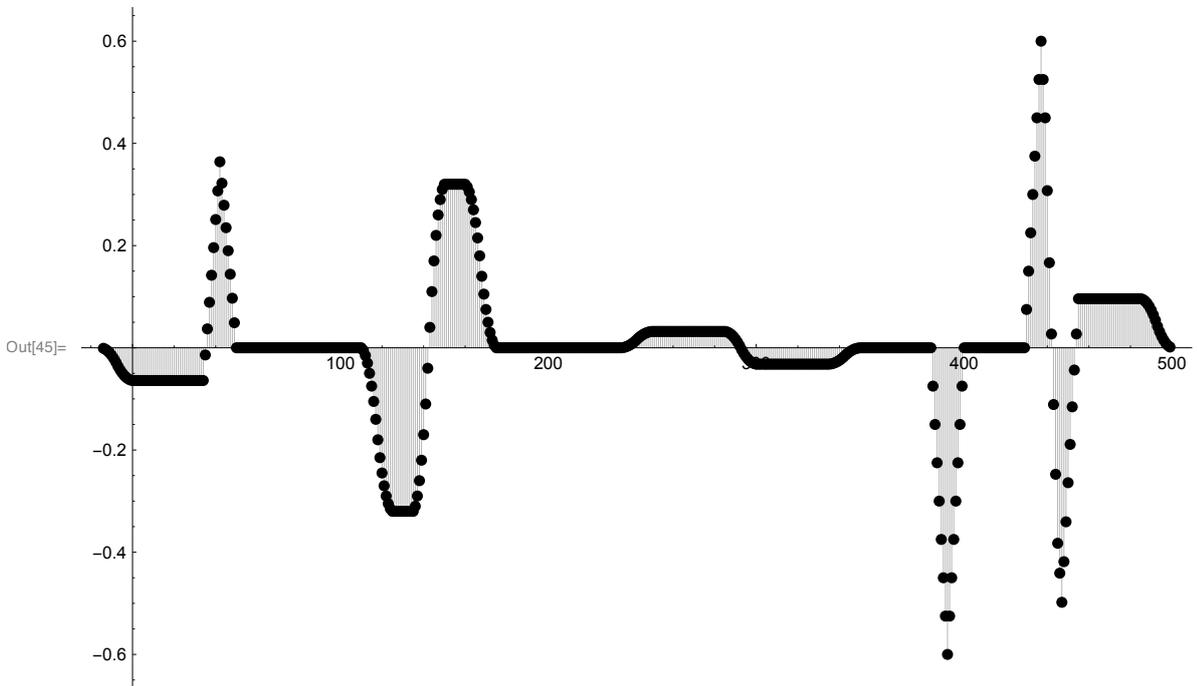
```
In[40]:= AHHG = convolve[AHH, spread[G, 2]];
ahhg = ListPlot[AHHG, opt[0.01], isl]
```



```
In[42]:= AHHH = convolve[AHH, spread[H, 2]];
         ahhh = ListPlot[AHHH, opt[0.01], isl]
```



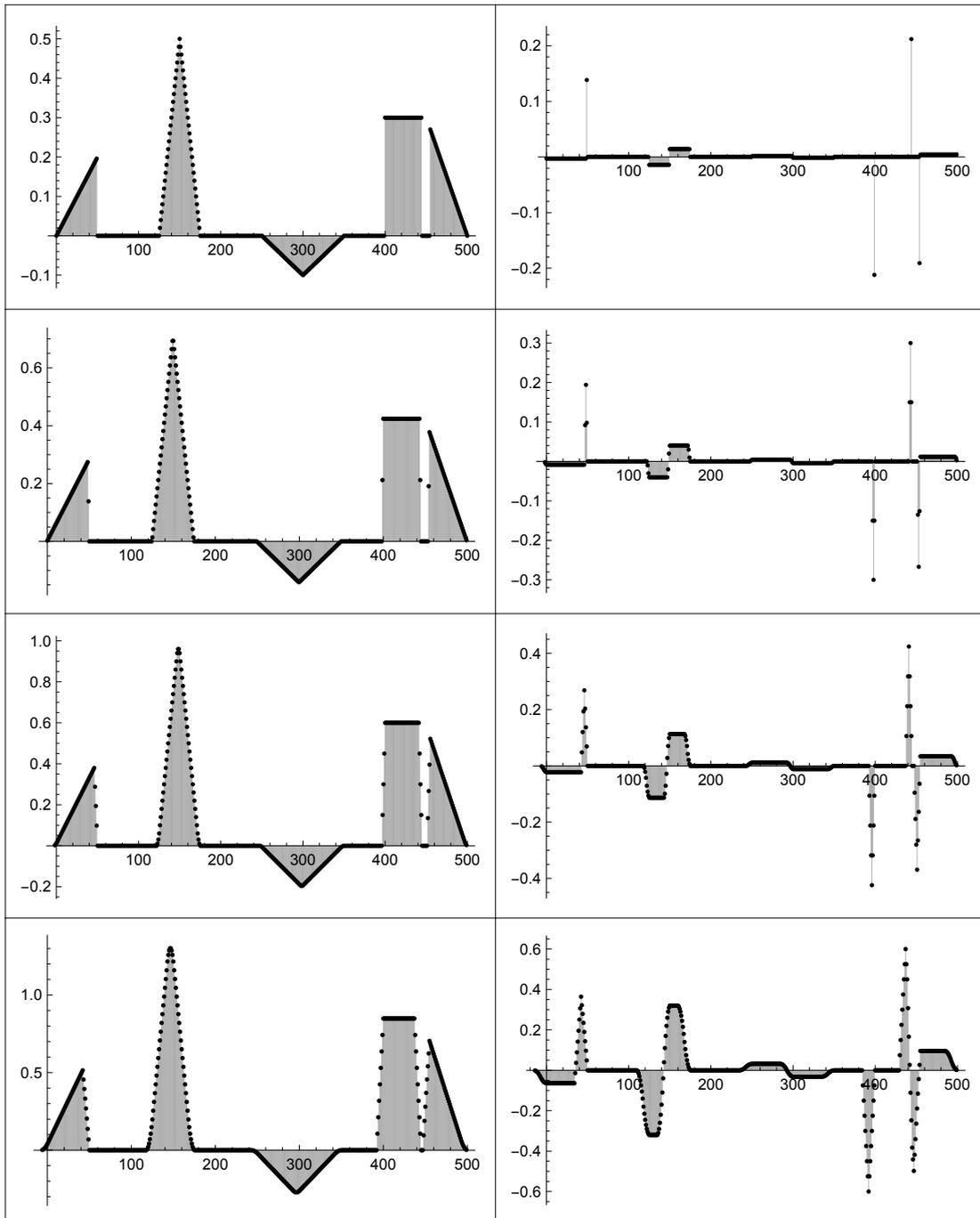
```
In[44]:= AHHHG = convolve[AHHH, spread[G, 3]];
         ahhhg = ListPlot[AHHHG, opt[0.01], isl]
```



summarizing the results

```
In[46]:= GraphicsGrid[{{a, ag}, {ah, ahg}, {ahh, ahhg}, {ahhh, ahhhg}}, Frame → All, isl]
```

Out[46]=



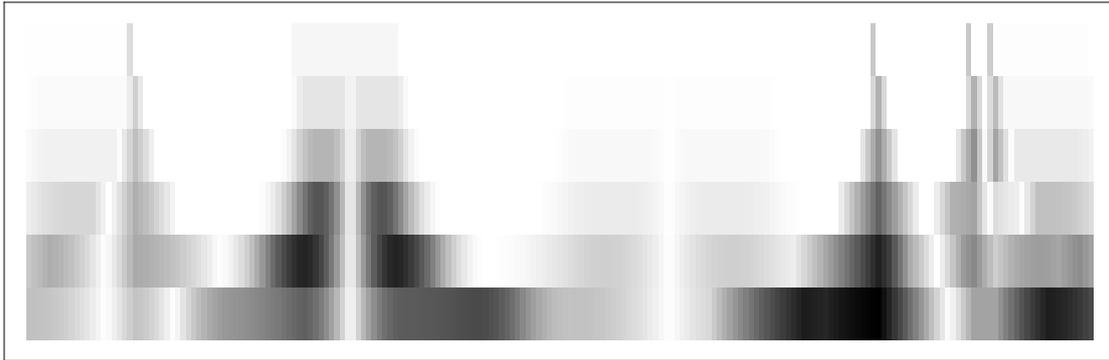
à-trous algorithm (scalogram version)

original signal (discretized)

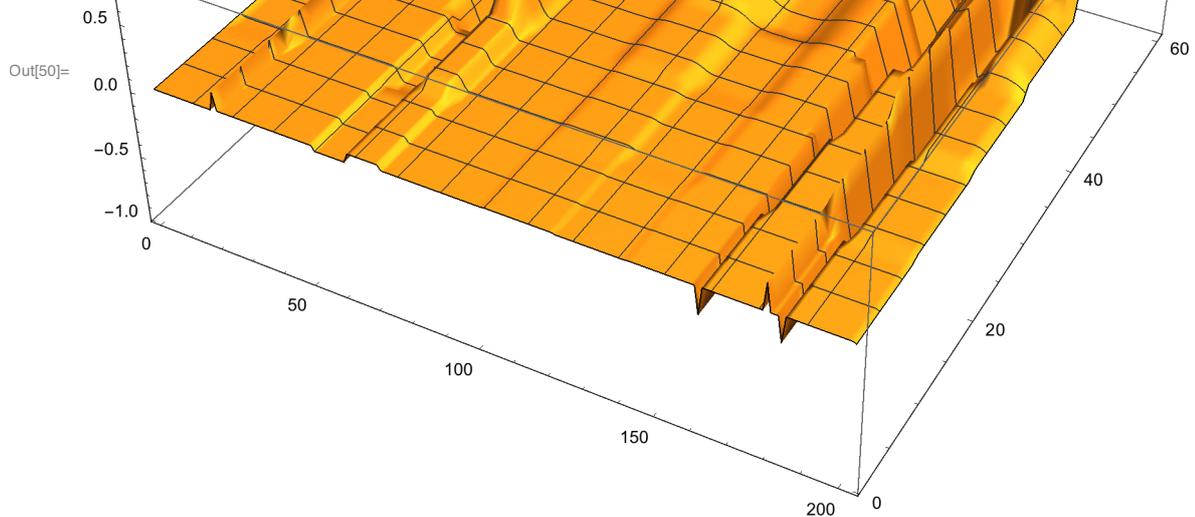
scalogram after applying a 5 level à-trous algorithm

```
In[47]:= B = Table[{x, f[x/20]}, {x, 0, 200}];
b = atrous[B, G, H, 5, 10];
bplot = ArrayPlot[b, is1]
```

Out[49]=



```
In[50]:= ListPlot3D[b, PlotRange -> All, is1]
```



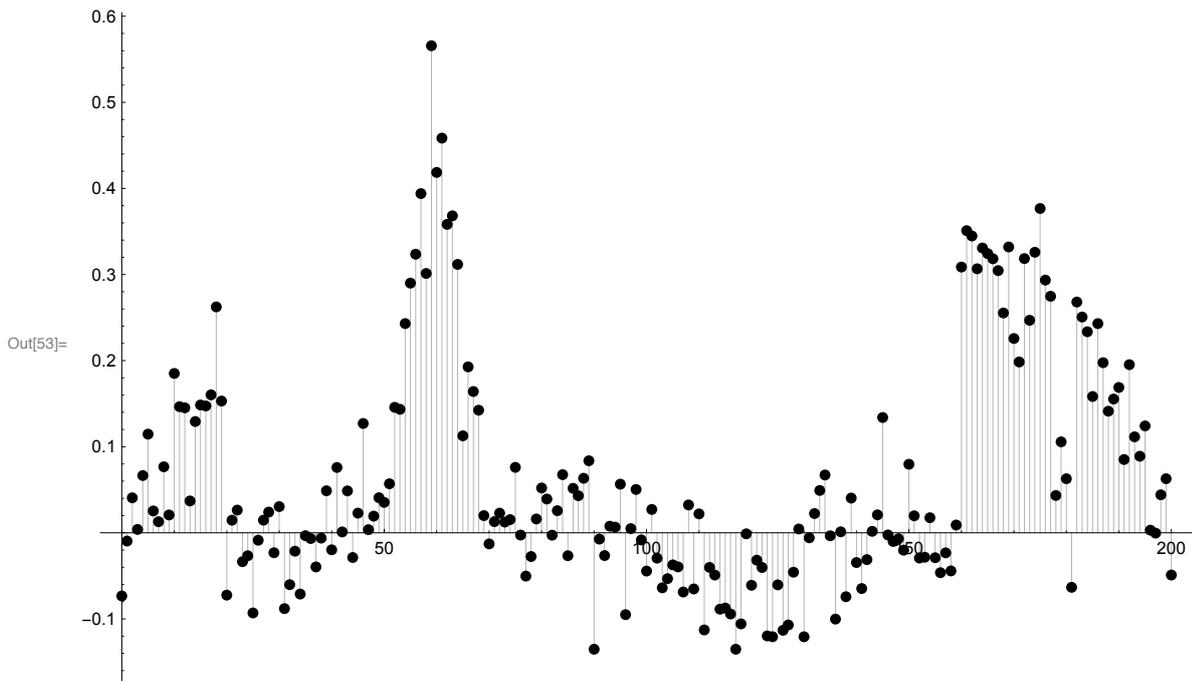
noised signal

```
In[51]:= data = RandomVariate[NormalDistribution[0, 0.05], 201];
```

```
In[52]:= Bn = Table[{x, f[x/20] + data[[x + 1]]}, {x, 0, 200}];
```

displaying the noised signal

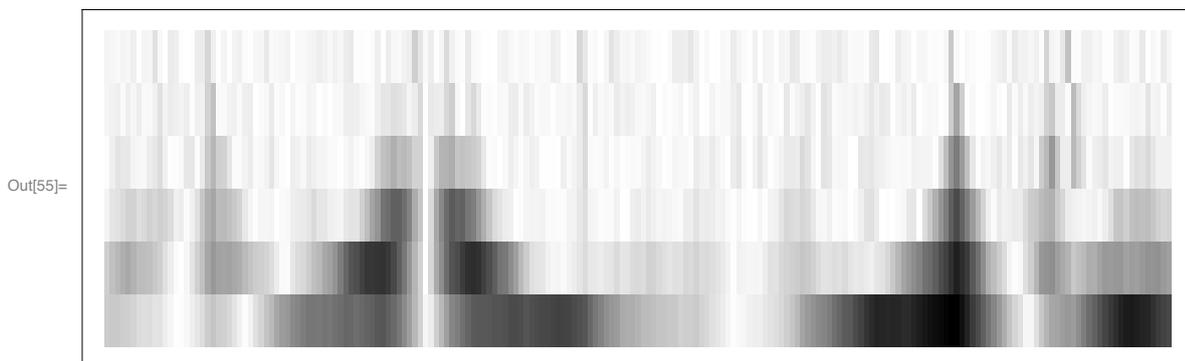
```
In[53]:= ListPlot[Bn, Filling -> Axis, opt[0.01], isl]
```



