Approximationsverfahren I:

Approximation univariater Funktionen

©R. Schaback, Göttingen

Stand:

May 4, 2010

Dieser Text entsteht im Laufe des WS 2009/2010 aus diversen Vorläufermanuskripten. Er ist bis auf Weiteres nur für die Studierenden meiner Vorlesung bestimmt. Ich werde nicht der Historie der Approximationstheorie folgen, sondern neuere Techniken wie z.B. wavelets in den Vordergrund stellen. Leider sind einige Teile noch auf Englisch, ebenso die Silbentrennung.

R. Schaback, May 4, 2010

Contents

Einführung **4**

1 Abstrakte Approximation **7**
1.1 Beste Approximationen 7
1.2 Diskrete Approximation 11

2 Tschebyscheff–Approximation **17**
2.1 Polynome 17
2.2 Alternanten und Referenzen 19
2.3 Remez–Algorithmus 22

3 Approximation in euklidischen Räumen **24**
3.1 Grundlagen 24
3.2 Orthogonalsysteme und Projektoren 28
3.3 Kompression und Beste n–Term–Approximation 31
3.4 Exkurs über Hilberträume 32
3.5 Beispiele von Orthogonalsystemen 35
3.6 Sätze von Weierstraß 46
3.7 Konvergenzgeschwindigkeit von Fourierreihen 54

4 Schnelle Algorithmen **56**
4.1 Trigonometrische Interpolation 56
4.2 Äquidistante Stützstellen 57
4.3 Schnelle Fouriertransformation 59
4.4 Fourier–Kompression 61
4.5 Interpolation in Tschebyscheff–Punkten 64
4.6 Diskrete Cosinustransformation 70
4.7 Baryzentrische Interpolationsformeln 75
4.8 Clenshaw–Curtis–Quadratur 79
4.9 Konvergenzgeschwindigkeit 80

5 Sampling **82**
5.1 Kardinale Interpolation 83
5.2 Skizze zur Fouriertransformation 84
5.3 Die sinc–Funktion 87
5.4 Bandbreitenbeschränkte Funktionen 89
5.5 Beste Approximation in L_2 mit sinc–Funktionen 90
5.6 Sampling Theorem 91
5.7 Berechnen von Shannon–Reihen 94
5.8 Fehlerabschätzung für sinc–Approximation 96

5.9 Aliasing 99
5.10 Direktes Shannon Sampling 99

6 Translationsinvariante Räume **102**
6.1 Grundlagen 103
6.2 Projektion 108
6.3 Approximationsordnung 109
6.4 Fehlerabschätzung 111
6.5 Strang–Fix–Bedingungen 112
6.6 B–Spline–Generatoren 114

7 Das Haar–Wavelet **117**
7.1 Summen und Differenzen 117
7.2 Haarsche Skalierungsfunktion 120
7.3 Multi-Skalen-Analyse 124
7.4 Das Haarsche Wavelet 125
7.5 Die schnelle Wavelet-Transformation 127

8 Allgemeine Wavelets **130**
8.1 Verfeinerbare Funktionen 131
8.2 Strang–Fix–Bedingungen 134
8.3 Wavelets 135
8.4 B–Spline wavelets 140
8.5 Orthogonale Wavelets 141
8.6 Skalierungsfunktionen aus Masken 145
8.7 Wavelets aus Masken 148
8.8 Die wavelets von Ingrid Daubechies 152
8.9 Die allgemeine Wavelet–Transformation 159
8.10 Biorthogonale Wavelets 164
8.11 Wavelet–Fehlerabschätzungen 170

9 Extras **172**
9.1 Fourier Transforms on $\mathbb{R}^d$ 172
9.2 Chebyshev Interpolation and DCT 181
9.3 Splines 191
9.4 B-Splines 198

Index **204**

Literatur **208**

Einführung

Approximationen sind Näherungen (lat.: *proximus*=der/das Nächste). Zu einem gegebenen Objekt versucht man, ein möglichst nahe gelegenes anderes Objekt (die "Approximation") zu finden. Was dabei "nah" heißen soll, wird im Allgemeinen durch eine Distanzfunktion oder eine Norm spezifiziert, und man wählt als Approximation normalerweise etwas Einfacheres als das zu approximierende Objekt. Ein typische Beispiel ist die Approximation von π durch 3.1415 oder die Ersetzung einer n–mal stetig differenzierbaren Funktion f durch den Anfang ihrer Taylorentwicklung

$$f(x) \approx \sum_{j=0}^{n} \frac{f^{(j)}(x_0)}{j!}(x - x_0)^j$$

um einen inneren Punkt x_0 ihres Definitionsbereichs.

Kurzum, Approximationen sind etwas extrem Praktisches, denn sie ersetzen oft ein unzugängliches Objekt f durch ein effizient berechenbares $A(f)$. Die Abbildung A, die einem Objekt f eine Approximation $A(f)$ zuordnet, wird oft als numerisches Verfahren realisiert, und man muß sich um den **Approximationsfehler**

$$\text{dist } (f, A(f)) \text{ oder } \|f - A(f)\|$$

kümmern. Oft spezifiziert man die Menge U, aus der man die Approximation wählen will (z.B. n–stellige Dezimalbrüche bei Approximationen von π, oder Polynome vom Maximalgrad n bei Taylorentwicklungen). Ist U eine Teilmenge des normierten Raums V, aus dem auch das zu approximierende Objekt f stammt, so kann man nach der Approximation $u^* \in U$ fragen, die das Minimierungsproblem

$$\min_{u \in U} \|f - u\|_V = \|f - u^*\|_V$$

löst. Das ist dann die **beste Approximation** an f bezüglich V, und man sieht, daß die Berechnung bester Approximationen sich auf **Optimierung** reduzieren läßt. Natürlich muß man dann beweisen, daß eine beste Approximation existiert und hoffentlich eindeutig ist, und man wird danach fragen, welche zusätzlichen Eigenschaften sie hat und wie man sie effizient ausrechnen kann. Im obigen Beispiel ist die Approximation durch das Taylorpolynom sicher nicht die bestmögliche, und sie setzt unnötigerweise eine hohe Differenzierbarkeit voraus.

Schon bei dem n der obigen Beispiele sieht man, daß die Approximationen oft von Parametern abhängen, die deren Komplexität steuern. Man hat oft geschachtelte Mengen

$$U_0 \subset U_1 \subset \ldots \subset U_n \subset U_{n+1} \subset \ldots$$

von Approximationen und berechnet zu jedem n eine Approximation $u_n \in U_n$ zu f. Natürlich tritt dann sofort die Frage auf, ob $\|f - u_n\|_V$ für $n \to \infty$ gegen Null konvergiert, und wenn ja, wie schnell. Das ist dann eine **asymptotische** Fragestellung, und auch diese Probleme sind natürlich von Interesse. Insbesondere besagt ein wichtiger Satz von Weierstraß, daß man jede stetige Funktion in $V = C[a, b]$ in der Maximumsnorm beliebig gut durch Polynome approximieren kann. Genauer: Zu jeder stetigen Funktion $f \in C[a, b]$ und zu jedem $\epsilon > 0$ gibt es ein Polynom p_n von einem unbekannten Grad n so daß $\|f - p_n\|_{\infty,[a,b]} < \epsilon$ gilt. Man kann also ohne große Verluste jede stetige Funktion durch ein approximierendes Polynom ersetzen.

Die Realisierung von Approximationen als Bestapproximationen kann aber leider sehr aufwendig sein. Einfachere Approximationen, wie z.B. die Taylorentwicklung, lassen sich als simple lineare Abbildungen $A : f \mapsto A(f)$ schreiben, und dann wird man darauf hoffen, trotz der einfachen Struktur der Approximation noch einen ziemlich kleinen Approximationsfehler zu erhalten. Das erfordert einiges an mathematischem Aufwand, aber ist oft lohnend. Die verschiedenen Restgliedformeln für Taylorentwicklungen sind ein typisches Beispiel.

Zwar kann man z.B. auch Approximationen von transzendenten Zahlen durch rationale Zahlen untersuchen, aber dieser Text wird sich auf die Approximation von **Funktionen** (oder Abbildungen) durch einfachere Funktionen beschränken. Fokussiert man auf reellwertige Funktionen, so stellt sich heraus, daß es einen riesigen Unterschied macht, ob man Funktionen von einer oder von mehreren Veränderlichen approximiert. Das spaltet diesen Text in zwei Teile: im Wesentlichen befaßt sich der erste Teil nur mit Funktionen **einer** Variablen. Dazu gibt es eine in den letzten hundert Jahren perfektionierte Theorie, die hier aber nicht in voller Breite ausgewalzt werden kann. Stattdessen sollen moderne lineare Approximationsverfahren behandelt werden, die in der Technik, z.B. bei Kompressionsverfahren wie MPEG oder JPEG, oder beim CAD, dem *Computer–Aided Design*, eine wichtige Rolle spielen. Die Einschränkung auf eine Veränderliche schließt Approximationen von Bildern oder Flächen aus, aber man kann sich auf zeitabhängige Funktionen (z.B. Tonsignale) oder auf Kurven konzentrieren.

In der Geschichte der Approximationstheorie spielten Approximationen durch algebraische und trigonometrische Polynome lange Zeit eine dominierende Rolle. Sie sind immer noch für das Grundverständis approximationstheoretischer Methoden unverzichtbar, und deshalb kommen sie hier auch an zentraler Stelle vor. Aber Polynome haben einige erhebliche Nachteile. Diese werden hier zu analysieren sein, und wir wenden uns dann den Splines und den wavelets als modernen Approximationen zu, die in mancher Hinsicht den Polynomen deutlich überlegen sind.

Der Text wird neben den Anfängervorlesungen auch gewisse Dinge aus der Vorlesung Numerische Mathematik I sowie Kenntnisse in MATLAB–Programmierung voraussetzen. Er hat Überschneidungen mit manchen Realisierungen der Vorlesung Numerische Mathematik II, wird die entsprechenden Passagen aber in voller Länge enthalten. Deshalb wird es manchen Lesern auffallen, daß viele Ähnlichkeiten zum zweiten Teil des Buches [4] bestehen. Dinge aus der Numerischen Mathematik I werden dort, wo sie benötigt werden, ohne Beweis bereitgestellt.

1 Abstrakte Approximation

Hier beginnen wir mit ganz allgemeinen Fragen, die noch nicht Bezug auf spezielle Eigenschaften von Approximationen nehmen. Wir beschränken uns auf reelle Vektorräume und lassen komplexe Approximationen sowie Approximationen in metrischen Räumen weg.

1.1 Beste Approximationen

Beginnen wir mit der Theorie der Best-Approximation in normierten Vektorräumen. Zur Erinnerung:

Definition 1.1. *Eine Funktion* $\|\cdot\| : V \to [0,\infty)$ *heißt* **Norm** *auf einem reellen Vektorraum* V*, falls*

1. $\|x\| = 0$ *genau dann, wenn* $x = 0$.
2. $\|x+y\| \le \|x\| + \|y\|$ *für* $x, y \in V$ *und*
3. $\|\lambda x\| = |\lambda|\|x\|$ *für* $x \in V$ *und* $\lambda \in \mathbb{R}$.

Definition 1.2. *Sei* V *ein normierter Vektorraum und* $U \subseteq V$ *eine nichtleere Teilmenge. Ein Element* $u^* \in U$ *heißt* **beste Approximation** *an ein Element* $f \in V$*, falls*

$$\|f - u^*\| \le \|f - u\| \qquad \text{für alle } u \in U.$$

Die Größe

$$d(f,U) := \inf_{u\in U} \|f-u\| \tag{1.3}$$

wird Abstand *oder genauer* **Minimalabstand** *von* f *zu* U *genannt. Die Menge* U *heißt* **Existenzmenge***,falls es zu jedem* $f \in V$ *(mindestens) eine beste Approximation an* $f \in V$ *aus* U *gibt. Sie heißt schließlich* **Tschebyscheff-Menge**[1]*, wenn es zu jedem* $f \in V$ *genau eine beste Approximation aus* U *gibt.*

Man mache sich klar, daß das Infimum in (1.3) nicht angenommen werden muß und $d(f,U)$ dennoch wohldefiniert ist. Als erstes einfaches Beispiel einer Existenzmenge notieren wir

Lemma 1.4. *Ist* U *eine kompakte Teilmenge eines normierten Raumes, so ist* U *eine Existenzmenge.*

[1] `http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Chebyshev.html`

Beweis: Dies folgt letztlich aus der Stetigkeit der Norm. Genauer definieren wir für $f \in V$ die Funktion $\varphi : V \to \mathbb{R}$ durch $\varphi(v) := \|f - v\|$. Dann ist φ vermöge

$$|\varphi(v) - \varphi(w)| = |\|f - v\| - \|f - w\|| \leq \|v - w\|$$

stetig und nimmt sein Minimum auf der kompakten Menge U an. □

Für Approximationen aus linearen Unterräumen gilt

Theorem 1.5. *Jeder endlich–dimensionale Unterraum U eines normierten, linearen Raums V ist eine Existenzmenge.*

Beweis: Als endlich dimensionaler Raum ist U abgeschlossen. Da $0 \in U$, muss die beste Approximation an ein $f \in V$ bereits aus der Menge

$$U_0 := \{u \in U : \|f - u\| \leq \|f - 0\|\}$$

kommen. Diese Menge ist wegen $\|u\| \leq \|u - f\| + \|f\| \leq 2\|f\|$ aber beschränkt. Man überzeugt sich leicht davon, dass sie auch abgeschlossen ist. Als beschränkte und abgeschlossene Teilmenge eines endlich dimensionalen Raumes U ist sie somit kompakt. Aus Lemma 1.4 folgt daher die Existenz einer besten Approximation an f aus U_0 und damit aus U. □

Die Bedingung an U, abgeschlossen zu sein, ist notwendig für einen Existenzsatz, denn andernfalls gibt es immer ein $f \in V$ mit $d(f, U) = 0$. Und für lineare Unterräume folgt Abgeschlossenheit nicht automatisch, wenn die Dimension unendlich ist. Man gebe dazu ein Beispiel an!

Nach diesen einfachen Fällen bemerken wir noch, dass der Minimalabstand stetig von den zu approximierenden Elementen abhängt.

Theorem 1.6. *Sei V ein normierter Raum und $U \subseteq V$ eine nichtleere Menge. Dann ist $d(\cdot, U)$ Lipschitz–stetig mit Lipschitzkonstanten 1, d.h. es gilt*

$$|d(f, U) - d(g, U)| \leq \|f - g\|$$

für alle $f,\ g \in V$.

Beweis: Seien $f, g \in V$ und $\epsilon > 0$ gegeben. Wir wählen ein $u_\epsilon \in U$, sodass $\|g - u_\epsilon\| \leq d(g, U) + \epsilon$ gilt. Damit folgt

$$d(f, U) \leq \|f - u_\epsilon\| \leq \|f - g\| + \|g - u_\epsilon\| \leq \|f - g\| + d(g, U) + \epsilon,$$

oder mit anderen Worten $d(f,U) - d(g,U) \leq \|f-g\| + \epsilon$. Vertauscht man die Rollen von f und g, so erhält man

$$|d(f,U) - d(g,U)| \leq \|f-g\| + \epsilon.$$

Da dies für beliebiges $\epsilon > 0$ gilt, folgt die Behauptung mit $\epsilon \to 0$. □

Jetzt wollen wir uns um die Eindeutigkeit der besten Approximation kümmern.

Definition 1.7. *Eine Teilmenge U eines reellen Vektorraums V heißt* **konvex***, wenn zu beliebigen Vektoren u, $v \in U$ und beliebigen Zahlen $\lambda \in [0,1]$ die* **Konvexkombination** $\lambda u + (1-\lambda)v$ *wieder in U liegt.*

Man mache sich klar, daß die Konvexkombinationen $\lambda u + (1-\lambda)v$ zu festem u und v genau die Verbindungsstrecke zwischen u und v darstellen. Ferner sind die abgeschlossenen Kugeln

$$K_r(u) := \{v \in V \ : \ \|u-v\|_V \leq r\}$$

immer konvex, insbesondere die **Einheitskugel** $K_1(0)$.

Lemma 1.8. *Ist U eine konvexe nichtleere Teilmenge eines normierten Vektorraums V, und ist $f \in V$ beliebig, so ist die Menge der besten Approximationen zu f aus U konvex.*

Beweis: Hat f keine beste Approximation, so ist nichts zu zeigen. Hat f zwei beste Approximationen u, $v \in U$, so folgt für alle $\lambda \in [0,1]$ sofort

$$\begin{aligned}
\|f - (\lambda u + (1-\lambda)v)\|_V &= \|(\lambda f + (1-\lambda)f) - (\lambda u + (1-\lambda)v)\|_V \\
&= \|\lambda(f-u) + (1-\lambda)(f-v)\|_V \\
&\leq \lambda\|f-u\|_V + (1-\lambda)\|f-v\|_V \\
&= \lambda d(f,U) + (1-\lambda)d(f,U) \\
&= d(f,U)
\end{aligned}$$

und deshalb ist jede Konvexkombination auch beste Approximation. □

Korollar 1.9. *Ist U eine nichtleere konvexe Teilmenge eines normierten Raumes V, und ist $f \in V$ gegeben, so hat f keine oder genau eine oder unendlich viele beste Approximationen bhezüglich U.*

Das folgende einfache Beispiel macht deutlich, dass man im Allgemeinen nicht mit Eindeutigkeit der besten Approximation rechnen kann. Ist nämlich

$V = \mathbb{R}^2$ und $U = \mathbb{R}$, so gilt für $f = (0,1)^T$ und $u = (x,0)^T \in U$ bezüglich der Unendlich-Norm

$$\|f - u\|_\infty = \max(|x|, |1|),$$

sodass jedes Element $u = (x,0)^T$ mit $|x| \leq 1$ beste Approximation an f ist.

Der Ausweg besteht darin, eine weitere Bedingung an die Norm zu stellen, und zwar gerade so, daß der Beweisgang von Lemma 1.8 nicht mehr funktioniert.

Definition 1.10. *Die Norm $\|\cdot\|$ eines normierten Raumes V heißt* **strikt konvex***, falls für alle $f \neq g \in V$ mit $\|f\| = \|g\| = 1$ stets $\|f + g\| < 2$ gilt.*

Offensichtlich ist die Unendlich-Norm auf $\mathbb{R}^2$ nicht strikt konvex, da z.B. die Vektoren $f = (1,1)^T$ und $g = (1,0)^T$ die Voraussetzung der Definition erfüllen, ihre Summe $f + g = (2,1)^T$ aber Norm 2 hat. Wir werden bald sehen, dass die euklidische Norm dagegen strikt konvex ist.

Theorem 1.11. *Sei V ein normierter Raum mit strikt konvexer Norm. Dann ist jede beste Approximation, wenn sie existiert, auch eindeutig. Ferner ist jeder endlich dimensionale Unterraum eine Tschebyscheff-Menge.*

Beweis: Wir müssen nur noch die Eindeutigkeit zeigen. Ist $f \in U$, so ist es selbst seine eindeutige beste Approximation. Ist $f \notin U$, so ist $\eta := d(f,U) > 0$, da das Infimum angenommen wird. Seien also $u_1 \neq u_2 \in U$ beste Approximationen an $f \in V$. Dann ist nach Lemma 1.8 auch $(u_1 + u_2)/2$ eine beste Approximation und daher

$$\|f - u_1 + f - u_2\| = 2\|f - \frac{u_1 + u_2}{2}\| = 2\eta.$$

Dies bedeutet aber, dass $(f - u_1)/\eta \neq (f - u_2)/\eta$ normierte Elemente sind, deren Summe die Norm 2 hat, was im Widerspruch zu der strikten Konvexität steht. □

Die abstrakte Approximation kann man erheblich weiter treiben, aber das lassen wir lieber bleiben, sonst bleibt keine Zeit für Konkretes.

1.2 Diskrete Approximation

Ist der zugrundeliegende normierte Raum endlichdimensional, so spricht man von **diskreter Approximation**. Nehmen wir der Einfachheit halber $V = \mathbb{R}^N$ an, so ist ein zu approximierendes Objekt $f \in V$ nichts anderes als ein N–Vektor. Diese Situation tritt z.B. dann auf, wenn zwar ursprünglich eine Funktion zu approximieren ist, man diese Funktion aber nur an N Stellen $x_1, \ldots, x_N$ kennt (oder benutzt) und dann den Vektor $(f(x_1), \ldots, f(x_N))^T \in \mathbb{R}^N$ approximiert. Approximiert wird dann durch andere Elemente des $\mathbb{R}^N$, und wenn man durch lineare Unterräume U approximiert, kann man sie als Bildmenge einer Abbildung bzw. mit einer $N \times n$–Matrix A als $U = A(\mathbb{R}^n)$ schreiben. Eine beste Approximation Au^* ist dann durch Minimieren von $\|f - Au\|$ über $u \in \mathbb{R}^n$ gegeben.

Diese simple Diskretisierungstechnik wird in einfachen Fällen zum Ausrechnen bester Approximationen benutzt, auch wenn man nicht mit linearen Unterräumen approximiert. Man transformiert das Approximationsproblem in ein endlichdimensionales Optimierungsproblem und wendet Algorithmen der **Optimierung** an. Diese sind dann in der Regel nicht optimal für das gegebene Approximationsproblem, aber sie liefern in vielen Fällen eine brauchbare "quick–and–dirty" Lösung.

Nur die Norm ist noch nicht spezifiziert. Man hat natürlich u.A. alle Normen

$$\|v\|_p^p := \sum_{j=1}^{N} |v_j|^p,\ 1 \le p < \infty,\ \|v\|_\infty := \max_{1 \le j \le N} |v_j|$$

zur Verfügung, und aus der Numerischen Analysis kennt man die **lineare Ausgleichsrechnung** nach der **Methode der kleinsten Quadrate** als den Spezialfall $p = 2$. Wir gehen darauf unten noch einmal etwas allgemeiner ein. Was passiert aber für die anderen p, etwa für $p = 1$ oder $p = \infty$, die **Maximumsnorm** oder **Tschebyscheff–Norm**?

Weil wir bei dieser Gelegenheit auch noch ein paar einführende Beispiele rechnen wollen, geben wir für diese beiden Fälle Standardrezepte an. Dazu schreiben wir f, u und A in Komponenten hin, was zum überbestimmten linearen Gleichungssystem $f = Au$ in der Form

$$f_j = \sum_{k=1}^{n} a_{jk} u_k,\ 1 \le j \le N$$

führt. Im Falle $p = \infty$ hat man dann das Minimierungsproblem

$$\min_{u \in \mathbb{R}^n} \max_{1 \le j \le N} \left| f_j - \sum_{k=1}^{n} a_{jk} u_k \right|,$$

das man auch als lineare diskrete **Tschebyscheff–Approximation** bezeichnet, während man für $p = 1$ auf

$$\min_{u\in\mathbb{R}^n} \sum_{j=1}^{N} \left| f_j - \sum_{k=1}^{n} a_{jk}u_k \right|$$

kommt. Diese Optimierungsaufgaben sehen nichtlinear aus, sind es aber nicht, wenn man sie etwas besser hinschreibt. Im Falle $p = \infty$ kann man

$$\eta := \max_{1\le j\le N} \left| f_j - \sum_{k=1}^{n} a_{jk}u_k \right|$$

einführen und bekommt die $2N$ Bedingungen

$$-\eta \le f_j - \sum_{k=1}^{n} a_{jk}u_k \le \eta,\ 1 \le j \le N.$$

Man minimiert η, wobei η und u freie Variablen sind, und das ist ein lineares Optimierungsproblem. Führt man den Vektor $\mathbf{1}_N \in \mathbb{R}^N$ mit N Einsen ein, so bekommt man in der üblichen komponentenweisen Notierung das Ungleichungssystem

$$\begin{pmatrix} A & -\mathbf{1}_N \\ -A & -\mathbf{1}_N \end{pmatrix} \begin{pmatrix} u \\ \eta \end{pmatrix} \le \begin{pmatrix} f \\ -f \end{pmatrix}$$

und den Zielfunktionsvektor $(\mathbf{0}_n^T, 1)^T$. Wir werden das unten in MATLAB implementieren. Man kann das besser machen, indem man das revidierte duale Simplexverfahren anwendet, aber das gehört in die Optimierung.

Auch für $p = 1$ gibt es diverse geschicktere Strategien, aber wir machen uns hier das Leben einfach. Wir führen $y := f - Au$ ein und trennen y in Positiv–und Negativteil $y = y^+ - y^-$ auf, wobei beide Teile nichtnegativ in allen Komponenten sind. Dann gilt

$$\|f - Au\|_1 = \|y\|_1 = \sum_{j=1}^{N} |y_j| = \sum_{j=1}^{N} (y_j^+ + y_j^-) = \mathbf{1}_N^T (y^+ + y^-)$$

und wir haben eine lineare Zielfunktion. Um in MATLAB leichter programmieren zu können, machen wir dieselbe Aufspaltung bei u. Die Nebenbedingung ist dann

$$A(u^+ - u^-) + y^+ - y^- = f$$

und wir betrachten $u^+,\ u^-\ y^+,\ y^-$ als nichtnegative Variablen. Das Ganze packen wir in ein MATLAB-m-file.

```
function [x, res] = discrapp(b, a, p)
% Diskrete Approximation b=a*x in p-Norm
% fuer p=1,2,\infty. Ausgabe x und res = b-a*x
% Benutzt linprog. Keineswegs optimal.
[m,n]=size(a);
if p==2
    % Ab hier L_2
    x=a\b;  % Loesung
end
% Ab hier Maximumsnorm
if p==Inf
    at=[a -ones(m,1) ; -a -ones(m,1)]; % Matrix
    bt=[b; -b]; % rechte Seite fuer Ungleichungen
    ft=zeros(n+1,1); % Zielfunktion
    ft(end,1)=1;
    xt=linprog(ft,at,bt); % Loesen
    x=xt(1:end-1,1);
end
if p==1
    % Ab hier L_1
    a1=[a -a -eye(m) eye(m) ]; % Matrix
    f1=[zeros(2*n,1);ones(2*m,1)]; % Zielfunktion
    x1=linprog(f1,[],[],a1,b,zeros(2*m+2*n,1),[]); % Loesung
    x=x1(1:n,1)-x1(n+1:2*n,1);
end
res=b-a*x; % Fehlervektor
return
```

Das Programm rekurriert auf `linprog` (man sehe in der MATLAB–Dokumentation nach) und baut die entsprechenden linearen Optimierungsprobleme wie im obigen Text auf. Damit kann man nun überbestimmte lineare Gleichungssysteme durch Approximation der rechten Seite durch die Matrixspalten näherungsweise lösen, und auch dazu gibt es ein MATLAB-m-file:

```
clear all;
close all;
% Programm fuer Diskrete Approximation bei Zufallsmatrizen
m=150 % Zeilenzahl
n=15  % Spaltenzahl
noise=0.01 % Rauschen
```

```
a=2*rand(m,n)-1; % Zufallsmatrix
x=2*rand(n,1)-1; % Loesung
b=a*x+noise*(2*rand(m,1)-1); % verrauschte rechte Seite
[x2, res2]=discrapp(b,a,2);
[xt, rest]=discrapp(b,a,Inf);
[x1, res1]=discrapp(b,a,1);
% Plot der Fehlervektoren
tnorm=max(abs(rest));
figure
plot(1:m,res2,'r.',1:m,rest,'b.',1:m,res1,'g.',...
    1:m,ones(m,1)*tnorm,'b',1:m,-ones(m,1)*tnorm,'b' ,'MarkerSize',20)
legend('L_2','L_\infty','L_1')
title('Diskrete Approximationsfehler im Bildraum')
% Plot der Fehler der Loesungsvektoren
figure
plot(1:n,x2-x,'r.',1:n,xt-x,'b.',1:n,x1-x,'g.','MarkerSize',20)
legend('L_2','L_\infty','L_1')
title('Diskrete Approximationsfehler im Urbildraum')
```

Die Ausgabe ist in den Abbildungen 1 und 2. Die horizontalen Linien geben den Maximalfehler in der Maximumsnorm an, und die beiden anderen Approximationen haben Ausreißer. Im Urbildraum sind die Fehler gegenüber der wahren Lösung nicht deutlich verschieden für die verschiedenen Normen.

Ein weiteres m–file macht das mit Funktionen:

```
clear all;
close all;
% Programm fuer Diskrete Approximation bei Funktionen
h=0.01; % Schrittweite der Funktionswerte
x=(-1:h:1)'; % Auswertungs- und Plotpunkte
k=7        % Polynomgrad
a=chebmat(x,k);  % wir nehmen die Tschebyscheff-Basis
                 % um die Matrix a zu berechnen
[m, n]= size(a);
b=1./(1+125*x.^2);  % Rungefunktion
% b=abs(x);
% b=exp(x);
[x2, res2]=discrapp(b,a,2);   % L_2-Approximation
```

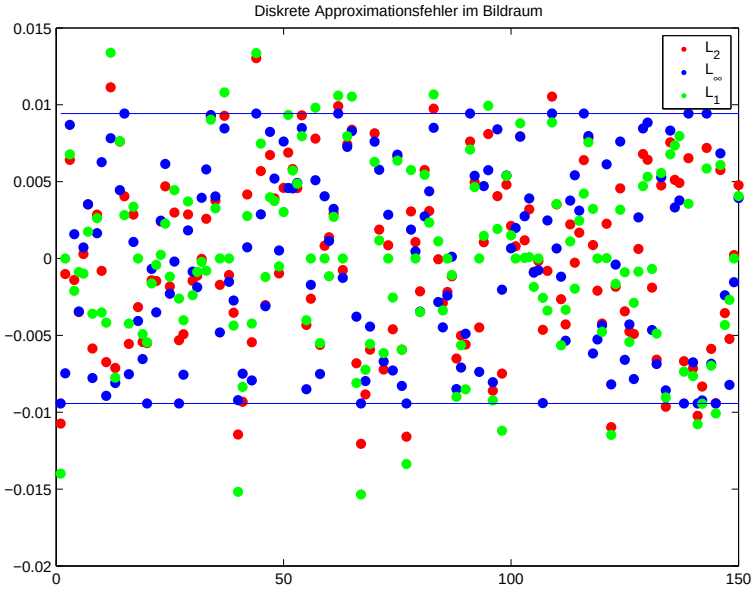

Figure 1: Approximationsfehler im Bildraum

```
[xt, rest]=discrapp(b,a,Inf); % Tschebyscheffapproximation
[x1, res1]=discrapp(b,a,1);   % L_1-Approximation
% Plot der Fehlervektoren
tnorm=max(abs(rest));         % Maximalfehler der T-Approximation
% Berechnung der Interpolation in aequidistanten Punkten
hi=2/k;                   % Schrittweite fuer aequidistante Punkte
xi=(-1:hi:1)';                % die Punkte dazu
fi=1./(1+125*xi.^2);          % die Funktionswerte dazu
resi=b-polyval(polyfit(xi,fi,k),x); % Fehler der Interpolation
% Berechnung der Interpolation in Tschebyscheff-Extremstellen,
% naiv programmiert
xit=cos((0:k)*pi/k);          % Tschebyscheff-Extremstellen
fit=1./(1+125*xit.^2);        % Werte dazu
resit=b-polyval(polyfit(xit,fit,k),x); % Interpolationsfehler
% Plotterei
figure
plot(x,res2,x,rest,x,res1,x,resi, x,resit,...
    x,ones(m,1)*tnorm,'b',x,-ones(m,1)*tnorm,'b' )
legend('L_2','L_\infty','L_1','I. ï¿½quidistant',...
           'I. Tschebyscheff')
title('Diskrete Approximationsfehler im Bildraum')
figure
```

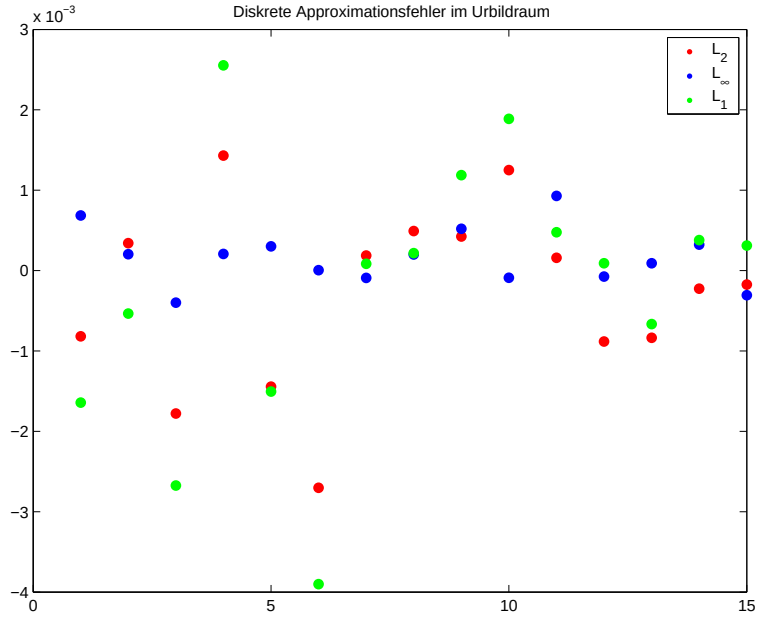

Figure 2: Approximationsfehler im Urbildraum

```
plot(x,b-res2,x,b-rest,x,b-res1,x,b-resi,x,b-resit,x,b)
legend('L_2','L_\infty','L_1','I. ï¿½quidistant',...
            'I. Tschebyscheff','Funktion')
```

Abbildung 3 zeigt die Approximation der Rungefunktion $f(x) = 1/(1 + 125x^2)$ durch Polynome vom Maximalgrad 11, und der entsprechende Fehler ist in Abb. 4 zu sehen. Die horizontalen Linien geben den Fehler der besten Tschebyscheff–Approximation an. Diese nimmt abwechselnd in 13 Punkten (Grad plus 2) ihre Extremstellen an, ganz ähnlich wie bei den Tschebyscheff–Polynomen. Dieses Phänomen wird **Alternation** genannt und wird uns noch beschäftigen. Die Interpolierende ist einmal zu äquidistanten Punkten und dann als Interpolation in den Extremstellen des Tschebyscheff–Polynoms vom Grad 11 gebildet. Man kann deutlich sehen, wie alle anderen Approximationen über das Limit der Tschebyscheff–Approximation hinausschießen. Aber der Fehler der Interpolation in den Tschebyscheff–Extremstellen ist am Rand erstaunlich klein, und wir werden sehen, daß diese Interpolation eine sehr gute, wenn auch nicht optimale Approximation ist, die sich obendrein extrem schnell und stabil ausrechnen läßt. Doch das wird noch eine lange Geschichte.