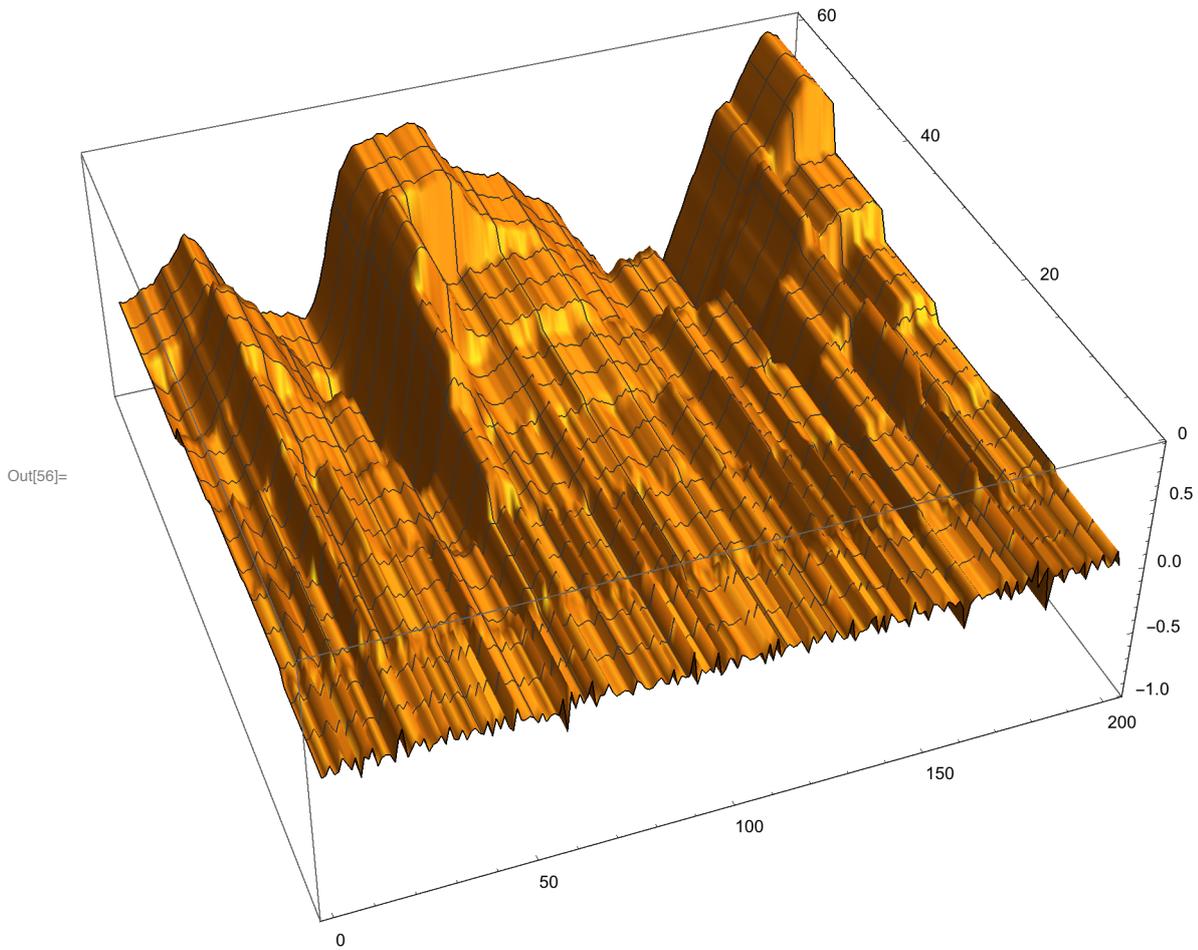
scalogram after applying a 5 level à-trous algorithm

```
In[54]:= bn = atrous[Bn, G, H, 5, 10];
```

```
bnplot = ArrayPlot[bn, isl]
```



```
In[56]:= ListPlot3D[bn, PlotRange -> All, is1]
```



applying first smoothing (Bspline filter) to the noised signal

```
In[57]:= S[n_] := pol2sig[z^(-n) ((1+z)/2)^(2n), z]
```

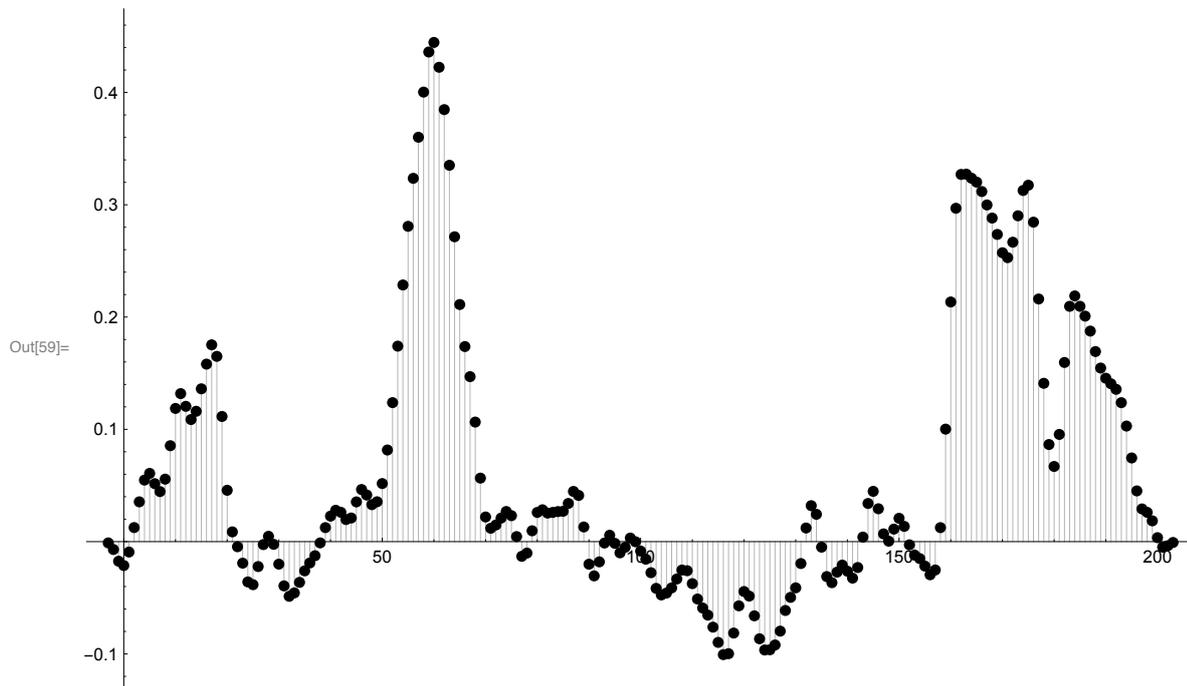
```
In[58]:= S[2]; show[S[2]]
```

Out[58]/MatrixForm=

$$\begin{pmatrix} -2 & -1 & 0 & 1 & 2 \\ \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \end{pmatrix}$$

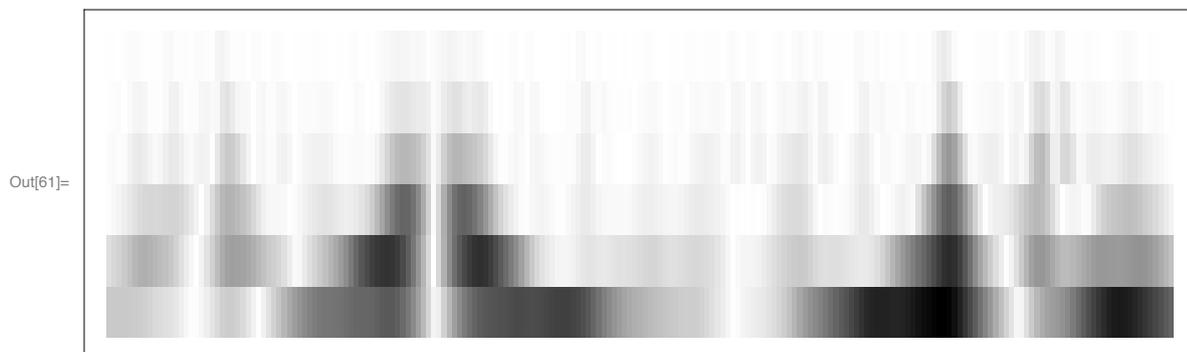
plotting the smoothed noised signal

```
In[59]:= Cn = convolve[Bn, S[3]]; ListPlot[Cn, opt[0.01], isl]
```

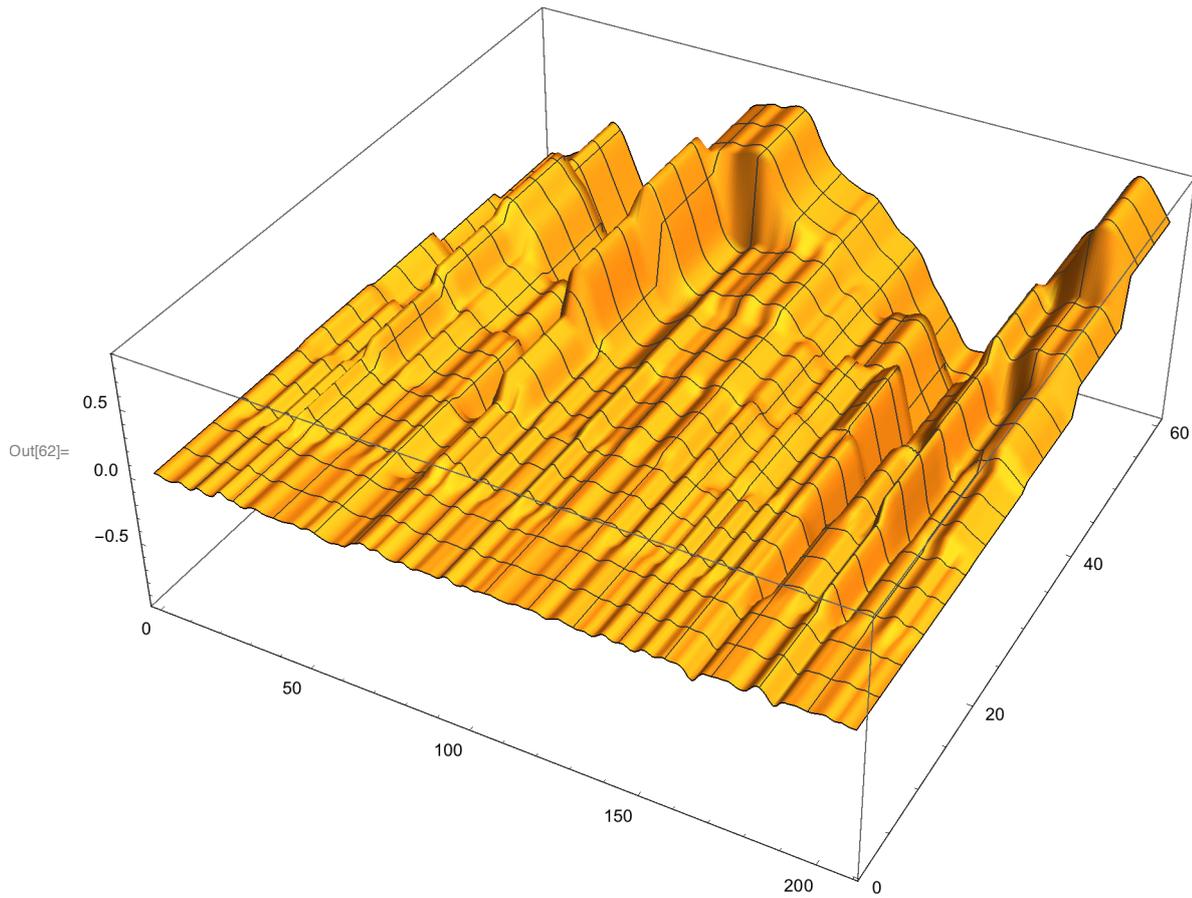


scalogram after applying a 5 level à-trous algorithm

```
In[60]:= c = atrous[Cn, G, H, 5, 10];  
cplot = ArrayPlot[c, isl]
```



```
In[62]:= ListPlot3D[c, PlotRange -> All, ImageSize -> Large]
```

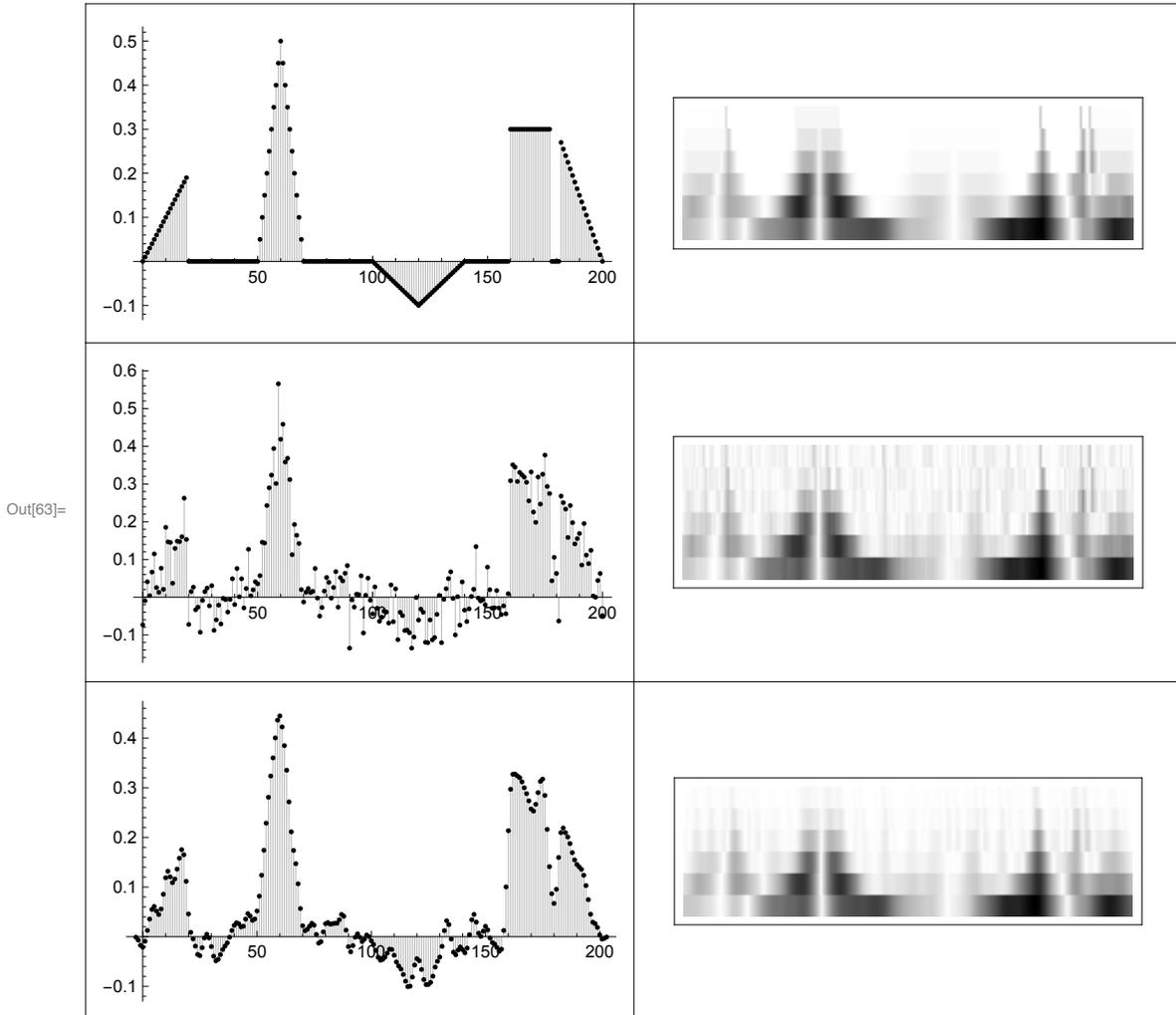


comparison of the results

```

In[63]:= GraphicsGrid[
  {{ListPlot[B, opt[0.01], isl], bplot}, {ListPlot[Bn, opt[0.01], isl], bnplot},
  {ListPlot[Cn, opt[0.01], isl], cplot}},
  Frame → All, isl]

```



Edge detection using wavelets

Computation of gradients

```
In[18]:= dir2vec[a_]:=Module[{aa},
  (*
  Computing the octant in which vector a={ax,ay} lies
  *)
  If[a=={0,0},Return[{1,0}]];
  aa=Mod[N[ArcTan[a[[1]],a[[2]]]+Pi/2],Pi,-Pi/2];
  Which[
  Abs[aa]<Pi/8,{1,0},
  Pi/8<aa<=3 Pi/8,{1,1},
  -3Pi/8<aa<= - Pi/8,{1,-1},
  True,{0,1}]
]
```

```
In[2]:= grad[A_,hfilter_,vfilter_]:=
  (*
  Computing the arrays for length (val) and direction (dir)
  of gradient vectors by filtering array A with filter hfilter and vfilter
  *)
  Module[{AA,val,dir,m,n,Ah,Av,h,v},
  {m,n}=Dimensions[A];
  AA=ArrayPad[A,{1,1}];
  Ah=ListConvolve[hfilter,AA];
  Av=ListConvolve[vfilter,AA];
  val=Table[0,{m},{n}];
  dir=Table[0,{m},{n}];
  Do[
  {h,v}={Ah[[x+1,y+1]],Av[[x+1,y+1]]};
  val[[x,y]]=N[Sqrt[h^2+v^2]];
  dir[[x,y]]=dir2vec[{h,v}],
  {y,1,n},{x,1,m}];
  {val,dir}
]
```

Determining edge vertices using gradient analysis

```

In[3]:= edges1[A_,hfilter_,vfilter_,level_]:=
(*
Detecting edge vertices in array A using gradient analysis.
The output cand is a 0-1-array of the same dimension as A
which shows all candidates.
Shown are all vertices which satisfy the edge vertex
criterion and where the local gradient maximum is
s ≥ "level"*average gradient
*)
Module[{m,n,val,dir,mean,VVal,Cand,
x,x1,x2,y,y1,y2,val0,val1,val2},
{m,n}=Dimensions[A];
{val,dir}=grad[A,hfilter,vfilter];
mean=Mean[Flatten[val]];
VVal=ArrayPad[val,{1,1}];
Cand=Table[0,{m},{n}];
Do[
{x1,y1}={x+1,y+1}+dir[[x,y]] ;
{x2,y2}={x+1,y+1}-dir[[x,y]];
val0=VVal[[x+1,y+1]];
val1=VVal[[x1,y1]];
val2=VVal[[x2,y2]];
If[
val0≥Max[val1,val2]&&val0≥level mean,
Cand[[x,y]]=1,
{x,1,m},{y,1,n}];
Cand
]

```

In[4]=

```

edges2[A_,B_,level_]:=
(*
Detecting egde vertices in array A using gradient analysis.
Input is given as two x- and y-filtered array A and B.
The output cand is a 0-1-array of the same dimension as A
which shows all candidates.
Shown are all vertices which satisfy the edge vertex criterion
and where the local gradient maximum is ≥ "level"*average gradient
*)
Module[{m,n,val,dir,mean,VVal,Cand,h,v,
x,x1,x2,y,y1,y2,val0,val1,val2},
{m,n}=Dimensions[A];
If[{m,n}≠Dimensions[B],
Throw["Dimensions don't match"]
];
val=Table[0,{m},{n}];
dir=Table[0,{m},{n}];
Do[
{h,v}={A[[x,y]],B[[x,y]]};
val[[x,y]]=N[Sqrt[h^2+v^2]];
dir[[x,y]]=dir2vec[{h,v}],
{x,1,m},{y,1,n}
];
mean=Mean[Flatten[val]];
VVal=ArrayPad[val,{1,1}];
Cand=Table[0,{m},{n}];
Do[
{x1,y1}={x+1,y+1}+dir[[x,y]] ;
{x2,y2}={x+1,y+1}-dir[[x,y]];
val0=VVal[[x+1,y+1]];
val1=VVal[[x1,y1]];
val2=VVal[[x2,y2]];
If[
val0≥Max[val1,val2]&&val0≥level mean,
Cand[[x,y]]=1],
{x,1,m},{y,1,n}
];
Cand
]

```

Test image

```
In[19]:= img = Import["~/Lehre/Wavelets-All/WTBV-10/CWT/zebras.jpg"]
```



Grayscale version of the test image

```
In[20]:= img = ColorConvert[img, "Grayscale"]
```



Smoothed image

```
In[21]:= img = GaussianFilter[img, 2]
```

```
Out[21]=
```



```
In[22]:= imdim = ImageDimensions[img]
```

```
Out[22]= {500, 357}
```

Naive gradient method

Horizontal and vertical Haar filter (high-pass)

```
In[23]:= h = {{1, -1}}; h // MatrixForm
```

```
v = {{1}, {-1}}; v // MatrixForm
```

```
Out[23]//MatrixForm=
```

$$\begin{pmatrix} 1 & -1 \end{pmatrix}$$

```
Out[24]//MatrixForm=
```

$$\begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Filtering the image with the Haar filter

```
In[25]:= imgh = ImageConvolve[img, h];
```

```
In[26]:= imgv = ImageConvolve[img, v];
```

```
In[30]:= IA = ImageAdjust; isl = ImageSize → Large;
```

Displaying the filtered image

```
In[31]:= GraphicsGrid[{{img_h, img_v}, {IA[img_h], IA[img_v]}}, isL]
```



Out[31]=

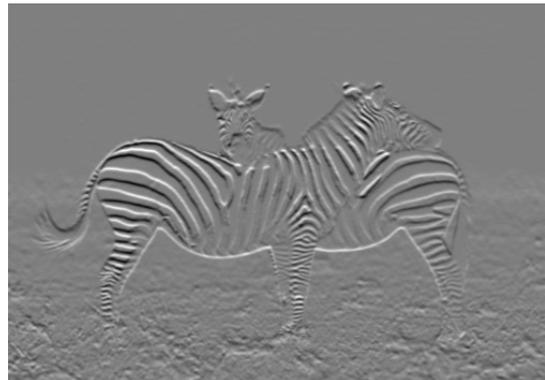
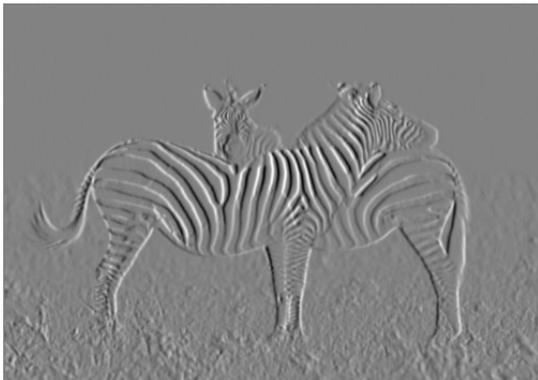


Image data as an array

```
In[32]:= imgdata = ImageData[img];
```

Edge detection based on the filtered image data

```
In[33]:= ImageDimensions[img]
```

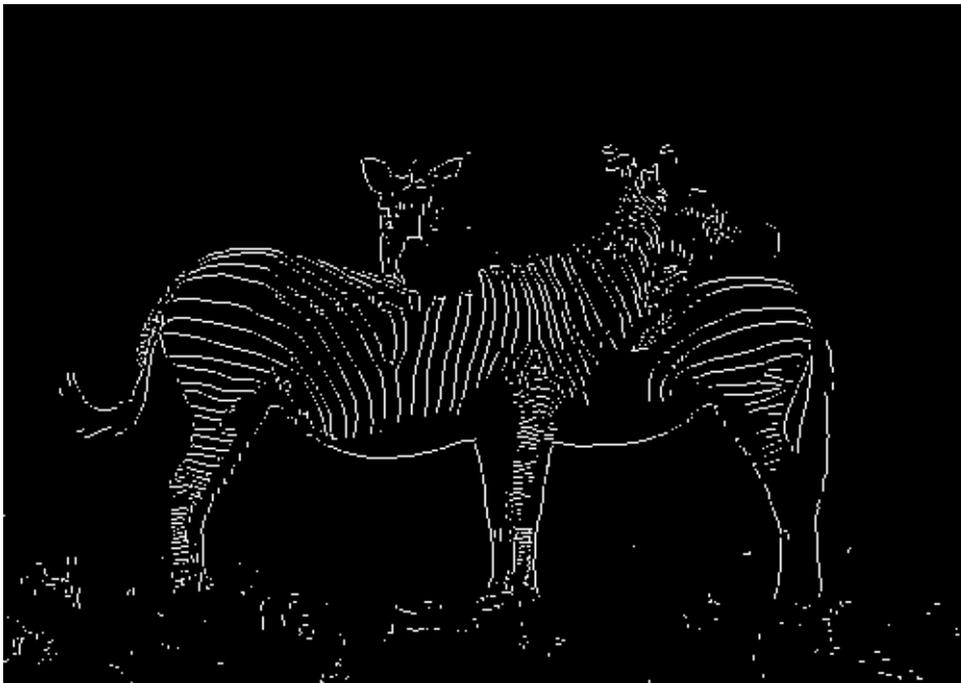
```
Out[33]:= {500, 357}
```

```
In[34]:= Dimensions[imgdata]
```

```
Out[34]:= {357, 500}
```

```
In[35]:= Image[edges1[imgdata, h, v, 3.0]]
```

```
Out[35]=
```



Gradients and the à-trous algorithm

Spreading (upsampling) a finite filter

```
In[36]:= spread[fil_]:=Most[Flatten[Map[{-#,0}&,fil]]]
```

Iterated spreading of a finite filter

```
In[37]:= spread[fil_,n_]:=Nest[spread[{-#}&,fil,n]
```

Example: double spreading of a filter of length 3

```
In[38]:= spread[{a, b, c}, 2]
```

```
Out[38]= {a, 0, 0, 0, b, 0, 0, 0, c}
```

One-level WT for images on approximation data

```
WTstep[AX_,AY_,H_,V_,h_,v_]:=
(*
One-level wavelet transform based on x-and y-filtered arrays AX and AY
using low-pass filters H and V to obtain the approximation data Ax and Ay
and using high-pass filters h and v to obtain the detail data Wx and Wy.
Filters are 2-dimensional!
*)
Module[
{Ax,Ay,Wx,Wy},
If[Dimensions[AX]#Dimensions[AY],Throw["dimensions don't match"]];
Ax=ListConvolve[H,AX,{1,1},0];
Ay=ListConvolve[V,AY,{1,1},0];
Wx=ListConvolve[h,AX,{1,1},0];
Wy=ListConvolve[v,AY,{1,1},0];
{Ax,Ay,Wx,Wy}
]
```

Bspline filters of length 2 and 3

```
In[39]:= spline1 = {1, 1} / 2
         spline2 = {1, 2, 1} / 4
```

```
Out[39]= {1/2, 1/2}
```

```
Out[40]= {1/4, 1/2, 1/4}
```

2D low-pass filters constructed from the spline filters

```
In[8]:= H[k_] := KroneckerProduct[spread[spline2, k], spread[spline1, k]];
```

```
In[9]:= V[k_] := KroneckerProduct[spread[spline1, k], spread[spline2, k]];
```

```
H[0] // MatrixForm
```

$$\begin{pmatrix} \frac{1}{8} & \frac{1}{8} \\ \frac{1}{4} & \frac{1}{4} \\ \frac{1}{8} & \frac{1}{8} \end{pmatrix}$$

```
V[2] // MatrixForm
```

$$\begin{pmatrix} \frac{1}{8} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{8} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{8} & 0 & 0 & 0 & \frac{1}{4} & 0 & 0 & 0 & \frac{1}{8} \end{pmatrix}$$

```
In[45]:= Clear[h, v]
```

Spreading the Haar high-pass filter

```
In[46]:= h[k_] := {spread[{1, -1}, k]};
         v[k_] := Transpose[h[k]];
```

```
In[48]:= h[2] // MatrixForm
```

```
Out[48]//MatrixForm=
( 1 0 0 0 -1 )
```

```
In[49]:= v[2] // MatrixForm
```

```
Out[49]//MatrixForm=
( 1
  0
  0
  0
 -1 )
```

Initialize the AX and AY arrays

```
In[50]:= AX = ImageData[img]; AY = ImageData[img];
```

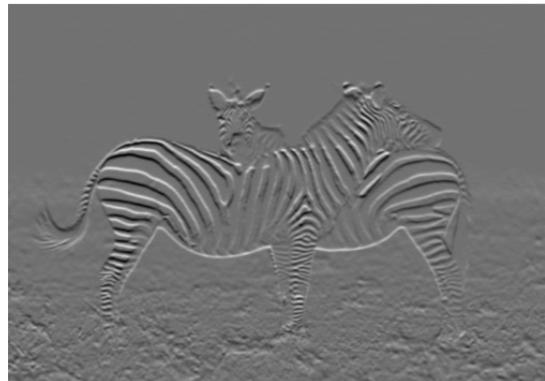
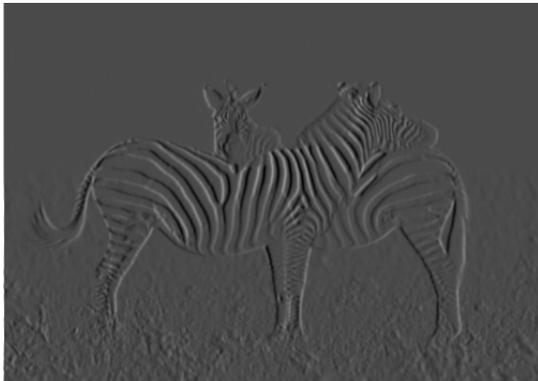
First level of the 2D wavelet transform (using unspreaded filters)

```
In[51]:= {A1x, A1y, W1x, W1y} = WTstep[AX, AY, H[0], V[0], h[0], v[0]];
```

```
In[69]:= GraphicsGrid[{{Image[A1x], Image[A1y]}, {IA[Image[W1x]], IA[Image[W1y]]}}, 1sl]
```



Out[69]=



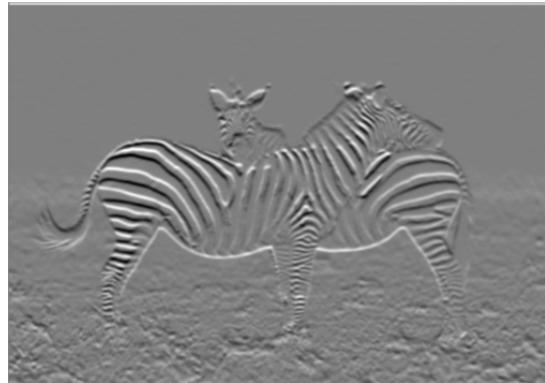
Second level of the 2-dim wavelet transform (using filters spreaded once)

```
In[53]:= {A2x, A2y, W2x, W2y} = WTstep[A1x, A1y, H[1], V[1], h[1], v[1]];
```

```
In[68]:= GraphicsGrid[{{Image[A2x], Image[A2y]}, {IA[Image[W2x]], IA[Image[W2y]]}}, isL]
```



Out[68]=



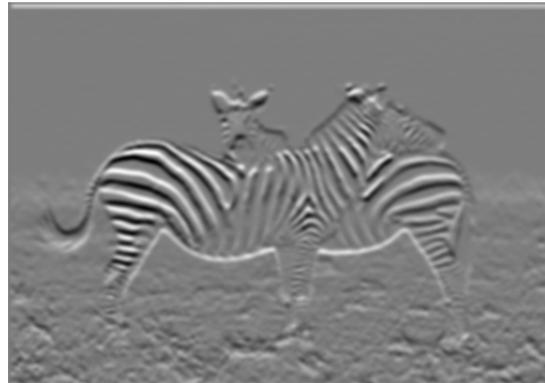
Third level of the 2D wavelet transform (using filters spreaded twice)

```
In[55]:= {A3x, A3y, W3x, W3y} = WTstep[A2x, A2y, H[2], V[2], h[2], v[2]];
```

```
In[67]:= GraphicsGrid[{{Image[A3x], Image[A3y]}, {IA[Image[W3x]], IA[Image[W3y]]}}, iSL]
```



Out[67]=



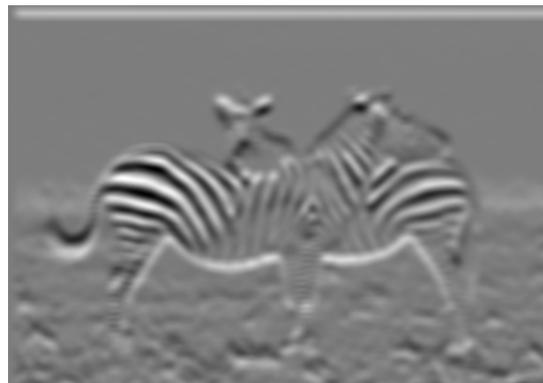
Fourth level of the 2D wavelet transform (using filters spreaded three times)

```
In[57]:= {A4x, A4y, W4x, W4y} = WTstep[A3x, A3y, H[3], V[3], h[3], v[3]];
```

```
In[66]:= GraphicsGrid[{{Image[A4x], Image[A4y]}, {IA[Image[W4x]], IA[Image[W4y]]}}, isL]
```



Out[66]=



Clipping the images to ignore boundary effects

```
In[59]:= cut = Sequence[{17, imdim[[2]]}, {17, imdim[[1]]}]
```

```
Out[59]= Sequence[{17, 357}, {17, 500}]
```

Edge detection based on level-1 detail data

```
In[60]:= kanten1 =  
ImageTake[Image[edges2[W1x, W1y, 2]], cut]
```

Out[60]=



Edge detection based on level-2 detail data

```
In[61]:= kanten2 =  
ImageTake[Image[edges2[W2x, W2y, 2]], cut]
```

Out[61]=



Edge detection based on level-3 detail data

```
In[62]:= kanten3 =  
ImageTake[Image[edges2[W3x, W3y, 2]], cut]
```