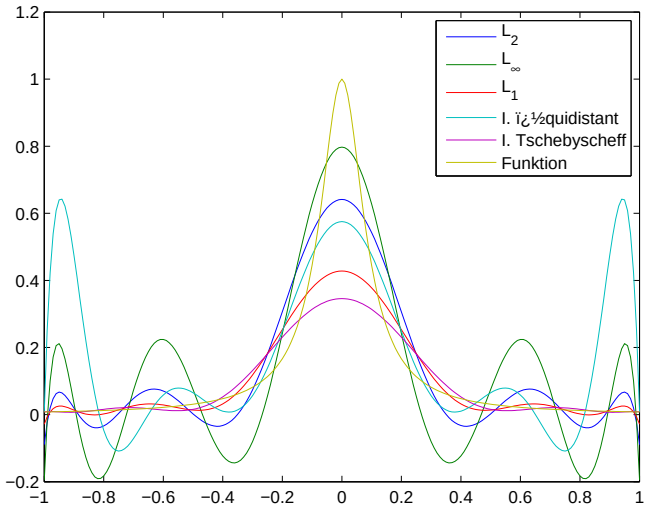

Figure 3: Approximation der Rungefunktion

2 Tschebyscheff–Approximation

Will man eine nicht direkt oder nur über Reihenentwicklungen zugängliche Funktion f wie $\exp(x)$, $\sqrt{x}$ oder $\cos(x)$ auf einem Rechner mit bestmöglicher Genauigkeit und möglichst effizient ausrechnen, so wird man zwecks Vermeidung von Ausreißern versuchen, eine beste Approximation u in der Tschebyscheff– oder Maximumsnorm auszurechnen. Mit Hilfe von Renormierungsformeln wie $\exp(M+x) = \exp(M)\exp(x)$ oder $\sqrt{M^2 x} = M\sqrt{x}$ beschränkt man sich auf ein endliches abgeschlossenes Intervall $I = [a, b]$. Obendrein wollen wir hier nur die Approximation mit Polynomen vom Maximalgrad n behandeln, und damit haben wir das Problem,

$$\min_{p \in \Pi_n} \|f - p\|_{\infty, I}$$

auszurechnen. Wir brauchen etwas Material über Polynome.

2.1 Polynome

In diesem Abschnitt fassen wir uns kurz, weil vieles aus der Numerischen Mathematik [4] bekannt ist.

Die Menge der reellwertigen algebraischen **Polynome** vom Grad höchstens n und mit reellen Koeffizienten wird mit $\Pi_n(\mathbb{R})$ bezeichnet. Wenn Werte und

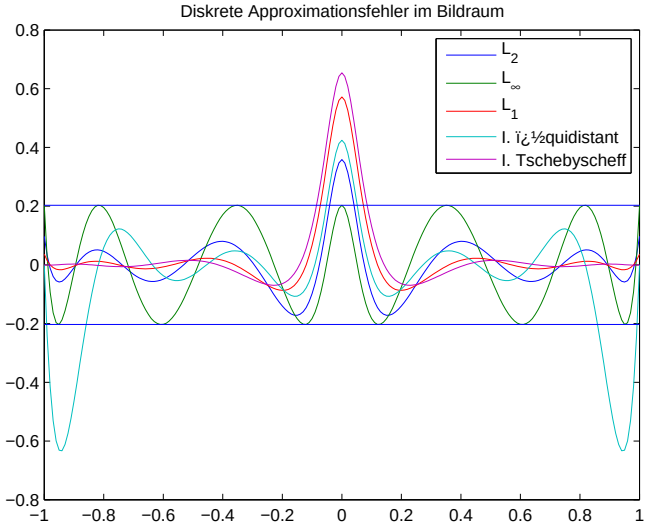

Figure 4: Fehler bei der Rungefunktion

Koeffizienten komplex sind, wird die Bezeichnung $\Pi_n(\mathbb{C})$ verwendet. Der komplexe Fall verhält sich über weite Strecken genau wie der reelle Fall, wird aber weiter unten wichtig, wenn wir trigonometrische Polynome behandeln.

Die **Monombasis** der reellwertigen algebraischen Polynome ist $1, x, x^2, \ldots, x^{n-1}, x^n$. Sie ist nur in einer Umgebung des Nullpunktes numerisch günstig. Wir schreiben damit Polynome als

$$p(x) = a_0 + a_1 x + \ldots + a_n x^n \in \Pi_n(\mathbb{R}).$$

Im Komplexen würde man die Basis natürlich als $1, z, z^2, \ldots, z^{n-1}, z^n$ mit einer komplexen Variablen z schreiben. Wir kommen darauf zurück. In beiden Fällen sichert der Fundamentalsatz der Algebra die Basiseigenschaft. Es gibt wesentlich bessere Basen als die Monombasis. aber darauf kommen wir erst im nächsten Kapitel zurück.

In MATLAB benutzt man die Routinen `polyval` und `polyfit`, um mit Polynomen in der Monombasis zu rechnen. Das ist wegen der miserablen Qualität der Monombasis nicht besonders empfehlenswert, aber es muß hier beschrieben werden, weil die Indizierung etwas ungewöhnlich ist. Ein Polynom P von Grad $\leq n$ wird in MATLAB durch einen Koeffizientenvektor $\mathtt{p} = (p_1, \ldots, p_{n+1})$ mit

$$P(x) = p_1 x^n + p_2 x^{n-1} + \ldots + p_{n+1} = \sum_{j=1}^{n+1} p_j x^{n+1-j}$$

dargestellt. Ist `x` ein Spaltenvektor aus M Punkten und `p` ein Koeffizientenvektor, so liefert `polyval(p,x)` einen M–Spaltenvektor durch Auswertung der obigen Formel. Um das als Matrixmultiplikation zu schreiben, kann man die $M \times (n+1)$–Vandermonde[2]–Matrix `V` mit Elementen x_k^{n+1-j} verwenden. Das wird in der Dokumentation von `polyfit` auch genau so beschrieben, wobei der Aufruf `p=polyfit(x,y,n)` einen Koeffizientenvektor `p` liefert, der sich durch `p=V\y` ergibt, also im Prinzip eine beste diskrete L_2–Approximation des Datenvektors `y`$\in \mathbb{R}^M$ durch Polynome vom Grad $\leq n$ in den M Punkten von `x` ist.

Es kann nicht schaden, sich mal klarzumachen, was in MATLAB durch

```
val=polyval(polyfit(x,y,n),z)
```

berechnet wird.

Für beliebige Basen $u_0, \ldots, u_n$ und ihre Werte in Punkten $x_1, \ldots, x_M$ werden wir das wie oben machen. Wir werden also immer eine $M \times (n+1)$–Matrix von Werten $u_j(x_k)$ erzeugen, d.h. die Punkte entsprechen den Zeilen und die Funktionen den Spalten. Wenn die u_j den Grad j haben sollen, werden wir die Spalten so sortieren, daß die Spalten mit u_n beginnen und mit u_0 enden, wie bei der Monombasis in MATLAB. Hier ist erst einmal die **Vandermondematrix**, aber wir werden später entsprechende Routinen für andere Basen bauen.

```
function V=vander(z,n)
% generates Vandermonde matrix for points z up to degree n
V(:,n+1) = ones(length(z),1);
for j = n:-1:1
    V(:,j) = z.*V(:,j+1);
end
```

2.2 Alternanten und Referenzen

Wir ahnen nach dem Beispiel der Abbildung 4, dass wir mit *Alternation* rechnen müssen, und das definieren wir so:

Definition 2.1. *Eine Punktmenge $x_0 < \ldots < x_{n+1}$ aus $n+2$ Punkten heißt* **Alternante** *zum obigen Approximationsproblem, wenn es ein Polynom $p \in \Pi_n$ und ein Vorzeichen $\sigma \in \{+1, -1\}$ gibt mit*

$$(f-p)(x_j)(-1)^j\sigma = \|f-p\|_{\infty,I},\ 0 \leq j \leq n+1. \tag{2.2}$$

[2] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Vandermonde.html`

Theorem 2.3. *Hat man eine Alternante mit einem Polynom p, so ist p eine beste Approximation. Die Umkehrung gilt auch: jede beste Approximation tritt als Alternante einer $n+2$–punktigen Teilmenge ihrer Extrema auf.*

Beweis: Wir beweisen nur den ersten Teil. Unter den Voraussetzungen der Definition 2.1 wollen wir zeigen, daß

$$\|f-p\|_{\infty,I} \leq \|f-q\|_{\infty,I}$$

für jedes Polynom $q \in \Pi_n$ gilt. Nehmen wir an, es gelte

$$\|f-p\|_{\infty,I} > \|f-q\|_{\infty,I}. \tag{2.4}$$

für ein $q \in \Pi_n$. Dann folgt

$$\begin{aligned}(q-p)(x_j)(-1)^j\sigma &= (q-f)(x_j)(-1)^j\sigma + (f-p)(x_j)(-1)^j\sigma \\ &= (q-f)(x_j)(-1)^j\sigma + \|f-p\|_{\infty,I} \\ &> 0,\ 1 \leq j \leq n+1,\end{aligned}$$

und nach dem Satz von Rolle muß $q-p$ dann mindestens $n+1$ verschiedene Nullstellen haben, was nicht sein kann, wenn nicht schon $q=p$ gilt, aber das kann wegen (2.4) oder der obigen Ungleichungen nicht sein. □.

Auf jeder $(n+2)$–punktigen Teilmenge $X = \{x_0, x_1, \ldots, x_{n+1}\}$ paarweise verschiedener Punkte kann man nun versuchen, eine Art Alternante hinzubekommen, indem man versucht, ein Polynom $p \in \Pi_n$ zu finden, sodaß

$$(f-p)(x_j)(-1)^j\sigma = \|f-p\|_{\infty,X},\ 0 \leq j \leq n+1 \tag{2.5}$$

gilt. Man spricht dann von einer **Referenz**, und der kleine, aber wesentliche Unterschied zu (2.2) besteht darin, daß in (2.5) rechts nur der Wert $\|f-p\|_{\infty,X}$ und nicht $\|f-p\|_{\infty,I}$ steht.

Theorem 2.6. *Man kann immer genau ein Polynom p mit (2.5) finden. Gilt (2.5), so ist p die eindeutig bestimmte beste Approximation zu f auf X, und es folgt die auf de la Vallée–Poussin[3] zurückgehende Einschließung*

$$\|f-p\|_{\infty,X} \leq \min_{q\in\Pi_n} \|f-q\|_{\infty,I} \leq \|f-p\|_{\infty,I} \tag{2.7}$$

des Fehlers der besten Approximation.

[3] `http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Vallee_Poussin.html`

Beweis: Wie in Theorem 2.3 folgt, daß ein p mit (2.5) eine beste Approximation auf X ist. Die eindeutige Lösbarkeit von (2.5) wollen wir konstruktiv beweisen, indem wir ein Verfahren angeben, das die eindeutige Lösung p von

$$(f-p)(x_j)(-1)^j = \eta,\ 0 \le j \le n+1$$

mit einem im Ansatz zunächst beliebigen, aber sich eindeutig ergebenden η konstruiert. Hat man nämlich irgendeine Lösung p, so bildet man die **Differenzenquotienten**

$$[x_0,\ldots,x_{n+1}](f-p) = [x_0,\ldots,x_{n+1}]f + 0 = \eta[x_0,\ldots,x_{n+1}](-1)^j$$

und sieht, daß

$$\eta = \frac{[x_0,\ldots,x_{n+1}]f}{[x_0,\ldots,x_{n+1}](-1)^j}$$

eindeutig bestimmt ist, und p muß die $n+2$ Daten

$$p(x_j) = f(x_j) - \eta(-1)^j,\ 0 \le j \le n+1$$

interpolieren. Das geht eindeutig mit einem Polynom des Grades $\le n+1$, aber weil der $(n+2)$–te Differenzenquotient der Daten verschwindet, ist die Lösung wegen der Newtonschen Interpolationsformel sogar in Π_n. Natürlich gilt dann auch $|\eta| = \|f-p\|_{\infty,X}$ und man hat eine eindeutige beste Approximation auf X und die Referenzeigenschaft (2.5) auf X.

Um (2.7) zu zeigen, braucht man nur die linke Ungleichung anzusehen. Die beste Approximation $p^* \in \Pi_n$ zu f auf ganz I existiert, aber weil p die beste Approximation auf X ist, folgt die Behauptung aus

$$\|f-p\|_{\infty,X} \le \|f-p^*\|_{\infty,X} \le \|f-p^*\|_{\infty,I} = \min_{q\in\Pi_n} \|f-q\|_{\infty,I}. \quad \square$$

Jetzt kann man ahnen, wie man eine beste Approximation auf I schrittweise konstruieren kann: man berechnet eine Folge von Referenzen, die die linke Seite von (2.7) allmählich immer größer macht, bis in (2.7) für eine Referenz X auf einer $(n+2)$–elementigen Teilmenge X Gleichheit herrscht. Dann ist man fertig, und gleichzeitig hat man auch einen noch ausstehenden Teil von Satz 2.3 bewiesen, denn die finale Referenz X ist notwendig eine Alternante. Es ist allerdings dann immer noch nicht bewiesen, daß **jede** beste Approximation eine Alternante enthält, aber das lassen wir offen.

2.3 Remez–Algorithmus

Der soeben grob skizzierte **Remez–Algorithmus** ist nach E. Y. Remez benannt[4][5], aber man muß noch sagen, wie man die Referenzen schrittweise verbessert. Hat man eine Referenz X, so weiß man, daß es eine Approximation gibt, die auf diesen Punkten alterniert und dort beste Approximation ist. Aber die Fehlerfunktion wird anderswo noch größer sein, wenn die Referenz noch keine Alternante ist. Dann gibt es einen Punkt $z \in I$ mit

$$\|f - p\|_{\infty,X} < |(f - p)(z)| = \|f - p\|_{\infty,I},$$

d.h. einen Punkt, an dem die Fehlerfunktion betragsmäßig maximal ist. In eine neue Referenz wird man diesen Punkt aufnehmen wollen, aber welchen der Punkte $x_0 < x_1 < \ldots < x_{n+1}$ der bisherigen Referenz wirft man raus? Grob gesagt: man nimmt erst einmal z in diese Folge auf, berechnet die Vorzeichen von $f-p$ in diesen nunmehr $n+3$ Punkten, und wirft dann einen Punkt $x_j \neq z$ heraus, so dass man in der Restfolge immer noch lauter strikte Zeichenwechsel hat. Genauer geht das mit einer Fallunterscheidung:

1. Gilt $x_{j-1} < z < x_j$ mit $2 \leq j \leq n+1$, so gilt entweder
   $\operatorname{sgn}((f-p)(x_j)) = \operatorname{sgn}((f-p)(z))$ (dann fliegt x_j heraus) oder
   $\operatorname{sgn}((f-p)(x_{j-1})) = \operatorname{sgn}((f-p)(z))$ (dann fliegt x_{j-1} heraus).

2. Gilt $z < x_1$ und $\operatorname{sgn}((f-p)(x_1)) = \operatorname{sgn}((f-p)(z))$, so fliegt x_1 heraus. Andernfalls fliegt x_{n+1} heraus.

3. Analog macht man das im Falle $x_{n+1} < z$.

Hier ist ein entsprechendes Programm, leider noch nicht poliert:

```
clear all;
close all;
% Programm fuer Remesverfahren
h=0.01 % Plot- und Rechenschrittweite
n=15;   % Polynomgrad
x=(-1:h:1)';  % Rechenpunkte
m=length(x);  % Anzahl davon
k=floor(2/(n*h+h)); % Wir wollen in n+2 Punkten starten
ind=1:k:m;  % die etwa aequidistant sind.
ind=ind(1:n+2); % Das sind dann die Punktindizes der Referenz
fx=abs(x); % die zu approximierende Funktion
```

[4] http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Remez.html
[5] http://en.wikipedia.org/wiki/Remez_algorithm

```
% fx=exp(x); % die zu approximierende Funktion
sig=((-1).^(1:n+2))'; % Vorzeichenvektor
while 1==1       % Remes-Schleife
    % wir wollen in den Punkten mit Indizes in ind die b.A. ausrechnen
    xi=x(ind);   % Punkte
    fxi=fx(ind); % Werte
    % Wir brauchen eta als Quotient zweier Differenzenquotienten
    dxc=polyfit(xi,fxi,n+1); % Fitten von f mit Grad n+1
    dxf=dxc(1);              % hoechster Koeff ist Differenzenquotient
    dxc=polyfit(xi,sig,n+1); % Fitten des Zeichenvektors
    dxs=dxc(1);              % hoechster Koeff ist Differenzenquotient
    seta=dxf/dxs;            % Vorzeichen mal eta ist der Quotient
    pf=chebyfit(xi,fxi-seta*sig,n); % Jetzt fitten wir mit Grad n
    res=fx-chebyval(pf,x);    % und dann muss das Residuum alternieren
    figure                   % in den n+2 Interpolationspunkten.
    plot(x,res,xi,res(ind),'rx',x,seta,x,-seta) % plotten
    hold on                  % Jetzt muessen wir das Extremum des Fehlers bestim
    [mf imf]=max(abs(res));
    mr=res(imf);
    plot(x(imf),mr,'ro')
    [abs(mr) abs(seta)]      % das sind die de la Vallee-Poussin Schranken
    disp('Hit Enter')
    pause
    if abs(mr)<abs(seta)*1.01 % Abbruchkriterium
        break;
    end
    % Hier kommt der Indextausch.
    if imf>max(ind) % max is outside right
        if res(ind(end))*mr >0   % same sign at end
            ind=[ind(1:end-1) imf]; % let new point replace
        else                      % different sign at end
            ind=[ind(2:end) imf]; % drop left point
        end
    else
        if imf<min(ind) % max is outside left
            if res(ind(1))*mr >0   % same sign at end
                ind=[imf ind(2:end)]; % let new point replace
            else                     % different sign at end
                ind=[imf ind(1:end-1)]; % drop right point
            end
        else
```

```
            % now the max is somewhere in the middle
            up=min(find(ind>imf));
            down=up-1;
            if res(ind(up))*mr >0 % same sign as upper point
                ind=[ind(1:down) imf ind(up+1:end)]; % replace upper point
            else % same sign as lower point
                ind=[ind(1:down-1) imf ind(up:end)]; % replace lower point
            end
        end
    end
end
```

Ohne Beweis geben wir das zum Programm passende Resultat an:

Theorem 2.8. *Beim Remez–Algorithmus werden $(n+2)$–punktige Referenzmengen X_j mit zugehörigen Polynomen $p_j \in \Pi_N$ konstruiert, für die*

$$\|f - p_j\|_{\infty,X_j} = \min_{p\in\Pi_n} \|f-p\|_{\infty,X_j} \leq \min_{p\in\Pi_n} \|f-p\|_{\infty,I}$$

für $j \to \infty$ mindestens linear und monoton wachsend gegen die obere Schranke konvergiert. Die Referenzmengen X_j und die Polynome p_j haben eine Teilfolge, die gegen die Punktmenge X^* und das Polynom p^* einer Alternante konvergiert. □

Zwei typische Ausgaben des Programms zeigen die Abbildungen 5 und 6. Es handelt sich um die Approximation von $|x|$ auf $[-1,1]$ durch Polynome vom Grad 15, d.h. mit Referenzen der Länge 17. Die horizontalen Linien geben die de la Valleé–Poussin–Schranke an, d.h. der finale Fehler liegt immer zwischen diesem Niveau und dem Betragsmaximum der Fehlerfunktion. Man kann zeigen, daß der Remez–Algorithmus in dieser einfachen Form eine Art von **Simplexverfahren** realisiert. Das ist nicht verwunderlich, weil wir oben schon gesehen haben, daß ein lineares Tschebyscheff–Approximationsproblem sich als lineares Optimierungsproblem schreiben läßt.

3 Approximation in euklidischen Räumen

3.1 Grundlagen

In diesem Abschnitt befassen wir uns mit dem extrem wichtigen Fall, dass die Norm durch ein Skalarprodukt, also eine bilineare, symmetrische und definite Form gegeben ist.

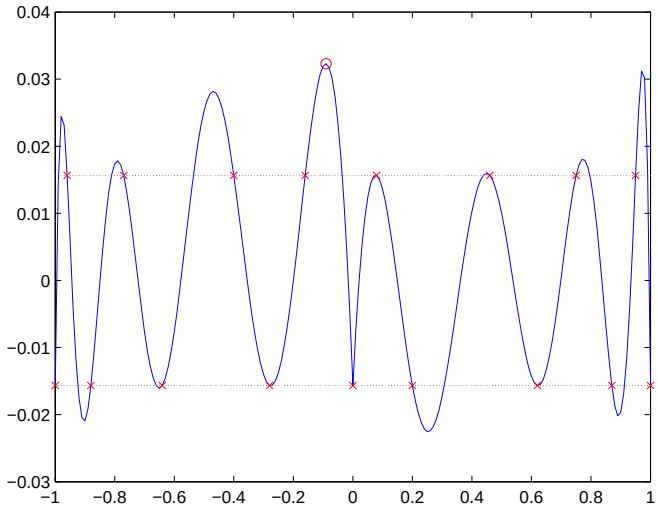

Figure 5: Schritt 12 des Remezverfahrens für $|x|$

Definition 3.1. *Ein normierter Raum V heißt* **euklidisch**[6] *oder auch* **Prä-Hilbertraum**[7]*, falls die Norm durch ein* **Skalarprodukt** *induziert wird, d.h. falls es eine bilineare, symmetrische und definite Form $(\cdot,\cdot) : V \times V \to \mathbb{R}$ gibt, sodass*

$$\|f\| = (f,f)^{1/2}, \qquad f \in V,$$

gilt. Ein vollständiger euklidischer Raum heißt **Hilbertraum**. *Zwei Elemente $f, g \in V$ heißen* **orthogonal**, *falls $(f,g) = 0$. Ein Element f heißt orthogonal zu einem Unterraum $U \subseteq V$, falls f orthogonal zu jedem Element aus U ist.*

In euklidischen Räumen kann man zwischen zwei von Null verschiedenen Elementen u, v einen Winkel φ definieren durch

$$\cos(\varphi) = \frac{(u,v)}{\|u\|\|v\|}.$$

Ferner gilt die **Parallelogrammgleichung**

$$\|u+v\|^2 + \|u-v\|^2 = 2\|u\|^2 + 2\|v\|^2 \text{ für alle } u, v \in V$$

und im Falle $(u,v) = 0$ auch der Satz des **Pythagoras**[8]

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

[6] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Euclid.html`
[7] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Hilbert.html`
[8] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Pythagoras.html`

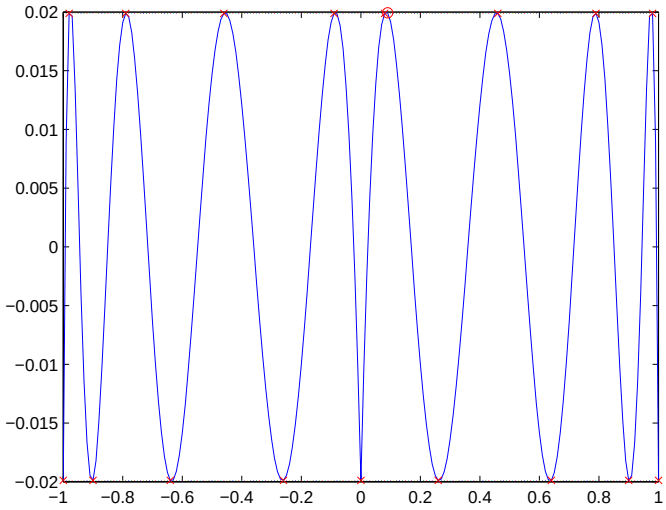

Figure 6: Ende des Remezverfahrens für $|x|$

Beides ist durch Ausmultiplizieren wie in

$$\|u+v\|^2 = (u+v, u+v) = (u,u) + 2(u,v) + (v,v) = \|u\|^2 + 2(u,v) + \|v\|^2$$

leicht zu beweisen.

Wie bereits vorher angedeutet, besitzen euklidische Räume strikt konvexe Normen:

Theorem 3.2. *Die Norm eines euklidischen Raumes ist strikt konvex.*

Beweis: Seien $f \neq g \in V$ mit $\|f\| = \|g\| = 1$ gegeben. Dann folgt aus der Parallelogrammgleichung sofort

$$\|f+g\|^2 < 2\|f\|^2 + 2\|g\|^2 = 4.$$

□

Damit können wir zusammen mit Satz 1.11 sofort eine wichtige Folgerung ziehen.

Korollar 3.3. *Ist V ein euklidischer Raum, dann ist jeder endlich dimensionaler Unterraum eine Tschebyscheff-Menge.*

Als nächstes kümmern wir uns um die Charakterisierungen bester Approximationen in euklidischen Räumen.

Theorem 3.4. *Sei V ein euklidischer Raum und $U \subseteq V$ ein Unterraum. Ein Element $u^* \in U$ ist beste Approximation aus U an ein $f \in V$ genau dann, wenn*

$$(f - u^*, u) = 0 \qquad \textit{für alle } u \in U. \tag{3.5}$$

Beweis: Nehmen wir zunächst an, dass ein $u^* \in U$ die Bedingung (3.5) erfüllt. Dann können wir jedes $u \in U$ als $u = u^* - \widetilde{u}$ mit $\widetilde{u} \in U$ schreiben, womit aber

$$\|f - u\|^2 = \|f - u^* + \widetilde{u}\|^2 = \|f - u^*\|^2 + 2(f - u^*, \widetilde{u}) + \|\widetilde{u}\|^2 \geq \|f - u^*\|^2$$

folgt. Also ist u^* beste Approximation an f aus U.

Ist andererseits $u^* \in U$ beste Approximation und gilt (3.5) nicht, so existiert ein $\widetilde{u} \in U$ mit $(f - u^*, \widetilde{u}) =: c \neq 0$. Wir suchen jetzt eine bessere Approximation als u^* auf der Geraden durch u^* mit Richtung $\widetilde{u}$. D.h. wir setzen $u_t = u^* - t\widetilde{u}$. Wie eben erhalten wir

$$\|f - u_t\|^2 = \|f - u^*\|^2 + 2ct + t^2\|\widetilde{u}\|^2 =: \|f - u^*\|^2 + \psi(t).$$

Wir erhalten also einen Widerspruch, wenn wir ein $t \in \mathbb{R}$ finden mit $\psi(t) < 0$. Da ψ eine nach oben geöffnete Parabel ist, ist der beste Kandidat gegeben durch $0 = \psi'(t) = 2c + 2t\|\widetilde{u}\|^2$, also durch $t^* = -c/\|\widetilde{u}\|^2$. Dieses t^* führt wegen $\psi(t^*) = -c^2/\|\widetilde{u}\|^2$ auch tatsächlich zum Erfolg. □

Da man in (3.5) insbesondere $u = u^*$ einsetzen kann, folgt sofort:

Korollar 3.6. *Unter den Voraussetzungen von Satz 3.4 gilt der Satz des Pythagoras*

$$\|f - u^*\|^2 + \|u^*\|^2 = \|f\|^2, \tag{3.7}$$

welcher die Stabilitätsabschätzungen

$$\|f - u^*\| \leq \|f\| \qquad \textit{und} \qquad \|u^*\| \leq \|f\| \tag{3.8}$$

zur Folge hat.

Mit (3.5) hat man außerdem ein Mittel an der Hand, die beste Approximation zu berechnen.

Korollar 3.9. *(Normalgleichungen) Ist* $\{u_1, \ldots, u_n\}$ *eine Basis von* U*, so gilt mit den Bezeichnungen und Voraussetzungen von Satz 3.4*

$$u^* = \sum_{j=1}^{n} c_j u_j,$$

wobei die $\{c_j\}$ *Lösung des Gleichungssystems*

$$\sum_{j=1}^{n} c_j (u_j, u_k) = (f, u_k), \qquad 1 \le k \le n,$$

sind.

Die hierbei auftretende Matrix $((u_j, u_k))_{j,k}$ nennt man **Gramsche Matrix**. Man kann leicht einsehen, daß sie symmetrisch und positiv definit ist.

Als Beispiel wollen wir die lineare Ausgleichsrechnung betrachten. Dabei ist $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ bei $m > n$ gegeben. D.h. das lineare Gleichungssystem $Ax = b$ ist überbestimmt und man versucht daher $\|Ax - b\|_2$ über alle $x \in \mathbb{R}^n$ zu minimieren. In diesem Zusammenhang bedeutet dies, dass $V = \mathbb{R}^m$ und $U = A(\mathbb{R}^n) = \{Ax : x \in \mathbb{R}^n\}$. Die Bedingung (3.5) führt zu

$$0 = (b - Ax^*, Ax) = x^T A^T (b - Ax^*) \qquad x \in \mathbb{R}^n,$$

was äquivalent zu den Normalgleichungen $A^T A x^* = A^T b$ aus der linearen Algebra ist. Aber es ist zu bedenken, daß man in der Praxis statt des Normalgleichungssystems eine QR–Zerlegung oder noch besser eine Singulärwertzerlegung von A verwendet.

3.2 Orthogonalsysteme und Projektoren

Am einfachsten sieht die Gramsche Matrix natürlich aus, wenn die u_j eine **Orthonormalbasis** bilden:

Definition 3.10. *Eine Basis* $u_1, \ldots, u_n$ *eines* n*–dimensionalen Teilraums* U *eines euklidischen Raums* V *heißt* **orthonormal***, wenn*

$$(u_j, u_k) = \delta_{jk}, \ 1 \le j, k \le n$$

gilt. Die Basis heißt **orthogonal***, wenn nur*

$$(u_j, u_k) = 0, \ 1 \le j < k \le n$$

gilt. Hat man eine unendliche Folge $u_1, \ldots, u_n, \ldots$ *linear unabhängiger Elemente, so spricht man analog von* **Orthonormalsystemen** *oder* **Orthogonalsystemen**.

Mit dem Schmidt–schen Orthonormalisierungsverfahren lässt sich aus jeder endlichen Basis eine orthonormale Basis berechnen, wie man aus der linearen Algebra weiß.

Definition 3.11. *Eine Abbildung $P : V \to V$ heißt* **Projektion**, *falls $P^2 = P$ gilt. Man bezeichnet diese Eigenschaft auch als* **Idempotenz**.

Theorem 3.12. *Sei V ein euklidischer Raum und U_n ein Unterraum der Dimension n mit einer Orthonormalbasis $\{u_1, \ldots, u_n\}$. Dann ist die eindeutig bestimmte beste Approximation an $f \in V$ aus U_n gegeben durch*

$$P_n f = \sum_{j=1}^{n} (f, u_j) u_j. \tag{3.13}$$

Der Operator $P_n : V \to U_n$ ist eine lineare Projektion mit Norm 1. Ferner gelten die Identitäten

$$\|P_n f\|^2 = \sum_{j=1}^{n} |(f, u_j)|^2, \qquad \|f - P_n f\|^2 = \|f\|^2 - \sum_{j=1}^{n} |(f, u_j)|^2. \tag{3.14}$$

Beweis: Die Darstellung für P_n folgt aus Folgerung 3.9, da $(u_j, u_k) = \delta_{jk}$ sofort $c_k = (f, u_k)$ impliziert. Da die beste Approximation an ein Element $f \in U_n$ aus U_n offensichtlich f selbst ist, ist $P|U_n$ die Identität, was $P_n^2 = P_n$ bedeutet. Aus (3.13) sieht man sofort, dass P_n linear ist. Aus der Stabilitätsabschätzung (3.8) folgt $\|P_n\| \leq 1$. Andererseits impliziert die Projektoreigenschaft $\|P_n\| \leq \|P_n\|^2$ also $1 \leq \|P_n\|$, womit wir also insgesamt $\|P_n\| = 1$ haben. Die Darstellung für $\|P_n f\|^2$ ergibt sich aus

$$\begin{aligned} \|P_n f\| &= \left(\textstyle\sum_{j=1}^{n} (f, u_j) u_j, \sum_{k=1}^{n} (f, u_k) u_k\right) = \sum_{j,k=1}^{n} (f, u_j)(f, u_k)(u_j, u_k) \\ &= \textstyle\sum_{j=1}^{n} |(f, u_j)|^2 \end{aligned}$$

unter Benutzung der Orthonormalität der Basis. Die Formel für $\|f - P_n f\|^2$ folgt schließlich aus dem Satz des Pythagoras. □

Hat man nun anstatt eines endlich dimensionalen Unterraumes eine (unendliche) Familie $\{u_j\}_{j \in \mathbb{N}}$ von orthonormalen Elementen, so kann man für ein gegebenes $f \in V$ zu jedem Unterraum $U_n := \operatorname{span}\{u_1, \ldots, u_n\}$ die beste Approximation $P_n f$ bilden, und es stellt sich die Frage, ob $P_n f$ gegen f konvergiert. Dies wird natürlich nicht für jede Folge von Unterräumen wahr sein, sondern nur für solche, die den gesamten Raum auch vollständig "ausschöpfen".

Definition 3.15. *Ein Orthonormalsystem $\{u_1, u_2, \ldots\}$ von Elementen eines euklidischen Raumes V heißt* **vollständig** *in V falls es zu jedem $f \in V$ eine Folge $f_n \in U_n := span\{u_1, \ldots, u_n\}$ mit $\|f - f_n\| \to 0$ für $n \to \infty$ gibt.*

Der Begriff "vollständiges Orthonormalsystem" ist historisch bedingt. Im Grunde ist er eher unglücklich, da es sich zum einen nicht um Vollständigkeit im klassischen Sinn handelt. Zum anderen beschreibt er eher eine Eigenschaft des Raumes $U_\infty := \cup U_n$, nämlich die Eigenschaft dicht in V zu sein:

Definition 3.16. *Eine Teilmenge U eines normierten Raumes V ist* **dicht** *in V, wenn V der Abschluß von U ist, d.h. zu jedem $v \in V$ gibt es eine gegen v konvergente Folge, die ganz in U liegt.*

Theorem 3.17. *Sei V ein euklidischer Raum und $\{u_1, u_2, \ldots\}$ ein Orthonormalsystem in V. Sei ferner $U_n := span\{u_1, \ldots, u_n\}$. Dann gilt die* Besselsche Ungleichung

$$\sum_{j=1}^{\infty} |(f, u_j)|^2 \leq \|f\|^2, \qquad f \in V.$$

Ferner sind die folgenden Eigenschaften äquivalent:

1. *$\{u_1, u_2, \ldots\}$ ist vollständig in V.*
2. *Jedes $f \in V$ lässt sich als Reihe*

$$f = \sum_{j=1}^{\infty} (f, u_j) u_j \tag{3.18}$$

   darstellen, und zwar im Sinne der Normkonvergenz der Partialsummen.

3. *Für jedes $f \in V$ gilt die* [9] Parsevalsche Gleichung

$$\sum_{j=1}^{\infty} |(f, u_j)|^2 = \|f\|^2.$$

Beweis: Die Besselsche Ungleichung folgt sofort aus $\|P_n f\| \leq \|f\|$ und der Darstellung in (3.14).

Die erste Eigenschaft impliziert die zweite, da zu $f \in V$ und $\epsilon > 0$ ein $N \in \mathbb{N}$ und $f_N \in U_N$ mit $\|f - f_N\| < \epsilon$ existiert. Dies bedeutet aber für $n \geq N$ auch

$$\|f - P_n f\| = \inf_{u \in U_n} \|f - u\| \leq \inf_{u \in U_N} \|f - u\| \leq \|f - f_N\| < \epsilon.$$

[9] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Parseval.html`

Also strebt $P_n f$ in der Norm des Raumes gegen f.