Out[62]=



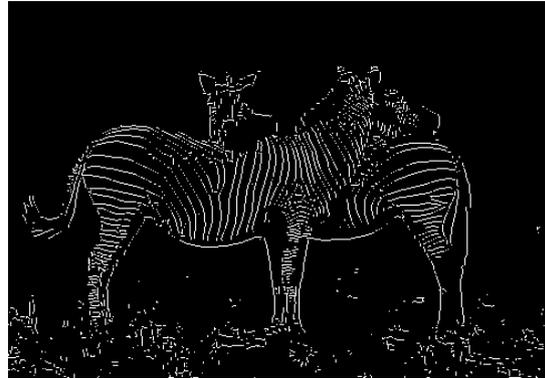
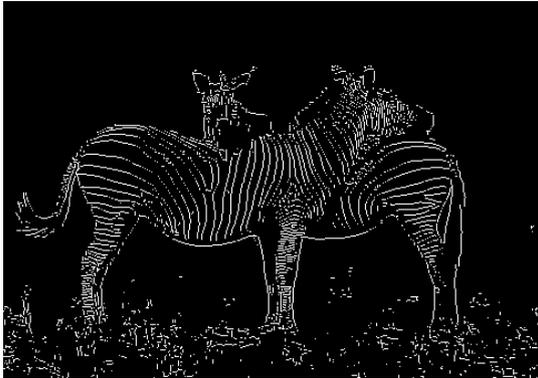
Edge detection based on level-4 detail data

```
In[63]:= kanten4 = ImageTake[  
Image[edges2[W4x, W4y, 2]], cut]
```

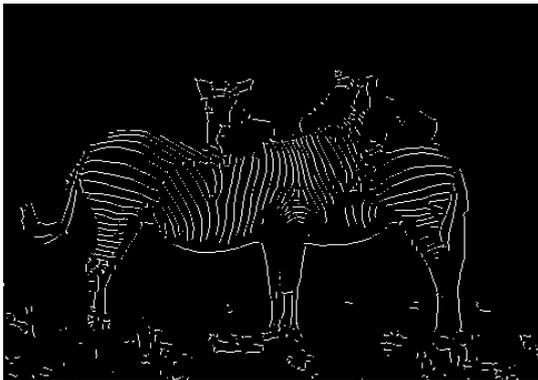
Out[63]=



```
In[65]:= GraphicsGrid[{{kanten1, kanten2}, {kanten3, kanten4}}, isL]
```



```
Out[65]=
```



Operations needed to compensate shifting effects of filtering in different resolution

```
In[12]:= RD[A_,k_]:=RotateRight[A,k];
RU[A_,k_]:=RotateLeft[A,k];
RR[A_,k_]:= (RD[A^T,k])^T;
RL[A_,k_]:= (RU[A^T,k])^T
```

```
RL[{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}}, 2] // MatrixForm
```

$$\begin{pmatrix} 3 & 1 & 2 \\ 6 & 4 & 5 \\ 9 & 7 & 8 \end{pmatrix}$$

Multiplying the data from edge detection at different resolution

```
In[70]:= Image[
  (ImageData[kanten1] *
   RU[RL[ImageData[kanten2], 1], 1]) * RU[RL[ImageData[kanten3], 3], 3]
]
```

```
Out[70]=
```



Weighted sum of the edge detection data

```
In[71]:= Image [  
  ImageData[kanten1] +  
  RU[RL[ImageData[kanten2], 1], 1] / 2 +  
  RU[RL[ImageData[kanten3], 3], 3] / 3 +  
  RU[RL[ImageData[kanten4], 7], 7] / 4]
```

Out[71]=



```
In[72]:= ImageAdjust[%]
```

Out[72]=



Two-level edge detection

```

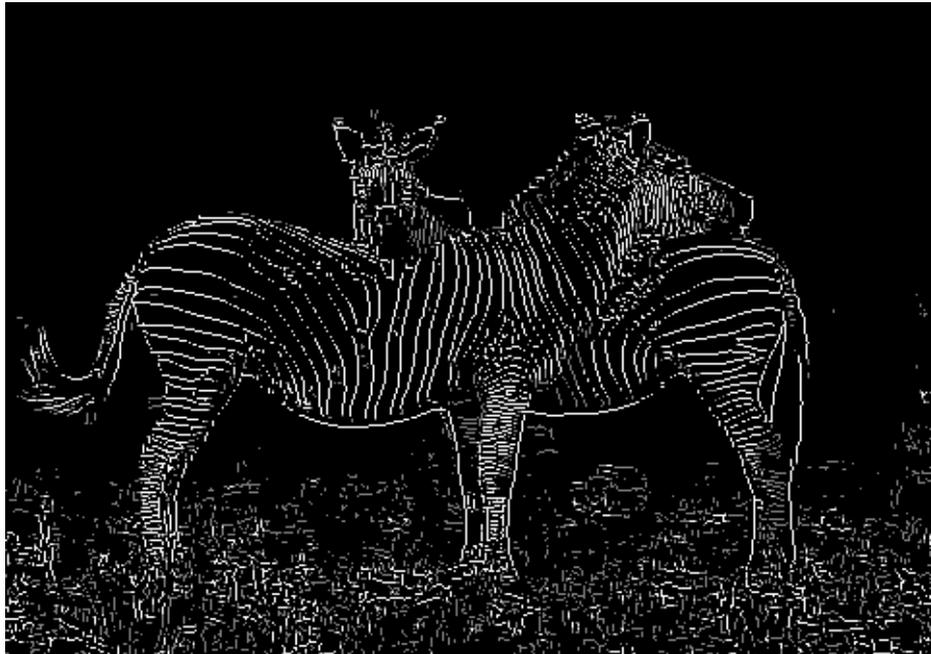
In[16]:= edges3[A_,B_,low_,high_] :=
  (*
  Detecting edge vertices using gradient analysis.
  Input is given as two x- and y-filtered data arrays A and B.
  Output is given in two arrays Clow and Chigh of same
  dimensions as A and B. Shown are all vertices which satisfy
  the edge vertex criterion and where the
  local gradient maximum is ≥ "level"*average gradient
  *)
  Module[{m,n,Val,Dir,mean,VVal,Clow,Chigh,h,v,
  x,x1,x2,y,y1,y2,val,val1,val2},
  {m,n}=Dimensions[A];
  If[{m,n}≠Dimensions[B],
  Throw["Dimensions don't match"]
  ];
  Val=Table[0,{m},{n}];
  Dir=Table[0,{m},{n}];
  Do[
  {h,v}={A[[x,y]],B[[x,y]]};
  Val[[x,y]]=N[Sqrt[h^2+v^2]];
  Dir[[x,y]]=dir2vec[{h,v}],
  {x,1,m},{y,1,n}
  ];
  mean=Mean[Flatten[Val]];
  VVal=ArrayPad[Val,{1,1}];
  Clow=Table[0,{m},{n}];
  Chigh=Table[0,{m},{n}];
  Do[
  {x1,y1}={x+1,y+1}+Dir[[x,y]];
  {x2,y2}={x+1,y+1}-Dir[[x,y]];
  val=VVal[[x+1,y+1]];
  val1=VVal[[x1,y1]];
  val2=VVal[[x2,y2]];
  If[
  val≥Max[val1,val2]&&val≥low mean,
  Clow[[x,y]]=1;
  If[val≥high mean,Chigh[[x,y]]=1]],
  {x,1,m},{y,1,n}
  ];
  {Clow,Chigh}
  ]

```

```
In[73]:= edges3[W1x, W1y, 1, 2];
```

```
In[74]:= ImageAdjust[ImageTake[Image[%[[1]] + %[[2]]], cut]]
```

Out[74]=



Canny-Iteration

```

In[17]:= canny[A_,B_,low_,high_,iter_]:=
(*
Uses edges3 for 2-level edge detection
by turning weak (low level) edge points
into strong (high level) edge vertices
*)
Module[{k,m,n,Alow,AAlow,Ahigh,AAhigh,Diff},
{m,n}=Dimensions[A];
{Alow,Ahigh}=edges3[A,B,low,high];
AAlow=ArrayPad[Alow,{1,1}];
AAhigh=ArrayPad[Ahigh,{1,1}];
Print["strong edge vertices: ",Total[Flatten[Ahigh]]];
Print["weak edge vertices: ",Total[Flatten[Alow]]];
Print["further edge vertices: "]
For[k=1,k<=iter,k++,
Diff=AAlow-AAhigh;
Do[
If[
Diff[[x,y]]==1,
Diff[[x,y]]=
Max[Take[AAhigh,{x-1,x+1},{y-1,y+1}]
],
{x,2,m+1},{y,2,n+1}
];
Print["round ",k," : ",Total[Flatten[Diff]]];
AAhigh=AAhigh+Diff
];
Take[AAhigh,{2,m+1},{2,n+1}]
]

```

Canny iteration of level-1 wavelet data

```
In[75]:= canny[W1x, W1y, 1.5, 3, 10];
```

```
strong edge vertices: 6249
```

```
weak edge vertices: 11911
```

```
further edge vertices:
```

```
round 1 : 803
```

```
round 2 : 351
```

```
round 3 : 221
```

```
round 4 : 145
```

```
round 5 : 94
```

```
round 6 : 66
```

```
round 7 : 39
```

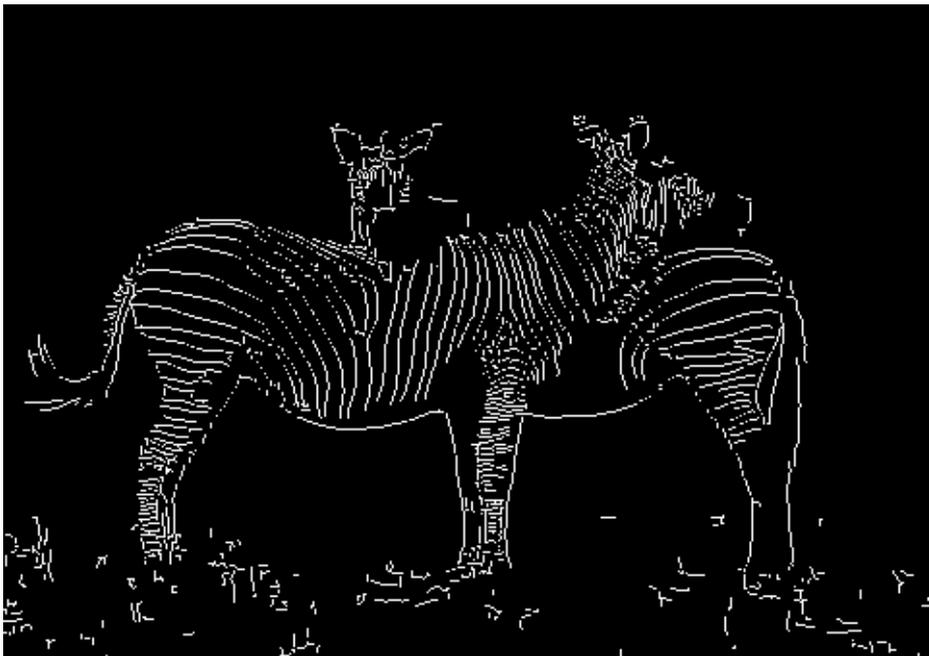
```
round 8 : 26
```

```
round 9 : 21
```

```
round 10 : 15
```

```
In[76]:= c1 = ImageAdjust[ImageTake[Image[%], cut]]
```

```
Out[76]=
```



Canny iteration of level-2 wavelet data

```
In[77]:= canny[W2x, W2y, 1.5, 3, 10];
```

```
strong edge vertices: 5382
```

```
weak edge vertices: 10186
```

```
further edge vertices:
```

```
round 1 : 589
```

```
round 2 : 274
```

```
round 3 : 183
```

```
round 4 : 128
```

```
round 5 : 89
```

```
round 6 : 67
```

```
round 7 : 45
```

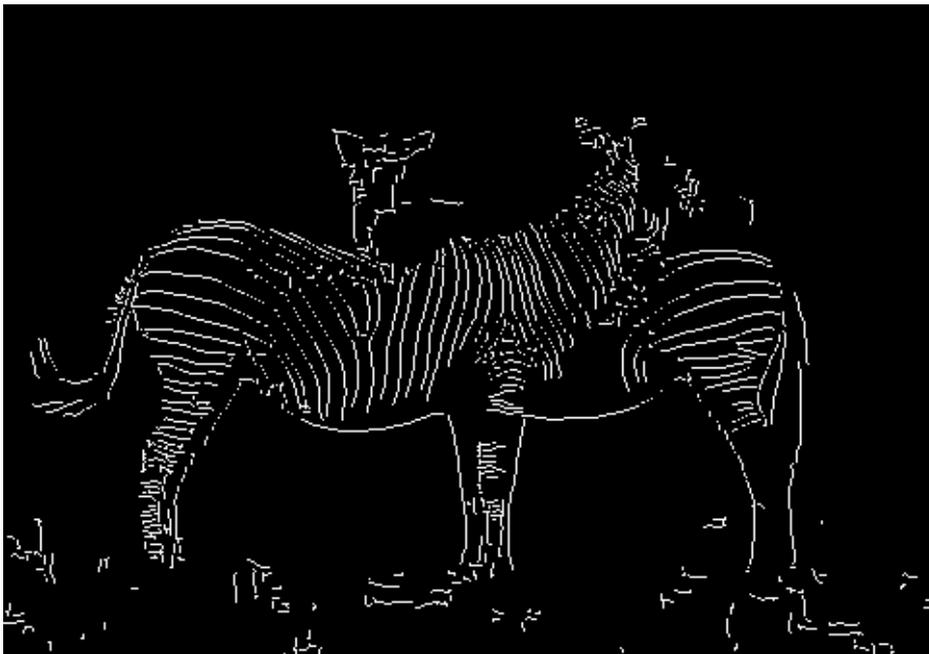
```
round 8 : 35
```

```
round 9 : 25
```

```
round 10 : 20
```

```
In[78]:= c2 = ImageAdjust[ImageTake[Image[%], cut]]
```

```
Out[78]=
```



Canny iteration of level-3 wavelet data

```
In[79]:= canny[W3x, W3y, 1.5, 3, 10];
```

```
strong edge vertices: 4245
```

```
weak edge vertices: 7368
```

```
further edge vertices:
```

```
round 1 : 289
```

```
round 2 : 128
```

```
round 3 : 81
```

```
round 4 : 60
```

```
round 5 : 45
```

```
round 6 : 39
```

```
round 7 : 32
```

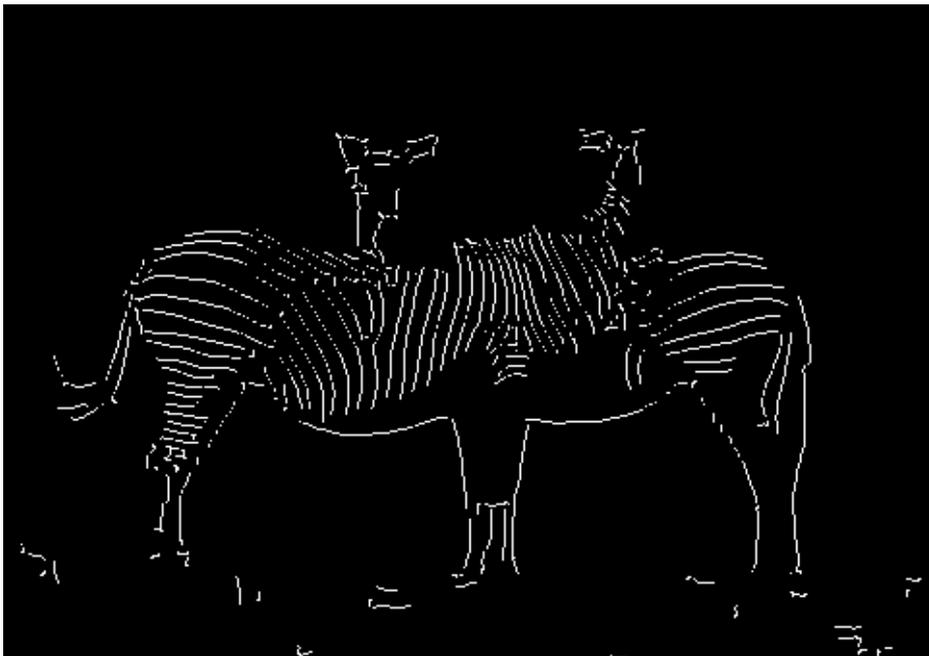
```
round 8 : 28
```

```
round 9 : 22
```

```
round 10 : 19
```

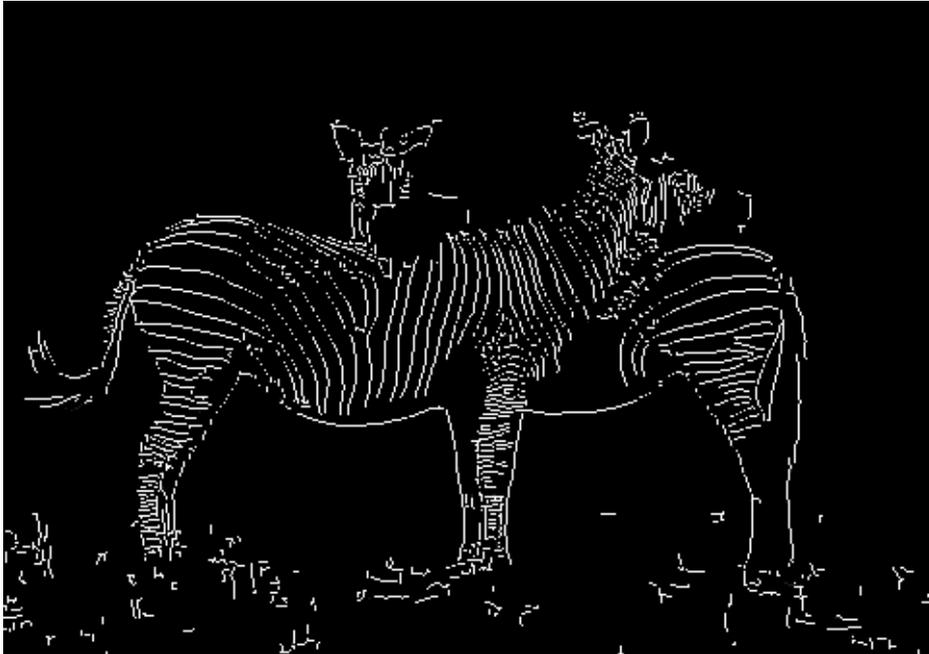
```
In[80]:= c3 = ImageAdjust[ImageTake[Image[%], cut]]
```

```
Out[80]=
```



Weighted average of the data from levels 1-3

```
In[81]:= Image [  
  ImageData[c1] +  
  RU[RL[ImageData[c2], 1], 1] / 2  
  RU[RL[ImageData[c3], 3], 3] / 4]
```



In[82]:= `EdgeDetect[img]`

Out[82]=



15. Joseph Fourier

Dieses Kapitel soll dem Schöpfer der Fourier-Theorie, dem französischen Mathematiker, Physiker und Politiker (!) Jean Baptiste Joseph FOURIER (1768–1830), die Ehre erweisen. Sein Leben zu Zeiten der französischen Revolution (zeitweise eingekerkert und zum Tod durch die Guillotine verurteilt) als Professor an der École Normale, Teilnehmer an Napoleons Expedition nach Ägypten und maßgeblich an deren wissenschaftlichen Auswertung beteiligt¹ . danach Präfekt des Départements Isère in Grenoble, und in der Zeit der Restauration Mitglied und schliesslich Sekretär der *Académie des sciences*, war reich an Höhen und Tiefen und war alles andere als die Routine-Karriere eines Wissenschaftlers.



¹ Fourier stand im Kontakt mit Jean-François CHAMPOLLION (1790–1832), dem Entzifferer der Hieroglyphenschrift (Stein von Rosetta!), und protegierte ihn, u.a., indem er dessen Einberufung zum Militärdienst verhinderte. Champollions Leben und Wirken war ebenfalls sehr aufregend, spannend beschrieben in dem Buch von Andrew Robinson *Wie der Hieroglyphen-Code geknackt wurde – Das revolutionäre Leben des Jean-François Champollion*, Philipp von Zabern, 2014.

Fouriers Name ist untrennbar mit der mathematischen Behandlung des Problems der Wärmeleitung in Festkörpern verbunden². Im Jahr 1807 stellte er seine völlig neuen Ideen unter dem Titel *Théorie mathématique des lois de la propagation de la chaleur et comparaison der résultats de cette théorie à des expériences exactes* erstmals der Pariser Académie des sciences vor. Allerdings wurden diese Ideen zunächst mit viel Skepsis aufgenommen und haben sich erst allmählich und nach der Veröffentlichung seines Hauptwerks *La théorie analytique de la chaleur* [Fou22] im Jahr 1822 durchgesetzt. Es wurde sicher nicht zu Unrecht in eine Liste von 30 *Büchern der Mathematik, die die Welt verändert haben* [SB06] aufgenommen. Eine ausführlichere Würdigung Fouriers und eine Darstellung der Geschichte der Fourierreihen und Wavelets findet man in [KL16].

Aus heutiger Sicht betrachtet waren Fouriers Neuerungen der Ausgangspunkt einer ungemein reichhaltigen und weit verzweigten Entwicklung in Mathematik, Mathematischer Physik und Signalverarbeitung. Es wurde nach dem anfänglichen Zögern schnell klar, dass die Bedeutung und Anwendbarkeit weit über das Lösen spezieller Differentialgleichungen hinaus reicht.

Die *Wärmeleitungsgleichung* in homogenen Medien ist eine (parabolische) partielle Differentialgleichung³:

$$\partial_t u(\vec{x}, t) = \kappa \Delta u(\vec{x}, t),$$

bei der $u(\vec{x}, t)$ die Temperatur an einem Raumpunkt $\vec{x} = (x_1, x_2, x_3)$ zum Zeitpunkt t angibt. $\Delta = \sum_{i=1}^3 \partial_{x_i}^2$ ist der Laplace-Operator bezüglich \vec{x} und die Konstante $\kappa > 0$ ist eine Materialkonstante (Diffusionskonstante), mit der die Temperaturleitfähigkeit des Mediums beschrieben wird. Im eindimensionalen Fall, und nur der soll uns hier interessieren, lautet die Gleichung:

$$\partial_t u(x, t) = \kappa \partial_x^2 u(x, t).$$

Fourier betrachtet diese von ihm aufgestellte Gleichung im Fall eines dünnen Stabes endlicher Länge, für den die eindimensionale Gleichung angenähert zutreffen soll. Auf diesem Stab, der die Länge 1 in passender Maßeinheit haben soll zum Zeitpunkt $t = 0$ eine Temperaturverteilung $f(x)$ mit $f(0) = f(1)$ bestehen soll. Man kann sich also den Stab zu einem Kreis gebogen denken und $f(x)$ als eine 1-periodische Funktion ansehen. Wie wird die Temperaturverteilung $u(x, t)$ zu einem Zeitpunkt $t > 0$ aussehen, wenn ab dem Zeitpunkt $t = 0$ die Wärmeleitungsgleichung allein das physikalische Geschehen bestimmt?

² Man kann lesen, dass er während seiner Zeit als Professor an der Militäarakademie mit dem Problem der effizienten Abkühlung von Kanonenrohren beim fortgesetzten Abfeuern befasst war.

³ Die *Wärmeleitungsgleichung*, auch *Diffusionsgleichung* genannt, ist eine der BIG THREE der partiellen Differentialgleichungen. Die anderen beiden – hier nur im homogenen Fall aufgeschrieben – sind die *Wellengleichung*, der Prototyp einer hyperbolischen partiellen Differentialgleichung:

$$\partial_t^2 u(x, t) = c^2 \Delta u(x, t),$$

sowie die *Laplace-Gleichung* als Prototyp einer elliptischen partiellen Differentialgleichung:

$$\Delta u(x, t) = 0.$$

THÉORIE
ANALYTIQUE
DE LA CHALEUR,
PAR M. FOURIER.



CHEZ FIRMIN DIDOT, PÈRE ET FILS,
LIBRAIRES POUR LES MATHÉMATIQUES, L'ARCHITECTURE HYDRAULIQUE
ET LA MARINE, RUE JACOB, N° 24.

1822.

210 THÉORIE DE LA CHALEUR.

La question que l'on vient de traiter est la première que nous ayons résolue dans la théorie de la chaleur, ou plutôt dans la partie de cette théorie qui exige l'emploi de l'analyse. Elle fournit des applications numériques très-faciles, soit que l'on fasse usage des tables trigonométriques ou des séries convergentes, et elle représente exactement toutes les circonstances du mouvement de la chaleur. Nous passerons maintenant à des considérations plus générales.

SECTION VI.

Développement d'une fonction arbitraire en séries trigonométriques.

207.

La question de la propagation de la chaleur dans un solide rectangulaire a conduit à l'équation $\frac{d^2 v}{dx^2} + \frac{d^2 v}{dy^2} = 0$; et si l'on suppose que tous les points de l'une des faces du solide ont une température commune, il faut déterminer les coefficients a, b, c, d, e , etc. de la série

$$a \cos. x + b \cos. 3x + c \cos. 5x + d \cos. 7x + \dots \text{ etc. ,}$$

en sorte que la valeur de cette fonction soit égale à une constante toutes les fois que l'arc x est compris entre $-\frac{1}{2}\pi$ et $+\frac{1}{2}\pi$. On vient d'assigner la valeur de ces coefficients; mais on n'a traité qu'un seul cas d'un problème plus général, qui consiste à développer une fonction quelconque

Ein zentraler Aspekt von Fouriers Herangehensweise ist der, dass die Funktion $f(x)$ eine *beliebige* 1-periodische Funktion sein kann. Auch über die Funktion $u(x, t)$ zu einem beliebigen Zeitpunkt $t > 0$ wird man also keine spezielleren Annahmen machen können, als über $f(x)$, also lediglich die 1-Periodizität. Fourier postuliert nun, dass sich *jede* 1-periodische Funktion in Form einer trigonometrischen Reihe darstellen lässt, dass man also (an Stelle von sin und cos die komplexen Exponentialfunktionen verwendend) für alle $t \geq 0$ schreiben kann:

$$u(x, t) = \sum_{k \in \mathbb{Z}} c_k(t) e^{2\pi i k x},$$

wobei die Koeffizienten $c_k(t)$ Funktionen der Zeit t sind. Wenn diese Funktion $u(x, t)$ der Wärmeleitungsgleichung genügen soll, so ergibt gliedweises Differenzieren (das man einfach einmal für legitim hält)

$$\sum_{k \in \mathbb{Z}} c_k'(t) e^{2\pi i k x} = \kappa \sum_{k \in \mathbb{Z}} c_k(t) (2\pi i k)^2 e^{2\pi i k x},$$

woraus sich durch Vergleich der Koeffizienten von $e^{2\pi i k x}$ auf beiden Seiten die gewöhnlichen Differentialgleichungen

$$c_k'(t) = -\kappa(2\pi k)^2 c_k(t) \quad (k \in \mathbb{Z})$$

für die Funktionen $c_k(t)$ ergeben. Diese Differentialgleichungen sind ganz einfach zu lösen:

$$c_k(t) = c_k(0) e^{-\kappa(2\pi k)^2 t} \quad (k \in \mathbb{Z}),$$

wobei die Anfangsbedingung

$$f(x) = u(0, x) = \sum_{k \in \mathbb{Z}} c_k(0) e^{2\pi i k x}$$

besagt, dass die Werte $c_k(0)$ nichts anderes sind als die Fourierkoeffizienten der Funktion $f(x)$, also

$$c_k(0) = c_f[k] = \int_0^1 f(y) e^{-\pi i k y} dy \quad (k \in \mathbb{Z}).$$

Damit schreibt sich die gesuchte Funktion $u(x, t)$ als

$$\begin{aligned} u(x, t) &= \sum_{k \in \mathbb{Z}} c_k(t) e^{2\pi i k x} \\ &= \sum_{k \in \mathbb{Z}} c_k(0) e^{-\kappa(2\pi k)^2 t} e^{2\pi i k x} \\ &= \sum_{k \in \mathbb{Z}} \int_0^1 f(y) e^{-\pi i k y} dy \cdot e^{-\kappa(2\pi k)^2 t} e^{2\pi i k x} \\ &= \int_0^1 \left[\sum_{k \in \mathbb{Z}} e^{2\pi i k(x-y)} e^{-\kappa(2\pi k)^2 t} \right] \cdot f(y) dy \end{aligned}$$

Das Integral erkennt man als eine Faltung. Schreibt man nämlich den von der Anfangsbedingung $f(y)$ völlig unabhängigen Summenausdruck als

$$g_t(x) = \sum_{k \in \mathbb{Z}} e^{2\pi i k x} e^{-\kappa(2\pi k)^2 t},$$

so ist

$$u(x, t) = (f \star g_t)(x).$$

Das ist eine höchst interessante Feststellung! Die Lösung $u(x, t)$ des Wärmeleitungsproblems ergibt sich aus der Anfangstemperaturverteilung $u(x, 0) = f(x)$ durch Faltung mit der der “universellen”, nicht von der Anfangsverteilung $f(t)$ abhängigen Funktion $g_t(x)$. Diese “magische” Funktion trägt den Namen *Greens Funktion für das Wärmeleitungsproblem*⁴. Diese Funktion kann man noch etwas umformen:

Lemma 15.1 Die 1-periodische Funktion

$$h_t(x) = \sum_{n \in \mathbb{Z}} e^{-(x-n)^2/2t}$$

mit $t > 0$ hat die Fourierreihe

$$\sqrt{2\pi t} \sum_{m \in \mathbb{Z}} e^{-2\pi^2 m^2 t} e^{2\pi i m x},$$

anders formuliert: für alle $m \in \mathbb{Z}$ gilt

$$\begin{aligned} c_{h_t}[m] &= \int_0^1 h_t(x) e^{-2\pi i m x} dx \\ &= \int_0^1 \sum_{n \in \mathbb{Z}} e^{-(x-n)^2/2t} e^{-2\pi i m x} dx \stackrel{!}{=} \sqrt{2\pi t} e^{-2\pi^2 m^2 t}. \end{aligned}$$

In der Tat:

$$\begin{aligned} \int_0^1 \sum_{n \in \mathbb{Z}} e^{-(x-n)^2/2t} e^{-2\pi i m x} dx &= \sum_{n \in \mathbb{Z}} \int_0^1 e^{-(x-n)^2/2t} e^{-2\pi i m x} dx \\ &= \sum_{n \in \mathbb{Z}} \int_n^{n+1} e^{-x^2/2t} e^{-2\pi i m x} dx \\ &= \int_{-\infty}^{\infty} e^{-x^2/2t} e^{-2\pi i m x} dx \\ &= e^{-2\pi^2 m^2 t} \int_{-\infty}^{\infty} e^{-\xi^2} dx \quad \text{mit } \xi = x/\sqrt{2t} + i\pi m\sqrt{2t} \\ &= \sqrt{2\pi t} e^{-2\pi^2 m^2 t}. \quad \square \end{aligned}$$

Damit ist also

$$g_t(x) = \frac{1}{\sqrt{4\pi\kappa t}} h_{2\kappa t}(x)$$

⁴ Benannt nach dem britischen Mathematiker und Physiker George GREEN (1793–1841)

und man kann die Lösung $u(x, t)$ des Wärmeleitungsproblems noch einfacher schreiben:

$$\begin{aligned} u(x, t) &= \frac{1}{\sqrt{4\pi\kappa t}} \int_0^1 h_{2\kappa t}(x-y) f(y) dy \\ &= \frac{1}{\sqrt{4\pi\kappa t}} \int_0^1 \sum_{m \in \mathbb{Z}} e^{-(x-y-m)^2/4\kappa t} f(y) dy \\ &= \frac{1}{\sqrt{4\pi\kappa t}} \sum_{m \in \mathbb{Z}} \int_m^{m+1} e^{-(x-y)^2/4\kappa t} f(y) dy \\ &= \frac{1}{\sqrt{4\pi\kappa t}} \int_{-\infty}^{\infty} e^{-(x-y)^2/4\kappa t} f(y) dy, \end{aligned}$$

wobei die 1-Periodizität der Funktion $f(t)$ ins Spiel gebracht wird. Das ist nun wieder ein Faltungsintegral:

$$u(x, t) = (f \star \mathfrak{g}_{2\kappa t})(x),$$

wobei

$$\mathfrak{g}_{\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-x^2/2\sigma^2}$$

die Dichtefunktion der Normalverteilung mit Varianz σ^2 ist.

Über die genaue Begründung der Wärmeleitungsgleichung, Fouriers Lösungs-
methode, deren Anwendung auf andere partielle Differentialgleichungen, sowie vielfältige
Lösungsbeispiele und Anwendungen geben die meisten Standardwerke der Ange-
wandten Mathematik, der Partiellen Differentialgleichungen und der Fourieranalysis
Auskunft, beispielsweise [Kam07] oder [Str86] oder [Col05]. Die Bedeutung der Ar-
beiten Fouriers würdigt ausführlich u.a. ein Artikel von Th. Narasimhan [Nar99], der
im Internet frei zugänglich ist.