Andererseits ist die zweite Eigenschaft nur ein Spezialfall der ersten, da hier die Funktionen $f_n = P_n f$ gewählt werden können. Schliesslich zeigt die Darstellung für $\|f - P_n f\|$ in (3.14) die äquivalenz der zweiten und dritten Eigenschaft. □

Die Parsevalsche Gleichung hat zusammen mit dem Satz des Pythagoras zur Konsequenz, dass wir den Fehler $\|f - P_n f\|$ ausdrücken können als

$$\|f - P_n f\|^2 = \sum_{j=n+1}^{\infty} |(f, u_j)|^2. \tag{3.19}$$

Man wird also darauf hoffen, daß die Entwicklungskoeffizienten (f, u_j) für große j sehr schnell klein werden. Aber bisher haben wir noch nicht mal die Vollständigkeit für Basen aus Polynomen bewiesen. Das folgt in Abschnitt 3.6.

3.3 Kompression und Beste n–Term–Approximation

Aber man kann an diser Stelle schon auf eine wichtige Technik zur **Kompression** hinweisen. Nehmen wir an, ein Element f eines euklidischen Raumes habe eine Darstellung (3.18) mit nur wenigen betragsmäßig großen Koeffizienten. Wenn man dann alle Terme mit kleinen Koeffizienten wegläßt, hat man eine sehr gute Approximation, denn das Quadrat des Fehlers ist die Quadratsumme der weggelassenen Koeffizienten. Das ist das Grundprinzip vieler Kompressionstechniken bis hin zu wavelets, und wir werden uns das Weglassen kleiner Koeffizienten noch öfter ansehen. Was man zu effizienten Kompressionstechniken noch braucht, sind schnelle Algorithmen zur Berechnung der Koeffizienten (**Analyse**) und zur Auswertung der Approximationen (**Synthese**). Auch das werden wir genauer studieren.

Wir gehen etwas allgemeiner vor und befassen uns mit Approximationen, die aus möglichst wenig Termen bestehen:

Definition 3.20. *Die* **beste** k**–Term–Approximation** *eines Elements f eines normierten Vektorraums V bezüglich eines n–dimensionalen linearen Unterraums U mit Basis $u_1, \ldots, u_n$ minimiert $\|f - u\|_V$ unter allen $u \in U$, die nur maximal k von Null verschiedene Koeffizienten haben.*

In diesem Zusammenhang verwendet man die etwas schlampige Notation $\|u\|_0$ für die Anzahl der von Null verschiedenen Koeffizienten von u in der

fest gewählten Basis. Das ist natürlich keine Norm, obwohl die Bezeichnung das suggeriert.

Natürlich müssen beste k–Term–Approximationen immer existieren, weil es nur endlich viele, nämlich $\binom{n}{k}$ Möglichkeiten gibt, k–dimensionale Unterräume von U auszuwählen. Die direkte Bestimmung bester k–Term–Approximationen ist aber im Allgemeinen kein lineares Approximationsproblem mehr, und schon im Falle univariater Polynome in $C[a,b]$ mit der Tschebyscheff–Norm sehr unangenehm. Im Falle euklidischer Räume ist das aber einfach:

Theorem 3.21. *Eine beste k–Term–Approximation zu Elementen f eines euklidischen Raums v durch einen Unterraum mit Orthonormalbasis $u_1, \ldots, u_n$ ist dadurch gegeben, daß man in der Projektion (3.13) alle bis auf die k betragsgrößten Koeffizienten wegläßt.*

Beweis: Jede k–Term-Approximation entsteht durch einen Projektor P_k auf einen speziellen k–dimensionalen Unterraum U_k von U. Wegen (3.14) folgt dann

$$\|f - P_k f\|^2 - \|f - P_n f\|_2^2 = \sum_{u_j \notin U_k} (f, u_j)^2,$$

d.h. der Unterschied zwischen der besten n–Term– und einer k–Term–Approximation, gemessen als Fehlernorm zum Quadrat, besteht genau in der Quadratsumme der weggelassenen Koeffizienten. Eine beste k–Term–Approximation bekommt man also, indem man die betragskleinsten $n-k$ Terme wegläßt. □

3.4 Exkurs über Hilberträume

In Definition 3.1 hatten wir Hilberträume als vollständige euklidische Räume definiert. Endlichdimensionale euklidische Räume sind immer isometrisch zu einem $\mathbb{R}^n$ und deshalb immer vollständig. Im unendlichdimensionalen Fall ist das nicht so. Beispielsweise ist es leicht (Aufgabe!) im Raum $C[a,b]$ unter dem Skalarprodukt

$$(f,g)_2 := \int_a^b f(t)g(t)dt \text{ für alle } f, g \in C[a,b]$$

eine nicht konvergente Cauchyfolge anzugeben. In der Funktionalanalysis wird aber bewiesen, daß sich alle normierten Räume V in einem speziellen Sinne vervollständigen lassen:

Theorem 3.22. *Zu jedem normierten Raum V über R gibt es einen vollständigen normierten Raum (einen* **Banachraum***) $\overline{V}$, der ein Bild von V*

unter einer Isometrie $I : V \to I(V) \subset \overline{V}$ enthält, und auf den sich jede stetige lineare Abbildung $A : V \to W$ mit Werten in einen Banachraum W zu einer stetigen linearen Abbildung $\overline{A} : \overline{V} \to W$ im Sinne von $\overline{A} \circ I = A$ fortsetzen läßt. Es gibt einen bis auf Isometrie eindeutigen kleinsten Raum $\overline{V}$ mit dieser Eigenschaft, und er wird die **Vervollständigung** *von V genannt.* □

In diesem Sinne läßt sich jeder euklidische Raum zu einem Hilbertraum vervollständigen. Im obigen Beispiel bekommt man den Raum $L_2[a, b]$ der im Lebesgue–Sine meßbaren und quadratintegrablen Funktionen heraus, und wegen der isometrischen Einbettung ändert sich das Skalarprodukt nicht.

In Hilberträumen kann man Projektoren auch auf unendlichdimensionalen Unterräumen definieren:

Theorem 3.23. *Für jeden abgeschlossenen Unterraum U eines Hilbertraums V gilt $V = U + U^\perp$ als direkte und orthogonale Summe. Ferner gibt es einen linearen Projektor P_U mit $U = P_U(V)$, und $Id - P_U$ ist ein Projektor von V auf $U^\perp$. Für jedes $f \in V$ ist $P_U(f)$ die beste Approximation von f bezüglich U. Ferner gilt $(U^\perp)^\perp = U$ und $P_{U^\perp} = Id - P_U$.*

Weil der Beweis aus purer Approximationstheorie besteht, soll er hier nicht fehlen.

Wir beweisen zuerst, daß jedes $f \in V$ genau eine beste Approximnation bezüglich U hat. Zu jedem $f \in$ gibt es eine sogenannte **Minimalfolge** $\{u_n\}_n$ von Approximationen aus U mit

$$\|f - u_n\|_V \to \inf_{u \in U} \|f - u\|_V \text{ für } n \to \infty.$$

Dann betrachten wir die Räume

$$U_n := \text{ span } \{u_1, \ldots, u_n\}$$

und bestimmen die beste Approximation u_n^* von f in U_n. Dann ist auch $\{u_n^*\}_n$ eine Minimalfolge, denn man hat

$$\inf_{u \in U} \|f - u\|_V \le \|f - u_n^*\|_V \le \|f - u_n\|_V \to \inf_{u \in U} \|f - u\|_V \text{ für } n \to \infty.$$

Für alle N folgt dann $(f - u_N^*) \perp U_N$ aus der Eigenschaft bester Approximationen, und insbesondere gilt $(f - u_N^*) \perp (u_N^* - u_n^*)$ für alle $n \le N$. Der Satz des Pythagoras liefert dann

$$\|f - u_n^*\|_V^2 = \|f - u_N^*\|_V^2 + \|u_N^* - u_n^*\|_V^2.$$

Das ergibt $\|u_N^* - u_n^*\|_V \leq \|f - u_n^*\|_V$ für alle $n \leq N$, und weil $\{u_n^*\}_n$ eine Minimalfolge ist, folgt daraus, daß $\{u_n^*\}_n$ eine Cauchyfolge ist. Wegen der Vollständigkeit von V und der Abgeschlossenheit von U hat sie einen Limes $u^* \in U$. Dafür gilt aber

$$\|f - u^*\|_V \leq \|f - u_n^*\|_V + \|u_n^* - u^*\|_V \to \inf_{u \in U} \|f - u\|_V \text{ für } n \to \infty$$

und deshalb ist u^* beste Approximation zu f in U. Wegen der strikten Konvexität der euklidischen Norm ist sie eindeutig, und weil wir die Abhängigkeit von f jetzt brauchen, verwenden wir die Bezeichnung $P_U(f)$ statt u^*.

Für zwei Elemente f und g aus V und Skalare α, β bilde man das Element $\alpha P_U(f) + \beta P_U(g) \in U$. Damit folgt

$$(\alpha f + \beta g - \alpha P_U(f) - \beta P_U(g), u)_V = \alpha(f - P_U(f), u)_V + \beta(g - P_U(g), u)_V = 0$$

für alle $u \in U$, und weil die Orthogonalitätsbedingung nach Satz 3.4 auch hinreichend ist, muß $\alpha P_U(f) + \beta P_U(g)$ die beste Approximation zu $\alpha f + \beta g$ sein. Also ist die Abbildung P_U linear, und sie ist wegen ihrer Definition als beste Approximation auch idempotent.

Der Rest ist jetzt wegen der orthogonalen Zerlegung

$$f = P_U(f) + (f - P_U(f))$$

von f in ein Element aus U und eines aus $U^\perp$ kein Problem mehr. Denn hat man ein $f \in (U^\perp)^\perp$, so folgt

$$(f - P_U(f), v) = 0 \text{ für alle } v \in U^\perp,$$

und weil $f - P_U(f)$ selbst in $U^\perp$ liegt, muß $f = P_U(f)$ gelten. Das beweist $(U^\perp)^\perp \subseteq U$, und weil $(U^\perp)^\perp \supseteq U$ trivial ist, folgt $(U^\perp)^\perp = U$. Der Projektor $Id - P_U$ bildet dann aber zu f die beste Approximation $f - P_U(f)$ bezüglich $U^\perp$, woraus $P_{U^\perp} = Id - P_U$ folgt. □

Die Existenz vollständiger Orthonormalsysteme im Sinne von Theorem 3.17 ist Für allgemeine Hilberträume nicht gesichert.

Definition 3.24. *Ein Hilbertraum heißt* **separabel**, *wenn er ein abzählbares vollständiges Orthonormalsystem besitzt.*

Die für die Praxis relevanten Hilberträume sind alle separabel, aber das muß man in jedem Falle erst beweisen. Wir machen das in Abschnitt 3.6, aber vorher brauchen wir die wichtigsten Spezialfälle.

3.5 Beispiele von Orthogonalsystemen

3.5.1 Orthogonalpolynome

Diese sollten die Leser schon aus der Numerischen Mathematik als Hintergrund der Gaußquadratur kennen.

Hat man zu einem Intervall $I := [a, b] \subset \mathbb{R}$ eine auf (a, b) stetige und positive Gewichtsfunktion w, für die alle Integrale

$$\int_I p(t)^2 w(t)dt \text{ für alle } p \in \Pi_n \text{ und } \text{ für alle } n \in \mathbb{N}_0$$

endlich sind, so kann man den Raum

$$L_{2,w}[a, b] := \left\{ f \in C[a, b] \ : \ \int_I f(t)^2 w(t)dt < \infty \right\}$$

definieren, der das Skalarprodukt

$$(f, g) := \int_I f(t)g(t)w(t)dt \text{ für alle } f, g \in L_{2,w}[a, b]$$

trägt und alle Polynome enthält. Durch rekursives Orthonormieren bekommt man dann eine orthonormale Basis $p_0, p_1, \ldots$ von Polynomen $p_j \in \Pi_j$ und man kann fragen, ob die Basis vollständig ist. Das schieben wir noch auf. Über Orthogonalpolynome gibt es Unmassen an klassischer Literatur[10], die wir hier weitgehend ignorieren (z.B. [5, 3, 6]). Die Nullstellen der Orthogonalpolynome liefern Stützstellen von Quadraturformeln, und deshalb beweist man in der numerischen Integration

Theorem 3.25. *Die Orthogonalpolynome p_j zu einer festen Gewichtsfunktion w auf (a, b) sind bis auf ihre Normierung eindeutig bestimmt, haben genau den Grad j und haben in (a, b) genau j einfache Nullstellen.* □

Egal wie die Gewichtsfunktion aussieht, man kann immer die Orthogonalpolynome durch eine 3–Term–Rekursion berechnen:

Theorem 3.26. *Mit geeigneten Koeffizienten a_j, b_j, c_j gilt*

$$p_{j+1}(x) = a_j x p_j(x) + b_j p_j(x) + c_j p_{j-1}(x)$$

für die Orthogonalpolynome zu einer festen Gewichtsfunktion.

[10] `http://en.wikipedia.org/wiki/Orthogonal_polynomials`

Beweis: Durch ein simples Induktionsargument ist klar, daß man immer eine Darstellung

$$p_{j+1}(x) = a_j x p_j(x) + b_j p_j(x) + c_j p_{j-1}(x) + \sum_{k=0}^{j-2} d_k p_k(x)$$

bekommt, aber es ist zu zeigen, daß die rechts stehende Summe verschwindet. Für alle m mit $0 \leq m \leq j-2$ kann man unter Ausnutzung der Orthogonalität folgendermaßen schließen:

$$\begin{aligned} 0 &= (p_{j+1}, p_m) \\ &= a_j(xp_j, p_m) + b_j(p_j, p_m) + c_j(p_{j-1}, p_m) + \sum_{k=0}^{j-2} d_k(p_k, p_m) \\ &= a_j(p_j, xp_m) + 0 + 0 + d_m(p_m, p_m) \\ &= d_m(p_m, p_m) = d_m \|p_m\|^2 \end{aligned}$$

und das beweist die Behauptung. □

Für die Gewichtsfunktion $w = 1$ auf $[-1, 1]$ bekommt man die (nicht normierten) **Legendre-Polynome** durch die 3-Term–Rekursion

$$(j+1)p_{j+1}(x) = (2j+1)xp_j(x) - jp_{j-1}(x),\ j \geq 1,\ p_0 = 1,\ p_1(x) = x.$$

Auch dafür haben wir eine kleine Routine

```
function V=leg(z,n)
% generates Legendre polynomials for points z up to degree n
V(:,n+1) = ones(length(z),1);
V(:,n) = z;
for j = 2:n % for degree j in V(:,n+1-j)
    V(:,n+1-j) = ((2*j-1)*z.*V(:,n+2-j)-(j-1)*V(:,n+3-j))/j;
end
```

und die Abbildung 7, die mit

```
% evaluate approximate Legendre basis
clear all;
close all;
n=55;
h=0.005;
x=(-1:h:1)'; % x points
B=leg(x,n);
for i=1:n+1
```

```
    plot(x,B(:,i));
    hold on
end
```

erzeugt wurde.

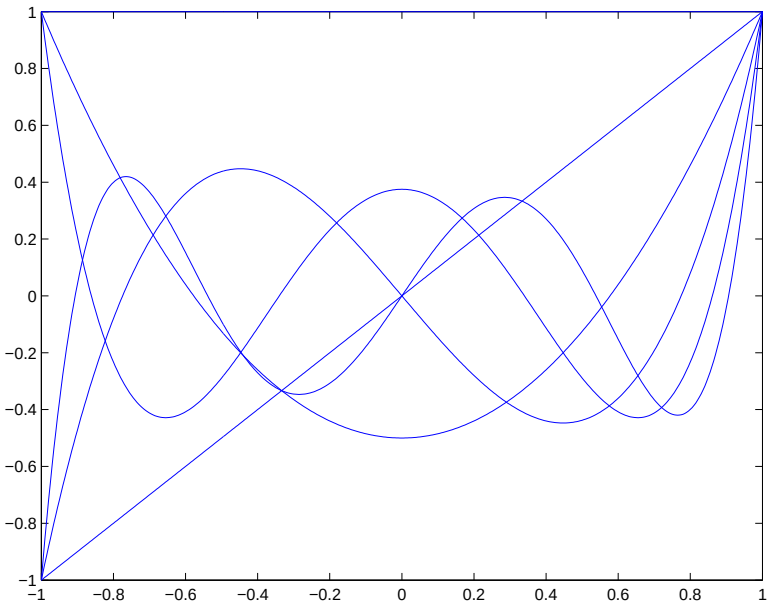

Figure 7: Legendre–Polynome

Für die Praxis sehr viel wichtiger sind die

3.5.2 Tschebyscheff-Polynome

Sie sind auf $[-1, 1]$ definiert durch

$$T_n(x) = \cos(n \arccos x), \qquad x \in [-1, 1],\ n \in \mathbb{N}_0.$$

und genügen der 3–Term–Rekursionsformel

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \qquad n \geq 1, \tag{3.27}$$

mit den Anfangsbedingungen $T_0(x) = 1$ und $T_1(x) = x$. Insbesondere ist $T_n \in \Pi_n(\mathbb{R})$ und hat den Grad n. Der Koeffizient vor x^n ist für $n \geq 1$ gegeben durch 2^{n-1}. Die Rekursionsformel lässt sich verallgemeinern zu

$$T_{m+n}(x) = 2T_m(x)T_n(x) - T_{|m-n|}(x), \qquad m, n \in \mathbb{N}_0. \tag{3.28}$$

Sie sind orthogonal auf $[-1,1]$ mit der Gewichtsfunktion

$$w(x) = \frac{1}{\sqrt{1-x^2}}$$

auf $(-1,1)$. Das sieht wegen der Singularität in $x = \pm 1$ kurios aus, aber mit der Substitution $x = \cos(\varphi)$ sieht man den Zusammenhang mit trigonometrischen Funktionen:

$$\begin{aligned} & \int_{-1}^{+1} T_n(x) T_m(x) w(x) dx \\ = & -\int_{\pi}^{-\pi} \frac{\cos(n\varphi)\cos(m\varphi)}{\sqrt{1-\cos^2(\varphi)}} \sin(\varphi) d\varphi \\ = & \int_{-\pi}^{+\pi} \cos(n\varphi)\cos(m\varphi) d\varphi. \end{aligned}$$

Die Orthogonalität der Cosinusfunktionen auf $[-\pi, +\pi]$ sehen wir uns gleich genauer an, aber erst berechnen wir die Wertematrix der Tschebyscheff–Polynome mit

```
function Ve=chebmat(x, k)
% Construct Chebyshev polynomial matrix
% for points x and degree k
n=k+1;
Ve(:,n) = ones(length(x),1);
Ve(:,n-1) = x;
for j = n-2:-1:1
    Ve(:,j) = 2*x.*Ve(:,j+1)-Ve(:,j+2);
end
```

und plotten Abbildung 8 mit

```
% evaluate Chebyshev basis
clear all;
close all;
n=25;
h=0.005;
x=(-1:h:1)'; % x points
B=chebmat(x,n);
for i=1:n+1
    plot(x,B(:,i));
    hold on
end
```

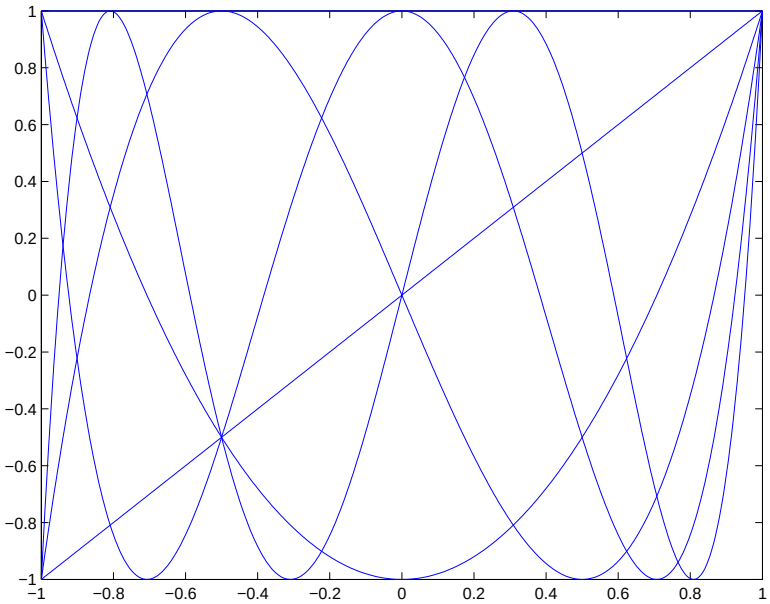

Figure 8: Tschebyscheff–Polynome

Man kann an vielen Beispielen sehen, daß die Tschebyscheffbasis der Monombasis numerisch weit überlegen ist. Hier ein kleines Beispiel:

```
clear all;
close all;
x=(-1:0.001:1)';
n=25;
y=abs(x);
p=polyfit(x,y,n);
c=chebyfit(x,y,n);
figure
plot(x,y-polyval(p,x),x,y-chebyval(c,x))
title('Fehlerfunktion')
figure
plot(p)
title('Koeffizienten Monombasis')
figure
plot(c)
title('Koeffizienten Tschebyscheffbasis')
```

Die Fehlerfunktionen in Abbildung 9 sind nicht zu unterscheiden, aber in den Abbildungen 10 und 11 sieht man, daß die Koeffizienten der Monombasis

riesig werden. Sie liegen in der Gegend von 10^6 und werden für höhere Grade größer, während die Koeffizienten in der Tschebyscheffbasis in der Gegend von 1 bleiben und für höhere Grade abklingen. Dabei ist zu beachten, daß die Koeffizienten der Polynome niederen Grades rechts stehen.

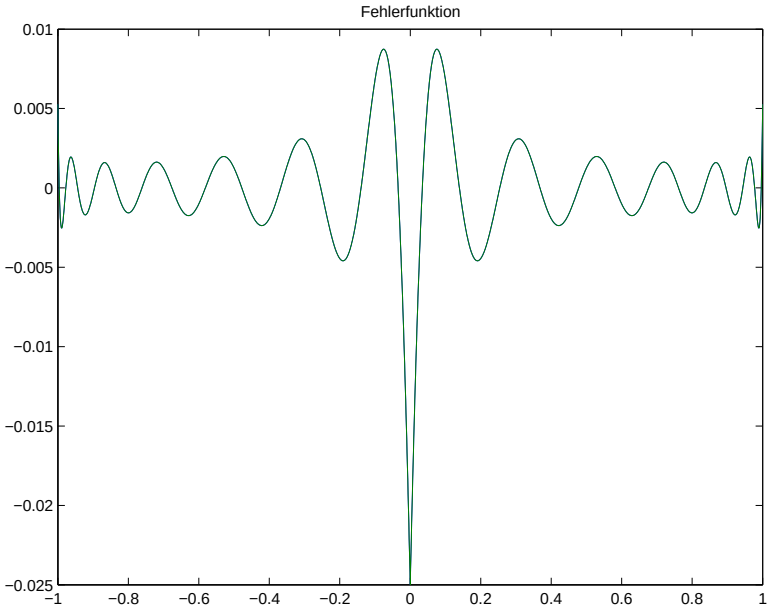

Figure 9: Fehlerfunktion bei Polynomapproximation von $|x|$

Die Interpolation in den Nullstellen oder Extremstellen der Tschebyscheff–Polynome ist ein wichtiges numerisches Verfahren, weil es sich stabil und sehr effizient implementieren läßt und andererseits als nichtoptimale Approximation nicht wesentlich schlechter als die beste Tschebyscheff–Appproximation ist. Wir gehen bei den schnellen Algorithmen im nächsten Kapitel noch einmal darauf ein, denn wir wollen dann auch den Zusammenhang mit der Clenshaw–Curtis–Quadratur und der schnellen diskreten Cosinustransfornation etwas genauer ansehen.

Aber wir entnehmen aus der Vorlseung Numerische Mathematik noch einen wichtigen Satz, der zeigt, daß die Interpolation in den Tschebyscheff–Nullstellen eine Minimaleigenschaft hat. Interpoliert man in Punkten

$$-1 \leq x_0 < x_1 < \ldots, x_n \leq 1 \tag{3.29}$$

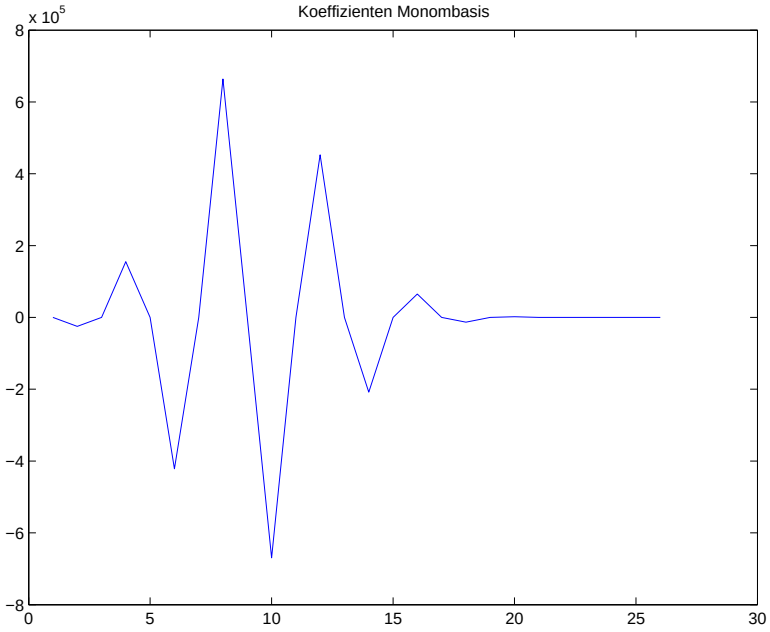

Figure 10: Koeffizienten der Monombasis

die Werte einer Funktion f durch ein Polynom P vom Grade $\leq n$, , so ist der Fehler als

$$f(x) - P(x) = [x_0, \ldots, x_n, x]f \cdot \prod_{j=0}^{n}(x - x_j)$$

zu schreiben. Das beweist man mit Hilfe der Newtonschen Interpolationsformel, die ja die dividierten Differenzen oder Differenzenquotienten $[x_0, \ldots, x_j]f$ verwendet, die man oben braucht.

Dann fragt man nach der optimalen Lage der Punkte:

Theorem 3.30. *Das Minimum von*

$$\left\| \prod_{j=0}^{n}(x - x_j) \right\|_{\infty,[-1,+1]}$$

unter allen Punktwahlen (3.29) wird für die $n+1$ *Nullstellen des Tschebyscheff–Polynoms* T_{n+1} *angenommen.*

Wir können eine Beweisskizze mit dem Alternantensatz angeben. Die Minimierung der obigen Größe ist nichts anderes, als die beste Approximation von x^{n+1} durch Polynome vom Grad $\leq n$ zu finden. Das Polynom $2^{-n}T_{n+1}$ hat aber gerade die Form einer Fehlerfunktion $x^{n+1} - P(x)$, und es alterniert in $n + 2$ Punkten mit Werten ± 1, ist also die Lösung des Problems. □

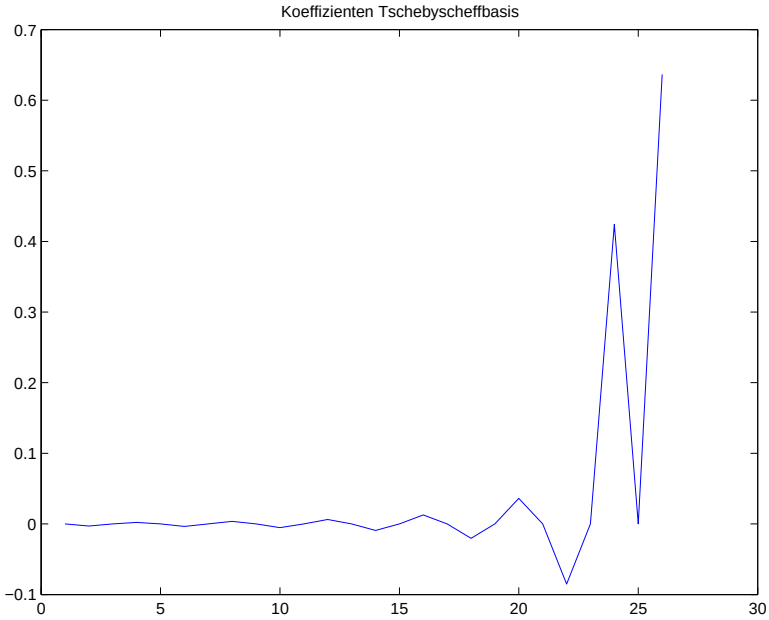

Figure 11: Koeffizienten der Tschebyscheffbasis

3.5.3 Trigonometrische Polynome

Auch hier sollte das Folgende aus der Numerischen Mathematik oder der reellen Analysis bekannt sein.

Definition 3.31. *Eine Funktion* $f : \mathbb{R} \to \mathbb{R}$ *heißt* **periodisch** *mit Periode* $h > 0$, *falls* $f(x + h) = f(x)$ *für alle* $x \in \mathbb{R}$.

Wir werden hier nur 2π–periodische Funktionen behandeln und alles auf $[-\pi, \pi]$ betrachten. Die einfachsten Beispiele 2π–periodischer Funktionen sind 1, $\cos x$, $\sin x$, $\cos 2x$, $\sin 2x$,.... Der Raum

$$C_{2\pi} := \{f \ : \ \text{stetig und } 2\pi - \text{periodisch auf } \mathbb{R}\}$$

mit dem Skalarprodukt

$$(f, g)_\pi := \frac{1}{\pi} \int_{-\pi}^{+\pi} f(t)g(t)dt \text{ für alle } f, g \in C_{2\pi}$$

ist ein euklidischer Raum.

Definition 3.32. *Die Elemente der Menge*

$$T_{\mathbb{R},m} := \left\{ T(x) = \frac{a_0}{2} + \sum_{j=1}^{m} (a_j \cos jx + b_j \sin jx) : a_j, b_j \in \mathbb{R} \right\} \tag{3.33}$$

heißen (reelle) trigonometrische Polynome vom Grad $\leq m$.

Theorem 3.34. *Der Raum $T_{\mathbb{R},m}$ ist ein $\mathbb{R}$-linearer Unterraum der Dimension $2m+1$ von $C_{2\pi}$, und die Funktionen*

$$\frac{1}{\sqrt{2}}, \cos(x), \sin(x), \ldots, \cos(mx), \sin(mx)$$

sind in $C_{2\pi}$ orthonormal.

Beweis: Die Orthonormalität beweist man mit den üblichen Additionstheoremen. Ist z.B. $j \neq k$, so gilt

$$\frac{1}{\pi}\int_0^{2\pi} \sin(jx)\sin(kx)dx = \frac{1}{2\pi}\int_0^{2\pi} (\cos(j-k)x - \cos(j+k)x)dx = 0.$$

Die übrigen Fälle gehen analog. □

Auch hier entsteht sofort die Frage, ob die trigonometrischen Polynome vollständig in $C_{2\pi}$ sind. Aber ohne weitere Argumentationen können wir alles anwenden, was wir bisher hergeleitet haben.

Korollar 3.35. *Sei $S_m f$ die beste Approximation aus $T_{\mathbb{R},m}$ an $f \in C_{2\pi}$ bzgl. $(\cdot,\cdot)_\pi$. Dann lässt sich $S_m f$ schreiben als*

$$S_m f(x) = \frac{a_0}{2} + \sum_{j=1}^{m} (a_j \cos jx + b_j \sin jx), \tag{3.36}$$

wobei

$$\begin{array}{rcll} a_j & = & (f(x), \cos(jx))_\pi = \frac{1}{\pi}\int_0^{2\pi} f(x)\cos(jx)dx, & 0 \le j \le m, \\ b_j & = & (f(x), \sin(jx))_\pi = \frac{1}{\pi}\int_0^{2\pi} f(x)\sin(jx)dx, & 1 \le j \le m. \end{array} \tag{3.37}$$

Wir werden später $S_m(f)$ als die m–te **Fourier–Partialsumme** bezeichnen und näher untersuchen.

Korollar 3.38. *Ist $f \in C_{2\pi}$ gerade, so ist $S_m f(x) = \frac{a_0}{2} + \sum_{j=1}^{m} \cos jx$ ebenfalls gerade.*

Beweis: Es gilt

$$\pi b_j = \int_{-\pi}^{\pi} f(x)\sin(jx)dx = \int_0^{\pi} f(x)\sin(jx)dx + \int_0^{\pi} f(-x)\sin(-jx)dx = 0,$$

da f gerade und der Sinus ungerade ist. □

Aufgabe: Man schreibe eine Routine in MATLAB, die zu einem Punktevektor $x \in \mathbb{R}^M$ und einem $n \geq 0$ eine $M \times (2n+1)$–Matrix baut, die es erlaubt, für die Basisdarstellung (3.36) eine Matrix–Vektor–Multiplikaton mit dem Vektor $(a_0, a_1, b_1, a_2, b_2, \ldots, a_n, b_n)^T$ zu verwenden. Aufrufe von sin und cos dürfen nur einmal (aber vektoriell) vorkommen.

Wir müssen für spätere Zwecke das Ganze noch ins Komplexe übersetzen. Die Eulersche[11] Formel $e^{ix} = \cos(x) + i\sin(x)$ liefert

$$S_m f(x) = \frac{a_0}{2} + \sum_{j=1}^{m} \frac{a_j - ib_j}{2} e^{ijx} + \frac{a_j + ib_j}{2} e^{-ijx} = \sum_{j=-m}^{m} c_j e^{ijx} \qquad (3.39)$$

mit

$$\begin{aligned} c_j &= \frac{a_j - ib_j}{2},\ 0 \leq j \leq n \\ c_{-j} &= \frac{a_j + ib_j}{2} = \overline{c_j},\ 0 \leq j \leq n. \end{aligned}$$

Setzen wir die Integrale aus (3.37) ein, so folgt, dass die Koeffizienten $c_j = c_j(f) =: \widehat{f}(j)$ jetzt die Form

$$\widehat{f}(j) = \frac{1}{2\pi} \int_0^{2\pi} f(x) e^{-ijx} dx, \qquad j \in \mathbb{Z}, \qquad (3.40)$$

annehmen. Sie treten in Abschnitt 3.7 als Fouriertransformierte wieder auf.

Die Darstellung (3.39) ist beinahe ein komplexes Polynom, denn es gilt

$$S_m f(x) = \sum_{j=-m}^{m} c_j e^{ijx} = e^{-imx} \sum_{j=0}^{2m} c_{j-m} \left(e^{ix}\right)^j.$$

Führt man die komplexe Variable $z = e^{ix}$ ein, so folgt

$$S_m f(x) = z^{-m} \sum_{j=0}^{2m-1} c_{j-m} z^j,$$

und wir werden diese Reduktion reeller trigonometrischer Polynome auf komplexe algebraische Polynome noch oft benutzen.