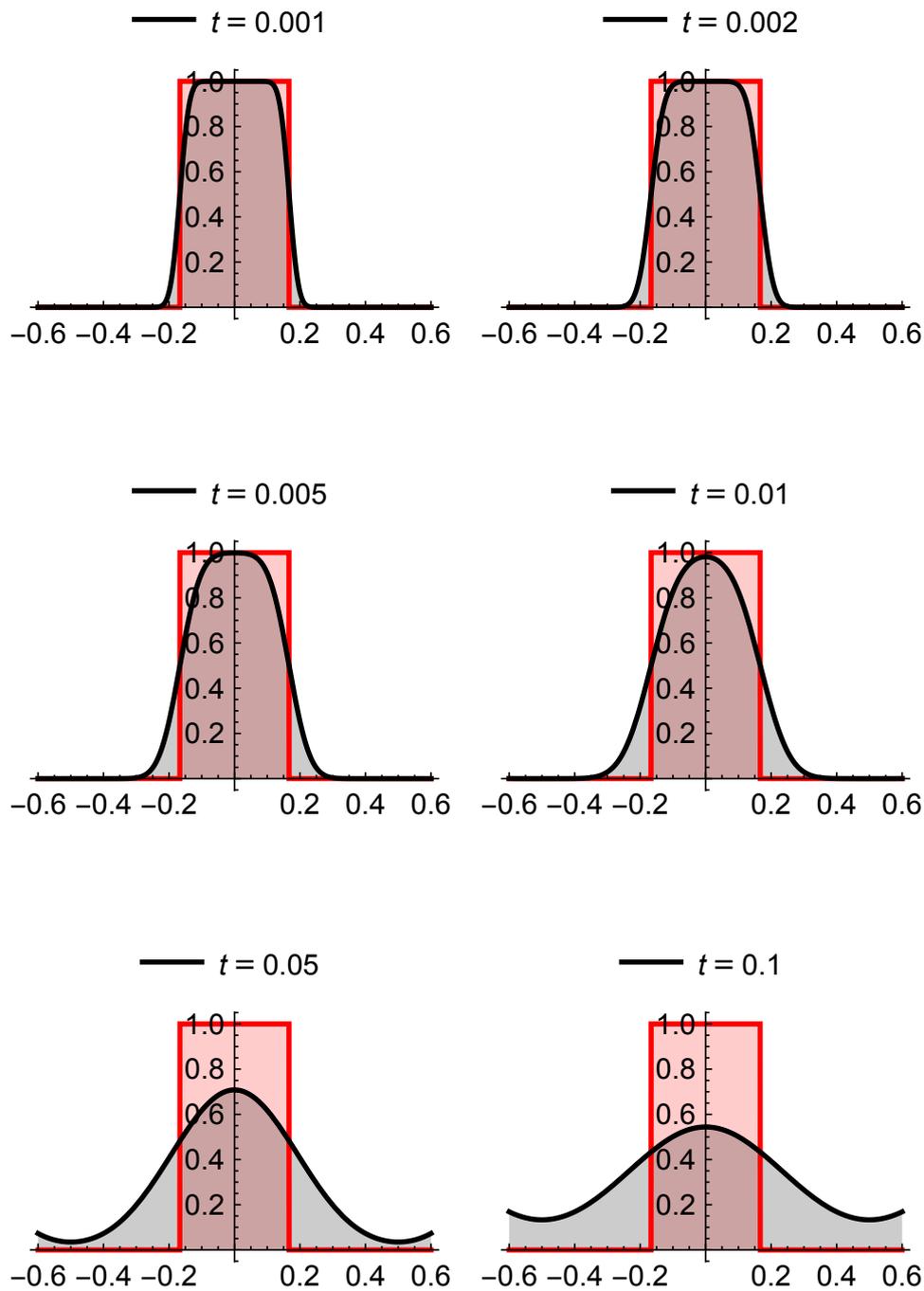


Abbildung 15.1: Zeitliches Verhalten der Wärmeausbreitung auf einem Ring vom Umfang 1 (periodische Situation) ausgehend von einer Box-Funktion der Breite 0.3 als Anfangsverteilung (rot) zu den Zeiten $t = \{0.001, 0.002, 0.005, 0.01, 0.05, 0.1\}$ (schwarz).

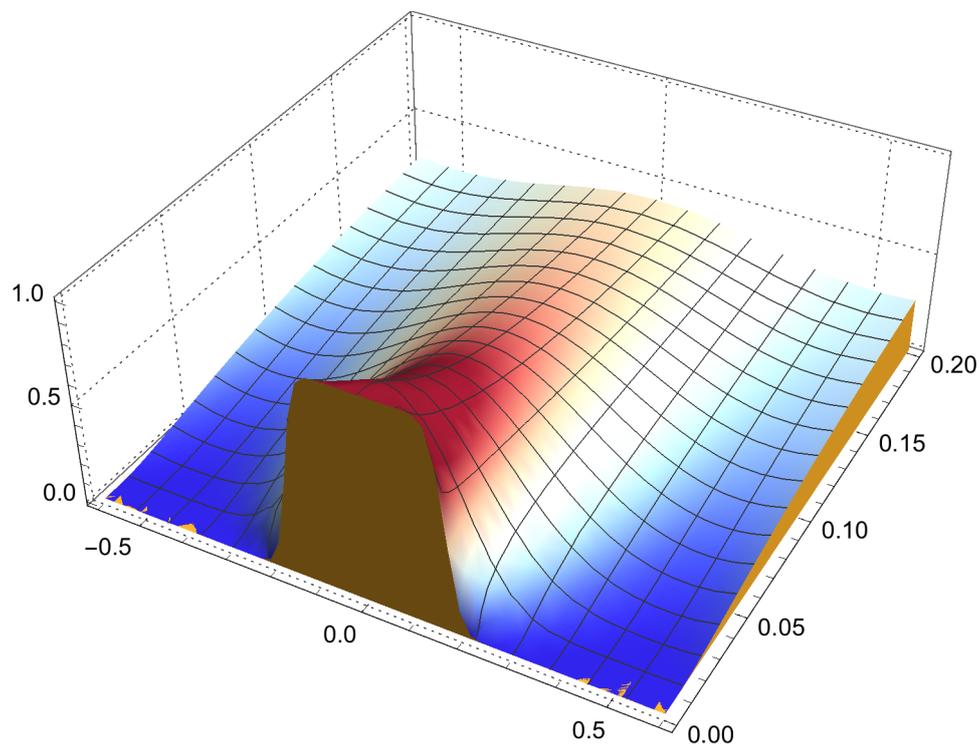


Abbildung 15.2: Zeitliches Verhalten der Wärmeausbreitung auf einem Ring vom Umfang 1 (periodische Situation) ausgehend von einer Box-Funktion der Breite 0.3 als Anfangsverteilung (rot) mit Zeiten $0.001 \leq t \leq 0.2$.

16. Anhang

16.1 Weitere Beispiele zur iterativen Konstruktion

Dieser Abschnitt enthält weitere Illustrationen zur iterativen Konstruktion mittels dyadischer Interpolation von Skalierungsfunktionen, sowie grafische Darstellungen der Waveletfunktionen für die folgenden Wavelets :

- Beylkin-18
- Vaidyanathan-30
- Cohen-Daubechies-Feauveau-9/7

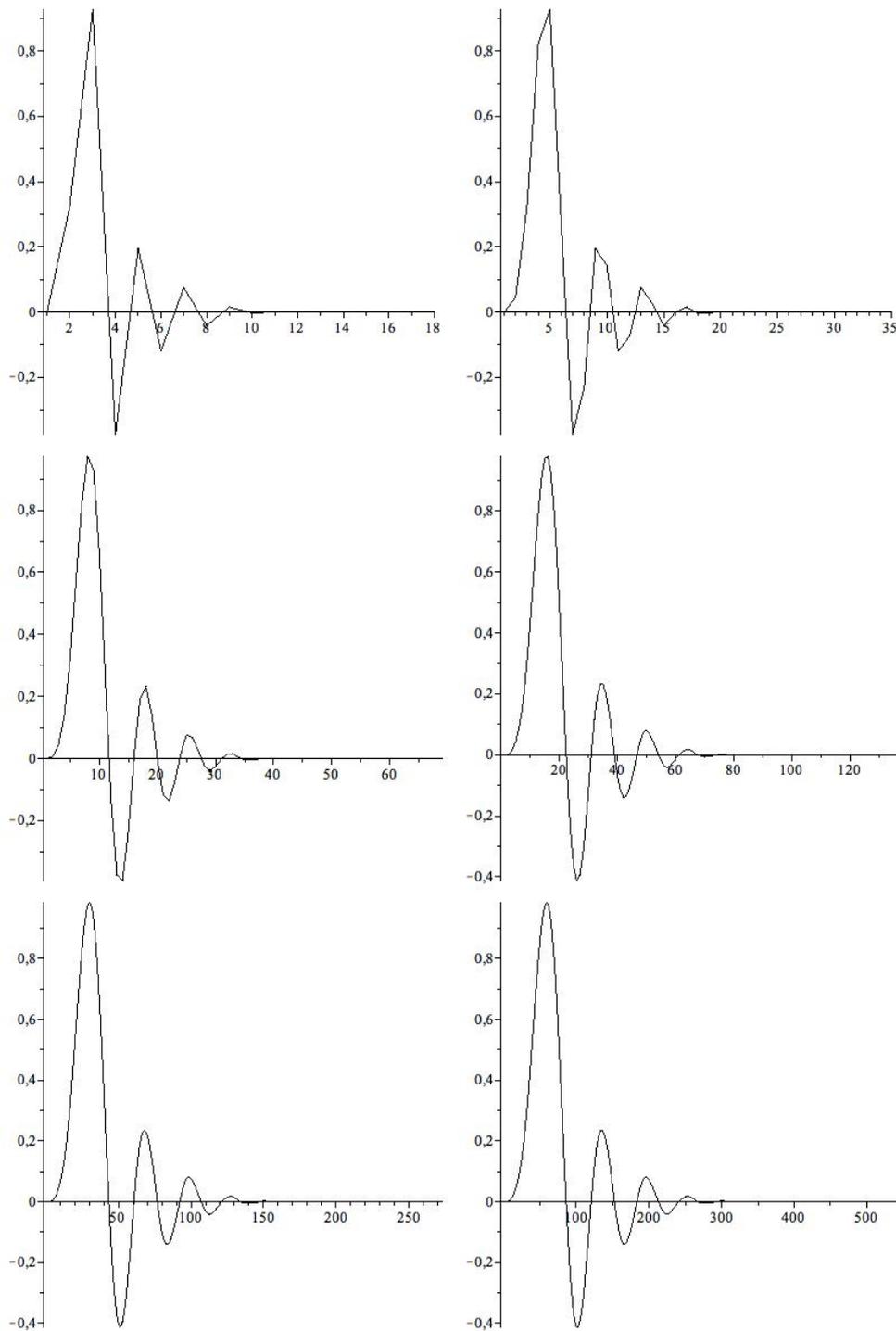


Abbildung 16.1: Iterative Konstruktion der Beylkin-18-Skalierungsfunktion mittels dyadischer Interpolation

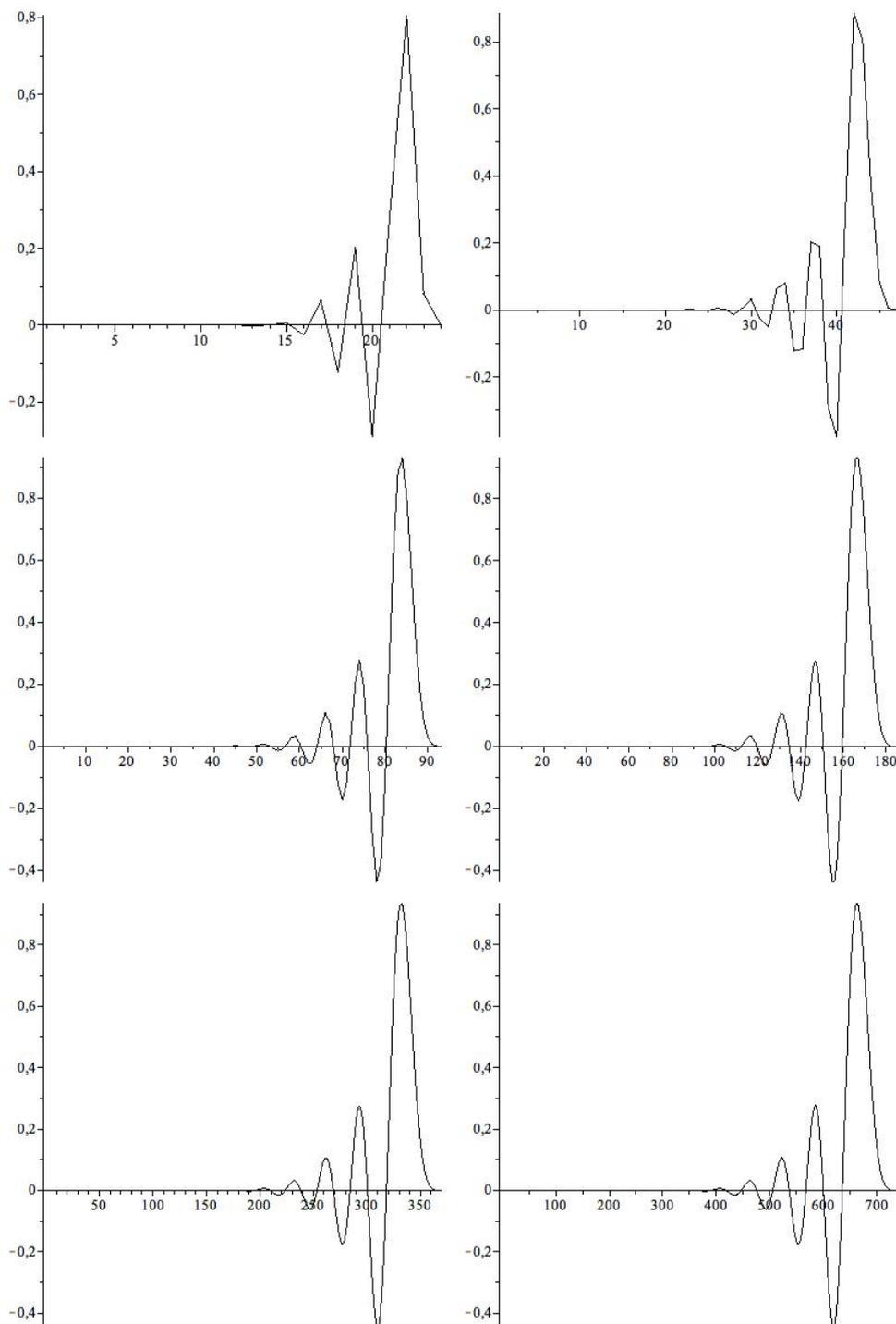


Abbildung 16.2: Iterative Konstruktion der Vaidyanathan-30-Skalierungsfunktion mittels dyadischer Interpolation

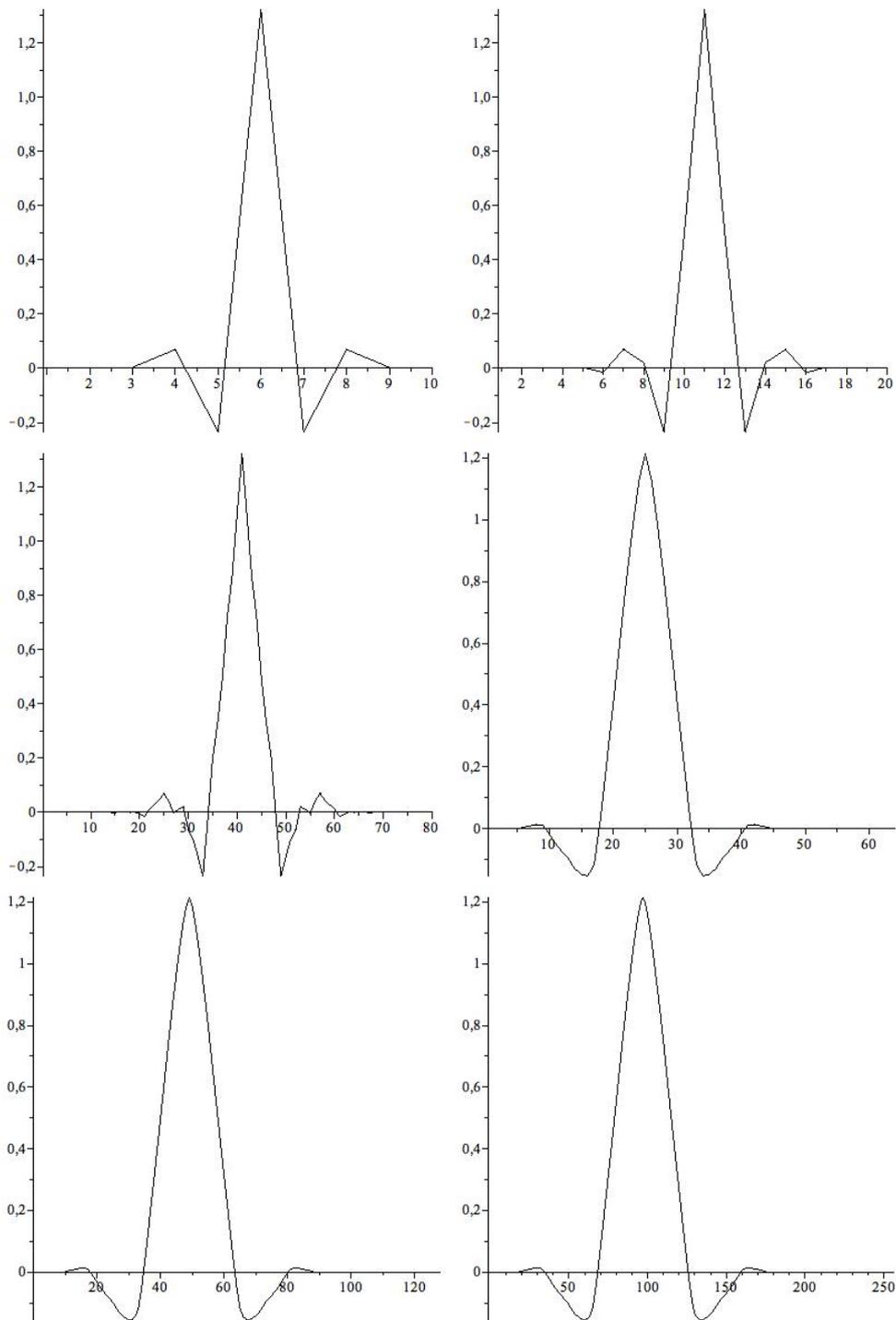


Abbildung 16.3: Iterative Konstruktion der Cohen-Daubechies-Feauveau-9/7-Skalierungsfunktion mittels dyadischer Interpolation

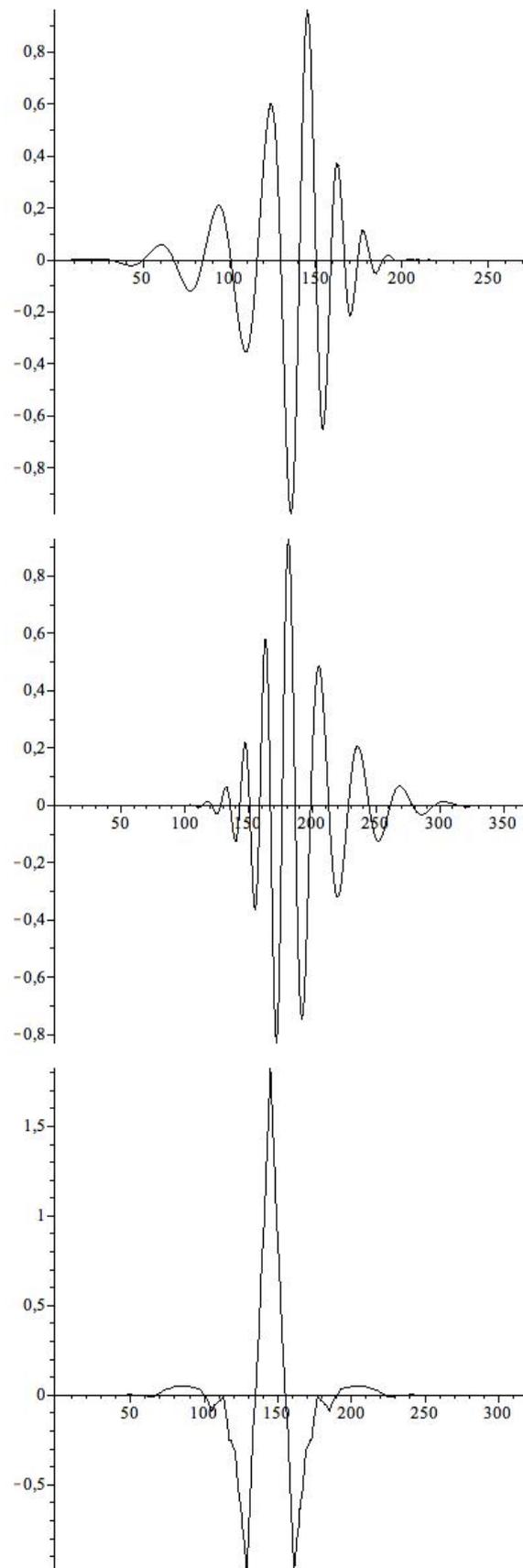
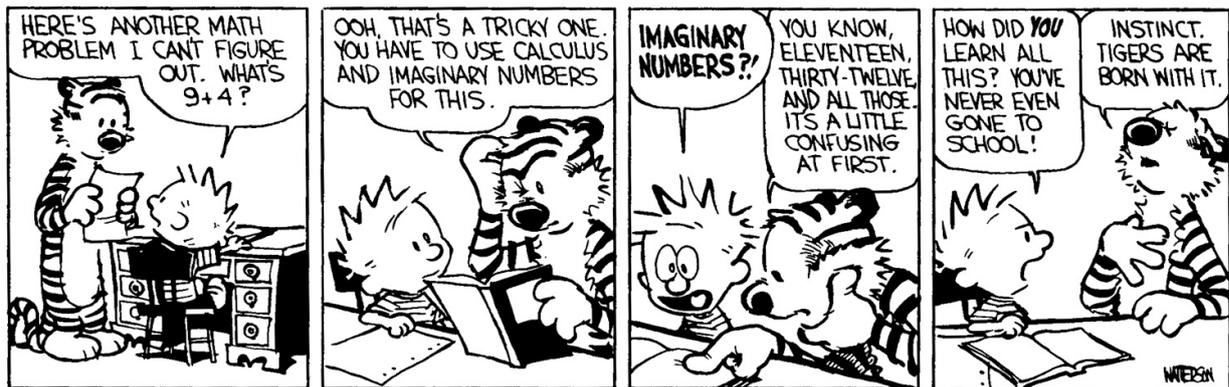


Abbildung 16.4: Waveletfunktionen zu den Filtern Belkin, Vaidyanathan, CFD-9/7

16.2 Fourier's World

Dieser Abschnitt enthält tabellarische Zusammenstellungen wichtiger Informationen und Daten zu den Themen

- Fourierreihen
- Fouriertransformation
- Diskrete Fouriertransformation
- Multiresolution



Fourier's world:

Square-integrable functions on $[0, 1]$ and square-summable bi-infinite sequences (**Fourier series**)

	time/space domain		frequency domain	
square-integrable functions	$\mathcal{L}^2([0, 1])$	name	ℓ^2	square-summable sequences
$\int_0^1 f(t) ^2 dt < \infty$	$f = [0, 1) \rightarrow \mathbb{C}$	objects	$\mathbf{x} = (x[m])_{m \in \mathbb{Z}}$	$\sum_{m \in \mathbb{Z}} x[m] ^2 < \infty$
$\langle f g \rangle = \int_0^1 f(t) \overline{g(t)} dt$		inner product	$\langle \mathbf{x} \mathbf{y} \rangle = \sum_{m \in \mathbb{Z}} x[m] \overline{y[m]}$	
$\ f\ ^2 = \langle f f \rangle = \int_0^1 f(t) ^2 dt$		norm	$\ \mathbf{x}\ ^2 = \langle \mathbf{x} \mathbf{x} \rangle = \sum_{m \in \mathbb{Z}} x[m] ^2$	
$d(f, g) = \ f - g\ $		distance (metric)	$d(\mathbf{x}, \mathbf{y}) = \ \mathbf{x} - \mathbf{y}\ $	
$\omega_k(t) = e^{2\pi i k t}$	$\{\omega_k\}_{k \in \mathbb{Z}}$	ON-basis	$\{\delta_k\}_{k \in \mathbb{Z}}$	$\delta_k[m] = \delta_{k,m}$
$c_f[k] = \langle f \omega_k \rangle$	$f(t) = \sum_{k \in \mathbb{Z}} c_f[k] \omega_k(t)$	basis representation	$\mathbf{x} = \sum_{k \in \mathbb{Z}} x[k] \delta_k$	$x[k] = \langle \mathbf{x} \delta_k \rangle$
$\widehat{\mathbf{x}}(t) = \sum_{k \in \mathbb{Z}} x[k] \omega_k(t)$	$\mathbf{X}(t) = \widehat{\mathbf{x}}(t)$	time/space $\xrightarrow{\mathcal{F}}$ frequency	$\widehat{\mathbf{f}}$	$\widehat{f}[k] = c_f[k] = \langle f \omega_k \rangle$
	$f(t) = \sum_{k \in \mathbb{Z}} \widehat{f}[k] \omega_k(t)$	Fourier's formula	$\mathbf{x} = \sum_{k \in \mathbb{Z}} \widehat{\mathbf{X}}[k] \delta_k$	$\widehat{\mathbf{X}}[k] = \langle \mathbf{X} \omega_k \rangle = x[k]$
	$s_{f,N}(t) = \sum_{k=-N}^N \widehat{f}[k] \omega_k(t)$	approximation	$\mathbf{s}_{x,N} = \sum_{k=-N}^N x[k] \delta_k$	
	$d(s_{f,N}, f) \xrightarrow{N \rightarrow \infty} 0$	convergence	$d(\mathbf{s}_{x,N}, \mathbf{x}) \xrightarrow{N \rightarrow \infty} 0$	
	$\langle \widehat{\mathbf{f}} \widehat{\mathbf{g}} \rangle = \langle \mathbf{f} \mathbf{g} \rangle$	Parseval/Plancherel	$\langle \widehat{\mathbf{x}} \widehat{\mathbf{y}} \rangle = \langle \mathbf{x} \mathbf{y} \rangle$	
	$(f \star g)(t) = \int_0^1 f(x) g(t-x) dx$	convolution	$(\mathbf{x} \star \mathbf{y})[m] = \sum_k x[k] y[m-k]$	
	$\widehat{(f \star g)}[m] = \widehat{f}[m] \cdot \widehat{g}[m]$	convolution theorem	$\widehat{(\mathbf{x} \star \mathbf{y})}(t) = \widehat{\mathbf{x}}(t) \cdot \widehat{\mathbf{y}}(t)$	

Important remark: The overall picture given by this scheme is correct, but note that there are nonetheless difficult convergence problems hidden in formulas which involve infinite sums (linear combinations) of functions – recall the Gibbs-phenomenon! For more on this see the lecture notes.

Fourier's world:

Square-integrable functions on \mathbb{R} (**Fourier transform**)

	time/space domain	name	frequency domain	
square-integrable functions	$\mathcal{L}^2(\mathbb{R})$		$\mathcal{L}^2(\mathbb{R})$	square-integrable functions
$\int_{\mathbb{R}} f(t) ^2 dt < \infty$	$f = \mathbb{R} \rightarrow \mathbb{C}$	objects	$f = \mathbb{R} \rightarrow \mathbb{C}$	$\int_{\mathbb{R}} f(t) ^2 dt < \infty$
	$\langle f g \rangle = \int_{\mathbb{R}} f(t) \overline{g(t)} dt$	inner product	$\langle f g \rangle = \int_{\mathbb{R}} f(t) \overline{g(t)} dt$	
e.g. Haar's basis $\{\psi_{j,k}(t)\}$	ON-basis	ON-basis	e.g. Shannon MRA	
	$\omega_s(t) = e^{2\pi i s t}$	exponentials	$\omega_t(s) = e^{2\pi i s t}$	
$\widehat{f}(s) = \langle f \omega_s \rangle$	$f(t) = \int_{\mathbb{R}} \widehat{f}(s) \omega_s(t) ds$	Fourier representation	$g(s) = \int_{\mathbb{R}} \check{g}(t) \omega_t(s) dt$	$\check{g}(t) = \langle g \overline{\omega_t} \rangle = \int_{\mathbb{R}} g(s) e^{2\pi i s t} ds$
	$f(t)$	time/space $\xrightarrow{\mathcal{F}}$ frequency	$\check{f}(s)$	$\widehat{f}(s) = \int_{\mathbb{R}} f(t) e^{-2\pi i s t} dt$
$\check{g}(t) = \int_{\mathbb{R}} g(s) e^{2\pi i s t} ds$	$\check{g}(t)$	time/space $\xleftarrow{\mathcal{F}^{-1}}$ frequency	$g(s)$	
	$\langle \widehat{f} \widehat{g} \rangle = \langle f g \rangle$	Parseval/Plancherel	$\langle \check{f} \check{g} \rangle = \langle f g \rangle$	
	$(f \star g)(t) = \int_0^1 f(x) g(t-x) dx$	convolution	$(f \star g)(s) = \int_0^1 f(y) g(s-y) dy$	
	$\widehat{(f \star g)}(s) = \widehat{f}(s) \cdot \widehat{g}(s)$	convolution theorem	$(f \star g)^\vee(t) = \check{f}(t) \cdot \check{g}(t)$	

Important remark: In reality the situation is not as “clean” as this scheme suggests. There are serious problems for which functions (in the proper sense) the Fourier transform is defined and is itself a Fourier transformable function. Or how to extend the class of Fourier transformable functions. The problem is with the integrals over an infinite domain and the fact that the exponentials cannot serve as a basis for the $\mathcal{L}^2(\mathbb{R})$ space. Indeed, a satisfactory theory only emerges if one makes the whole space big enough so that exponentials have a Fourier transform. The price to pay is that one must introduce “generalized functions” (a.k.a. *distributions*). This gives rise to a beautiful theory, see some remarks in this direction in the lecture notes and references cited there. For practical purposes it is important to note that the common technique of *sampling* of signals can be discussed in a satisfactory way only if such an extension beyond \mathcal{L}^2 -spaces is made.

By the way: when discussing wavelets we will see methods how orthogonal bases of the space $\mathcal{L}^2(\mathbb{R})$ can be constructed (and used!).

Fourier's world:

Finite-dimensional spaces (Discrete Fourier Transform)

	time/space domain		frequency domain
$\mathbb{Z}_N = \{0, 1, \dots, N-1\}$	\mathbb{C}^N	name	\mathbb{C}^N
$\mathbf{x} : \mathbb{Z}_N \rightarrow \mathbb{C}$	$\mathbf{x} = (x[m])_{m \in \mathbb{Z}_N}$	objects	$\mathbf{x} : \mathbb{Z}_N \rightarrow \mathbb{C}$
	$\langle \mathbf{x} \mathbf{y} \rangle = \sum_{m \in \mathbb{Z}_N} x[m] \overline{y[m]}$	inner product	$\langle \mathbf{x} \mathbf{y} \rangle = \sum_{m \in \mathbb{Z}_N} x[m] \overline{y[m]}$
	$\{ \delta_k \}_{k \in \mathbb{Z}_N}$	ON-bases	$\{ \omega_k \}_{k \in \mathbb{Z}_N}$
$\delta_k = (\delta_{m,k})_{m \in \mathbb{Z}_N}$	$\delta_k = \frac{1}{\sqrt{N}} \sum_{m \in \mathbb{Z}_N} \xi^{-k \cdot m} \omega_m$	basis transformation	$\omega_k = \frac{1}{\sqrt{N}} \sum_{m \in \mathbb{Z}_N} \xi^{k \cdot m} \delta_m$
	$\mathbf{x} = \begin{cases} \sum_{k \in \mathbb{Z}} x[k] \delta_k \\ \sum_{k \in \mathbb{Z}} \langle \mathbf{x} \omega_k \rangle \omega_k \end{cases}$	basis representation	$\hat{\mathbf{x}} = \begin{cases} \sum_{k \in \mathbb{Z}} \overline{\langle \mathbf{x} \omega_k \rangle} \delta_k \\ \sum_{k \in \mathbb{Z}} \langle \mathbf{x} \delta_k \rangle \omega_k \end{cases}$
	\mathbf{x}	\mathcal{F} time/space \rightleftharpoons frequency \mathcal{F}^{-1}	$\hat{\mathbf{x}}$
	$\langle \hat{\mathbf{x}} \hat{\mathbf{y}} \rangle = \langle \mathbf{x} \mathbf{y} \rangle$	Parseval/Plancherel	
	$(\mathbf{x} \star \mathbf{y})[m] = \sum_{k \in \mathbb{Z}_N} x[k] y[m-k]$	convolution	
	$\widehat{(\mathbf{x} \star \mathbf{y})}[m] = \hat{\mathbf{x}}[m] \cdot \hat{\mathbf{y}}[m]$	convolution theorem	

Important remark: The matrix of the basis transformation is the DFT-matrix of order N :

$$DFT_N = [\xi^{k \cdot \ell}]_{0 \leq k, \ell < N} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{N-1} \\ 1 & \xi^2 & \xi^4 & \dots & \xi^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{N-1} & \xi^{(N-1)2} & \dots & \xi^{(N-1)(N-1)} \end{bmatrix}$$

Observe that $\xi^N = 1$, so that in this matrix $\xi^{m \cdot n}$ with $m \geq N$ can be replaced by $\xi^{m \bmod N}$. The inverse matrix DFT_N^{-1} , which is the matrix of the inverse transformation, is obtained by simply replacing ξ by ξ^{-1} everywhere. In proper mathematical terms: the transformation matrices

DFT_N are *unitary*, which means that their inverse is the conjugate transpose of the matrix. since the DFT_N matrices are symmetric, one can forget about transposing, only taking complex conjugates elementwise does the inversion.