Aber es gibt noch eine andere Reduktion. Sehen wir uns dazu den Spezialfall **gerader** trigonometrischer Polynome an, d.h. solche mit $f(-x) = f(x)$ für

[11] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Euler.html`

alle $x \in \mathbb{R}$. Diese haben wegen Korollar 3.38 keine Sinusterme, und deshalb können wir sie mit der Substitution $\cos(\varphi) = x$ über

$$S_m f(\varphi) = \frac{a_0}{2} + \sum_{j=1}^{m} \cos j\varphi = \frac{a_0}{2} + \sum_{j=1}^{m} T_j(x)$$

als algebraische Polynome auf $[-1, 1]$ schreiben. Umgekehrt ist jedes algebraische Polynom auf $[-1, 1]$ in ein gerades trigonometrisches Polynom überführbar, wenn wir das Ganze rückwärts machen.

Wir gehen deshalb jetzt noch einmal auf die Tschebyscheff–Polynome zurück.

Theorem 3.41. *Mit dem Skalarprodukt*

$$(f, g)_T := \frac{2}{\pi} \int_{-1}^{+1} \frac{f(x)g(x)}{\sqrt{1-x^2}} dx$$

gilt

$$(T_j, T_k)_T = \begin{cases} 0 & j \neq k, \\ 1 & j = k \geq 1, \\ 2 & j = k = 0. \end{cases}$$

Beweis: Wir erhalten mit der Substitution $x = \cos(t)$, dass

$$\begin{array}{rcl} (T_j, T_k)_T & = & \frac{2}{\pi} \int_{-1}^{1} \cos(j \arccos x) \cos(k \arccos x) \frac{dx}{\sqrt{1-x^2}} \\ & = & \frac{2}{\pi} \int_0^{\pi} \cos(jt) \cos(kt) dt \\ & = & \frac{1}{\pi} \int_0^{2\pi} \cos(jt) \cos(kt) dt \\ & = & (\cos(jt) \cos(kt))_\pi, \end{array}$$

wobei wir in der vorletzten Gleichung noch ausgenutzt haben, dass der Integrand gerade und 2π-periodisch ist. Der Rest folgt also aus Satz 3.34. □

Die Substitution $x = \cos(t)$ liefert also die Brücke zwischen den Tschebyscheff–Polynomen und den geraden trigonometrischen Polynomen. Daher lässt sich also die beste Approximation $P_n f$ aus $\Pi_n(\mathbb{R})$ an $f \in C[-1, 1]$ bzgl. $(\cdot, \cdot)_T$ darstellen als

$$P_n f(x) = \frac{r_0}{2} + \sum_{j=1}^{n} r_j T_j(x),$$

wobei die Koeffizienten gegeben sind als

$$\begin{array}{rcl} r_j = r_j(f) & = & (f, T_j)_T = \frac{2}{\pi} \int_{-1}^{1} f(x) T_j(x) \frac{dx}{\sqrt{1-x^2}} \\ & = & \frac{1}{\pi} \int_0^{2\pi} f(\cos t) \cos(jt) dt. \\ & = & a_j(\widetilde{f}) \end{array}$$

mit der geraden Funktion $\widetilde{f} = f \circ \cos$. Deshalb stimmt die Projektion von f auf die Tschebyscheff–Polynome mit der Projektion von $\widetilde{f}$ auf die trigonometyrischen Polynome überein.

3.6 Sätze von Weierstraß

Nach diesem endlichdimensionalen Vorgeplänkel ist es an der Zeit, über Vollständigkeit trigonometrischer oder algebraischer Polynome nachzudenken. Das führt zu zwei fundamentalen Ergebnissen der Approximationstheorie, die man auch als Weierstraß[12] –Sätze bezeichnet:

- Jede stetige Funktion $f \in C[a,b]$ kann beliebig gut durch algebraische Polynome approximiert werden.
- Jede stetige, 2π-periodische Funktion kann beliebig gut durch trigonometrische Polynome approximiert werden.

In beiden Fällen wird zunächst in der **Maximumsnorm** oder **Tschebyscheff-Norm**

$$\|f\|_\infty := \|f\|_{L_\infty[a,b]} = \max_{x\in[a,b]} |f(x)|$$

argumentiert und dann zu anderen Normen übergegangen. Ferner werden wir uns wann immer möglich auf das Einheitsintervall $[0,1]$ zurückziehen.

Zur Herleitung dieser Resultate benutzen wir Korovkin-Operatoren.

Definition 3.42. *Eine Abbildung $K : C[0,1] \to C[0,1]$ heißt* monoton*, falls für alle $f, g \in C[0,1]$ mit $f(x) \le g(x)$, $x \in [0,1]$, auch $Kf(x) \le Kg(x)$, $x \in [0,1]$, folgt. Eine Folge $K_n : C[0,1] \to C[0,1]$, $n \in \mathbb{N}$, monotoner, linearer Operatoren heißt* Korovkin-Folge*, falls*

$$\lim_{n\to\infty} \|K_n f_j - f_j\|_\infty = 0$$

für $f_j(x) := x^j$, $j = 0,1,2$.

Das Erstaunliche an einer Korovkin-Folge ist, dass aus der gleichmäßigen Konvergenz $K_n f \to f$ auf $f \in \Pi_2(\mathbb{R})$ die gleichmäßige Konvergenz auf ganz $C[0,1]$ folgt.

Theorem 3.43. *Ist $\{K_n\}$ eine Korovkin-Folge auf $C[0,1]$, so gilt*

$$\lim_{n\to\infty} \|K_n f - f\|_\infty = 0$$

für alle $f \in C[0,1]$.

[12] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Weierstrass.html`

Beweis: Als stetige Funktion ist f auf dem kompakten Intervall $[0,1]$ gleichmäßig stetig. D.h. zu $\epsilon > 0$ existiert ein $\delta > 0$, sodass $|f(x) - f(y)| < \epsilon/3$ für alle $x, y \in [0,1]$ mit $|x-y| < \delta$. Sei $t \in [0,1]$ fest. Dann gilt einerseits $|f(x) - f(t)| < \epsilon/3$, falls $|x-t| < \delta$, und andererseits

$$|f(x) - f(t)| \leq 2\|f\|_\infty \leq 2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2,$$

falls $|x-t| \geq \delta$. Beides zusammen gibt

$$|f(x) - f(t)| \leq \frac{\epsilon}{3} + 2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2, \qquad x \in [0,1],$$

oder

$$\underbrace{f(t) - \frac{\epsilon}{3} - 2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2}_{p_t(x):=} \leq f(x) \leq \underbrace{f(t) + \frac{\epsilon}{3} + 2\|f\|_\infty \left(\frac{x-t}{\delta}\right)^2}_{q_t(x):=}. \quad (3.44)$$

Da K_n monoton ist folgt somit

$$K_n p_t(x) \leq K_n f(x) \leq K_n q_t(x), \qquad x \in [0,1]. \quad (3.45)$$

Nun sind p_t und q_t beides quadratische Polynome, sodass die Anwendung der Korovkin-Operatoren auf sie gleichmäßig in x gegen sie konvergiert. Tatsächlich ist diese Konvergenz sogar gleichmässig in x und t. Zum Beispiel erhalten wir mit $f_j(x) = x^j$ aus

$$\begin{aligned} K_n q_t(x) - q_t(x) &= (K_n f_0(x) - f_0(x)) \left[f(t) + \tfrac{\epsilon}{3} + \tfrac{2t^2\|f\|_\infty}{\delta^2}\right] \\ &\quad + (K_n f_1(x) - f_1(x)) \left[\tfrac{-4t\|f\|_\infty}{\delta^2}\right] \\ &\quad + (K_n f_2(x) - f_2(x)) \left[\tfrac{2\|f\|_\infty}{\delta^2}\right], \end{aligned}$$

dass

$$\begin{aligned} |K_n q_t(x) - q_t(x)| &\leq \left(\|f\|_\infty + \tfrac{\epsilon}{3} + 2\tfrac{\|f\|_\infty}{\delta^2}\right) \|K_n f_0 - f_0\|_\infty \\ &\quad + \tfrac{2\|f\|_\infty}{\delta^2} \left(2\|K_n f_1 - f_1\|_\infty + \|K_n f_2 - f_2\|_\infty\right), \end{aligned}$$

was offensichtlich gleichmäßig in x und t aus $[0,1]$ gegen Null strebt. Da es sich für p_t analog verhält, finden wir also ein $n_0 \in \mathbb{N}$, sodass für $n \geq n_0$ und $x, t \in [0,1]$ gilt

$$|K_n q_t(x) - q_t(x)| \leq \frac{\epsilon}{3}, \quad |K_n p_t(x) - p_t(x)| \leq \frac{\epsilon}{3}.$$

Dies impliziert aber zusammen mit (3.45)

$$p_t(x) - \frac{\epsilon}{3} \leq K_n f(x) \leq q_t(x) + \frac{\epsilon}{3}$$

und dies wiederum mit (3.44)

$$p_t(x) - q_t(x) - \frac{\epsilon}{3} \leq f(x) - K_n f(x) \leq q_t(x) - p_t(x) + \frac{\epsilon}{3}.$$

Da die letzte Formel für beliebige $t, x \in [0,1]$ gilt, können wir insbesondere $t = x$ setzten. Weil $q_x(x) - p_x(x) = 2\epsilon/3$ ist, erhalten wir schließlich

$$|f(x) - K_n f(x)| \leq \epsilon$$

für alle $n \geq n_0$ und $x \in [0,1]$. □

Damit ist die Hauptarbeit auf dem Weg zum Beweis des Weierstraßschen Approximationssatzes getan. Wir benötigen letztlich nur noch eine Folge von Korovkin-Operatoren, die $C[0,1]$ auf die Polynome abbildet. Kurioserweise werden uns diese Operatoren im Computer–Aided Design wieder begegnen, und dort dienen sie nicht der Theorie, sondern der Praxis.

Theorem 3.46. *Die Bernstein*[13]*–Operatoren* $B_n : C[0,1] \to \Pi_n(\mathbb{R})$,

$$B_n f(x) := \sum_{j=0}^{n} \binom{n}{j} f\left(\frac{j}{n}\right) x^j (1-x)^{n-j}, \qquad x \in [0,1],$$

bilden eine Korovkin-Folge auf $C[0,1]$.

Beweis: Offensichtlich sind die B_n linear und monoton. Ferner folgt aus dem binomischen Lehrsatz sofort $B_n f_0 = f_0$. Für f_1 gilt wegen $\binom{n}{j}\frac{j}{n} = \binom{n-1}{j-1}$ für $n \geq 1$, dass

$$B_n f_1(x) = \sum_{j=1}^{n} \binom{n}{j} \frac{j}{n} x^j (1-x)^{n-j} = x \sum_{j=0}^{n-1} \binom{n-1}{j} x^j (1-x)^{n-1-j} = f_1(x).$$

ähnlich beweist man

$$\sum_{j=2}^{n} \binom{n}{j} \frac{j(j-1)}{n(n-1)} x^j (1-x)^{n-j} = x^2, \qquad x \in [0,1],$$

für $n \geq 2$, sodass aus

$$\frac{j^2}{n^2} = \frac{j(j-1)}{n^2} + \frac{j}{n^2} = \frac{n-1}{n}\frac{j(j-1)}{n(n-1)} + \frac{1}{n}\frac{j}{n}$$

[13] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Bernstein_Sergi.html`

für f_2 jetzt

$$\begin{aligned} B_n f_2(x) &= \frac{n-1}{n}\sum_{j=2}^{n}\binom{n}{j}\frac{j(j-1)}{n(n-1)}x^j(1-x)^{n-j} + \frac{1}{n}\sum_{j=1}^{n}\binom{n}{j}\frac{j}{n}x^j(1-x)^{n-j} \\ &= \frac{n-1}{n}x^2 + \frac{x}{n} \end{aligned}$$

folgt. Dies bedeutet aber vermöge

$$|f_2(x) - B_n f_2(x)| = \left|\frac{1}{n}x^2 - \frac{x}{n}\right| \leq \frac{2}{n}$$

gleichmäßige Konvergenz. □

Definition 3.47. *Man nennt die Polynome*

$$B_{j,n}(x) := \binom{n}{j}x^j(1-x)^{n-j},\ 0 \leq j \leq n$$

Bernsteinpolynome *auf* $[0,1]$.

Sie haben die Eigenschaften

$$\begin{aligned} B_{j,n}(x) &\in [0,1] \\ B_{j,n} &\text{ hat Grad } n \text{ für alle } j,\ 0 \leq j \leq n \\ \sum_{j=0}^{n} B_{j,n}(x) &= 1 \\ B_{j,n+1}(x) &= (1-x)B_{j,n}(x) + xB_{j-1,n}(x) \end{aligned}$$

für alle $x \in [0,1]$, wenn man die oben undefinierten Bernsteinpolynome als Null definiert (Beweis als Aufgabe). Abbildung 12 zeigt ihren Verlauf.

Damit haben wir das erste Hauptresultat dieses Abschnittes bewiesen:

Theorem 3.48 (Weierstraß)**.** *Zu jedem $f \in C[a,b]$ und jedem $\epsilon > 0$ gibt es ein Polynom p, sodass $\|f-p\|_\infty < \epsilon$ gilt.*

Beweis: Für $[a,b] = [0,1]$ folgt dies unmittelbar aus Satz 3.43 und Satz 3.46. Den allgemeinen Fall führt man auf diesen per linearer Transformation zurück. Mit $f \in C[a,b]$ ist $g(s) = f((b-a)s+a) \in C[0,1]$. Ist q das Polynom, dass g auf $[0,1]$ bis auf $\epsilon > 0$ approximiert, so ist $p(t) = q(\frac{t-a}{b-a})$, $t \in [a,b]$ das gesuchte Polynom für f. □

Die $L_p[a,b]$-Norm, die durch

$$\|f\|_p^p := \|f\|_{L_p[a,b]}^p = \int_a^b |f(x)|^p dx$$

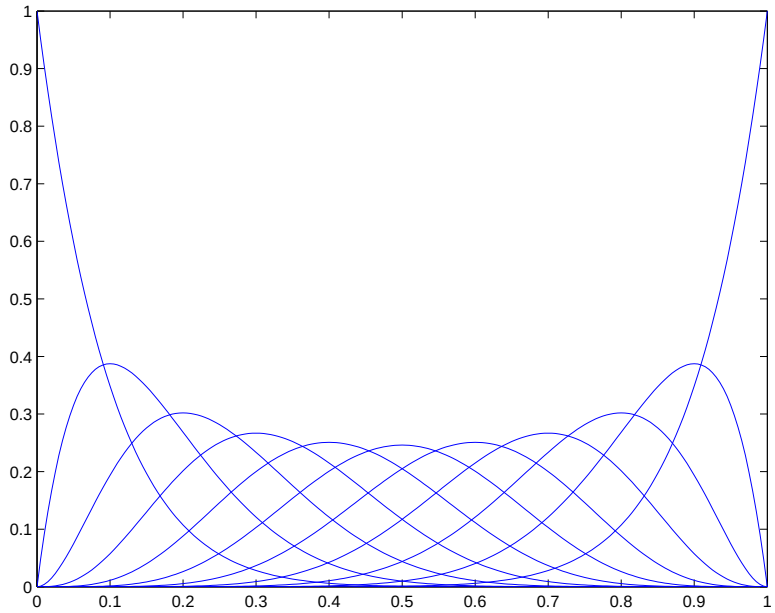

Figure 12: Die 11 Bernstein–Polynome vom Grad 10

definiert ist, lässt sich auf dem kompakten Intervall $[a, b]$ immer durch die Tschebyscheff-Norm vermöge

$$\|f\|_p^p = \int_a^b |f(x)|^p dx \leq \|f\|_\infty^p \int_a^b dx = \|f\|_\infty^p (b-a)$$

abschätzen. Dies bedeutet aber

Korollar 3.49. *Zu $f \in C[a, b]$ sei $p_n \in \Pi_n(\mathbb{R})$ so gewählt, dass $\|f - p_n\|_\infty \to 0$ für $n \to \infty$. Dann gilt auch $\|f - p_n\|_p \to 0$ für $n \to \infty$ für jedes $1 \leq p < \infty$. Konvergenz in der Tschebyscheff-Norm zieht also Konvergenz in jeder anderen L_p-Norm nach sich. Aus dem Spezialfall $p = 2$ folgt, daß die Legendre–Polynome vollständig in $C[-1, +1]$ und in $L_2[-1, +1]$ sind.*

Die Konvergenzgeschwindigkeit der Approximation durch Bernsteinoperatoren ist aber leider fürchterlich schlecht. Das sieht man, wenn man folgendes Programm ausführt:

```
% Bernstein Fit
clear all;
close all;
z=(0:0.001:1)';
fz=datfun(z);
val=zeros(length(z),1);
```

```
noise=0.00;
for n=1:150
    x=0:1/n:1;
    fx=datfun(x)';
    fx=fx.*(1+noise*(2*rand(size(fx))-1));
    val=0;
    B=bern(z,n);
    val=B*fx;
    figure(1)
    plot(z,val,z,fz)
    legend(sprintf('Degree: %d, Error = %e',n,norm(val-fz,Inf)),1)
    xlim([0 1]);
    ylim([0 1]);
    nn(n)=n;
    vn(n)=norm(val-fz,Inf);
    % F(n)=getframe;
end
% figure
% movie(F,1,5)
% movie2avi(F,'bernsteinmovie.avi','fps',10)
figure
loglog(nn,vn)
legend('Error',0)
```

Das Programm verwendet eine Routine `datfun.m` zum Erzeugen der Funktionswerte und eine Routine `bern.m` zum Erzeugen der Werte der Bernsteinpolynome, aber das wollen wir hier nicht weiter erklären. Man kann die Bernsteinpolynome durch ihre typische Rekursionsformel stabil berechnen, etwa mit

```
function V=bern(x,n)
% Bernstein basis of degree n on data vector x
V=zeros(length(x),n+1);
V(:,1)=ones(length(x),1);
for k=1:n
    V(:,k+1)=x.*V(:,k);
    for j=k:-1:2
        V(:,j)=(1-x).*V(:,j)+x.*V(:,j-1);
    end
    V(:,1)=(1-x).*V(:,1);
end
```

Jetzt wollen wir noch ein äquivalentes Resultat für periodische Funktionen und trigonometrische Polynome beweisen.

Definition 3.50. *Der Vektorraum der stetigen und 2π-periodischen Funktionen $f : \mathbb{R} \to \mathbb{R}$ wird mit $C_{2\pi}$ bezeichnet. Dementsprechend besteht $C^k_{2\pi}$ aus k-fach differenzierbaren, 2π-periodischen Funktionen.*

Die Rolle der Polynome übernehmen jetzt die trigonometrischen Polynome.

Lemma 3.51. *Das Produkt zweier trigonometrischer Polynome ist wieder eine trigonometrisches Polynom.*

Beweis: Dies folgt unmittelbar aus der Definition trigonometrischer Polynome und den folgenden Gleichungen:

$$\begin{aligned} \cos(jx)\cos(kx) &= \tfrac{1}{2}\left[\cos((j-k)x) + \cos((j+k)x)\right], \\ \sin(jx)\sin(kx) &= \tfrac{1}{2}\left[\cos((j-k)x) - \cos((j+k)x)\right], \\ \sin(jx)\cos(kx) &= \tfrac{1}{2}\left[\sin((j+k)x) + \sin((j-k)x)\right], \end{aligned}$$

die man leicht verifiziert. □

Lemma 3.52. *Sei $f \in C[0,\pi]$ und $\epsilon > 0$. Dann existiert ein gerades trigonometrisches Polynom T, sodass $\|f - T\|_\infty < \epsilon$ gilt.*

Beweis: Die Funktion g sei definiert durch $g(x) = f(\arccos(x))$, $x \in [-1,1]$. Dann ist $g \in C[-1,1]$, und nach Satz 3.48 existiert ein algebraisches Polynom $p(x) = \sum_{j=0}^n c_j x^j$ mit $|f(\arccos(x)) - p(x)| < \epsilon$ für alle $x \in [-1,1]$. Dies bedeutet aber $|f(x) - p(\cos x)| < \epsilon$ für $x \in [0,\pi]$. Nach Lemma 3.51 ist $T(x) := p(\cos x) = \sum_{j=0}^n c_j \cos^j(x)$ das gesuchte gerade trigonometrisches Polynom. □

Nach diesen Vorbereitungen können wir den zweiten Weierstraßschen Satz beweisen. Die Tschebyscheff-Norm bezieht sich hier auf das Intervall $[0, 2\pi]$. Da die betrachteten Funktionen aber 2π-periodisch sind, gibt sie auch das Maximum auf ganz $\mathbb{R}$ an.

Theorem 3.53 (Weierstraß)**.** *Zu jedem $f \in C_{2\pi}$ und jedem $\epsilon > 0$ existiert ein trigonometrisches Polynom T, sodass $\|f - T\|_\infty < \epsilon$ gilt.*

Beweis: Die Funktionen $f(x) + f(-x)$ und $(f(x) - f(-x))\sin x$ sind beide gerade. Zu $\epsilon > 0$ existieren nach Lemma 3.52 also zwei gerade trigonometrische Polynome T_1 und T_2, sodass

$$|f(x)+f(-x)-T_1(x)| < \epsilon/2 \text{ und } |(f(x)-f(-x))\sin x - T_2(x)| < \epsilon/2 \quad (3.54)$$

für alle $x \in [0, \pi]$ gilt. Nun sind die Funktionen innerhalb der Beträge aber offensichtlich gerade, sodass sich ihr Wert beim übergang von x zu $-x$ nicht ändert. Also gilt (3.54) für alle $x \in [-\pi, \pi]$ und, da alle Funktionen 2π-periodisch sind, sogar für alle $x \in \mathbb{R}$. Wir schreiben (3.54) jetzt in der Form

$$f(x) + f(-x) = T_1(x) + \alpha_1(x) \text{ und } (f(x) - f(-x)) \sin x = T_2(x) + \alpha_2(x) \tag{3.55}$$

mit gewissen Funktionen α_1, α_2, die $|\alpha_1(x)|, |\alpha_2(x)| < \epsilon/2$ für alle x erfüllen. Nun multiplizieren wir die erste Gleichung in (3.55) mit $\sin^2 x$ und die zweite mit $\sin x$, addieren die Ergebnisse und dividieren das Resultat noch durch zwei. Dies führt zu

$$f(x) \sin^2 x = T_3(x) + \beta(x) \tag{3.56}$$

mit einem trigonometrischen Polynom T_3 und $|\beta(x)| < \epsilon/2$ für alle x. Die Konstruktion, die zu (3.56) geführt hat lässt sich aber für jedes beliebige $f \in C_{2\pi}$ durchführen, insbesondere auch für $f(\cdot + \frac{\pi}{2})$, was

$$f(x + \frac{\pi}{2}) \sin^2 x = T_4(x) + \gamma(x),$$

mit einem trigonometrischen Polynom T_4 und $|\gamma(x)| < \epsilon/2$ bedeutet. Ersetzt man in dieser Formel x durch $x - \frac{\pi}{2}$, so wird sie zu

$$f(x) \cos^2 x = T_5(x) + \delta(x), \tag{3.57}$$

wobei $T_5(x) = T_4(x + \frac{\pi}{2})$ und $|\delta(x)| < \epsilon/2$ ist. Addieren wir schließlich (3.56) und (3.57), so erhalten wir

$$f(x) = T_3(x) + T_5(x) + \delta(x) + \beta(x),$$

sodass $T = T_3 + T_5$ wegen $|\delta(x) + \beta(x)| < \epsilon$ das gesuchte trigonometrische Polynom ist. □ Natürlich zieht diese Konvergenz in der Unendlich-Norm wieder die Konvergenz in jeder anderen $L_p[0, 2\pi]$-Norm nach sich. Und im Spezialfall $p = 2$ folgt

Korollar 3.58. *Die trigonometrischen Polynome sind vollständig in $C_{2\pi}$ und die Tschebyscheff–Polynome sind vollständig in $C[-1, +1]$ unter dem zugehörigen inneren Produkt.* □

Als **Stone–Weierstraß–Sätze**[14] kann man das Ganze noch erheblich verallgemeinern, aber das werden wir zugunsten praxisbezogener Überlegungen bleiben lassen.

[14] `http://en.wikipedia.org/wiki/Stone-Weierstrass_theorem`

3.7 Konvergenzgeschwindigkeit von Fourierreihen

Die beste Approximation $S_m(f)$ aus (3.36) an eine 2π–peridische stetige Funktion f konvergiert also nach Korollar 3.58 gegen f in der $\|\cdot\|_\pi$-Norm, d.h. es gilt

$$f(x) = \frac{a_0}{2} + \sum_{j=1}^{\infty}(a_j \cos jx + b_j \sin jx) \tag{3.59}$$

und das ist eine Darstellung von f als **Fourierreihe**[15].

Man beachte, dass die Konvergenz nur in der $\|\cdot\|_\pi$-Norm, also letztlich im Mittel garantiert ist. Dies bedeutet insbesondere, dass die Gleichheit in (3.59) nicht als punktweise Identität zu verstehen ist. Im Gegenteil, es gibt Beispiele, wo die Reihe auf der rechten Seite von (3.59) an einzelnen Punkten divergiert. Wir werden gleich hinreichende Bedingungen herleiten, die auch punktweise Konvergenz garantieren.

Die Darstellung (3.59) von f wird mt (3.39) zu

$$f(x) = \sum_{j=-\infty}^{\infty} \widehat{f}(j)e^{ijx}. \tag{3.60}$$

Definition 3.61. *Die Zahlen $a_j(f)$, $b_j(f)$, $\widehat{f}(j)$ heißen* **Fourier-Koeffizienten** *von f. Die Abbildung $\widehat{f} : C_{2\pi} \to \mathbb{C}$ heißt auch (diskrete)* **Fourier-Transformation** *von f.*

Man kann die in Abschnitt 4.3 besprochene schnelle Fourier-Transformation benutzen, um die Fourier-Koeffizienten $\widehat{f}(j)$ näherungsweise effizient zu berechnen. Wir wollen hier aber auf Details verzichten, sie folgen später. Auch lassen wir Petitessen zum Konvergenzbegriff der obigen biinfiniten Reihe weg und kümmern uns stattdessen um Fragen der Konvergenzgeschwindigkeit und der punktweisen Konvergenz.

Theorem 3.62. *Sei $f \in C^k_{2\pi}$. Dann gilt*

$$\|f - S_m f\|_\pi \leq \frac{1}{(m+1)^k}\|f^{(k)} - S_m(f^{(k)})\|_\pi = \mathcal{O}(m^{-k}) \qquad \text{für } m \to \infty.$$

Beweis: Durch partielle Integration und auf Grund der 2π-Periodizität finden wir

$$c_j(f^{(k)}) = \frac{1}{2\pi}\int_0^{2\pi} f^{(k)}(x)e^{-ijx}dx = \frac{ij}{2\pi}\int_0^{2\pi} f^{(k-1)}(x)e^{-ijx} = (ij)c_j(f^{(k-1)}),$$

[15] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Fourier.html`

was per Induktion zu $c_j(f^{(k)}) = (ij)^k c_j(f)$ führt. Aus (3.19) erhalten wir damit

$$\begin{aligned} \|f - S_m f\|_\pi^2 &= \textstyle\sum_{|j|\geq m+1} |c_j(f)|^2 \leq \sum_{|j|\geq m+1} |j|^{-2k} |c_j(f^{(k)})|^2 \\ &\leq \frac{1}{(m+1)^{2k}} \|f^{(k)} - S_m f^{(k)}\|_\pi^2 \end{aligned}$$

und $\|f^{(k)} - S_m f^{(k)}\|_\pi$ konvergiert immer noch gegen Null, was den $o(m^{-k})$ Teil rechtfertigt. □

Diese genauere Konvergenzaussage erlaubt uns jetzt auch auf gleichmäßige Konvergenz zu schließen.

Korollar 3.63. *Zu $f \in C_{2\pi}^1$ ist die Fourier-Reihe gleichmäßig konvergent.*

Beweis: Aus dem Beweis von Satz 3.62 wissen wir bereits $c_j(f') = (ij)c_j(f)$. Daher gilt für die Ableitung von $S_m f$, dass

$$(S_m f)'(x) = \sum_{j=-m}^{m} c_j(f) \left(e^{ijx}\right)' = \sum_{j=-m}^{m} c_j(f)(ij)e^{ijx} = S_m(f')(x).$$

Da $f - S_m f$ senkrecht auf allen trigonometrischen Polynomen vom Grad $\leq m$ steht, folgt insbesondere $0 = (f - S_m f, 1)_\pi$, d.h das Integral über $f - S_m f$ verschwindet auf $[0, 2\pi]$. Also hat $f - S_m f$ in $[0, 2\pi]$ eine Nullstelle x^*. Der Hauptsatz der Differential- und Integralrechnung und die Cauchy-Schwarzsche Ungleichung liefern

$$\begin{aligned} |f(x) - S_m f(x)| &= |\textstyle\int_{x^*}^x (f - S_m f)'(t)dt| \leq \int_{x^*}^x |(f' - S_m f')(t)|dt \\ &\leq \sqrt{|x - x^*|}\sqrt{\pi}\|f' - S_m f'\|_\pi \leq \sqrt{2\pi}\sqrt{\pi}\|f' - S_m f'\|_\pi, \end{aligned}$$

und der letzte Ausdruck strebt gleichmäßig in x gegen Null. □

Korollar 3.64. *Zu $f \in C_{2\pi}^k$ konvergiert die Fourier-Reihe mindestens wie*

$$\|f - S_m f\|_\infty = o(m^{1-k}) \qquad m \to \infty.$$

Das ist nicht das beste denkbare Resultat, aber es sollte hier reichen. Optimale Konvergenzraten liefern die Sätze von Jackson, die man neben ihren Umkehrungen, den Sätzen von Bernstein, in der Standardliteratur zur univariaten Approximationstheorie finden kann. Wichtig ist vielmehr, daß man beobachten kann, daß hohe Konvergenzordnungen nur möglich sind, wenn man hohe Differenzierbarkeitsvoraussetzungen macht. Dieses Grundprinzip der Approximationstheorie wird uns noch oft wieder begegnen.

Wenn wir die obige Argumentation etwas verallgemeinern, so bekommen wir ineinandergeschachtelte Räume

$$H^k_{2\pi} := \left\{ f \text{ mit (3.59) und } \sum_{j=1}^{\infty} j^{-2k}(a_j^2 + b_j^2) < \infty \right\} \supset H^{k+1}_{2\pi}$$

die man auch als **Sobolevräume** bezeichnet und die man mit der verallgemeinerten Differentiationsabbildung D durch

$$D \ : \ H^k_{2\pi} \to H^{k-1}_{2\pi}, \ k \geq 1$$

verknüpfen kann, die formal die Einzelterme differenziert und die entstehenden Faktoren in die Koeffizienten schiebt. Für $k > 1$ ist das die normale Differentiation, aber für $k = 1$ kann man die übliche Differentiation nicht termweise ausführen.

Die Entwicklungen in Orthonormalsysteme sind ein Spezialfall der **harmonischen Analyse**[16], die man noch sehr viel weiter treiben kann, aber dazu haben wir keine Zeit.

4 Schnelle Algorithmen

In diesem Abschnitt geht es um praktisch wichtige schnelle Approximationsverfahren wie die schnelle Fourier–und Cosinustransformation. Letztere spielt bei den JPEG–Kompressionsmethoden eine zentrale Rolle. Aber zuerst müssen wir noch etwas Theorie nachholen. Wir sind zu Anfang wieder etwas geizig mit Details, weil sich hier vieles aus der Numerischen Mathematik oder der reellen Analysis wiederholt.

4.1 Trigonometrische Interpolation

Die Behandlung reeller trigonometrischer Polynome $T \in T_{\mathbb{R},m}$ aus Theorem 3.34 wird wesentlich erleichtert, wenn man wie in (3.39) auf Seite 44 vorgeht und ein reelles trigonometrisches Polynom bis auf einen komplexen "Phasenfaktor" e^{-imx} in ein komplexes algebraisches Polynom p überführt. Dabei sind die Koeffizienten keine beliebigen komplexen Zahlen, aber das wollen wir für einen Moment ignorieren und etwas allgemeiner vorgehen.

[16] `http://de.wikipedia.org/wiki/Harmonische_Analyse`

Definition 4.1. *Die Elemente der Menge*

$$T_{\mathbb{C},n-1} := \left\{ T : T(x) = \sum_{j=0}^{n-1} c_j e^{ijx} : c_j \in \mathbb{C} \right\} \tag{4.2}$$

heißen (komplexe) trigonometrische Polynome vom Grad $\leq n-1$.

Der Raum $T_{\mathbb{C},n-1}$ ist ein $\mathbb{C}$-linearer, endlich dimensionaler Raum. Er hat für $n \in \mathbb{N}$ die Dimension n über $\mathbb{C}$, und damit weist man leicht nach, dass der Raum $T_{\mathbb{R},m}$ die Dimension $2m+1$ über $\mathbb{R}$ hat.

Wie im Reellen zeigt man, dass es zu $n \in \mathbb{N}$ paarweise verschiedenen komplexen Stützstellen $z_0, \ldots, z_{n-1}$ und komplexen Stützwerten $f_0, \ldots, f_{n-1}$ genau ein komplexes algebraisches Polynom p vom Grad $\leq n$ gibt mit $p(z_j) = f_j$, $0 \leq j \leq n-1$. Das kann man für Punkte auf dem Einheitskreisrand anwenden, und wir werden insbesondere den äquidistanten Fall unten behandeln.

Beim Rückgang ins Reelle bekommt man dann auch zu beliebigen paarweise verschiedenen Stützstellen $x_0, \ldots, x_{2m} \in [0, 2\pi)$ und reellen Daten $f_0, \ldots, f_{2m}$ genau ein reelles trigonometrisches Polynom $T \in T_{\mathbb{R},m}$ mit $T(x_j) = f_j$, $0 \leq j \leq 2m$. Das wollen wir hier nicht nachrechnen, es verläuft genau nach dem obigen Umrechnungsprozeß zwischen komplexen und reellen Koeffizienten.

4.2 Äquidistante Stützstellen

Auf dem Einheitskreis sind die äquidistanten Stützstellen durch die *n-ten Einheitswurzeln*

$$\zeta_n := e^{\frac{2\pi i}{n}} \tag{4.3}$$

beschreibbar. Sie erfüllen offensichtlich die Beziehungen

$$\zeta_n^n = 1, \quad \zeta_n^{j+k} = \zeta_n^j \zeta_n^k, \quad \zeta_{mn}^{mk} = \zeta_n^k, \; \zeta_n^{-j} = \overline{\zeta_n^j} \tag{4.4}$$

sowie

$$\frac{1}{n} \sum_{j=0}^{n-1} \zeta_n^{mj} = \begin{cases} 1 & \text{falls } m \text{ Vielfaches von } n, \\ 0 & \text{sonst.} \end{cases} \tag{4.5}$$

Letzteres ist etwas weniger offensichtlich, aber es folgt aus der Summationsformel für Partialsummen der geometrischen Reihe. Ist m kein Vielfaches

von n, so gilt $\zeta_n^m \neq 1$ und man hat

$$\begin{aligned}\sum_{j=0}^{n-1} \zeta_n^{mj} &= \sum_{j=0}^{n-1} (\zeta_n^m)^j \\ &= \frac{1-(\zeta_n^m)^n}{1-\zeta_n^m} \\ &= \frac{1-\zeta_n^{mn}}{1-\zeta_n^m} = 0.\end{aligned}$$

Ist aber m ein Vielfaches von n, so ist die Summe gleich n, weil alle Summanden Eins sind.

Von zentraler Bedeutung ist folgendes Resultat:

Theorem 4.6. *Sind für $n \in \mathbb{N}$ die äquidistanten reellen Stützstellen $x_j = \frac{2\pi j}{n}$, $0 \leq j \leq n-1$ bzw die äquidistanten komplexen Stützstellen $\zeta_n^0 = e^{ix_0}, \ldots, \zeta_n^{n-1} = e^{ix_{n-1}}$ und die Stützwerte $f_0, \ldots, f_{n-1} \in \mathbb{C}$ gegeben, so erfüllt das eindeutig bestimmte komplexe trigonometrische Interpolationspolynom*

$$p(x) = \sum_{j=0}^{n-1} c_j e^{ijx}, \qquad x \in [0, 2\pi),$$

die Gleichung

$$f_k = \sum_{j=0}^{n-1} c_j \zeta_n^{jk} \qquad 0 \leq k \leq n-1, \tag{4.7}$$

und hat die Koeffizienten

$$c_j = \frac{1}{n} \sum_{k=0}^{n-1} f_k \zeta_n^{-jk}, \qquad 0 \leq j \leq n-1. \tag{4.8}$$

Beweis: Man setzt (4.8) in das Polynom ein und rechnet (4.7) aus, wobei man (4.5) verwendet. Genauer: für $0 \leq k < n$ hat man

$$\begin{aligned}p(x_k) &= \sum_{j=0}^{n-1} c_j e^{ijx_k} \\ &= \sum_{j=0}^{n-1} \left(\frac{1}{n} \sum_{m=0}^{n-1} f_m \zeta_n^{-jm} \right) \zeta_n^{jk} \\ &= \sum_{m=0}^{n-1} f_m \frac{1}{n} \sum_{j=0}^{n-1} \zeta_n^{j(k-m)} = f_k. \quad \square\end{aligned}$$

4.3 Schnelle Fouriertransformation

Die explizite Formel (4.8) zur Berechnung der Koeffizienten der trigonometrischen Interpolanten erlaubt es, jeden einzelnen Koeffizienten, sofern die Potenzen der Einheitswurzeln vorab bekannt sind, in $\mathcal{O}(n)$ Operationen auszurechnen, sodass man insgesamt $\mathcal{O}(n^2)$ Operationen benötigt, um die Interpolante komplett zu bestimmen. Dies ist im Vergleich zu den üblichen $\mathcal{O}(n^3)$ Operationen, die normalerweise zum Lösen des zugehörigen Gleichungssystem benötigt werden, bereits eine merkliche Verbesserung. Trotzdem lässt sich dieses Resultat noch weiter verbessern.

Bei der Bildung der Summen in (4.8) treten bei geradem $n = 2m$ bei mehreren verschiedenen Funktionswerten f_k numerisch die gleichen (oder nur im Vorzeichen verschiedenen) Faktoren $\zeta_n^{-jk} = e^{-\frac{2\pi i j k}{n}}$ auf. Genauer gilt

$$\zeta_n^{-j(k+m)} = \zeta_n^{-jk}\zeta_n^{-jm} = (-1)^j \zeta_n^{-jk}.$$

ähnliches gilt natürlich auch für die diskrete Fourier Synthese. Diese Tatsache kann man ausnutzen, um durch geschicktes Zusammenfassen der Terme die Anzahl der Multiplikationen zu reduzieren. Auf dieser Tatsache beruht die *schnelle Fourier-Transformation* (englisch: *Fast Fourier Transform* oder *FFT*).

Bleiben wir bei geradem $n = 2m$, so gilt für die Koeffizienten mit geradem Index $j = 2\ell$ offenbar

$$\begin{aligned} c_{2\ell} &= \tfrac{1}{n}\textstyle\sum_{k=0}^{n-1} f_k \zeta_n^{-2\ell k} = \tfrac{1}{n}\sum_{k=0}^{m-1}\left(f_k\zeta_n^{-2\ell k} + f_{k+m}\zeta_n^{-2\ell(k+m)}\right) \\ &= \tfrac{1}{m}\sum_{k=0}^{m-1} \underbrace{\frac{f_k + f_{k+m}}{2}}_{f_k^{(1)}} \zeta_m^{-\ell k}, \end{aligned}$$

während für ungeraden Index $j = 2\ell + 1$ anlog

$$c_{2\ell+1} = \frac{1}{m}\sum_{k=0}^{m-1} \frac{f_k - f_{k+m}}{2}\zeta_n^{-(2\ell+1)k} = \frac{1}{m}\sum_{k=0}^{m-1} \underbrace{\frac{f_k - f_{k+m}}{2}\zeta_n^{-k}}_{f_{m+k}^{(1)}} \zeta_m^{-\ell k}$$

folgt. Statt einer Fourier-Transformation der Länge n hat man nun also zwei Fourier-Transformationen der Länge $n/2$, eine für die Koeffizienten mit geradem Index und eine für die Koeffizienten mit ungeradem Index. Ist n nicht nur gerade, sondern eine Zweierpotenz $n = 2^p$, so lässt sich dieser Prozess iterieren. Geht man wieder davon aus, dass die Potenzen der Einheitswurzeln

vorliegen, so ergibt sich für die Anzahl der komplexen Multiplikation und Additionen offenbar $M(n) = n/2 + 2M(n/2)$, bzw. $A(n) = n + 2A(n/2)$, was sich beides zu $\mathcal{O}(n \log n)$ auflösen lässt. Die Anzahl der Multiplikationen ist tatsächlich noch geringer, wenn man berücksichtigt, dass in jedem Schritt $\zeta^{-0} = 1$ vorkommt. Dies ändert aber nicht das asymptotische Verhalten.

Hier ist ein simples Paar von MATLAB-m-files zur FFT.

```
function c=myfft(f)
% primitive fft implementation for powers of two
n=length(f);
if n==1
    c=f;
    return
end
m=n/2;
fplus=(f(1:m)+f(m+1:n))/2;
fminus=exp(-2*pi*(0:m-1)'/n).*(f(1:m)-f(m+1:n))/2;
ceven=myfft(fplus);
codd =myfft(fminus);
c(1+2*(0:m-1),1)=ceven;
c(2+2*(0:m-1),1)=codd;
return

function c=myifft(f)
% primitive inverse fft implementation for powers of two
n=length(f);
if n==1
    c=f;
    return
end
m=n/2;
fplus=f(1:m)+f(m+1:n);
fminus=exp(+2*pi*(0:m-1)'/n).*(f(1:m)-f(m+1:n));
ceven=myifft(fplus);
codd =myifft(fminus);
c(1+2*(0:m-1),1)=ceven;
c(2+2*(0:m-1),1)=codd;
return
```

und ein Treiberprogramm

```
clear all;
close all;
% Demo zur FFT, NAIV programmiert
n=64; % das wird der Grad
f=complex(rand(n,1),rand(n,1));
c=myfft(f);
ft=myifft(c);
% [f ft]
fehler=norm(ft-f)
```

.

4.4 Fourier–Kompression

Wir wollen jetzt durchspielen, wie man ganz naiv ein periodisches Signal komprimiert, indem man

1. mit schneller Fouriertransformation die Fourierkoeffizienten ausrechnet (Fourier–Analyse),
2. davon die betragsmäßig kleinsten wegwirft, und dann
3. mit der schnellen Fouriertransformation zurücktransformiert (Fourier–Synthese).

Die dazu nötigen Bausteine haben wir schon. Wir fassen ein Signal aus n reellen Zahlen (eine `.wav`–Datei enthält nichts anderes) als Werte $f(2\pi k/n),\ 0 \leq k < n$ einer 2π–periodischen Funktion auf und kümmern uns nicht darum, ob diese Funktionswerte eine glatte periodische Fortsetzung haben. Wir scheren uns auch nicht um die trigonometrische Interpolation, sondern stecken die Werte als $f_k := f(2\pi k/n),\ 0 \leq k < n$ in die Gleichung (4.8). Dabei verschenken wir n reelle Freiheitsgrade, weil wir keine Imaginärteile haben. Dann rechnen wir mit der schnellen Fouriertransformation die komplexen Zahlen c_j aus, setzen die mit kleinem Betrag auf Null und transformieren mit (4.7) zurück. Dabei haben wir das Problem, sicherzustellen, daß das Ergebnis auch nach dem Nullsetzen gewisser c_j wieder reell ist. Aber wir

haben

$$\begin{aligned}\overline{c_j} &= \frac{1}{n}\sum_{k=0}^{n-1} f_k \zeta_n^{jk} \\ &= \frac{1}{n}\sum_{k=0}^{n-1} f_k \zeta_n^{jk-nk} \\ &= \frac{1}{n}\sum_{k=0}^{n-1} f_k \zeta_n^{-(n-j)k} \\ &= c_{n-j}\end{aligned}$$

für $1 \leq j \leq n-1$, und c_0 ist notwendig reell. Unsere Wegwerfstrategie müssen wir so einrichten, daß für die betragsgleichen Paare (c_j, c_{n-j}), $1 \leq j \leq n-1$ entweder beide Koeffizienten oder keiner weggeworfen wird, d.h. die Beziehung $\overline{c_j} = c_{n-j}$ sollte auch nach dem Nullsetzen gelten. Denn dann könen wir mit einem analogen Argument sehen, daß dann das Ergebnis von (4.7) wieder reell ist. Ein passendes Programm ist

```
function [g, prozent]=fftcomp(f, tol)
% primitiver FFT Kompressor mit relativer Toleranz tol
% fuer die  FFT Koeffizienten
y=fft(f); % mache blind die fft
tolloc=max(abs(y))*tol;  % absolute Toleranz berechnen
ind=find(abs(y)<tolloc); % Indizes zum Nullsetzen finden
y(ind)=0.0;              % Nullsetzen
g=real(ifft(y));         % Rücktransformieren, sicherheitshalber Realteil nehmen
prozent=100*(length(y)-length(ind))/length(y);
```

mit einem Treiberprogramm für Funktionen

```
close all;
clear all;
% Einfaches Treiberprogramm zur Kompression mit FFT
n=4096; % Punktezahl
tol=0.01; % relative Toleranz fuer die Fourierkoeffizienten
x=(0:2*pi/(n-1):2*pi)'; % Stützstellen
% f=exp(cos(x));          % Funktion darauf
f=abs(cos(x));          % Funktion darauf
[g proz]=fftcomp(f,tol); % FFT Kompressor aufrufen
figure                   % und Fehler plotten
plot(x,f-g)
proz_behalten=proz
```

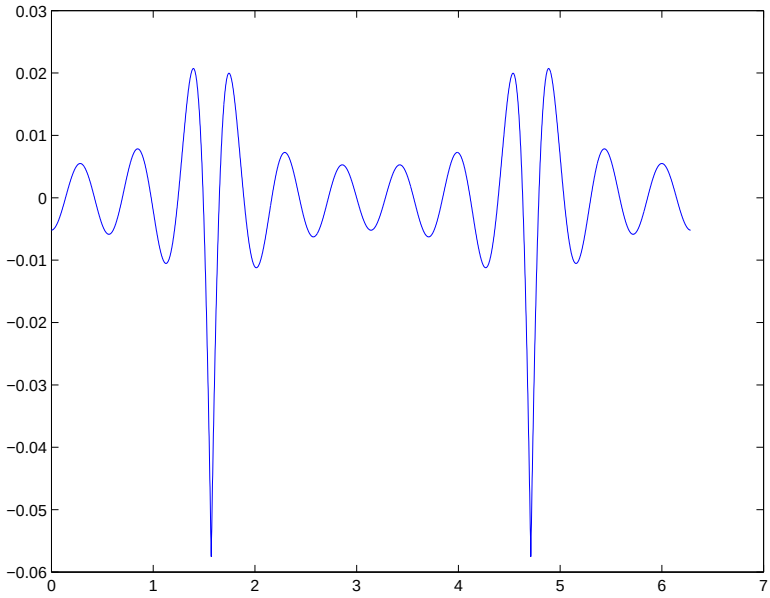

Figure 13: Fehler der Fourier–Kompression von $|\cos(x)|$ mit 0.28% der Koeffizienten

und der typischen Ausgabe in Abbildung 13.

Ein Programm zur Kompression von Tonsignalen ist

```
close all;
clear all;
% now we read a wave file
% [y,Fs,bits] = wavread('Bells.wav');
[y,Fs,bits] = wavread('Trumpet1.wav');
% [y,Fs,bits] = wavread('Toilet.wav');
% [y,Fs,bits] = wavread('Beeth5th.wav');
subplot(1,2,1)
plot(y);
title('Full Signal')
% soundsc(y,Fs,bits); % it sometimes does not work to play two
% disp('Hit a key')   % sound files in one m-file
% pause;
% return
tol=0.1 % relative tolerance in Fourier space
[z, proz]=fftcomp(y,tol); % do FT compression
proz_behalten=proz
subplot(1,2,2)
```

```
plot(z);
title('Compressed Signal')
soundsc(z,Fs,bits); % it sometimes does not work to play two
disp('Hit a key')   % sound files in one m-file
pause;
```

und der Ausgabe in Abbildung 14.

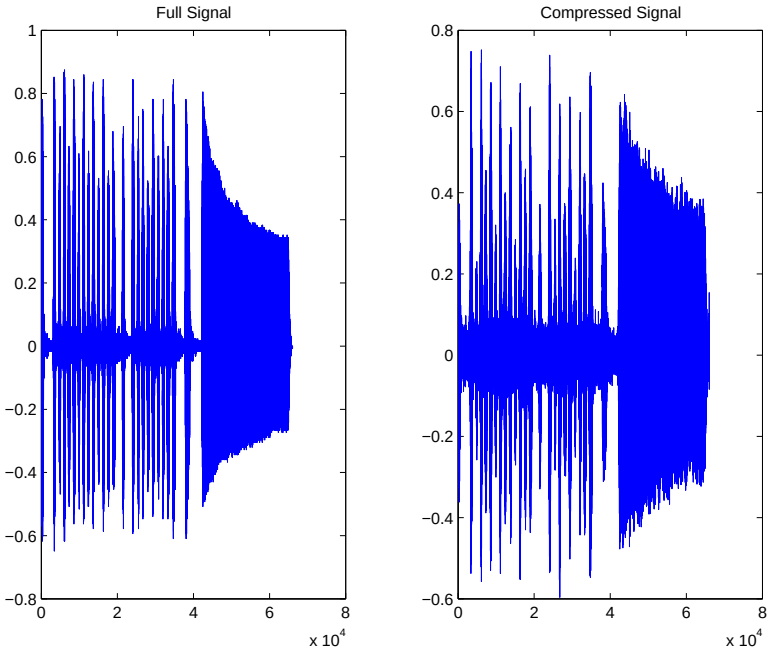

Figure 14: Sound–Kompression eines Trompetensignals mit 4.8% der Koeffizienten

4.5 Interpolation in Tschebyscheff–Punkten

Wir untersuchen jetzt die Interpolation von Polynomen in den Nullstellen oder Extremstellen von Tschebyscheff–Polynomen.

Der folgende Satz zeigt, daß man durch Interpolation in den Nullstellen des Tschebyscheff-Polynoms $T_{n+1}(x)$ für jede stetige Funktion die Güte der besten Tschebyscheff-Approximation bis auf einen Faktor $\log n$ erreichen kann:

Theorem 4.9. *Gegeben seien paarweise verschiedene Punkte $x_0, \ldots, x_n$ im Intervall $I := [-1, +1]$, und zugehörige Basisfunktionen $\omega_j^{(n)}(x)$ der La-*

grange[17] *–Interpolation in den Punkten* $x_0, \ldots, x_n$. *Zu beliebigem* $f \in C(I)$ *sei*

$$L_n(f)(x) := \sum_{j=0}^{n} \omega_j^{(n)}(x) f(x_j)$$

die Lagrange-Interpolierende von f *in* $x_0, \ldots, x_n$ *und* $\varepsilon_n := f - L_n f$ *sei die zugehörige Fehlerfunktion. Ist* $\eta_n(f) = \|f - T_n(f)\|_\infty$ *die Güte der besten Tschebyscheff-Approximation* $T_n(f)$ *von* f *bezüglich des Polynomraums* Π_n, *so gilt*

$$\|\varepsilon_n\|_\infty \leq \eta_n(f) \left(1 + \left\| \sum_{j=0}^{n} |\omega_j^{(n)}(x)| \right\|_\infty \right). \tag{4.10}$$

Wenn man die Nullstellen des Tschebyscheff-Polynoms T_{n+1} *als Punkte* x_j *wählt, haben die sogenannten* **Lebesgue–Konstanten**[18]

$$v_n := 1 + \left\| \sum_{j=0}^{n} |\omega_j^{(n)}(x)| \right\|_\infty$$

die Abschätzung

$$v_n \leq 1 + \frac{\pi}{2} + \frac{1}{2} \log(4n - 3).$$

Genauer gilt

$$v_n = 1 + \frac{1}{n+1} \sum_{j=0}^{n} \tan\left(\frac{j + \frac{1}{2}}{n+1} \cdot \frac{\pi}{2}\right)$$

mit schwierigerem Beweis und schwierigerer Abschätzung.

Beweis von Satz 4.9: Da der Operator L_n die Polynome n-ten Grades fest läßt, gilt

$$\begin{aligned}
\varepsilon_n = f - L_n(f) &= f - T_n(f) + T_n(f) - L_n(f) \\
&= f - T_n(f) + L_n(T_n(f) - L_n(f)) \\
&= f - T_n(f) + \sum_{j=0}^{n} \omega_j^{(n)}(x)(T_n(f) - L_n(f))(x_j) \\
&= f - T_n(f) + \sum_{j=0}^{n} \omega_j^{(n)}(x)((T_n(f)(x_j) - f(x_j))
\end{aligned}$$

[17] `http://www.gap-system.org/~history/Biographies/Lagrange.html`
[18] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Lebesgue.html`

und durch einfache Abschätzung folgt (4.10):

$$\begin{aligned} |\varepsilon_n(x)| &\leq |f(x) - T_n(f)(x)| + \sum_{j=0}^{n} |\omega_j^{(n)}(x)|\, |T_n(f)(x_j) - f(x_j)| \\ &\leq \eta_n(f)\left(1 + \textstyle\sum_{j=0}^{n} |\omega_j^{(n)}(x)|\right). \end{aligned}$$

Von hier ab lassen wir obere Indizes weg, die nur von n abhängen. Um das asymptotische Verhalten von v_n zu klären, müssen zunächst die Lagrange-Basisfunktionen $\omega_j(x)$ für die Interpolation in den $n+1$ Nullstellen

$$x_j := \cos\, \varphi_j \;\text{ mit }\; \varphi_j := \frac{(2j+1)\pi}{2n+2},\; 0 \leq j \leq n \tag{4.11}$$

des $(n+1)$-ten Tschebyscheff-Polynoms bestimmt werden. Zunächst wird der für die Theorie der Fourier-Reihen wichtige **Dirichlet–Kern**[19]

$$D_n(x) := \frac{1}{2}\,\frac{\sin\,(n+\frac{1}{2})\,x}{\sin\,\frac{x}{2}} = \frac{1}{2} + \sum_{j=1}^{n} \cos\, jx \tag{4.12}$$

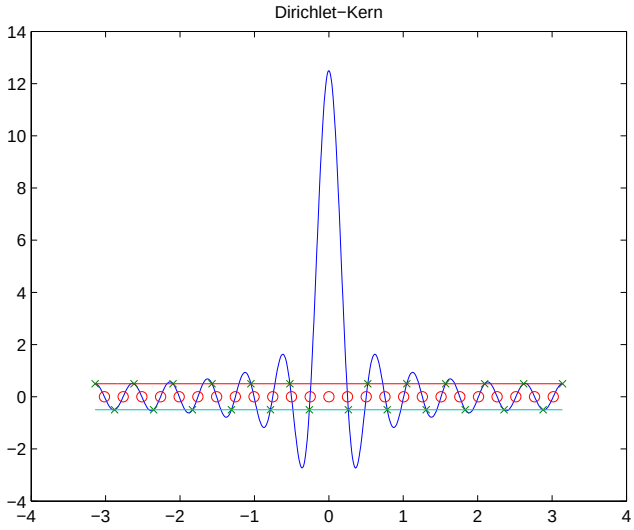

Figure 15: Dirichlet–Kern für $n = 12$

definiert (siehe Abbildung 15). Er ist ein gerades trigonometrisches Polynom, weil man durch Induktion leicht die Identität in (4.12) beweisen kann, und

[19] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Dirichlet.html`

erfüllt $D_n(0) = n + 1/2$ sowie für $j = 1, 2, \ldots, 2n + 1$ auch

$$D_n\left(\frac{j\pi}{n+1}\right) = \frac{1}{2}\,\frac{\sin\dfrac{(2n+1)j\pi}{2n+2}}{\sin\dfrac{j\pi}{2n+2}} = \frac{1}{2}\,\frac{\sin\left(j\pi - \dfrac{j\pi}{2n+2}\right)}{\sin\dfrac{j\pi}{2n+2}} = \frac{1}{2}(-1)^{j+1}.$$

In Abbildung 15 sind die Linien für die Werte $\pm\frac{1}{2}$ eingezeichnet, die diese Eigenschaft illustrieren.

Mit den Größen φ_j aus (4.11) ist das trigonometrische Polynom

$$u_j(\varphi) := \frac{1}{n+1}\,\left(D_n(\varphi + \varphi_j) + D_n(\varphi - \varphi_j)\right)$$

eine gerade Funktion und nimmt die Werte

$$\begin{aligned} u_j(\varphi_k) &= \frac{1}{n+1}\,\left(D_n(\varphi_k + \varphi_j) + D_n(\varphi_k - \varphi_j)\right) \\ &= \frac{1}{n+1}\,\left(D_n\left(\frac{(k+j+1)\pi}{n+1}\right) + D_n\left(\frac{|k-j|\pi}{n+1}\right)\right) \\ &= \delta_{jk} \end{aligned}$$

für $0 \le j, k \le n$ an. Dasselbe gilt per definitionem auch für $\omega_j(\cos\,\varphi)$, und deshalb folgt

$$\omega_j(\cos\,\varphi) = u_j(\varphi) = \frac{1}{n+1}\,\left(D_n(\varphi + \varphi_j) + D_n(\varphi - \varphi_j)\right).$$

Die Werte $\varphi \pm \varphi_j$ haben die Form

$$\varphi \pm \varphi_j = \varphi \pm \frac{(2j+1)\pi}{2n+2} = \varphi \pm \frac{\pi}{2n+2} \pm \frac{j\pi}{n+1},$$

und wenn wir $\psi = \varphi + \frac{\pi}{2n+2}$ setzen, bekommen wir die Werte

$$\psi + \frac{j\pi}{n+1},\ 0 \le j \le n$$

und

$$\psi - 2\frac{\pi}{2n+2} - \frac{j\pi}{n+1},\ 0 \le j \le n,$$

somit alle

$$\psi + \frac{j\pi}{n+1},\ -n - 1 \le j \le n,$$

und das sind $2n+2$ äquidistante Werte mit Abstand $\pi/(n+1)$. Zur Bestimmung von v_n ist also D_n an $2n+2$ äquidistanten Punkten mit Abstand $\pi/(n+1)$ auszuwerten, d.h. man kann v_n über

$$v_n \leq 1 + \frac{1}{n+1} \max_{\psi \in \mathbb{R}} \sum_{j=0}^{2n+1} |D_n(\psi + \frac{j\pi}{n+1})| \tag{4.13}$$

abschätzen. Die obige Summe ist aber eine Funktion mit Periode $h = \pi/(n+1)$, weil sie eine Summe über die $2n+2$ möglichen Verschiebungen einer festen 2π–periodischen Funktion mit Schrittweite h ist. Deshalb kann man die Bildung des Maximums auf die ψ mit $|\psi| \leq h/2$ einschränken.

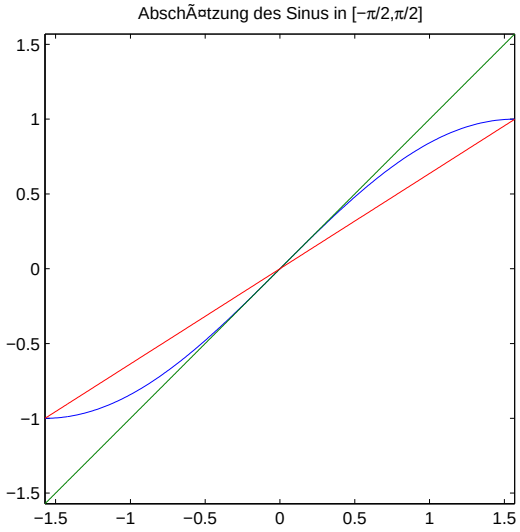

Figure 16: Abschätzung des Sinus

Für das Weitere ist wichtig, daß man

$$\frac{2}{\pi}|x| \leq |\sin\ x| \leq |x|$$

für alle $|x| \leq \pi/2$ hat (siehe Abbildung 16). In $[-h,h] = [-\pi/(2n+2),\ +\pi/(2n+2)]$ gilt dann

$$|D_n(\varphi)| \leq \frac{(n+\frac{1}{2})|\varphi|}{2 \cdot \frac{2}{\pi}\frac{|\varphi|}{2}} = (2n+1)\frac{\pi}{4}$$

wegen

$$(n+\frac{1}{2})|\varphi| \le (n+\frac{1}{2})\frac{\pi}{2n+2} < \frac{\pi}{2}.$$

Das deckt den Term mit $j = 0$ in (4.13) ab.

Weil der Nenner von D_n kritisch ist, sollte man sich ansehen, wo $(\psi + \frac{j\pi}{n+1})/2$ für $j \ge 1$ liegt. Man hat für $1 \le j \le 2n+1$ die Einschließung

$$0 < \frac{\pi}{4n+4} \le \frac{(2j-1)\pi}{4n+4} \le \frac{\psi + \frac{j\pi}{n+1}}{2} \le \frac{(2j+1)\pi}{4n+4} \le \frac{(4n+3)\pi}{4n+4} < \pi \quad (4.14)$$

und das ist eine Einteilung in Intervalle der Form

$$\left[\frac{(2j-1)\pi}{4n+4}, \frac{(2j+1)\pi}{4n+4}\right] \subseteq \left[\frac{\pi}{4n+4}, \pi - \frac{\pi}{4n+4}\right]$$

die sich bei $\pi/2$ überlappen, aber ansonsten spiegelsymmetrisch zu π liegen und weder 0 noch π enthalten. In diesen Intervallen schätzt man den Zähler von D_n immer durch 1 ab und betrachtet nur den Nenner. Dessen Argument ist für $1 \le j \le n$ nach oben abschätzbar durch

$$\frac{\psi + \frac{j\pi}{n+1}}{2} \le +\frac{\pi}{4n+4} + \frac{j\pi}{2n+2} = \frac{(2j+1)\pi}{4n+4} \le \frac{(2n+1)\pi}{4n+4} < \frac{\pi}{2}$$

und nach unten durch

$$\frac{\pi}{4n+4} \le \frac{(2j-1)\pi}{4n+4} = -\frac{\pi}{4n+4} + \frac{j\pi}{2n+2} \le \frac{\psi + \frac{j\pi}{n+1}}{2}.$$

Daraus folgt

$$D_n((\psi + \frac{j\pi}{n+1})/2) \le \frac{1}{2}\frac{1}{\frac{2}{\pi}\frac{(2j-1)\pi}{4n+4}} = \frac{n+1}{2j-1}$$

für $1 \le j \le n$.

Wir betrachten jetzt $j = n+1$. Das relevante Intervall ist

$$\left[\frac{(2n+1)\pi}{4n+4}, \frac{(2n+3)\pi}{4n+4}\right]$$

und liegt symmetrisch zu $\pi/2$. Das Minimum des Sinus darauf liegt auf beiden Grenzen links und rechts, so daß wir diesen Fall wie oben behandeln können.

Im Falle $n+2 \le j \le 2n+1$ liegt das Ganze so, daß man nicht die linken, sondern die rechten Intervallenden einsetzen muss, weil dort der Sinus monoton fällt, denn diese Intervalle liegen immer rechts von $\pi/2$. Man hat nach (4.14) die Abschätzung

$$D_n((\psi + \frac{j\pi}{n+1})/2) \le \frac{1}{2}\frac{1}{\frac{2}{\pi}\frac{(2j+1)\pi}{4n+4}} = \frac{n+1}{2j+1}$$

für $n+2 \le j \le 2n+1$.

Jetz bauen wir das zusammen. Wir erhalten

$$\begin{aligned} v_n &\le 1 + \frac{1}{n+1}\left(\frac{(2n+1)\pi}{4} + \sum_{j=1}^{n+1}\frac{n+1}{2j-1} + \sum_{j=n+2}^{2n+1}\frac{n+1}{2j+1}\right) \\ &\le 1 + \frac{\pi}{2} + \sum_{j=1}^{2n+1}\frac{1}{2j-1} \end{aligned}$$

Die Summe ist eine Untersumme mit Schrittweite Eins für ein Integral über $1/(2x-1)$, und deshalb gilt

$$\begin{aligned} v_n &\le 1 + \tfrac{\pi}{2} + \int_1^{2n+2}\frac{1}{2x-1}dx \\ &= 1 + \tfrac{\pi}{2} + \frac{1}{2}\log(4n-3). \quad \square \end{aligned}$$

Ein ähnliches Ergebnis hat man auch für die Interpolation in den Tschebyscheff–Extremstellen. Beide Interpolationen haben schnelle Transformationen, und diese beiden Transformationen sind Varianten der schnellen Cosinustransformation. Die schnellen Algorithmen für die Tschebyscheff–Interpolation in den Tschebyscheff–Nullstellen behandeln wir in Abschnitt 9.2.1. Wegen der Querverbindung zum Oxforder `chebfun`–System[20] bleiben wir hier erst noch bei der Tschebyscheff–Interpolation in den Extremstellen.

4.6 Diskrete Cosinustransformation

Wenn wir nicht wie in MATLAB indizieren, sollten wir bei Interpolation n–ten Grades die $n+1$ Extremstellen $x_k = \cos(\pi k/n),\ 0 \le k \le n$ von T_n nehmen und Funktionswerte $f_0, \ldots . f_n$ in diesen Punkten durch eine Linearkombination der Tschebyscheff–Polynome $T_0, \ldots, T_n$ interpolieren. Das bedeutet, ein System der Form

$$f_k = \sum_{j=0}^{n}\alpha_j T_j(x_k) = \sum_{j=0}^{n}\alpha_j \cos\left(\frac{\pi jk}{n}\right),\ 0 \le k \le n \qquad (4.15)$$

[20] `http://www2.maths.ox.ac.uk/chebfun/`

zu lösen. Die Koeffizientenmatrix ist symmetrisch, und es wäre schön, wenn sie bis auf die Diagonale eine Orthogonalmatrix wäre, weil man dann die Inverse kennen würde und man sich nur noch um schnelle Algorithmen kümmern müßte. Wegen

$$\cos\left(\frac{\pi jk}{n}\right) = \cos\left(\frac{2\pi jk}{2n}\right) = \frac{e^{\frac{2\pi jk}{2n}} + e^{-\frac{2\pi jk}{2n}}}{2} = \frac{\zeta_{2n}^{jk} + \zeta_{2n}^{-jk}}{2}$$

müßte man das mit einer Variante der schnellen Fouriertransformation und einiger Rechnung hinbekommen.

Es folgt jetzt ein Trick, der es für alle Varianten der Cosinustransformation und speziell auch der Interpolation mit Tschebyscheff–Polynomen in Tschebyscheff–Extremstellen oder Nullstellen erlaubt, ein Orthogonalitätsargument auszuführen. Die Idee ist, sich die Vektoren mit Einträgen $T_j(x_k)$, nämlich

$$v^k := (T_0(x_k), \ldots, T_n(x_k))^T \in \mathbb{R}^{n+1}$$

anzusehen und sie als Eigenvektoren passend gebastelter tridiagonaler symmetrischer Matrizen der Form

$$\begin{pmatrix} \alpha & -\beta & 0 & 0 & \ldots & 0 \\ -\beta & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & -1 & \ldots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \end{pmatrix}$$

zu schreiben, wobei das untere rechte Ende der Matrix analog gebaut ist. Wir halten k und n fest und sehen uns die Lage "mitten" in der Matrix an. Soll λ_k der Eigenwert zum Eigenvektor v^k sein, so muß man erreichen, daß

$$-T_{j-1}(x_k) + 2T_j(x_k) - T_{j+1}(x_k) = \lambda_k T_j(x_k)$$

gilt. Wegen der Rekursionsformel der Tschebyscheffpolynome ist das dasselbe wie

$$(2 - 2x_k)T_j(x_k) = \lambda_k T_j(x_k)$$

und das klappt, wenn man $\lambda_k := 2 - 2x_k \in [0, 4]$ setzt, und zwar für alle Punktwahlen aus $[-1, +1]$. Aber wenn man ans obere oder untere Ende der Matrix geht, muß man Modifikationen anbringen, die von der Punktwahl abhängen. Das werden wir gleich sehen.

Gehen wir in die zweite Zeile. Dort geht alles gut, wenn wir unseren zukünftigen Eigenvektor etwas modifizieren:

$$v^k := (T_0(x_k)/\beta, T_1(x_k) \ldots, T_n(x_k))^T.$$

Die erste Zeile muß dann

$$\alpha\frac{T_0(x_k)}{\beta} - \beta T_1(x_k) = \lambda_k\frac{T_0(x_k)}{\beta}$$

erfüllen, und das ist

$$\frac{\alpha}{\beta} - \beta x_k = \frac{\lambda_k}{\beta} = \frac{2-2x_k}{\beta}$$

was mit $\beta = \sqrt{2}$ und $\alpha = 2$ funktioniert. Das ging noch ohne Annahmen über die Lage der Punkte. Am unteren Ende ist alles etwas komplizierter, aber klappt auch. Man setzt

$$v^k := (T_0(x_k)/\beta, T_1(x_k), \ldots, T_{n-1}(x_k), T_n(x_k)/\beta)^T$$

und braucht

$$-\beta T_{n-1}(x_k) + \alpha\frac{T_n(x_k)}{\beta} = \lambda_k\frac{T_n(x_k)}{\beta}.$$

Das ist

$$-\beta^2 T_{n-1}(x_k) + \alpha T_n(x_k) = \lambda_k T_n(x_k),$$

und jetzt setzen wir unsere Tschebyscheff–Extremstellen ein. Wir bekommen erst

$$-\beta^2\cos\left(\tfrac{\pi(n-1)k}{n}\right) + \alpha\cos\left(\tfrac{\pi nk}{n}\right) \;=\; \lambda_k\cos\left(\tfrac{\pi nk}{n}\right)$$

und mit Hilfe des cosinus–Additionstheorems

$$-\beta^2\cos\left(\frac{\pi k}{n}\right) + \alpha = \lambda_k = 2 - 2x_k = 2 - 2\cos\left(\frac{\pi k}{n}\right).$$

Wieder sehen wir, daß $\beta = \sqrt{2}$ und $\alpha = 2$ funktioniert. Also:

Theorem 4.16. *Sind $x_0, \ldots, x_n$ die Extremstellen von T_n, so sind die Vektoren*

$$v^k := (T_0(x_k)/\sqrt{2}, T_1(x_k), \ldots, T_{n-1}(x_k), T_n(x_k)/\sqrt{2})^T$$

für $0 \le k \le n$ orthogonal. Das schreibt man auch als

$$\begin{aligned} \sum_{k=0}^{n}{}'' T_i(x_k)T_j(x_k) &= 0 \text{ für alle } 0 \le i < j \le n \\ \sum_{k=0}^{n}{}'' T_k(x_i)T_k(x_j) &= 0 \text{ für alle } 0 \le i < j \le n \end{aligned} \tag{4.17}$$

wobei der Doppelstrich an der Summe andeutet, daß der erste und letzte Term zu halbieren ist.