These DFT_N matrices are used for the actual computations with coefficient vectors in the standard basis:

$$\hat{\mathbf{x}} = \begin{bmatrix} \langle \hat{\mathbf{x}}, \boldsymbol{\omega}_0 \rangle \\ \langle \hat{\mathbf{x}}, \boldsymbol{\omega}_1 \rangle \\ \langle \hat{\mathbf{x}}, \boldsymbol{\omega}_2 \rangle \\ \vdots \\ \langle \hat{\mathbf{x}}, \boldsymbol{\omega}_{N-1} \rangle \end{bmatrix} = DFT_N \cdot \mathbf{x} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \xi & \xi^2 & \dots & \xi^{N-1} \\ 1 & \xi^2 & \xi^4 & \dots & \xi^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi^{N-1} & \xi^{(N-1)2} & \dots & \xi^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}$$

The DFT -matrices are highly structured and this matrix×vector product can be much more efficiently than using the $\approx N^2$ multiplications that the formula seems to require. This is accomplished by the **Fast Fourier Transform (FFT)** algorithm.

MRA defn

nesting separation completeness $\{0\} \searrow \dots V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \dots \nearrow \mathcal{L}^2(\mathbb{R})$
 scaling $V_j = D_{2^j} V_0$ ($j \in \mathbb{Z}$) $V_0 = \overline{\text{span}}(T_k \phi)_{k \in \mathbb{Z}}$ orthogonality of translates $\langle \phi(t) | T_k \phi(t) \rangle = \delta_{0,k}$

scaling + wavelet

scaling fn $\phi(t)$ scaling id $\phi(t) = \sum_k h_k \phi_1, k(t) = \sum_k h_k \sqrt{2} \phi(2t - k)$ FT: $\hat{\phi}(s) = m_0(s/2) \cdot \hat{\phi}(s/2)$
 wavelet fn $\psi(t)$ wavelet id $\psi(t) = \sum_k g_k \phi_{1,k}(t) = \sum_k g_k \sqrt{2} \phi(2t - k)$ FT: $\hat{\psi}(s) = m_1(s/2) \cdot \hat{\phi}(s/2)$

filter + transfer fn

filter $(h_k)_{k \in \mathbb{Z}}$ $m_0(s) = \frac{1}{\sqrt{2}} h(e^{-2i\pi s}) = \frac{1}{\sqrt{2}} \sum_k h_k e^{-2i\pi s k}$ $|m_0(s)|^2 + |m_0(s + 1/2)|^2 \equiv 1$
 dual filter $\begin{cases} (g_k)_{k \in \mathbb{Z}} \\ g_k = (-1)^k h_{1-k} \end{cases}$ $m_1(s) = \frac{1}{\sqrt{2}} g(e^{-2i\pi s}) = \frac{1}{\sqrt{2}} \sum_k g_k e^{-2i\pi s k}$ $|m_1(s)|^2 + |m_1(s + 1/2)|^2 \equiv 1$

spaces + DWT

approximation spaces $V_j = \overline{\text{span}}(\phi_{j,k})_{k \in \mathbb{Z}}$ orthogonality $\{\psi_{j,k}\}_{j,k \in \mathbb{Z}}$ is ON-basis of $\mathcal{L}^2(\mathbb{R})$
 wavelet (detail) spaces $W_j = \overline{\text{span}}(\psi_{j,k})_{k \in \mathbb{Z}}$ $V_{j+1} = V_j \oplus W_j$ $V_j \perp W_j$
 $\phi_{j,k}(t) = \sum_\ell h_\ell h_{\ell-2k} \phi_{j+1,\ell}(t)$ approx coeffs $a_{j,k} = \langle f | \phi_{j,k} \rangle$ DWT approx $a_{j,k} = \sum_\ell h_{\ell-2k} a_{j+1,\ell}$
 $\psi_{j,k}(t) = \sum_\ell g_\ell g_{\ell-2k} \phi_{j+1,\ell}(t)$ detail coeffs $d_{j,k} = \langle f | \psi_{j,k} \rangle$ DWT detail $d_{j,k} = \sum_\ell g_{\ell-2k} a_{j+1,\ell}$

vanishing moments

always $|\hat{\phi}(n)| = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$ and $\hat{\psi}(0) = \int \psi(t) = 0$

N vanishing moments:
 $\int t^k \psi(t) dt = 0$ ($0 \leq k < N$)
 \rightarrow low/highpass properties, smoothness, quality of reconstruction, compression

equivalent assertions are
 ψ -moments $\int t^k \psi(t) dt = 0$ ($0 \leq k < N$)
 h -moments $\sum_n h_n n^k = 0$ ($0 \leq k < N$)
 QMF $m_0^{(k)}(1/2) = 0$ ($0 \leq k < N$)
 transfer fn $m_0(s) = \left(\frac{1+e^{-2\pi i s}}{2} \right)^N L(s)$

17. Literatur

Lehrbücher und Monografien

- [Add98] Paul S. Addison. *The Illustrated Wavelet Transform Handbook*. Taylor & Francis, 1998.
- [Bän05] Werner Bäni. *Wavelets. Eine Einführung für Ingenieure*. 2nd edition. Oldenbourg, 2005.
- [BEL07] Jöran Bergh, Fredrik Ekstedt, and Martin Lindberg. *Wavelets mit Anwendungen in Signal- und Bildverarbeitung*. Springer, 2007.
- [Bla98] Christian Blatter. *Wavelets – Eine Einführung*. Advances Lectures in Mathematics. Vieweg, 1998 (cited on page 455).
- [BN01] Albert Boggess and Francis J. Narcowich. *A First Course in Wavelets with Fourier Analysis*. 3rd edition. Prentice-Hall, 2001.
- [BB08] Wilhelm Burger and Mark J. Burge. *Digital Image Processing. An Algorithmic Introduction Using Java*. Springer, 2008.
- [Bur96] Barbara Burke Hubbard. *The World According to Wavelets*. A.K. Peters, 1996 (cited on page 219).
- [Car50] H. S. Carslaw. *Introduction to the Theory of Fourier's Series and Integrals*. 3rd edition. Dover, 1950.
- [Col05] Matthew P. Coleman. *An Introduction to Partial Differential Equations with Matlab*. Applied Mathematics and Nonlinear Science. Chapman and Hall/CRC, 2005 (cited on page 563).
- [Dau92] Ingrid Daubechies. *Ten Lectures on Wavelets*. Volume 61. CBMS-NSF Lectures Notes. SIAM, Philadelphia, PA, 1992 (cited on page 269).

- [Eld15] Yonina C. Eldar. *Sampling Theory*. Cambridge University Press, 2015 (cited on page 180).
- [Eps03] Charles L. Epstein. *Introduction to the Mathematics of Medical Imaging*. Pearson Prentice Hall, 2003.
- [Fou22] Joseph Fourier. *Théorie analytique de la chaleur*. Paris: Firmin Didot, Père et Fils, 1822 (cited on pages 114, 558).
- [GW99] Claude Gasquet and Patrick Witomski. *Fourier Analysis and Applications: Filtering, Numerical Computation, Wavelets*. Texts in Applied Mathematics. Springer, 1999 (cited on page 113).
- [Gol61] Richard R. Goldberg. *Fourier Transforms*. Cambridge Tracts in Mathematics. Cambridge University Press, 1961.
- [GW08] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. 3rd edition. Pearson Prentice-Hall, 2008.
- [Hog06] S. G. Hoggar. *Mathematics of Digital Images. Creation, Compression, Restoration, Recognition*. Cambridge University Press, 2006.
- [Hol95] Matthias Holschneider. *Wavelets. An Analysis Tool*. Claredon Press, Oxford, 1995 (cited on page 455).
- [JMR01] Stéphane Jaffard, Yves Meyer, and Robert D. Ryan. *Wavelets. Tools for Science & Technology*. Society for Industrial and Applied Mathematics, 2001 (cited on pages 219, 455).
- [JC01] Arne Jensen and Anders la Cour-Harbo. *Ripples in Mathematics. The Discrete Wavelet Transform*. Springer, 2001.
- [Jer98] Abdal J. Jerri. *The Gibbs Phenomenon in Fourier Analysis, Splines and Wavelet Approximations*. Volume 446. Mathematics and Its Applications. Kluwer Academic Publishers, 1998.
- [KL16] Gilles Kahane and Pierre Gilles Lemarié-Rieusset. *Séries de Fourier et ondelettes*. Volume 3. Nouvelle bibliothèque mathématique. Paris: Cassini, 2016 (cited on pages 455, 558).
- [Kam07] David W. Kammler. *A First Course in Fourier Analysis*. 2nd edition. Cambridge University Press, 2007 (cited on pages 113, 563).
- [Mal09] Stéphane Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd edition. Academic Press, 2009 (cited on pages 408, 455).
- [MP08] Martin Mohlenkamp and Maria Cristina Pereyra. *Wavelets, Their Friends, and What They Can Do for You*. EMS Series of Lectures in Mathematics. European Mathematical Society, 2008.
- [Nie99] Yves Nievergelt. *Wavelets Made Easy*. Birkhäuser, 1999.
- [PW00] Donald B. Percival and Andrew T. Walden. *Wavelet Methods for Time Series Analysis*. Cambridge Series in Statistical and Probabilistic Methods. Cambridge University Press, 2000.

- [PW12] Maria Cristina Pereyra and Lesley A. Ward. *Harmonic analysis*. Volume 63. Student Mathematical Library. From Fourier to wavelets, IAS/Park City Mathematical Subseries. American Mathematical Society, Providence, RI; Institute for Advanced Study (IAS), Princeton, NJ, 2012, pages xxiv+410 (cited on page 113).
- [RB98] Raghuvver M. Rao and Ajit S. Bopardikar. *Wavelet Transforms. Introduction to Theory and Applications*. Addison-Wesley, 1998 (cited on page 408).
- [RW98] Howard L. Resnikoff and Raymond O. Wells Jr. *Wavelet Analysis. The Scalable Structure of Information*. Springer, 1998.
- [Sal02] David Salomon. *A Guide to Data Compression Methods*. New York: Springer-Verlag, 2002 (cited on page 407).
- [SM10] David Salomon and G. Motta. *Handbook of Data Compression*. Springer, 2010 (cited on page 407).
- [SB06] Jean-Jacques Samuëli and Jean-Claude Boudenot. *Trente livres de mathématique qui ont changé le monde*. Éditions ellipses, 2006 (cited on page 558).
- [Say12] Khalid Sayood. *Introduction to Data Compression*. 4th edition. Morgan Kaufmann, Elsevier, 2012 (cited on page 408).
- [Sch65] Laurent Schwartz. *Théorie des distributions*. Editions Hermann, 1965 (cited on page 176).
- [Ste70] E. M. Stein. *Topics in Harmonic Analysis Related to Littlewood-Paley Theory*. Princeton University Press, 1970 (cited on page 220).
- [SGV97] Walter Strampp, Victor Ganzha, and Evgenij Vorozhtsov. *Höhere Mathematik mit Mathematica 4. Funktionentheorie, Fourier- und Laplace-transformationen*. Lehrbuch Computeralgebra. Vieweg, 1997.
- [Str86] Gilbert Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986 (cited on page 563).
- [Van08] Patrick J. Van Fleet. *Discrete Wavelet Transforms*. J. Wiley, 2008.
- [Wal98] James S. Walker. *A Primer on Wavelets and their Scientific Applications*. Chapman and Hall/CRC, 1998.
- [Wal02] David F. Walnut. *An Introduction to Wavelet Analysis*. Birkhäuser, 2002 (cited on page 59).

Zeitschriftenartikel

- [Cal64] A.P. Calderón. “Intermediate spaces and interpolation, the complex method”. In: *Studia Mathematica* 24 (1964), pages 113–190 (cited on page 455).
- [Can86] J. Canny. “A computational approach to edge detection”. In: *IEEE Trans. Patt. Recog. and Mach. Intell.* 6 (1986), pages 961–1005 (cited on page 455).

- [CDF93] Albert Cohen, Ingrid Daubechies, and Jean Christophe Feauveau. “Biorthogonal bases of compactly supported wavelets”. In: *Comm. Pure and Applied Math.* 45.5 (1993), pages 485–560 (cited on page 221).
- [Dau88] Ingrid Daubechies. “Orthonormal bases for compactly supported wavelets”. In: *Comm. Pure. Appl. Math.* 41 (1988), pages 909–996 (cited on pages 221, 269).
- [Dau93] Ingrid Daubechies. “Orthonormal bases for compactly supported wavelets II”. In: *SIAM J. Math. Anal.* 24.23 (1993), pages 499–519 (cited on pages 221, 269).
- [Eng08] Shlomo Engelberg. “Edge Detection Using Fourier Coefficients”. In: *Amer. Math. Monthly* 115 (2008), pages 499–513 (cited on page 127).
- [Gib98a] Josiah W. Gibbs. “Fourier’s series”. In: *Nature* 59.1522 (Dec. 1898), page 200.
- [Gib98b] Josiah W. Gibbs. “Fourier’s series”. In: *Nature* 59.1539 (Apr. 1898), page 606.
- [GM84] Alex Grossmann and Jean Morlet. “Decomposition of Hardy functions into square integrable wavelets of constant shape”. In: *SIAM J. Math. Anal.* 15 (1984), pages 723–736 (cited on pages 220, 455).
- [Haa10] Alfréd Haar. “Zur Theorie der orthogonalen Funktionensysteme”. In: *Mathematische Annalen* 69 (1910), pages 331–371 (cited on page 219).
- [LWW13] Demetrio Labate, Guido Weiss, and Edward Wilson. “Wavelets”. In: *Notices of the AMS* 60.1 (2013), pages 66–76.
- [Mal89a] Stéphane Mallat. “A theory for multiresolution signal decomposition: The wavelet representation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.7 (1989), pages 674–693 (cited on page 221).
- [Mal89b] Stéphane Mallat. “Multiresolution approximation and wavelet orthonormal bases of $L^2(\mathbb{R})$ ”. In: *Transactions Amer. Math. Soc.* 315 (1989), pages 69–87 (cited on page 221).
- [MZ92] Stéphane Mallat and Sifen Zhong. “Characterization of signal from multiscale edges”. In: *IEEE Trans. Patt. Recog. and Mach. Intell.* 14 (1992), pages 710–732 (cited on page 455).
- [Nyq28] Harry Nyquist. “Certain topics in telegraph transmission theory”. In: *AIEE Trans.* 47.2 (1928), pages 617–644 (cited on page 180).
- [SP96] Amir Said and William A. Pearlman. “A new, fast, and efficient image codec based on set partitioning in hierarchical trees”. In: *IEEE Trans. Circ. Syst. Video Technol.* 6.3 (1996), pages 243–250 (cited on page 408).
- [Sha48] Claude E. Shannon. “Communication in the presence of noise”. In: *Proc. Inst. Radio Eng.* 37 (1948), pages 10–21 (cited on page 180).
- [Str83] Jan-Olov Strömberg. “A modified Franklin system and higher-order spline systems on \mathbb{R} as unconditional bases for Hardy spaces”. In: *Conference on Harmonic Analysis in Honor of Antoni Zygmund II* (1983), pages 475–494 (cited on page 220).