Wir haben die erste Gleichung schon bewiesen, aber die zweite folgt durch Transponieren der Matrix. □

Das ist eine "diskrete" Orthogonalitätseigenschaft der Tschebyscheff–Polynome. Wenn man das Ganze als Diskretisierung eines Integrals sieht, bemerkt man auch eine Ähnlichkeit mit der iterierten Trapezregel.

Man muß aber noch

$$c_j := \sum_{k=0}^{n}{}'' T_j^2(x_k) \text{ für alle } 0 \le j \le n$$

ausrechnen, um weiterzukommen. Klar ist $c_0 = c_n = n$, und für $1 \le j < n$ kann man $1 + \cos(2\varphi) = 2\cos^2(\varphi)$ für

$$\sum_{k=0}^{n}{}'' T_j^2(x_k) \;=\; \frac{n}{2} + \frac{1}{2}\sum_{k=0}^{n}{}'' \cos\left(\frac{2\pi jk}{n}\right)$$

benutzen. Nach (4.17) für $i = 0$ verschwindet die rechts stehende Summe für $1 \le 2j \le n$. Ersetzt man j durch $n - j$, so ändert sich in der rechts stehenden Summe nichts, und deshalb gilt $c_j = n/2$ für $1 \le j < n$.

Unsere Erfahrungen mit der Doppelstrichsumme zeigen uns, daß wir statt (4.15) eher

$$f_k = \sum_{j=0}^{n}{}'' \beta_j T_j(x_k) = \sum_{j=0}^{n}{}'' \beta_j \cos\left(\frac{\pi jk}{n}\right),\ 0 \le k \le n \tag{4.18}$$

schreiben sollten, und zwecks Inversion mit Hilfe von (4.17) sollten wir folgendes ausrechnen:

$$\begin{aligned}
\sum_{k=0}^{n}{}'' T_m(x_k) f_k &= \sum_{k=0}^{n}{}'' T_m(x_k) \sum_{j=0}^{n}{}'' \beta_j T_j(x_k) \\
&= \sum_{j=0}^{n}{}'' \beta_j \sum_{k=0}^{n}{}'' T_m(x_k) T_j(x_k) \\
&= \sum_{j=0}^{n}{}'' \beta_j \delta_{jm} c_m \\
&= \frac{n}{2}\beta_m,\ 0 \le m \le n.
\end{aligned}$$

Theorem 4.19. *Die Inversion der Tschebyscheff–Interpolation in (4.18) ist gegeben durch*

$$\beta_m = \frac{2}{n}\sum_{k=0}^{n}{}'' T_m(x_k) f_k,\ 0 \le m \le n. \quad \square$$

Dieses Paar von Formeln bezeichnet man in der Fassung

$$\begin{aligned} f_k &= \sum_{j=0}^{n}{}'' \beta_j \cos\left(\frac{\pi jk}{n}\right),\ 0 \le k \le n \\ \beta_m &= \frac{2}{n}\sum_{k=0}^{n}{}'' f_k \cos\left(\frac{\pi mk}{n}\right),\ 0 \le m \le n \end{aligned}$$

auch als DCT-I, die erste Variante der diskreten **Cosinustransformation**[21]. Die Ähnlichkeit mit der diskreten Fouriertransformation ist nicht zu übersehen, aber wir haben noch keinen schnellen Algorithmus dafür.

Wenn man die Auswertung der DCT-I–Formeln für gerade n direkt splittet, um einen schnellen Algorithmus zu bekommen, erhält man leider keine zwei DCT-I–Formeln der Länge $n/2$, sondern braucht eine andere DCT–Formel, die wir bisher nicht hergeleitet haben, aber die in Abschnitt 9.2.1 vorkommt. Man kann aber auch den etwas ineffizienten Umweg über eine FFT der Länge $2n$ gehen, indem man wie folgt aufteilt

$$\begin{aligned} & \sum_{k=0}^{n}{}'' f_k \cos\left(\frac{\pi mk}{n}\right) \\ = \ & \frac{f_0}{2} + (-1)^m \frac{f_n}{2} + \sum_{k=1}^{n-1} f_k \cos\left(\frac{\pi mk}{n}\right) \\ = \ & \frac{f_0}{2} + (-1)^m \frac{f_n}{2} + \frac{1}{2}\sum_{k=1}^{n-1} f_k \left(e^{\frac{2\pi imk}{2n}} + e^{-\frac{2\pi imk}{2n}}\right) \\ = \ & \frac{f_0}{2} + (-1)^m \frac{f_n}{2} + \frac{1}{2}\sum_{k=1}^{n-1} f_k \left(\zeta_{2n}^{mk} + \zeta_{2n}^{-mk}\right) \\ = \ & \frac{f_0}{2} + (-1)^m \frac{f_n}{2} + \frac{1}{2}\sum_{k=1}^{n-1} f_k \left(\zeta_{2n}^{mk} + \zeta_{2n}^{m(2n-k)}\right) \\ = \ & \frac{f_0}{2} + (-1)^m \frac{f_n}{2} + \frac{1}{2}\left(\sum_{k=1}^{n-1} f_k \zeta_{2n}^{mk} + \sum_{k=1}^{n-1} f_k \zeta_{2n}^{m(2n-k)}\right) \\ = \ & \frac{f_0}{2} + (-1)^m \frac{f_n}{2} + \frac{1}{2}\left(\sum_{k=1}^{n-1} f_k \zeta_{2n}^{mk} + \sum_{j=n+1}^{2n-1} f_{2n-j} \zeta_{2n}^{mj}\right) \\ = \ & \frac{1}{2}\sum_{k=0}^{2n-1} g_k \zeta_{2n}^{mk} \end{aligned}$$

mit den Elementen

$$g_0 = f_0, g_1 = f_1, \ldots, g_{n-1} = f_{n-1}, g_n = f_n, g_{n+1} = f_{n-1}, \ldots, g_{2n-1} = f_1.$$

Die letzte Summe folgt dann aus einer schnellen Fouriertransformation der Länge $2n$.

[21] `http://en.wikipedia.org/wiki/Discrete_cosine_transform`

4.7 Baryzentrische Interpolationsformeln

Wir behandeln jetzt noch die **baryzentrische Auswertung** von Interpolationsformeln, weil sie bei Interpolation in Tschebyscheff–Punkten stabiler als andere Techniken ist. Dazu halten wir uns an [1] und betrachten den allgemeinen Fall der Interpolation von Werten $f(x_0), \ldots, f(x_n)$ in Stützstellen $x_0 < x_1 < \ldots x_n$ in Lagrange–Form

$$p(x) = \sum_{j=0}^{n} f(x_j)\ell_j(x),\ \ell_j(x) = \prod_{0 \le k \ne j \le n} \frac{x - x_k}{x_j - x_k},\ 0 \le j \le n.$$

Mit

$$\ell(x) := \prod_{j=0}^{n} (x - x_j)$$

und

$$w_j := \prod_{0 \le k \ne j \le n} \frac{1}{x_j - x_k}$$

haben wir

$$\ell_j(x) = \frac{\ell(x) w_j}{x - x_j}$$

und bekommen

$$p(x) = \ell(x) \sum_{j=0}^{n} \frac{f(x_j) w_j}{x - x_j}$$

für alle x außerhalb der Stützstellen. Weil die Interpolation exakt ist für konstante Funktionen, folgt

$$1 = \ell(x) \sum_{j=0}^{n} \frac{w_j}{x - x_j}$$

und

$$p(x) = \sum_{j=0}^{n} f(x_j) \frac{\frac{w_j}{x - x_j}}{\sum_{k=0}^{n} \frac{w_k}{x - x_k}} \tag{4.20}$$

als gewichtete Summe der Funktionswerte. Die Faktoren w_j haben dabei die alternative Form

$$\frac{1}{w_j} = \ell'(x_j)$$

wegen

$$1 = \ell_j(x_j) = \ell'(x_j) w_j$$

nach der Regel von l'Hôspital[22]. Die Auswerteformel (4.20) wäre besonders stabil, wenn alle Gewichte bei $f(x_j)$ nichtnegativ wären, aber das ist im

[22] `http://www.gap-system.org/~history/Biographies/De_L%27Hopital.html`

Allgemeinen nicht der Fall. Immerhin ist sie relativ effizient, weil sie nur $\mathcal{O}(n)$ Operationen erfordert. Das ist bei Auswertung an einzelnen Punkten optimal.

Für äquidistante Punkte in $[0,1]$ bekommt man nach dem Herauskürzen der von j unabhängigen Terme

$$w_j = (-1)^j \binom{n}{j},\ 0 \le j \le n,$$

was wegen der starken Variation mit j zu einer instabilen Formel führt. Die Lage ist wesentlich besser bei Interpolation in den Tschebyscheff–Nullstellen. Man hat $\ell = 2^{-n} T_{n+1}$ und bekommt nach kurzer Rechnung und Beseitigung der nur von n abhängigen Faktoren das Ergebnis

$$w_j = (-1)^j \sin\left(\frac{(2j+1)\pi}{2n+2}\right),\ 0 \le j \le n.$$

Bei Interpolation in den Tschebyscheff–Extremstellen erhält man unter Weglassung von skalaren Faktoren und mit der Substitution $\cos(\varphi) = x$ das gerade trigonometrische Polynom

$$\ell(x) = \sin(n\varphi)\sin(\varphi) = \frac{T_{n-1}(x) + T_{n+1}(x)}{2},$$

weil es vom Grade $n+1$ ist und an den Stellen $\varphi_k = k\pi/n,\ 0 \le k \le n$ bzw. $x_k = \cos(\varphi_k)$ verschwindet. Ableiten ergibt

$$\ell'(x) \ = \ -\frac{n\cos(n\varphi)\sin(\varphi) + \sin(n\varphi)\cos(\varphi)}{\sin(\varphi)}$$

und in den Punkten mit Index $1 \le k < n$ folgt

$$\begin{aligned} \ell'(x_k) &= -n\cos(n\varphi_k) \\ &= -n(-1)^k. \end{aligned}$$

In den Punkten mit $k = 0$ und $k = n$ bekommt man mit der l'Hôspital–schen Regel

$$\begin{aligned} \ell'(x_k) &= -2n\cos(n\varphi_k) \\ &= -2n(-1)^k. \end{aligned}$$

Das ergibt die Faktoren

$$w_k := (-1)^k \left\{ \begin{array}{ll} \frac{1}{2} & k = 0,\ k = n \\ 1 & 1 \le k < n \end{array} \right\}$$

nach Weglassen irrelevanter von k unabhängiger Multiplikatoren.

Theorem 4.21. *Die Auswertung der Interpolante vom Grade $\leq n$ in den $n+1$ Extremstellen von T_n kann durch*

$$p(x) = \sum_{j=0}^{n}{}'' f(x_j) \frac{\frac{(-1)^j}{x-x_j}}{\sum_{k=0}^{n} \frac{(-1)^k}{x-x_k}}$$

außerhalb der Stützstellen geschehen.

Das Erstaunliche an der obigen Formel ist, daß nur die Stützstellen eingehen, die Tschebyscheff-Polynome kommen gar nicht vor.

Das Programm

```
clear all;
close all;
% Test fuer gewichtete Evaluation der T-Interpolierenden
n=640; % Grad des Interpolationspolynoms
xv=cos(pi*(0:n)'/n); % T-Extremstellen
sx=((-1).^(0:n))'; % Vorzeichenvektor
fxv=datfun(xv);      % Daten dort
plot(xv,fxv,'.')
title('Funktionswerte')
% Wir wollen mit der Auswertung ueber Koeffizienten
% vergleichen, hier trivial programmiert, sorry
mat=cheby(xv,n); % deshalb Matrix aufbauen
co=mat\fxv;      % Koeffizienten ausrechnen
% Jetzt auswerten
np=1510; % Zahl der Auswertungspunkte
hp=2/np; % Schrittweite
for i=1:np  % Schleife ueber Evaluationspunkte
    % Punkte, die hoffentlich nicht exakt auf den T-Extremstellen liegen
    xe(i,1)=-1+hp/2+(i-1)*hp; % Aufpunkt x
    xw=xv-xe(i,1); % Vektor der x_j-x
    w=sx./xw;      % Gewichte w_j
    w(1,1)=0.5*w(1,1); % Randkorrektur
    w(end,1)=0.5*w(end,1);
    sm(i,1)=sum(w); % Summe der Gewichte
    ww=w/sm(i,1); % relative Summe
    sp(i,1)=ww'*fxv; % Wert des Polynoms
    st(i,1)=sum(abs(ww)); % das wird dann der Funktionswert
    % der Lebesgue-Funktion
```

```
end
figure
plot(xe,sp-datfun(xe))
title('Fehler bei gewichteter Auswertung')
figure
plot(xe,st)
title('Lebesgue Funktion')
emat=cheby(xe,n);
val=emat*co;
figure
plot(xe,val-datfun(xe))
title('Fehler bei Auswertung ueber Koeffizienten')
figure
plot(xe,sp-val)
title('Fehler zwischen beiden Auswerteverfahren')
```

implementiert und testet diese gewichtete Auswertung, wobei als Vergleich die Auswertung über die Koeffizienten mit herangezogen wurde (bei sehr simpler und nicht nachahmenswerter Programmierung).

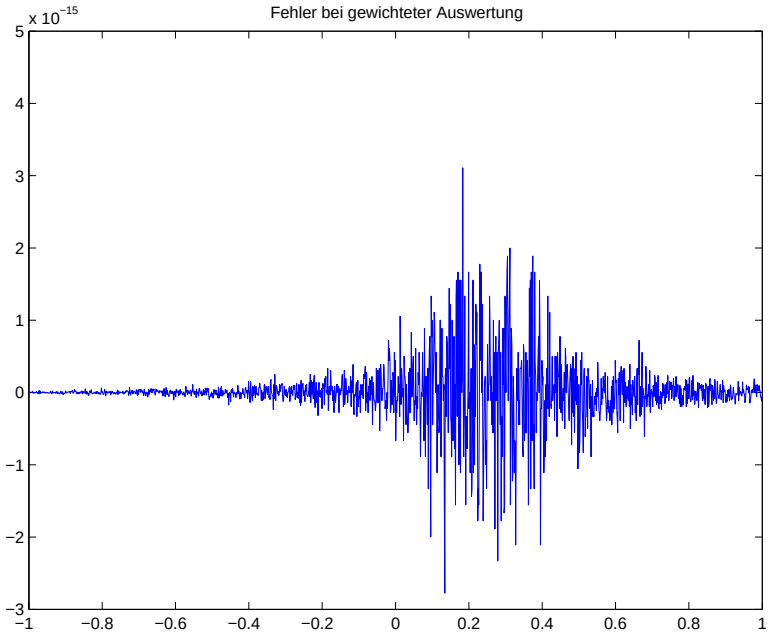

Figure 17: Fehlerfunktion bei Auswertung durch Mittelung

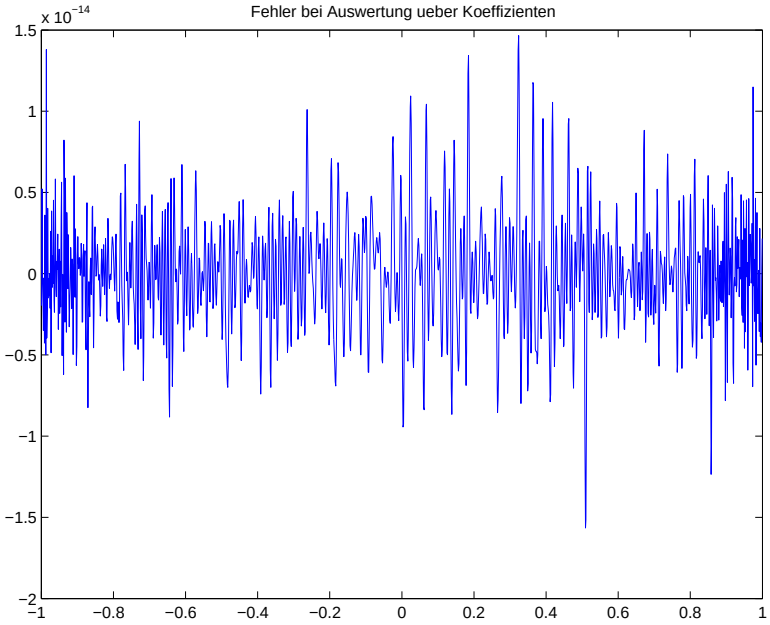

Figure 18: Fehlerfunktion bei Auswertung durch Koeffizienten

4.8 Clenshaw–Curtis–Quadratur

Diese hocheffiziente Methode zur numerischen Integration basiert auf der exakten Integration einer Darstellung

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n T_n(x) \tag{4.22}$$

einer Funktion auf $[-1, 1]$ als Reihe nach Tschebyscheff–Polynomen. Ob wir diese Reihe irgendwo abbrechen, oder ob wir sie durch Interpolation in den Tschebyscheff–Nullstellen oder –Extremstellen berechnen, spielt zunächst keine Rolle. Man braucht dazu die exakte Integration von

$$\begin{aligned} \tau_n &:= \int_{-1}^{+1} T_n(x)dx \\ &= \int_0^{\pi} \cos(n\varphi)\sin(\varphi)d\varphi \end{aligned}$$

mit partieller Integration. Es folgt

$$\begin{aligned}
\tau_n &:= [-\cos(n\varphi)\cos(\varphi)]_0^\pi - n\int_0^\pi \sin(n\varphi)\sin(\varphi)d\varphi \\
&= (-1)^n + 1 + n[\sin(n\varphi)\sin(\varphi)]_0^\pi + n^2\int_0^\pi \cos(n\varphi)\sin(\varphi)d\varphi \\
&= (-1)^n + 1 + 0 + n^2\tau_n, \\
\tau_{2n-1} &= 0,\ n \geq 1 \\
\tau_{2n} &= \frac{2}{1-(2n)^2},\ n \geq 0
\end{aligned}$$

wobei man den Fall $n = 1$ getrennt über $2\sin(\varphi)\cos(\varphi) = \sin(2\varphi)$ ausrechnen muß.

Damit wird das Integral von (4.22) zu

$$\int_{-1}^{+1} f(x)dx = a_0 + 2\sum_{n=1}^{\infty} \frac{a_{2n}}{1-(2n)^2}.$$

Weil die Koeffizienten a_{2n} für glatte Funktionen sehr schnell abfallen, braucht man in der Praxis nur wenige Terme zu berücksichtigen.

Das folgende Programm berechnet π mit der Clenshaw–Curtis–Quadratur, und zwar über

$$2\int_{-1}^{+1} \frac{1}{1+x^2}dx = 4\arctan(1) = \pi.$$

```
% Clenshaw-Curtis demo :
% calculate pi via integration
clear all;
close all;
n=16;
x=cos((0:n)*pi/n)';
fx=1./(1+x.^2);
cx=myiDCTI(fx);
dx=cx(1:2:end);
dx(2:end)=2*dx(2:end)./(1-(2*(1:n/2)').^2);
format long g
2*sum(dx)
pi
```

4.9 Konvergenzgeschwindigkeit

Wir untersuchen am Ende dieses Kapitels noch die Konvergenzgeschwindigkeit von Reihenentwicklungen nach Tschebyscheff–Polynomen, wenn diese näherungsweise

über Interpolation ausgerechnet wurden. Wir brauchen bei festem Grad n die Abbildungen

$$\begin{array}{rcccll} A_n & : & f & \mapsto & A_n(f) & = & \text{beste T-Approximation} \\ S_n & : & f & \mapsto & S_n(f) & = & \text{Fourier–Partialsumme} \\ I_n & : & f & \mapsto & I_n(f) & = & \text{Interpolante in T-Nullstellen oder Extrema} \\ Y_n & : & f & \mapsto & Y_n(f) & = & \text{Taylorpolynom in Null.} \end{array}$$

Wir arbeiten entweder auf $[-1,+1]$ oder mit $x = \cos(\varphi)$ auf $[0,\pi]$. Aus Satz 4.9 wissen wir, daß

$$\|f - I_n(f)\|_{\infty,[-1,+1]} \leq C\log(n)\|f - A_n(f)\|_{\infty,[-1,+1]}$$

gilt. Ferner haben wir auch

$$\begin{array}{rcl} \|f - A_n(f)\|_{\infty,[-1,+1]} & = & \|f(\cos(\varphi)) - A_n(f)(\cos(\varphi))\|_{\infty,[0,\pi]} \\ & \leq & \|f(\cos(\varphi)) - S_n(f(\cos())))(\varphi)\|_{\infty,[-\pi,+\pi]} \\ & \leq & C_f(n+1)^{1-k} \end{array}$$

für alle $f \in C^k[-1,+1]$ nach Korollar 3.64. Das ergibt den ersten Teil von

Theorem 4.23. *Für alle $f \in C^k[-1,+1]$ genügt die Tschebyscheff–Interpolation vom Grade $\leq n$ einer Fehlerabschätzung*

$$\|f - I_n(f)\|_{\infty,[-1,+1]} \leq C(f)(n+1)^{1-k}\log(n).$$

Die Clenshaw–Curtis–Quadratur, angewendet auf die durch $I_n(f)$ berechnete Interpolation durch Tschebyscheff–Polynome, hat höchstens den doppelten Fehler.

Beweis: Die Clenshaw–Curtis–Quadratur integriert $I_n(f)$ exakt. Also ist der Fehler gleich

$$\begin{array}{rcl} & & \left|\int_{-1}^{+1} f(x)dx - \int_{-1}^{+1} I_n(f)(x)dx\right| \\ & = & \left|\int_{-1}^{+1} (f(x) - I_n(f)(x))dx\right| \\ & \leq & 2\|f - I_n(f)\|_{\infty,[-1,+1]} \\ & \leq & 2C(f)(n+1)^{1-k}\log(n). \quad \square \end{array}$$

Für beliebig oft differenzierbnare Funktionen f auf $[-1,+1]$ könnte man auch mit der Taylorentwicklung vergleichen. Es folgt

$$\begin{array}{rcl} \|f - I_n(f)\|_{\infty,[-1,+1]} & \leq & C\log(n)\|f - A_n(f)\|_{\infty,[-1,+1]} \\ & \leq & C\log(n)\|f - Y_n(f)\|_{\infty,[-1,+1]} \\ & \leq & C\log(n)\dfrac{1}{(n+1)!}\|f^{(n+1)}\|_{\infty,[-1,+1]}. \end{array}$$

wenn man das übliche Restglied einsetzt und ausnutzt, daß die Zwischenstelle ξ immer zwischen 0 und dem Aufpunkt x liegt, sodaß man immer $|x - \xi| \leq 1$ hat. Das Problem ist hierbei aber der Term $\|f^{(n+1)}\|_{\infty,[-1,+1]}$, der etwa für $f(x) = 1/(2-x)$ wie $n!$ wächst. Man bekommt aber immer noch eine ausgezeichnete Fehlerabschätzung, wenn die Ableitungen "nur" geometrisch, d.h wie K^n für ein $K > 1$ ansteigen.

Die Funktionentheorie hilft dabei weiter. Wenn f als Funktion einer komplexen Variablen z in einer Umgebung eines Kreises um 0 mit Radius $R > 1$ noch holomorph (komplex differenzierbar) ist, hat man nach der **Integralformel von Cauchy**[23] die Darstellung

$$f^{(k)}(z) = \frac{(-1)^k k!}{2\pi i} \int_{|z|=R} \frac{f(\zeta)}{(\zeta - z)^{k+1}} d\zeta, \quad \text{für alle } k \geq 0,\ |z| \leq 1.$$

Dann folgt durch Abschätzung

$$\begin{aligned} |f^{(k)}(z)| &\leq \frac{k!}{2\pi} 2\pi R \frac{\|f\|_{\infty,|z|=R}}{(R-1)^{k+1}} \\ \frac{|f^{(k)}(z)|}{k!} &\leq C(f,R) \frac{R}{(R-1)^{k+1}} \quad \text{für alle } k \geq 0. \end{aligned}$$

Theorem 4.24. *Ist f eine Funktion, die auf einem Kreis mit Radius $R > 2$ um 0 in $\mathbb{C}$ komplex differenzierbar ist, so folgt*

$$\|f - I_n(f)\|_{\infty,[-1,+1]} \leq C(f,R) \log(n) (R-1)^{-n} \textit{ für alle } n \geq 0. \quad \square$$

Dieses Ergebnis ist nicht optimal, aber einigermaßen lehrreich, weil man eine (fast) geometrische Konvergenz hat, die umso besser wird, je größer der Differenzierbarkeitskreis ist. Ein besseres Ergebnis geht auf S.N. Bernstein zurück und wird hier weggelassen.

5 Sampling

Wir betrachten Interpolationsaufgaben auf einer biinfiniten Folge äquidistanter Punkte, d.h. auf $\mathbb{Z}$ oder $h\mathbb{Z}$ mit $h > 0$. So etwas ist der Standardfall in der **digitalen Signalverarbeitung**, weil man äquidistante diskrete Zeitreihen als Ergebnis einer Analog-Digital-Wandlung eines Signals bzw einer Funktion f bekommt. Man nennt dann die Werte $f(jh)$ für $j \in \mathbb{Z}$ ein **Sampling** von f. Aus einem digitalen Sampling eines ehemaligen Analogsignals

[23] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Cauchy.html`

bestehen unkomprimierte `wav`–Dateien[24]. Ein **AD–Wandler**[25] (Analog–Digital-Wandler) macht aus einem Analogsignal ein Sampling, während ein **DA–Wandler**[26] aus einem Sampling wieder ein Analogsignal macht. Die angegebenen Links beziehen sich dabei auf die Elektronik.

Es geht im folgenden aber darum, aus einem Sampling die Funktion **mathematisch** wieder zu rekonstruieren. Diese Rekonstruktion ist der Normalfall beim Hören einer CD oder eines MP3-komprimierten Signals nach der digitalen Dekompression.

5.1 Kardinale Interpolation

In Anlehnung an die Lagrange-Interpolation macht man das am einfachsten durch Verschieben und Skalieren einer **kardinalen** Funktion $K : \mathbb{R} \to \mathbb{R}$ mit

$$K(j) = \delta_{j0},\ j \in \mathbb{Z}.$$

Die Interpolation einer Funktion f auf $\mathbb{R}$ in den Punkten von $\mathbb{Z}$ ist dann einfach durch

$$K_{1,f}(x) := \sum_{j\in\mathbb{Z}} f(j)K(x-j),\ x \in \mathbb{R}$$

gegeben, wobei man aber noch die Konvergenz der Reihe sicherstellen muß. Auf $h\mathbb{Z}$ verwendet man entsprechend

$$K_{h,f}(x) := \sum_{j\in\mathbb{Z}} f(jh)K\left(\frac{x-jh}{h}\right),\ x \in \mathbb{R}. \tag{5.1}$$

Für kardinale Funktionen K hat man diverse Kandidaten, z.B. die Hutfunktion

$$K(t) := \begin{cases} 1-|t| & |t| \le 1 \\ 0 & \text{sonst} \end{cases}$$

oder die sinc-Funktion

$$\operatorname{sinc}(x) := \frac{\sin(\pi x)}{\pi x},\ x \in \mathbb{R}.$$

Man mache sich klar, daß letztere analytisch und sogar eine **ganze** Funktion im Sinne der Funktionentheorie ist, denn die vermeintliche Singularität in der Null ist hebbar. Man kann sich auch kardinale Funktionen aus Splines

[24] `http://en.wikipedia.org/wiki/WAV`
[25] `http://de.wikipedia.org/wiki/Analog-Digital-Umsetzer`
[26] `http://de.wikipedia.org/wiki/Digital-Analog-Umsetzer`

festen Grades bauen, aber das wollen wir hier nicht vertiefen. Aus physikalischen und mathematischen Gründen, die wir noch herzuleiten haben, interessiert man sich besonders für die Rekonstruktion mittels der kardinalen sinc–Funktion. Dabei untersuchen wir schließlich Abschätzungen des Fehlers $f - K_{h,f}$ und klären später, für welche K und f man die kardinalen Interpolanten überhaupt hinschreiben und stabil auswerten kann.

Das geht nicht ohne die Theorie der **Fouriertransformation**, die in Abschnitt 9.1 sehr allgemein und multivariat behandelt wird, für die wir aber im Folgenden eine Kurzversion anbieten.

5.2 Skizze zur Fouriertransformation

Wir verwenden hier formell die symmetrische **Fouriertransformation**[27] in einer Variablen als Formelpaar

$$\begin{aligned} \hat{f}(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} f(x) e^{-ix\omega} dx, \ \omega \in \mathbb{R} \\ f(x) &= \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} \hat{f}(\omega) e^{ix\omega} d\omega, \ x \in \mathbb{R}, \end{aligned} \tag{5.2}$$

das allerdings sehr erklärungsbedürftig ist. Man schreibt die zweite Formel oft in der Form

$$g^\wedge(x) := \frac{1}{\sqrt{2\pi}} \int_{\mathbb{R}} g(\omega) e^{ix\omega} d\omega, \ x \in \mathbb{R}$$

als **inverse Fouriertransformation**. Man nennt $\hat{f}$ die **Fouriertransformierte** zu f und $g^\wedge$ die **inverse Fouriertransformierte** zu g.

Wir beginnen unsere Erklärung mit der Parallelität zu Fourierreihen. Würde man das zweite Integral unter Wegfall des konstanten Faktors durch eine Summe über ganze Zahlen ersetzen, so bekäme man

$$f(x) = \sum_{k \in \mathbb{Z}} \hat{f}(k) e^{ikx} = \sum_{k \in \mathbb{Z}} \hat{f}(k) \left(\cos(kx) + i \sin(kx) \right),$$

was wie in (3.59) und (3.60) die Reproduktion einer 2π–periodischen Funktion f aus ihren Fourierkoeffizienten $\{\hat{f}(k)\}_{k \in \mathbb{Z}}$ beschreibt. Die Umkehrung ist dann ein Integral (3.40), das die Fourierkoeffizienten $\{\hat{f}(k)\}_{k \in \mathbb{Z}}$ aus f berechnet. Die Variable k steht dabei für eine Frequenz $2\pi k$ aks Vielfaches der Grundfrequenz 2π, und die Transformationen vermitteln zwischen komplexwertigen 2π–periodischen Funktionen einer Variablen x im **Ortsraum** und Folgen $\{\hat{f}(k)\}_{k \in \mathbb{Z}} \in \mathbb{C}^{\mathbb{Z}}$ mit Indizes $k \in \mathbb{Z}$ aus dem **Frequenzraum**.

[27] `http://de.wikipedia.org/wiki/Fourier-Transformation`

Analog dazu vermittelt das Formelpaar (5.3) eine Transformation zwischen nichtperiodischen komplexwertigen Funktionen, wobei f als Funktion der Ortsvariablen x aus dem Ortsraum $\mathbb{R}$ und $\hat{f}$ als Funktion der Frequenzvariablen ω aus $\mathbb{R}$ als Frequenzraum zu sehen ist.

Unklar ist aber noch, auf welchen Funktionenräumen die Abbildungen aus (5.3) definierbar und dann zueinander invers sind. Man kann mit einer stetigen Funktion f aus

$$L_1(\mathbb{R}) := \left\{ f \ : \ \mathbb{R} \to \mathbb{C} \ : \ \int_{\mathbb{R}} |f(t)| dt < \infty \right\}$$

in die erste Formel gehen und bekommt eine auf $\mathbb{R}$ beschränkte Funktion $\hat{f}$ heraus, von der man zeigen kann, daß sie stetig ist. Genauso geht das auch mit der zweiten Formel, aber man hat als Ergebnis der ersten Formel nicht automatisch auch $\hat{f} \in L_1(\mathbb{R}^d)$, um f mit der zweiten Formel wiederzubekommen. Das komplette Formelpaar, das ja eine Transformation und ihre Inverse darstellt, erfordert demnach Funktionen f und $\hat{f}$, die beide stetig und beschränkt auf $\mathbb{R}$ sind, in $L_1(\mathbb{R})$ liegen und dann über das Formelpaar verknüpft sind. Aber es ist nicht klar, für welche Funktionen so etwas gilt.

Man macht sich das Leben etwas einfacher, indem man das Formelpaar zuerst auf einem ziemlich "kleinen" Raum von **Testfunktionen** betrachtet und dort beweist, daß beide Formeln gelten und invers zueinander sind. Das funktioniert z.B. für den auf Laurent Schwartz[28] zurückgehenden Raum $\mathcal{S}$ der Funktionen, die auf $\mathbb{R}$ unendlich oft differenzierbar sind und bei $\pm\infty$ schneller als jedes Polynom abklingen. Das tut zum Beispiel die **Gaußglocke** [29] $f(x) = \exp(-x^2)$.

Dann beweist man, daß der Testraum dicht in

$$L_2(\mathbb{R}) := \left\{ f \ : \ \mathbb{R} \to \mathbb{C} \ : \ \int_{\mathbb{R}} |f(t)|^2 dt < \infty \right\}$$

liegt und beide Formeln als Abbildungen auf $L_2(\mathbb{R}^d) \cap \mathcal{S}$ in der L_2–Norm, die auf dem inneren Produkt

$$(f,g)_{L_2(\mathbb{R})} := \int_{\mathbb{R}} f(x)\overline{g(x)}dx \text{ für alle } f,g \in L_2(\mathbb{R})$$

basiert, stetig sind. Wenn man nun noch in der Definition von $L_2(\mathbb{R})$ das Lebesgue–Integral verwendet und damit $L_2(\mathbb{R})$ zu einem Hilbertraum macht,

[28] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Schwartz.html`
[29] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Gauss.html`

sind beide Formeln als Abbildungen stetig auf $L_2(\mathbb{R})$ fortsetzbar. Sie sind dort invers zueinander und es gilt die **Parseval-Plancherel-sche Gleichung**[30] [31]

$$(f,g)_{L_2(\mathbb{R})} = \int_{\mathbb{R}} f(t)\overline{g(t)}dt = \int_{\mathbb{R}} \hat{f}(\omega)\overline{\hat{g}(\omega)}d\omega = (\hat{f},\hat{g})_{L_2(\mathbb{R})}$$

für alle $f, g \in L_2(\mathbb{R})$. Man mache sich aber klar, daß bei dieser abstrakten Fortsetzung das Formelpaar (5.3) nicht als klassische Integration zu verstehen ist, sondern als Grenzwert der Anwendung der Formeln auf gewisse Cauchyfolgen aus dem Testraum, die f bzw. $\hat{f}$ in $L_2(\mathbb{R})$ approximieren. In allen praktischen Fällen, die uns über den Weg laufen, werden wir die Integrale aber klassisch ausrechnen können.

Im Folgenden sind einige formelle Rechenregeln nützlich. Beide Abbildungen aus (5.3) sind natürlich linear. Angewendet auf gerade Funktionen liefern sie reellwertige Funktionen, und man kann dann statt $e^{\pm ix\omega}$ in beiden Formeln $\cos(x\omega)$ schreiben, was die Parallelität zur Cosinustransformation illustriert. Sofern die Ableitungen existieren und fouriertransformierbar sind, gilt

$$\begin{aligned}(\hat{f})^{(k)}(\omega) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}(-ix)^k f(x)e^{-ix\omega}dx,\ \omega\in\mathbb{R}\\ f^{(k)}(x) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\hat{f}(\omega)(i\omega)^k e^{ix\omega}d\omega,\ x\in\mathbb{R}.\end{aligned}$$

Man kann die zweite Gleichung folgendermaßen interpretieren: Klingt $|\hat{f}|$ bei Unendlich schnell ab, und zwar so schnell, daß $|\hat{f}(\omega)\omega^k|$ noch quadratisch integrabel ist, so hat f eine k–te Ableitung, die in $L_2(\mathbb{R})$ liegt. Die Umkehrung gilt auch. Die Lage ist wie bei Fourierreihen: die Differenzierbarkeit einer Funktion wird durch das Abklingverhalten der Fouriertransformierten bestimmt.

Bei Translation um $t \in \mathbb{R}$ verhält sich die Fouriertransformation wie

$$\begin{aligned}\widehat{f(\cdot - t)}(\omega) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} f(x-t)e^{-i\omega x}dx\\ &= e^{-i\omega t}\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} f(y)e^{-i\omega y}dy\\ &= e^{-i\omega t}\hat{f}(\omega).\end{aligned}$$

Dann brauchen wir noch das Verhalten der Fouriertransformation gegenüber multiplikativer Skalierung $f_h(x) := f(x/h)$ im Urbildbereich für festes $h > 0$.

[30] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Parseval.html`
[31] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Plancherel.html`

Wir haben

$$\begin{aligned}\widehat{f_h}(\omega) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} f(x/h)e^{-i\left(\frac{x}{h}\right)h\omega}dx \\ &= \frac{h}{\sqrt{2\pi}}\int_{\mathbb{R}} f(y)e^{-iyh\omega}dy \\ &= h\hat{f}(h\omega) \\ &= h\widehat{f_{1/h}}(\omega) \text{ für alle } \omega \in \mathbb{R}.\end{aligned}$$

Weiter unten haben wir oft folgendes auszurechnen:

$$f\left(\frac{\cdot - jh}{h}\right)^{\wedge}(\omega) = he^{-ijh}\hat{f}(\omega h),$$

und das folgt durch Kombination der obigen beiden Rechenregeln.

Der Vorsicht halber sollte erwähnt werden, daß die Fouriertransformation gegenüber (5.3) manchmal mit anderen Faktoren oder anderen Skalierungen definiert wird, z.B.

$$\begin{aligned}\hat{f}(\omega) &= \int_{\mathbb{R}} f(x)e^{-2\pi ix\omega}dx,\ \omega \in \mathbb{R} \\ f(x) &= \int_{\mathbb{R}} \hat{f}(\omega)e^{2\pi ix\omega}d\omega,\ x \in \mathbb{R},\end{aligned} \tag{5.3}$$

was zu keinen wesentlichen Änderungen, aber manchmal zu etwas Verwirrung[32][33] führt.

Wir definieren schließlich noch die **charakteristische Funktion** zu einer Menge T als

$$\chi_T(t) := \left\{ \begin{array}{ll} 1 & t \in T \\ 0 & t \notin T \end{array} \right\}.$$

5.3 Die sinc–Funktion

Definition 5.4. *Wie schon oben vorweggenommen, wird*

$$S_{h,f}(t) := \sum_{k\in\mathbb{Z}} f(kh)\mathrm{sinc}\left(\frac{t}{h} - k\right) \tag{5.5}$$

zu einer Funktion $f : R \to \mathbb{C}$ und zu $h > 0$ die **Shannon-Reihe**[34] *genannt, und die Abbildung $f \mapsto S_{h,f}$ ist der* **Shannon-Operator**.

[32] http://en.wikipedia.org/wiki/Fourier_transform

[33] http://de.wikipedia.org/wiki/Fourier-Transformation

[34] http://www-history.mcs.st-andrews.ac.uk/Biographies/Shannon.html

Die Konvergenz dieser Reihe und der Definitionsbereich des Operators werden später geklärt. Wir müssen erst einmal nachsehen, was wir über die sinc–Funktion herausbekommen können.

Lemma 5.6. *Für jedes feste $x \in \mathbb{R}$ gilt*

$$\begin{aligned} \operatorname{sinc}\left(\frac{t-x}{h}\right) &= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega}e^{-ix\omega}d\omega \\ &= \frac{h}{\sqrt{2\pi}}\left(e^{-ix\omega}\chi_{[-\frac{\pi}{h},+\frac{\pi}{h}]}(\omega)\right)^{\vee}(t) \\ \operatorname{sinc}\left(\frac{\cdot-x}{h}\right)^{\wedge}(\omega) &= \frac{h}{\sqrt{2\pi}}e^{-ix\omega}\chi_{[-\frac{\pi}{h},+\frac{\pi}{h}]}(\omega). \end{aligned}$$

Beweis: Die erste Gleichung folgt aus

$$\begin{aligned} & \int_{-\pi/h}^{\pi/h} e^{i(t-x)\omega}d\omega \\ = & \int_{-\pi/h}^{\pi/h} e^{-i(t-x)\omega}d\omega \\ = & \frac{-1}{i(t-x)}e^{-i(t-x)\omega}|_{-\pi/h}^{+\pi/h} \\ = & \frac{-1}{i(t-x)}\left(e^{-i(t-x)\pi/h} - e^{+i(t-x)\pi/h}\right) \\ = & \frac{2i\sin((t-x)\pi/h)}{i(t-x)} \\ = & \frac{2\pi}{h}\frac{\sin((t-x)\pi/h)}{(t-x)\pi/h} \\ = & \frac{2\pi}{h}\operatorname{sinc}\left(\frac{t-x}{h}\right) \end{aligned}$$

und ist bis auf den Faktor $1/\sqrt{2\pi}$ eine inverse Fouriertransformation. Daraus folgt dann auch der Rest. □

Lemma 5.7. *Die Funktionen* $\operatorname{sinc}\left(\frac{t}{h}-k\right)$ *liegen in* $L_2(\mathbb{R})$ *und erfüllen die Orthogonalitätsrelation*

$$\left(\operatorname{sinc}\left(\frac{t}{h}-j\right), \operatorname{sinc}\left(\frac{t}{h}-k\right)\right)_{L_2(\mathbb{R})} = h\delta_{jk},\ j,k \in \mathbb{Z},\ h>0.$$

Insbesondere sind die Funktionen $s_{k,h}(t) := \frac{1}{\sqrt{h}}\operatorname{sinc}\left(\frac{t}{h}-k\right)$ *orthonormal in* $L_2(\mathbb{R})$.

Proof: Mit der Plancherel-Gleichung und dem vorigen Lemma folgt

$$\begin{aligned}
& \left(\operatorname{sinc}\left(\frac{t}{h}-j\right), \operatorname{sinc}\left(\frac{t}{h}-k\right)\right)_{L_2(\mathbb{R})} \\
= & \left(\frac{h}{\sqrt{2\pi}} e^{-ijh\omega} \chi_{[-\frac{\pi}{h},+\frac{\pi}{h}]}(\omega), \frac{h}{\sqrt{2\pi}} e^{+ikh\omega} \chi_{[-\frac{\pi}{h},+\frac{\pi}{h}]}(\omega)\right)_{L_2(\mathbb{R})} \\
= & \frac{h^2}{2\pi} \int_{-\pi/h}^{\pi/h} e^{+i(k-j)h\omega} d\omega \\
= & h\delta_{jk}.
\end{aligned}$$

□

5.4 Bandbreitenbeschränkte Funktionen

Wir wollen auch noch ausrechnen, was herauskommt, wenn wir eine beliebige L_2–Funktion u gegen eine skalierte und verschobene sinc–Funktion integrieren:

$$\begin{aligned}
& \left(u(t), \operatorname{sinc}\left(\frac{t-x}{h}\right)\right)_{L_2(\mathbb{R})} \\
= & \left(\hat{u}(\omega), \operatorname{sinc}\left(\frac{\cdot - x}{h}\right)^{\vee}(\omega)\right)_{L_2(\mathbb{R})} \\
= & \frac{h}{\sqrt{2\pi}} \int_{-\pi/h}^{\pi/h} \hat{u}(\omega) e^{+ix\omega} d\omega
\end{aligned}$$

Das wäre gleich $hu(x)$, wenn die Integrationsgrenzen nicht endlich wären. Aber wir können einen Raum von Funktionen betrachten, für den das klappt:

Definition 5.8. *Der Raum BLF_τ der* **bandbreitenbeschränkten Funktionen** *(bandlimited functions) mit* **Grenzfrequenz** *τ bestehe aus allen Funktionen u, die sich als inverse Fouriertransformierte*

$$u(x) := \frac{1}{\sqrt{2\pi}} \int_{-\tau}^{\tau} v(\omega) e^{ix\omega} d\omega$$

von Funktionen $v \in L_2[-\tau, \tau]$ schreiben lassen.

Solche Funktionen sind immer analytisch und liegen in $L_2(\mathbb{R})$. Ihre Fouriertransformierte verschwindet außerhalb des Intervalls $[-\tau, \tau]$.

Lemma 5.9. *Für Funktionen u aus $BLF_{\pi/h}$ und alle $x \in \mathbb{R}$ gilt die* **Reproduktionsgleichung**

$$u(x) = \left(u, \frac{1}{h}\operatorname{sinc}\left(\frac{\cdot - x}{h}\right)\right)_{L_2(\mathbb{R})}.$$

□

Obwohl wir das nicht adäquat vertiefen können, sollte bemerkt werden, daß $BLF_{\pi/h}$ unter dem $L_2(\mathbb{R})$-Skalarprodukt ein **Hilbertraum** mit positiv definitem **reproduzierendem Kern**

$$\Phi(t,x) := \frac{1}{h}\mathrm{sinc}\left(\frac{t-x}{h}\right)$$

ist, der obendrein die bemerkenswerte Gleichung

$$\Phi(x,y) = (\Phi(x,\cdot),\Phi(y,\cdot))_{L_2(\mathbb{R})}$$

erfüllt. Der Raum $BLF_{\pi/h}$ ist ferner auch ein abgeschlossener Unter-Hilbertraum von $L_2(\mathbb{R})$, denn mit dem **Abschneideoperator** (truncation operator)

$$\mathrm{Trunc}_\tau(u) := (\chi_{[-\tau,\tau]}\hat{u})^\vee$$

können wir beliebige Funktionen $u \in L_2(\mathbb{R})$ auf Funktionen aus BLF_τ abbilden, und das ist klar ein lineare und stetige Abbildung, sogar ein **Projektor**.

Damit erhalten wir für ganz allgemeine Funktionen $u \in L_2(\mathbb{R})$ und alle $x \in \mathbb{R}$ die Gleichung

$$\begin{aligned} & \left(u(t), \frac{1}{h}\mathrm{sinc}\left(\frac{t-x}{h}\right)\right)_{L_2(\mathbb{R})} \\ = \quad & \frac{1}{\sqrt{2\pi}}\int_{-\pi/h}^{\pi/h} \hat{u}(\omega)e^{+ix\omega}d\omega \\ = \quad & \mathrm{Trunc}_{\pi/h}(u)(x). \end{aligned}$$

5.5 Beste Approximation in L_2 mit sinc–Funktionen

Wir können jetzt auch ausrechnen, was die Orthogonalprojektion P_h von $L_2(\mathbb{R})$ auf den span der orthogonalen sinc–Funktionen

$$s_{k,h}(x) := \frac{1}{\sqrt{h}}\mathrm{sinc}\left(\frac{x-kh}{h}\right),\ k \in \mathbb{Z}$$

ist. Sie berechnet natürlich die beste $L_2(\mathbb{R})$–Approximation aus diesem span. Man hat

$$\begin{aligned}(P_h u)(x) &= \sum_{k\in\mathbb{Z}} \left(u, \frac{1}{\sqrt{h}}\mathrm{sinc}\left(\frac{\cdot - kh}{h}\right)\right)_{L_2(\mathbb{R})} \frac{1}{\sqrt{h}}\mathrm{sinc}\left(\frac{x-kh}{h}\right)\\ &= \frac{1}{h}\sum_{k\in\mathbb{Z}} \left(u, \mathrm{sinc}\left(\frac{\cdot - kh}{h}\right)\right)_{L_2(\mathbb{R})} \mathrm{sinc}\left(\frac{x-kh}{h}\right)\\ &= \sum_{k\in\mathbb{Z}} \mathrm{Trunc}_{\pi/h}(u)(kh)\mathrm{sinc}\left(\frac{x-kh}{h}\right)\\ &= S_{h,\,\mathrm{Trunc}_{\pi/h}(u)}(x)\end{aligned}$$

und es gilt notwendig die Parceval'sche Gleichung für Orthogonalentwicklungen in der Form

$$\|P_h u\|^2_{L_2(\mathbb{R})} = h\sum_{k\in\mathbb{Z}} \Big(\ \mathrm{Trunc}_{\pi/h}(u)(kh)\Big)^2$$

für alle $u \in L_2(\mathbb{R})$. Setzt man hier Funktionen $u \in BLF_{\pi/h}$ ein, so folgt auch

$$\|P_h u\|^2_{L_2(\mathbb{R})} = h\sum_{k\in\mathbb{Z}} u(kh)^2$$

und

$$P_h(u)(x) = \sum_{k\in\mathbb{Z}} u(kh)\mathrm{sinc}\left(\frac{x-kh}{h}\right) = S_{h,u}(x).$$

Theorem 5.10. *Der Shannon-Operator, wenn man ihn auf $BLF_{\pi/h}$ einschränkt, ist der Projektor der besten Approximation auf $BLF_{\pi/h}$ auf den span der orthonormalen sinc–Funktionen $s_{k,h}$ für $k \in \mathbb{Z}$. Die beste Approximation zu einem $u \in L_2(\mathbb{R})$ ist die Shannon–Reihe zu $Trunc_{\pi/h}(u)$. Die Konvergenz der Reihe des Projektionsoperators $P_h(u)$ findet im Sinne der L_2–Norm statt, und die Quadrate der Beträge der Koeffizienten sind summierbar.* □

5.6 Sampling Theorem

Aber das alles reicht nicht aus, um das berühmte **Sampling Theorem**[35] zu beweisen, das nicht nur Shannon[36] sondern auch Whittaker[37], Kotelnikov[38] oder Nyquist zugeschrieben wird.

[35] http://en.wikipedia.org/wiki/Nyquist%E2%80%93Shannon_sampling_theorem
[36] http://www-history.mcs.st-andrews.ac.uk/Biographies/Shannon.html
[37] http://www-history.mcs.st-andrews.ac.uk/Biographies/Whittaker_John.html
[38] http://www-history.mcs.st-andrews.ac.uk/Biographies/Kotelnikov.html

Theorem 5.11. *Alle Funktionen $u \in BLF_{\pi/h}$ sind durch ihre Shannon-Reihe im L_2–Sinne exakt reproduzierbar, d.h. es gilt*

$$u(x) = \sum_{k\in\mathbb{Z}} u(kh)\mathrm{sinc}\left(\frac{x-kh}{h}\right) = S_{h,u}(x)$$

für alle Funktionen $u \in BLF_{\pi/h}$.

Was fehlt, ist daß die orthogonalen sinc–Funktionen $s_{k,h}$ in $BLF_{\pi/h}$ **vollständig** sind, d.h. $u = P_h u$ für alle $u \in BLF_{\pi/h}$ gilt. Insbesondere muß man ausschließen können, daß es eine nichtverschwindende Funktion $u \in BLF_{\pi/h}$ gibt, deren Werte $u(kh)$ für $k \in \mathbb{Z}$ alle Null sind.

Dazu brauchen wir ein Hilfsmittel:

Theorem 5.12. *(Allgemeine* **Poisson'sche Summenformel***)* [39]
Es gilt

$$\frac{1}{\sqrt{2\pi}} \sum_{k\in\mathbb{Z}} \hat{u}(k)e^{ikx} = \sum_{j\in\mathbb{Z}} u(x+2\pi j)$$

im L_2-Sinne, sofern u in L_1 ist und die 2π-periodische rechte Seite auf $[0, 2\pi]$ gleichmäßig konvergiert und in $L_2[0, 2\pi]$ liegt.

Die Formel gilt auch unter anderen Voraussetzungen, und gegebenenfalls auch in einem stärkeren Sinne. Die Standardform ist die für $x = 0$, d.h.

$$\sum_{k\in\mathbb{Z}} \hat{u}(k) = \sqrt{2\pi} \sum_{j\in\mathbb{Z}} u(2\pi j),$$

die aber mit Vorsicht zu genießen ist, weil sie punktweise und nicht im L_2-Sinne gemeint ist. Unter den obigen schwachen Voraussetzungen ist nur klar, daß

$$\sum_{k\in\mathbb{Z}} |\hat{u}(k)|^2 < \infty$$

gilt. Man sieht an der Standardform, daß man auf einer Seite über das Gitter $\mathbb{Z}$, auf der anderen Seite über das Gitter $2\pi\mathbb{Z}$ summiert. Die Kristallographen reden vom *reziproken* Gitter im Fourierraum, wenn sie Beugung von Röntgenstrahlen am Kristallgitter untersuchen, um aus den Beugungsbildern auf das Gitter zu schließen.

Wir geben hier eine Beweisskizze an. Die rechte Seite ist unter den gegebenen Voraussetzungen in eine Fourierreihe entwickelbar, und im L_2–Sinne gilt

$$g(x) := \sum_{j\in\mathbb{Z}} u(x+2\pi j) = \sum_{k\in\mathbb{Z}} c_k e^{ikx}$$

[39] `http://www-history.mcs.st-andrews.ac.uk/Biographies/Poisson.html`

mit Fourierkoeffizienten

$$\begin{aligned} c_k &= \frac{1}{2\pi}\int_{-\pi}^{\pi} g(t)e^{-itk}dt \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_{j\in\mathbb{Z}} u(t+2\pi j)e^{-itk}dt \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi}\sum_{j\in\mathbb{Z}} u(t+2\pi j)e^{-i(t+2\pi j)k}dt \\ &= \frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} u(t)e^{-itk}dt \\ &= \frac{1}{\sqrt{2\pi}}\hat{u}(k). \quad \square \end{aligned}$$

Wir geben noch eine Variante an. Dazu setzen wir $v := \hat{u}(\cdot/h)$ und vertauschen in Satz 5.12 die Voraussetzungen bzgl. u und $\hat{u}$. Es folgt

$$\hat{v}(\omega) = h\hat{\hat{u}}(h\omega) = hu(-h\omega)$$

und

$$\begin{aligned} \frac{1}{\sqrt{2\pi}}\sum_{k\in\mathbb{Z}}\hat{v}(k)e^{ikx} &= \sum_{j\in\mathbb{Z}} v(x+2\pi j) \\ \frac{h}{\sqrt{2\pi}}\sum_{k\in\mathbb{Z}} u(-kh)e^{ikx} &= \sum_{j\in\mathbb{Z}}\hat{u}\left(\frac{x+2\pi j}{h}\right) \\ \frac{h}{\sqrt{2\pi}}\sum_{k\in\mathbb{Z}} u(kh)e^{-ikhy} &= \sum_{j\in\mathbb{Z}}\hat{u}\left(y+\frac{2\pi j}{h}\right). \end{aligned}$$

Das gilt ebenfalls im L_2-Sinne, und zwar wenn $\hat{u}$ in L_1 ist und die $2\pi/h$-periodische rechte Seite auf $[0,2\pi/h]^d$ gleichmäßig konvergiert und in $L_2[0,2\pi/h]^d$ liegt. Man sieht an dieser Form, daß die linke Seite über ein h–Gitter summiert, während rechts über das reziproke $2\pi/h$–Gitter summiert wird. Für h gegen Null oder Unendlich wird das eine Gitter feiner, wenn das andere gröber wird.

Um das Sampling Theorem zu beweisen, nehmen wir ein $u \in BLF_{\pi/h}$ her

und zeigen, daß $\hat{u}$ und $\hat{S}_{h,u}$ in L_2 gleich sind. Also

$$\begin{aligned}
\hat{S}_{h,u}(\omega) &= \left(\sum_{k\in\mathbb{Z}} u(kh)\mathrm{sinc}\left(\frac{x-kh}{h}\right)\right)^{\wedge}(\omega) \\
&= \sum_{k\in\mathbb{Z}} u(kh)\mathrm{sinc}\left(\frac{x-kh}{h}\right)^{\wedge}(\omega) \\
&= \frac{h}{\sqrt{2\pi}}\chi_{-\pi/h,\pi/h}(\omega)\sum_{k\in\mathbb{Z}} u(kh)e^{-ikh\omega} \\
&= \frac{\sqrt{h}}{\sqrt{2\pi}}\chi_{-\pi/h,\pi/h}(\omega)\frac{\sqrt{2\pi}}{\sqrt{h}}\sum_{j\in\mathbb{Z}^d}\hat{u}\left(\omega+\frac{2\pi j}{h}\right) \\
&= \hat{u}(\omega),
\end{aligned}$$

wobei wir die Poisson'sche Summenformel in der zuletzt genannten Form benutzt haben. Die erforderlichen Voraussetzungen für die obige Schlußweise sind gegeben, sofern man ein $u \in BLF_{\pi/h}$ verwendet, aber das wollen wir nicht im Detail nachrechnen. □

5.7 Berechnen von Shannon–Reihen

Das Ausrechnen einer Shannon–Reihe (5.5) ist nicht ganz unproblematisch. Wir stellen uns hier auf den Standpunkt, daß wir die Funktion f nur in einem Intervall $[-K,+K]$ zur Verfügung haben, und wir machen dort ein Sampling mit einer Schrittweite $h = K/n$, so daß wir $2n+1$ Punkte der Form $kh \in [-K,+K],\ -n \le k \le n$ mit Werten $f(kh)$ haben. Die Auswertung der Reihe

$$S_{h,f}(t) := \sum_{-n\le k\le n} f(kh)\mathrm{sinc}\left(\frac{t}{h}-k\right)$$

in einem festen Punkt t wird dann den Aufwand $\mathcal{O}(n) = \mathcal{O}(K/h)$ haben, und wenn man das in mehr als $2n+1$ Punkten macht, bekommt man den Aufwand $\mathcal{O}(n^2)$.

Um dies zu verbessern, sollte man die Auswertung als Faltung schreiben, die simultan in $\mathcal{O}(n)$ Punkten auswertet und dann den Aufwand $\mathcal{O}(n\log n)$ hat. Man kann das als **Upsampling** machen, indem man zwischen je zwei Samplingpunkten kh und $(k+1)h$ je $m-1$ neue Werte berechnet, die eine Schrittweite h/m haben. Die Auswertung als Faltung erfolgt dann für ein festes $i,\ 1 \le i \le m-1$ simultan in den $2n+1$ Punkten $jh+ih/m,\ -n \le j \le n$.

Mit einem festen i, $0 < i < m$ bildet man den Vektor

$$p_{j-k} = \operatorname{sinc}\left(\frac{jh + ih/m}{h} - k\right) = \operatorname{sinc}\left(j - k + \frac{i}{m}\right)$$

und bekommt zur Auswertung der Shannon–Reihe in den Punkten $jh+ih/m$ mit $-n \le j \le n$ die Faltung

$$(c * p)_j := \sum_{k=-n}^{n} c_k p_{j-k}.$$

Damit dies für $-n \le j \le n$ klappt, braucht man in p die Werte mit Indizes $-2n \le j - k \le 2n$. Die Faltung wird also zwischen einem $c \in \mathbb{R}^{2n+1}$ und einem $p \in \mathbb{R}^{4n+1}$ ausgeführt und hat dann nach MATLAB–Konvention automatisch $(2n+1)+(4n+1)-1 = 6n+1$ Ergebnisse, von denen wir aber nur die "mittleren" $2n+1$ brauchen. Man bekommt insgesamt $(m-1)*(2n+1)$ neue Werte durch Aufruf von $m-1$ Faltungsoperationen der Komplexität $\mathcal{O}(n \log n)$, und hat also den Gesamtaufwand $\mathcal{O}(m\, n \log n)$ für $\mathcal{O}(mn)$ Punkte. Hier ist ein passendes Programm, und die Ausgabe dazu sind die Abbildungen 19 und 20.

```
% testing Shannon sampling by convolution
clear all;
close all;
K=1;          % data in [-K,K]
n=7;
h=K/n;        % with this stepsize
xc=(-K:h:K)'; % points for data sampling
freq=2;       % frequency for the following
c=sin(freq*xc*2*pi);
m=15;     % divisor for upsampling at (Z + i/m)*h
ind=(-2*n:2*n)'; % the basic points for the p vector
hu=h*1/m; % stepsize for upsampling
y=zeros(m*(2*n+1),1); % the resulting vector
y(1:m:m*(2*n+1))=c;  % and the data
for i=1:m-1
    p=sinc(ind+i/m); % form p from sinc
    cv=conv(c,p);    % we convolve and store
    y(1+i:m:i+m*(2*n+1))=cv(2*n+1:4*n+1,1);
end
x=((-K):hu:(K))'; % this is where we want to plot
dx=sin(freq*x*2*pi);  % data there
```

```
dy=y(1:length(dx));   % result there
ny=length(dy)
nx=length(dx)
figure
plot(xc,c,'o',x,dy,x,dx);
legend('Data','Expansion','True function')
title('Evaluation of Shannon series')
figure
plot(x,dx-dy);
title('Error of Shannon series')
```

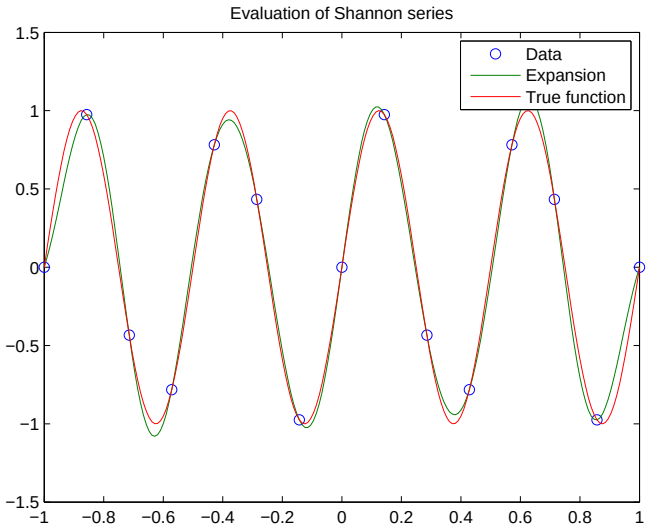

Figure 19: Reproduktion des Sinus durch die Shannonreihe

Für die schnelle Realisierung der Faltung durch die FFT gibt es eine Übungsaufgabe.

5.8 Fehlerabschätzung für sinc–Approximation

Aus dem Shannon-Theorem folgt eine ziemlich einfache, aber nützliche Fehlerabschätzung:

Theorem 5.13. *Die beste Approximation $P_h(u)$ einer beliebigen Funktion $u \in L_2(\mathbb{R}^d)$ durch orthonormale sinc–Funktionen $s_{k,h}$ hat den Fehler*

$$\|u - P_h(u)\|^2_{L_2(\mathbb{R})} = \|u - Trunc_{\pi/h}(u)\|^2_{L_2(\mathbb{R})} = \int_{|\omega| \geq \pi/h} |\hat{u}(\omega)|^2 d\omega.$$

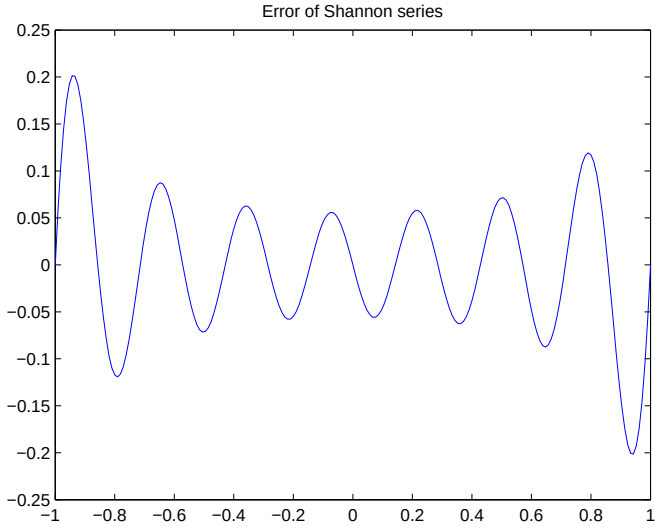

Figure 20: Fehlerfunktion dazu

Zum Beweis benutzen wir, daß nach dem Shannon-Theorem auch

$$Trunc_{\pi/h}(u) = P_h(Trunc_{\pi/h}(u)) = P_h(u)$$

gilt, und daraus folgt

$$u - P_h(u) = u - Trunc_{\pi/h}(u) + Trunc_{\pi/h}(u) - P_h(u) = u - Trunc_{\pi/h}(u).\square \tag{5.14}$$

Wie in der Approximationstheorie üblich, wollen wir das in Fehlerabschätzungen umsetzen, die etwas mit der Glätte der zu approximierenden Funktionen zu tun haben. Dazu

Definition 5.15. *Der Raum*

$$W_2^\tau(\mathbb{R}^d) := \{u \in L_2(\mathbb{R}^d) \ : \ \int_{\mathbb{R}^d} |\hat{u}(\omega)|^2(1+\|\omega\|^2)^\tau d\omega < \infty\}$$

heißt **Sobolevraum** *der Ordnung τ auf $\mathbb{R}^d$. Er ist ein Hilbertraum mit dem inneren Produkt*

$$(u,v)_{W_2^\tau(\mathbb{R}^d)} := \int_{\mathbb{R}^d} \hat{u}(\omega)\overline{\hat{v}(\omega)}(1+\|\omega\|^2)^\tau d\omega.$$

Man mache sich klar, daß die Funktionen $u \in W_2^\tau(\mathbb{R}^d)$ die Eigenschaft haben, daß alle Ableitungen bis zur Ordnung τ noch als $L_2(\mathbb{R})$-Funktionen existieren. Zwar kann man diese Räume auch für nicht–ganzzahlige τ definieren, aber das soll hier nicht vertieft werden.

Theorem 5.16. *Die beste Approximation $P_h(u)$ einer beliebigen Funktion $u \in W_2^\tau(\mathbb{R}^d)$ durch orthonormale sinc–Funktionen $s_{k,h}$ hat den Fehler*

$$\|u - P_h(u)\|_{L_2(\mathbb{R})} \leq \frac{h^\tau}{\pi^\tau} \|u\|_{W_2^\tau(\mathbb{R}^d)}.$$

Das beweist man durch Einsetzen in

$$\begin{aligned} & \int_{|\omega| \geq \pi/h} |\hat{u}(\omega)|^2 d\omega \\ = & \int_{|\omega| \geq \pi/h} |\hat{u}(\omega)|^2 \frac{(1+|\omega|^2)^\tau}{(1+|\omega|^2)^\tau} d\omega \\ \leq & \left(\frac{h}{\pi}\right)^{2\tau} \int_{\mathbb{R}} |\hat{u}(\omega)|^2 (1+|\omega|^2)^\tau d\omega \\ = & \left(\frac{h}{\pi}\right)^{2\tau} \|u\|^2_{W_2^\tau(\mathbb{R}^d)} \square \end{aligned}$$

Aber das ist wegen (5.14) auch genau der Abschneidefehler, der durch den Operator $\mathrm{Trunc}_{\pi/h}$ entsteht, denn danach findet ein fehlerfreies Shannon-Sampling von $\mathrm{Trunc}_{\pi/h}(u)$ statt.

Das wird in der Technik auch genau so realisiert. Ein gegebenes Signal u wird

1. durch ein Tiefpaßfilter[40] bandbreitenbeschränkt, d.h. die hohen Frequenzen werden abgeschnitten, d.h. die Abbildung Trunc_ω wird mit geeignetem ω angewendet.

2. Dann wird mit der Schrittweite h ein sampling durchgeführt.

Gilt dann

$$\frac{\pi}{h} \geq \omega, \text{ d.h. } h \leq \frac{\pi}{\omega},$$

so wird das tiefpaßgefilterte Signal (nicht aber das Originalsignal) exakt reproduzierbar, und der Gesamtfehler ist gleich dem Abschneidefehler. Die Nachrichtentechniker verwenden statt ω immer eine "Abschneidefrequenz" F mit $2\pi F = \omega$ und eine "Abtastfrequenz"[41] f mit $f = 1/h$. Dann hat man

$$f \geq 2F$$

[40] `http://de.wikipedia.org/wiki/Tiefpass`
[41] `http://de.wikipedia.org/wiki/Abtastrate`

zu fordern, d.h. die Abtastfrequenz muß das Doppelte der Abschneidefrequenz sein. Die halbe Abtastfrequenz wird auch **Nyquist**–Frequenz genannt. Sie muss dann größer als die Abschneidefrequenz sein, wenn man keinen sampling-Fehler haben will.

5.9 Aliasing

Es ist wichtig, hohe Frequenzen abzuschneiden, **bevor** man ein Sampling macht. Man bekommt sonst Artefakte, die man als **Aliasing** bezeichnet. Statt des vorgegebenen Signals sieht oder hört man eines mit anderer Frequenz, ein "Alias"–Signal.

Wir analysieren, was passiert, wenn man ein hochfrequentes Signal $f(x) = g(x)\cos(\Omega x)$ mit einer hohen Frequenz Ω und einem "netten" und bei $\pm\infty$ abklingenden konvergenzerzeugenden Faktor g einem sampling in $h\mathbb{Z}$ unterwirft. Man nimmt die Werte

$$\begin{aligned} g(hj)\cos(\Omega hj) &= g(hj)\cos(\Omega hj - 2\pi kj) \\ &= g(hj)\cos((\Omega - \tfrac{2\pi k}{h})hj) \end{aligned}$$

und das sind Werte von unendlich vielen möglichen Signalen mit Aliasfrequenzen $\Omega - \frac{2\pi k}{h}$, ebenfalls abgetastet an den Stellen jh. Die arme Shannon–Reihe kann gar nicht wissen, welches der Signale gemeint ist, und sie wird sicher keine Frequenzen oberhalb π/h liefern, sondern im Falle $(2k-1)\pi/h < \Omega < (2k+1)\pi/h$ nur die Aliasfrequenz $\Omega - \frac{2\pi k}{h} \in (-\pi/h, \pi/h)$.[42]

5.10 Direktes Shannon Sampling

Wenn man von einer gegebenen Funktion $u \in L_2(\mathbb{R})$ ausreichend viel voraussetzt, kann man durchaus die Shannon-Reihe $S_{h,u}$ bilden, ohne vorher eine Abschneideoperation auszuführen. Ab hier setzen wir deshalb noch voraus, dass u und $\hat{u}$ bei Unendlich mindestens quadratisch abklingen, d.h. es gilt

$$|u(t)| \leq C|t|^{-2} \text{ für alle } |t| > K$$

mit positiven Konstanten C und K, und analog für die Fourier-Transformierte. Wir untersuchen jetzt die Shannon-Reihe zu u, nicht die zu $\mathrm{Trunc}_{\pi/h}(u)$. Und wir untersuchen die Konvergenz des Fehlers $u(t) - S_{h,u}(t)$ für $h \to 0$. Das quadratische Abklingen garantiert zunächst, daß sowohl u als auch $\hat{u}$ in L_1 liegen und dann folgt, daß sowohl $\hat{u}$ als auch u in L_∞ liegen, weil man die Fourier-Transformation anwenden kann. Aber aus dem Abklingen folgt

[42] `http://www.dsptutor.freeuk.com/aliasing/AD102.html`

auch, daß die Shannon-Reihe punktweise absolut konvergent ist. Das beweist man mit

$$\begin{aligned} |S_{h,u}(t)| &= \left| \sum_{k\in\mathbb{Z}} u(kh)\mathrm{sinc}\left(\frac{t}{h}-k\right)\right| \\ &\leq \sum_{k\in\mathbb{Z}} |u(kh)| \\ &\leq \frac{C}{h^2}\sum_{k>0} k^{-2} + const. \\ &\leq \frac{C\pi^2}{6h^2} + const.. \end{aligned}$$

Wir stellen mit Lemma 5.6 die Shannon-Reihe zu u neu dar als

$$\begin{aligned} S_{h,u}(t) &= \sum_{j\in\mathbb{Z}} u(jh)\mathrm{sinc}\left(\frac{t}{h}-j\right) \\ &= \sum_{j\in\mathbb{Z}} u(jh)\frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega}e^{-ijh\omega}d\omega \\ &= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega}\underbrace{\sum_{j\in\mathbb{Z}} u(jh)e^{-ijh\omega}}_{=:g(-h\omega)}d\omega \end{aligned}$$

wobei wir die Summe mit dem Integral vertauschen können, weil wir quadratisches Abklingen von u vorausgesetzt haben. Die innere Summe

$$g(\eta) := \sum_{j\in\mathbb{Z}} u(jh)e^{ij\eta}$$

sehen wir uns näher an. Sie ist 2π-periodisch und hat die komplexen Fourierkoeffizienten $u(jh)$.

In unserer Situation können wir die Poisson'sche Summenformel anwenden mit $\hat{v}(\omega) = u(h\omega)$, also

$$v(t) = u(h\omega)^\vee(t) = u(h\omega)^\wedge(-t) = \frac{1}{h}\hat{u}(-\omega/h).$$

Wir bekommen, wenn $\hat{u}$ hinreichend nett ist, die Beziehung

$$\begin{aligned} g(\eta) &= \sum_{j\in\mathbb{Z}} u(jh)e^{ij\eta} \\ &= \frac{\sqrt{2\pi}}{h}\sum_{j\in\mathbb{Z}} \hat{u}\left(\frac{-\eta-2\pi j}{h}\right) \end{aligned}$$

und weiter

$$\begin{aligned}
S_{h,u}(t) &= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega} g(-h\omega)d\omega \\
&= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega}\frac{\sqrt{2\pi}}{h}\sum_{j\in\mathbb{Z}}\hat{u}\left(\frac{h\omega-2\pi j}{h}\right)d\omega \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\pi/h}^{\pi/h} e^{it\omega}\sum_{j\in\mathbb{Z}}\hat{u}\left(\omega-\frac{2\pi j}{h}\right)d\omega \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}}\int_{-\pi/h-2\pi j/h}^{\pi/h-2\pi j/h} e^{it\left(\eta+\frac{2\pi j}{h}\right)}\hat{u}\left(\eta\right)d\eta \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}} e^{\frac{2\pi itj}{h}}\int_{-\pi/h-2\pi j/h}^{\pi/h-2\pi j/h} e^{it\eta}\hat{u}\left(\eta\right)d\eta \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}} e^{-\frac{2\pi itj}{h}}\int_{\frac{(2j-1)\pi}{h}}^{\frac{(2j+1)\pi}{h}} e^{it\eta}\hat{u}\left(\eta\right)d\eta.
\end{aligned}$$

Zusammen mit der Fouriertransformationsgleichung für u folgt

$$\begin{aligned}
u(t) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\hat{u}(\omega)e^{+i\omega t}d\omega \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}}\int_{\frac{(2j-1)\pi}{h}}^{\frac{(2j+1)\pi}{h}} e^{it\eta}\hat{u}\left(\eta\right)d\eta \\
u(t)-S_{h,u}(t) &= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}}\left(1-e^{-\frac{2\pi itj}{h}}\right)\int_{\frac{(2j-1)\pi}{h}}^{\frac{(2j+1)\pi}{h}} e^{it\eta}\hat{u}\left(\eta\right)d\eta.
\end{aligned}$$

Theorem 5.17. *Die obige Gleichung gilt bei mindestens quadratischem Abklingen von u und $\hat{u}$ bei Unendlich, und wenn zusätzlich noch die periodische Funktion $\sum_{j\in\mathbb{Z}}\hat{u}\left(\frac{\eta-2\pi j}{h}\right)$ in L_2 liegt und gleichmässig konvergiert. Ferner hat man dann die vereinfachte Fehlerabschätzung*

$$|u(t)-S_{h,u}(t)| \leq \frac{\sqrt{2}}{\sqrt{\pi}}\int_{|\eta|\geq\pi/h}|\hat{u}\left(\eta\right)|d\eta.$$

Der obige Satz gilt auch allgemeiner, weil man die Gleichung umschreiben kann zu

$$\begin{aligned}
u(t)-S_{h,u}(t) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\hat{u}\left(\eta\right)e^{it\eta}\sum_{j\in\mathbb{Z}}\left(1-e^{-\frac{2\pi itj}{h}}\right)\chi_{[\frac{(2j-1)\pi}{h},\frac{(2j+1)\pi}{h}]}(\eta)d\eta \\
&= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\hat{u}\left(\eta\right)e^{it\eta}\underbrace{\sum_{j\in\mathbb{Z}\setminus\{0\}}\left(1-e^{-\frac{2\pi itj}{h}}\right)\chi_{[-1/2,+1/2]}\left(\frac{h\eta}{2\pi}-j\right)}_{=:K(h\eta/2\pi,2\pi t/h)}d\eta
\end{aligned}$$

und die Funktion K gleichmässig beschränkt, bis auf ihre Sprungstellen beliebig oft differenzierbar, und lokal in $L_2 \cap L_1$ ist. Es gilt

$$\begin{aligned} K(\eta,t) &= \sum_{j\in\mathbb{Z}} \left(1-e^{-itj}\right) \chi_{[-1/2,+1/2]}\left(\eta-j\right) \\ &= 1-e^{-it\cdot round(\eta)} \end{aligned}$$

fast überall. Deshalb kommt man mit den Abklingvoraussetzungen aus.

Theorem 5.18. *Wenn man die obige Situation auf Funktionen aus dem Sobolevraum $W_2^\tau(\mathbb{R}^d)$ einschränkt, bekommt man ein Konvergenzverhalten wie h^τ für $h \to 0$.*

Das folgt mit der oben schon verwenden Technik zur Abschätzung des Abschneidefehlers. □

6 Translationsinvariante Räume

Dieser Abschnitt lehnt sich stark an den Artikel [2] von Kurt Jetter und Gerlind Plonka an. Wir wollen das Sampling generalisieren auf das Studium von Reihen

$$\sum_{k\in\mathbb{Z}} c_k \varphi\left(\frac{x-kh}{h}\right) \tag{6.1}$$

die durch Verschieben eines einzigen "Generators" φ auf einem Gitter $h\mathbb{Z}$ der Schrittweite $h>0$ erzeugt werden. Dabei beginnen wir mit $h=1$ und verschieben die Skalierung auf später. Es geht also um Reihen

$$\varphi_c(x) := \sum_{k\in\mathbb{Z}} c_k \varphi(x-k)$$

und ihre Approximationen an gegebene Funktionen f auf $\mathbb{R}$. Schon die Summierbarkeit so einer Reihe ist ein Problem, und deshalb müssen wir vorsichtig anfangen. Das Fernziel wird sein, Funktionen durch solche Reihen zu approximieren und den Approximationsfehler abzuschätzen. Gleichzeitig ist das eine gute Vorbereitung auf die wavelets in den Folgekapiteln.

Es sollte aber vorab noch geklärt werden, warum man diese Verallgemeinerung überhaupt braucht. Dazu vergleichen wir die Reihen (5.1) für die kardinalen Funktionen

$$\begin{aligned} \varphi(x) &:= \operatorname{sinc}(x) \\ \varphi(x) &:= \begin{cases} 1-|x| & |x|\le 1 \\ 0 & \text{sonst} \end{cases} \end{aligned} \tag{6.2}$$

und sehen, daß man bei der sinc–Reihe sehr viele Terme braucht, um eine brauchbare Approximation an einer festen Stelle x zu erreichen, während für die Hutfunktion nur zwei Werte zu berücksichtigen sind. Deshalb ist die Reihe für die Hutfunktion sehr viel schneller auswertbar als die Shannon–Reihe. Aber der Fehler verhält sich anders: wir wissen nach den Sätzen 5.16 und 5.18, daß die Shannon–Reihe beliebig gute Konvergenzordnungen erreichen kann, während wir aus der Numerik wissen, daß stückweise linear Interpolation nur bestenfalls $\mathcal{O}(h^2)$ bekommen kann. Wir brauchen also ein "dazwischenliegendes" Verfahren, das eine schnell abklingende Funktion φ benutzt, aber dennoch hohe Konvergenzordnungen möglich macht.

6.1 Grundlagen

Wir verallgemeinern hier, was wir über das Shannon-Sampling gelernt haben. Statt einer kardinalen Funktion wie in (6.2) betrachten wir allgemeine "Generatoren" $\varphi \in L_2(\mathbb{R})$ und den in $L_2(\mathbb{R})$ genommenen Abschluß des spans ihrer Translate:

$$S_\varphi := \overline{span\{\varphi(\cdot - k) \ : \ k \in \mathbb{Z}\}} \tag{6.3}$$

Definition 6.4. *Der Raum S_φ aus (6.3) ist der von φ erzeugte* **principal shift–invariant space** *(PSI).*

Dieser Raum ist wohldefiniert. Das Fernziel ist, ihn genau zu charakterisieren, den Projektor P_φ von $L_2(\mathbb{R})$ auf S_φ auszurechnen und dessen Fehlerverhalten zu studieren. Vorab würden wir aber gerne wissen, welche Bedingungen an einen infiniten Koeffizientenvektor $c = \{c_k\}_{k\in\mathbb{Z}}$ man stellen muß, um sicherzustellen, daß die über die **diskrete Faltung** $*$ definierte Funktion

$$\varphi_c(x) := (c * \varphi)(x) := \sum_{k\in\mathbb{Z}} c_k \varphi(x-k)$$

punktweise auswertbar ist bzw. noch in $L_2(\mathbb{R})$ und damit in S_φ liegt. Das ist unter verschiedenen Voraussetzungen machbar, die wir hier teilweise aufzählen, die sich aber nicht gegenseitig ausschließen. Eine wichtige Bedingung wird sein, daß der biinfinite Koeffizientenvektor im klassischen Hilbertraum

$$\ell_2 := \left\{ \{c_k\}_{k\in\mathbb{Z}} \in \mathbb{C}^{\mathbb{Z}} \ : \ \sum_{k\in\mathbb{Z}} |c_k|^2 < \infty \right\}$$

liegt. Aber wir fangen etwas einfacher an.

Situation 1: Für endliche Koeffizientenvektoren c liegt φ_c immer in $L_2(\mathbb{R})$ und damit in S_φ. Ist φ punktweise auswertbar, so auch φ_c.

Situation 2: Die Funktion φ habe kompakten Träger in $[-K, K]$, d.h. $\varphi(x) = 0$ für alle $|x| > K$. Dann kommen nur die k mit

$$x - K \leq k \leq x + K$$

in der Summe für festes x vor. Somit ist zumindestens für stetige φ die Summe finit auswertbar, und sie liegt in $L_2[a, b]$ auf allen endlichen Intervallen $[a, b]$.

Wir sehen uns jetzt die **globale** L_2–Norm von φ_c an und bekommen

$$\begin{aligned}
\|\varphi_c\|_2^2 &= \textstyle\int_{\mathbb{R}} \varphi_c^2(x)dx \\
&= \int_{\mathbb{R}} \left(\sum_{k\in\mathbb{Z}} c_k \varphi(x-k) \right)^2 dx \\
&= \sum_{j\in\mathbb{Z}} \int_j^{j+1} \left(\sum_{k\in\mathbb{Z}} c_k \varphi(x-k) \right)^2 dx \\
&= \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{k\in\mathbb{Z}} c_k \varphi(x-j-k) \right)^2 dx \\
&= \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{m\in\mathbb{Z}} c_{m-j} \varphi(x-m) \right)^2 dx.
\end{aligned}$$

Die inneren Indices m können mindestens auf $-K \leq m \leq K+1$ eingeschränkt werden, weil das Integral über $\varphi(x-m)$ verschwindet, sofern $-m \geq K$ oder $-m+1 \leq -K$ gilt. Deshalb

$$\|\varphi_c\|_2^2 = \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{m=-K}^{K+1} c_{m-j} \varphi(x-m) \right)^2 dx.$$

Im inneren Teil kann nun die Cauchy–Schwarz–Ungleichung angewendet werden:

$$\begin{aligned}
\|\varphi_c\|_2^2 &\leq \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{m=-K}^{K+1} c_{m-j}^2 \right) \left(\sum_{n=-K}^{K+1} \varphi(x-n)^2 \right) dx \\
&= \sum_{j\in\mathbb{Z}} \left(\sum_{m=-K}^{K+1} c_{m-j}^2 \right) \int_0^1 \sum_{n=-K}^{K+1} \varphi(x-n)^2 dx \\
&\leq (2K+2) \left(\sum_{j\in\mathbb{Z}} c_j^2 \right) \int_0^1 \sum_{n=-K}^{K+1} \varphi(x-n)^2 dx \\
&\leq (2K+2) \|c\|_{\ell_2}^2 \|\varphi\|_2^2
\end{aligned}$$

weil beim Summieren jedes der c_j^2 maximal $(2K+2)$–mal vorkommen kann.

Theorem 6.5. *Im Falle $\varphi \in L_2(\mathbb{R})$ mit kompaktem Träger und $c \in \ell_2$ gilt $\varphi_c := c * \varphi \in L_2(\mathbb{R})$.*

Wir rechnen für den allgemeineren Fall die Fouriertransformierte formal aus

$$\hat{\varphi}_c(\omega) = \hat{\varphi}(\omega) \sum_{k \in \mathbb{Z}} c_k e^{-ik\omega} =: \hat{\varphi}(\omega)\sigma_c(\omega)$$

und bekommen eine 2π–periodische Funktion σ_c, die man manchmal auch als **Symbol** von c bezeichnet, obwohl sie nichts anderes als die vom Koeffizientenvektor $c \in \ell_2$ erzeugte 2π–periodische Funktion aus L_2 ist. Deren Fourierkoeffizienten sind die c_k, denn sie ist so definiert, und es folgt wegen der Parcevalschen Gleichung auch

$$\|c\|_{\ell_2} = \|\sigma_c\|_{L_{2,2\pi}}.$$

Daran kann man ablesen, daß unter der Voraussetzung $c \in \ell_2$ die 2π–periodische Funktion σ_c noch in $L_{2,2\pi}$ liegt. Es folgt:

Situation 3:

Theorem 6.6. *Gilt $c \in \ell_2$ und ist σ_c eine beschränkte 2π–periodische Funktion, so gilt $\varphi_c \in L_2(\mathbb{R})$.*

Aber man kann auch folgendermaßen weiterarbeiten:

$$\begin{aligned} \|\varphi_c\|_2^2 &= \int_{\mathbb{R}} |\hat{\varphi}(\omega)|^2 |\sigma_c(\omega)|^2 d\omega \\ &= \sum_{j \in \mathbb{Z}} \int_{-\pi}^{\pi} |\hat{\varphi}(\omega + 2\pi j)|^2 |\sigma_c(\omega + 2\pi j)|^2 d\omega \\ &= \int_{-\pi}^{\pi} |\sigma_c(\omega)|^2 \sum_{j \in \mathbb{Z}} |\hat{\varphi}(\omega + 2\pi j)|^2 d\omega \\ &=: \int_{-\pi}^{\pi} |\sigma_c(\omega)|^2 [\varphi, \varphi](\omega) d\omega \end{aligned} \tag{6.7}$$

mit dem wichtigen **Klammerprodukt**

$$[\varphi, \psi](\omega) := \sum_{j \in \mathbb{Z}} \hat{\varphi}(\omega + 2\pi j) \overline{\hat{\psi}(\omega + 2\pi j)},$$

das, wenn es existiert, eine 2π–periodische Funktion ist.

Situation 4:

Theorem 6.8. *Gilt $c \in \ell_2$ und ist das Klammerprodukt $[\varphi,\varphi](\omega)$ punktweise existent, meßbar und gleichmäßig beschränkt, so gilt $\varphi_c \in L_2(\mathbb{R})$.*

Es sieht zwar nach Spielerei aus, aber wir wollen mal die Fourierkoeffizienten von $[\varphi,\psi]$ ausrechnen:

$$\begin{aligned}
& \int_{-\pi}^{\pi} [\varphi,\psi](\omega) e^{-ik\omega} d\omega \\
= & \int_{-\pi}^{\pi} \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\overline{\hat{\psi}(\omega+2\pi j)} e^{-ik(\omega+2\pi j)} d\omega \\
= & \int_{\mathbb{R}} \hat{\varphi}(\omega)\overline{\hat{\psi}(\omega)} e^{-ik\omega} d\omega \\
= & \int_{\mathbb{R}} \varphi(x-k)\overline{\psi(x)} dx \\
= & \int_{\mathbb{R}} \hat{\varphi}(\omega)\overline{\hat{\psi}(\omega) e^{ik\omega}} d\omega \\
= & \int_{\mathbb{R}} \varphi(x)\overline{\psi(x+k)} dx.
\end{aligned}$$

Rückwärts gerechnet folgt daraus, daß alle Fourierkoeffizienten des Klammerprodukts $[\varphi,\psi]$ immer berechenbar sind, wenn ψ und φ in $L_2(\mathbb{R})$ liegen. Wir machen neben

$$(\phi,\psi)_{L_2} = \int_{-\pi}^{\pi} [\varphi,\psi](\omega) d\omega$$

ein paar einfache Beobachtungen:

Theorem 6.9. *Die Translate einer Funktion $\varphi \in L_2(\mathbb{R}^d)$ sind orthogonal, wenn $[\varphi,\varphi]$ in L_2 liegt und konstant ist. Sie sind orthonormal, wenn $[\varphi,\varphi]$ konstant gleich $1/2\pi$ ist.*

Theorem 6.10. *Haben φ und ψ kompakten Träger, so ist das Klammerprodukt ein trigonometrisches Polynom.*

Theorem 6.11. *Sind f und φ beide in $L_2(\mathbb{R})$ und liegt das Klammerprodukt $[f,\varphi]$ in $L_{2,2\pi}$, so ist f orthogonal zu S_φ genau dann, wenn das Klammerprodukt verschwindet.*

Das wirft die Frage auf, wann das Klammerprodukt eine L_2–Funktion ist. Sicher dann wenn die Folge der Fourierkoeffizienten in ℓ_2 liegt. Und man kann zeigen, daß das bei geeigneten Abklingbedingungen and ψ und φ zutrifft. Da wir aber auch wissen, daß die Translate der sinc–Funktion orthonormal sind, kann es also auch sehr schlecht abklingende φ geben, die orthogonale Translate haben bzw. deren Klammerprodukt noch in L_2 liegt.

Situation 5: Für die L_2–Funktion φ gelte, daß das Klammerprodukt $[\varphi,\varphi]$ in $L_{2,2\pi}$ liegt.

Wir wollen untersuchen, wann man ein c finden kann, so daß die Translate von $\psi := \varphi_c$ orthonormal sind. Wir haben folgendes zu erfüllen:

$$\begin{aligned} 1/2\pi &= [\varphi_c, \varphi_c](\omega) \\ &= \sum_{j\in\mathbb{Z}} \hat{\varphi}_c(\omega+2\pi j)\overline{\hat{\varphi}_c(\omega+2\pi j)} \\ &= \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\sigma_c(\omega+2\pi j)\overline{\hat{\varphi}(\omega+2\pi j)\sigma_c(\omega+2\pi j)} \\ &= \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\sigma_c(\omega)\overline{\hat{\varphi}(\omega+2\pi j)\sigma_c(\omega)} \\ &= |\sigma_c(\omega)|^2 \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\overline{\hat{\varphi}(\omega+2\pi j)} \\ &= |\sigma_c(\omega)|^2 [\varphi,\varphi](\omega). \end{aligned} \tag{6.12}$$

Theorem 6.13. *Erfüllt der Generator φ die Bedingung $0 < 1/[\varphi,\varphi] \in L_1$, so existiert eine Funktion $\psi := c * \varphi$ mit $c \in \ell_2$, so daß die Translate von ψ orthonormal sind.*

Klar, denn man nehme die Funktion $f(\omega) := 1/\sqrt{2\pi[\varphi,\varphi](\omega)} \in L_2$ her und wähle c als den biinfiniten Vektor ihrer Fourierkoeffizienten. Dann gilt die oben durchgerechnete Gleichung.

Situation 6: Man setzt oft voraus, daß das Klammerprodukt punktweise und als 2π–periodische L_2–Funktion existiert und zwischen zwei positive Schranken einschließbar ist:

$$0 < A^2 \le [\varphi,\varphi](\omega) \le B^2. \tag{6.14}$$

Diese Situation wird manchmal auch "stabil" genannt. Aus (6.7) bekommt man dann sofort

$$A^2\|c\|_{\ell_2}^2 = A^2\|\sigma_c\|_{L_2}^2 \le \|\varphi_c\|_{L_2}^2 = \int_{-\pi}^{\pi} |\sigma_c(\omega)|^2 [\varphi,\varphi](\omega) \le B^2\|\sigma_c\|_{L_2}^2 = B^2\|c\|_{\ell_2}^2$$

bzw. die "frame"–Relation

$$A\|c\|_{\ell_2} \le \|\varphi_c\|_{L_2} \le B\|c\|_{\ell_2},$$

die ausdrückt, daß die ℓ_2–Norm der Koeffizienten äquivalent ist zur L_2–Norm auf dem Teilraum von S_ϕ, der aus allen φ_c mit $c \in \ell_2$ erzeugt wird. Das wird uns bei wavelets wieder begegnen....

Theorem 6.15. *Es sei $\varphi \in L_2(\mathbb{R})$ ein Generator, so daß das Klammerprodukt $[\varphi,\varphi]$ in $L_{2,2\pi}$ liegt und der Stabilitätsabschätzung genügt. Dann hat der Raum S_φ die alternativen Schreibweisen*

$$\begin{aligned} \{f \in L_2(\mathbb{R}) &: \hat{f} = \tau\cdot\hat{\varphi},\ \tau \in L_{2,2\pi}\} =: S_1 \\ \{f \in L_2(\mathbb{R}) &: f = \varphi_c,\ c \in \ell_2\} =: S_2. \end{aligned}$$

Beweis: Beide Räume liegen in S_φ, wenn man die Definitionen zunächst auf endliche Folgen c und trigonometrische Polynome τ einschränkt. Mit (6.7) kann man dann aber auch im Falle von S_2 wie folgt abschätzen:

$$A^2\|c\|_{\ell_2}^2 \;\leq\; \|f\|_{L_2(\mathbb{R})}^2 \;=\; \|\varphi_c\|_{L_2(\mathbb{R})}^2 \;\leq\; B^2\|c\|_{\ell_2}^2.$$

Damit kann man zum Abschluß übergehen. Die Situation von S_1 ist analog wegen $\tau = \sigma_c$ für $f = \varphi_c$ und $\|c\|_{\ell_2} = \|\tau\|_{L_{2,2\pi}}$. □

6.2 Projektion

Wir wollen jetzt die L_2–Projektion von $L_2(\mathbb{R})$ auf S_φ ausrechnen, wie bei der Shannon-Situation. Der Projektor, nennen wir ihn P_φ, muss existieren, und im Falle eines orthogonalen Generators ist er auch klassisch ausrechenbar. Für jede L_2–Funktion f muss $f - P_\varphi f$ auf allen $\varphi(\cdot - k)$ senkrecht stehen, und wir nehmen nach Theorem 6.15 an, dass er über Koeffizienten $c_f \in \ell_2$ mit $P_\varphi f = c_f * \varphi$ parametrisierbar ist.

Es folgt

$$\begin{aligned}
0 &= (f - P_\varphi f, \varphi(\cdot - k))_{L_2(\mathbb{R})} \\
(f, \varphi(\cdot - k))_{L_2(\mathbb{R})} &= (c_f * \varphi, \varphi(\cdot - k))_{L_2(\mathbb{R})} \\
\int_{-\pi}^{\pi} [f, \varphi](\omega) e^{-ik\omega} d\omega &= \int_{-\pi}^{\pi} [c_f * \varphi, \varphi](\omega) e^{-ik\omega} d\omega \\
&= \int_{-\pi}^{\pi} \sum_{j\in\mathbb{Z}} \sigma_c(\omega)\hat{\varphi}(\omega + 2\pi j)\overline{\hat{\varphi}(\omega + 2\pi j)} e^{-ik\omega} d\omega \\
&= \int_{-\pi}^{\pi} \sigma_c(\omega)[\varphi, \varphi](\omega) e^{-ik\omega} d\omega
\end{aligned}$$

d.h. als Gleichung in $L_2(\mathbb{R})$

$$[f, \varphi](\omega) = \sigma_c(\omega)[\varphi, \varphi](\omega)$$

weil die Fourierkoeffizienten gleich sind. Also ist der Projektor so definiert, daß man die Fourierkoeffizienten c_k von

$$\frac{[f, \varphi](\omega)}{[\varphi, \varphi](\omega)}$$

ausrechnen muß. Mit anderen Worten:

$$P_\varphi f = \sum_{k\in\mathbb{Z}} c_k \varphi(\cdot - k),\; c_k = \int_{-\pi}^{\pi} \frac{[f, \varphi](\omega)}{[\varphi, \varphi](\omega)} e^{-ik\omega} d\omega$$

oder im Fourierraum

$$(P_\varphi f)^\wedge(\omega) \;=\; \frac{[f, \varphi](\omega)}{[\varphi, \varphi](\omega)} \hat{\varphi}(\omega).$$

Man braucht diese Gleichung später bei der wavelet–Konstruktion.

6.3 Approximationsordnung

Wir wollen jetzt die Projektion skalieren. Statt auf die Shifts $\varphi(\cdot - k)$ projizieren wir für kleine $h > 0$ auf die Shifts von $\frac{1}{h}\varphi((\cdot - hk)/h)$ indem wir den Projektor

$$P_{\varphi,h}(f)(x) := P_\varphi(f(\cdot h))(x/h) \tag{6.16}$$

nehmen. Diese Art der Skalierung wird in der Literatur auch "**stationär**" genannt. Definiert man den Projektor so, ergibt sich die Orthogonalität

$$\begin{aligned}(f - P_{\varphi,h}(f), \tfrac{1}{h}\varphi((\cdot - kh)/h))_{L_2(\mathbb{R})} &= \frac{1}{h}\int_{\mathbb{R}} (f(x) - P_{\varphi,h}(f)(x))\overline{\varphi(x/h - k)}dx \\ &= \frac{1}{h}\int_{\mathbb{R}} (f(x) - P_\varphi(f(\cdot h))(x/h))\overline{\varphi(x/h - k)}dx \\ &= \int_{\mathbb{R}} (f(hy) - P_\varphi(f(\cdot h))(y))\overline{\varphi(y - k)}dy \\ &= 0.\end{aligned}$$

Genauso rechnen wir den Fehler aus, und zwar

$$\begin{aligned}\|f - P_{\varphi,h}(f)\|^2_{L_2(\mathbb{R})} &= \int_{\mathbb{R}} |f(x) - P_\varphi(f(\cdot h))(x/h)|^2 dx \\ &= h\int_{\mathbb{R}} |f(hy) - P_\varphi(f(\cdot h))(y)|^2 dy \\ &= h\|f_h - P_\varphi(f_h)\|^2_{L_2(\mathbb{R})}\end{aligned}$$

mit $f_h(x) := f(xh)$.

Ziel des Ganzen ist, beim Grenzübergang $h \to 0$ noch eine Konvergenz des Fehlers gegen Null zu erreichen, und zwar mit irgendeiner Potenz von h.

Definition 6.17. *Das Projektionsverfahren hat bezüglich eines Unterraums W von $L_2(\mathbb{R})$ die Approximationsordnung m, wenn für alle $f \in W$ eine Abschätzung der Form*

$$\|f - P_{\varphi,h}(f)\|_{L_2(\mathbb{R})} \le C_f h^m$$

mit einer von h unabhängigen Konstanten C_f gilt.

Definition 6.18. *Für beliebige positive κ kann man den* **Sobolewraum**

$$W_2^\kappa(\mathbb{R}) := \{f \in L_2(\mathbb{R}) \ : \ \int_{\mathbb{R}} |\hat{f}(\omega)|^2(1 + |\omega|^2)^k d\omega < \infty\}$$

mit dem Skalarprodukt

$$(f, g)_{W_2^\kappa(\mathbb{R})} := \int_{\mathbb{R}} \hat{f}(\omega)\overline{\hat{g}(\omega)}(1 + |\omega|^2)^k d\omega$$

definieren.

Der obige Raum besteht aus allen Funktionen, die durch Fouriertransformation definierte verallgemeinerte Ableitungen bis zur Ordnung κ haben, die noch in $L_2(\mathbb{R})$ liegen. Wir haben solche Räume schon bei den Fourierreihen gesehen, dort aber im periodischen Fall, und beim Shannon–Sampling. Man bedenke, dass hier auch Werte wie $\kappa = \pi$ oder $\kappa = \sqrt{2}$ möglich sind.

In vielen Situationen (auch dieses kennen wir schon von den Fourierreihen her) haben gutartige Approximations- oder Interpolationsprozesse in $W_2^m(\mathbb{R})$ die Ordnung m.

Theorem 6.19. *Gilt*

$$\|f - P_\varphi f\|_2 \le C|f|_m \tag{6.20}$$

für alle $f \in W_2^m(\mathbb{R})$ mit der Seminorm

$$|f|_m^2 := \int_{\mathbb{R}} |\hat{f}(\omega)|^2 |\omega|^{2m} d\omega,$$

so hat der Projektor $P_{\varphi,h}$ die Approximationsordnung m im Raum $W_2^m(\mathbb{R})$.

Der **Beweis** folgt aus einem einfachen Skalierungsargument:

$$\begin{aligned}
\|f - P_{\varphi,h} f\|_2^2 &= h\|f_h - P_\varphi f_h\|_2^2 \\
&\le Ch|f_h|_m^2 \\
&= Ch\|\hat{f}_h(\omega)|\omega|^m\|_2^2 \\
&= Ch\left\|\frac{1}{h}\hat{f}(\frac{\omega}{h})|\omega|^m\right\|_2^2 \\
&= Ch\frac{1}{h^2}h^{2m}\int_{\mathbb{R})} |\hat{f}(\frac{\omega}{h})^2|\frac{\omega}{h}|^{2m} d\omega \\
&\le Ch^{2m}\left\||\hat{f}(\omega)|^2|\omega|^m\right\|_2^2 \\
&= Ch^{2m}|f|_m^2 \\
&\le Ch^{2m}\|f\|_{W_2^m(\mathbb{R})}^2.
\end{aligned}$$

□

Wären Polynome in $L_2(\mathbb{R})$, so könnte man aus (6.20) schließen, daß Polynome bis zum Grade $m-1$ durch den Projektor noch exakt reproduziert werden. Viele Darstellungen der Fehleranalyse in translationsinvarianten Räumen gehen den Umweg über Reproduktion von Polynomen, aber das wollen wir uns nicht ohne Not antun.

Wir rechnen die Approximationsordnung für den Shannon–Fall noch einmal

vor. Es folgt

$$\begin{aligned} P_s &:= P_{sinc} \\ \hat{P}_s &= \hat{f} \cdot \hat{\chi}_{[-\pi,\pi]} \\ \|f - P_s f\|_2^2 &= \|\hat{f} - \hat{P}_s f\|_2^2 \\ &= \int_{|\omega|\geq\pi} |\hat{f}(\omega)|^2 d\omega \\ &\leq \int_{|\omega|\geq\pi} |\hat{f}(\omega)|^2 \frac{|\omega|^{2m}}{\pi^{2m}} d\omega \\ &\leq \frac{1}{\pi^{2m}} \int_{\mathbb{R}} |\hat{f}(\omega)|^2 |\omega|^{2m} d\omega. \end{aligned}$$

□

Wir benutzen das, um auf den Fehler anderer Projektoren zu schließen.

Theorem 6.21. *Gilt*

$$\|f - P_\varphi f\|_2 \leq C_{\varphi,s} |f|_m$$

für alle bandbreitenbeschränkten $f \in P_s(L_2(\mathbb{R}))$, so hat $P_{\varphi,h}$ die Approximationsordnung m in $W_2^m(\mathbb{R})$.

Beweis: Wir schätzen folgendermaßen ab:

$$\begin{aligned} \|f - P_\varphi f\|_2 &\leq \|f - P_s f\|_2 + \|P_s f - P_\varphi P_s f\|_2 + \|P_\varphi P_s f - P_\varphi f\|_2 \\ &\leq C_s |f|_m + C_{\varphi,s} |P_s f|_m + \|P_\varphi\| \|P_s f - f\|_2 \\ &\leq C_s |f|_m + C_{\varphi,s} |f|_m + C_s |f|_m \end{aligned}$$

weil die Projektoren die Norm 1 in L_2 haben und $|P_s f|_m \leq |f|_m$ gilt. □

6.4 Fehlerabschätzung

Unter den Voraussetzungen des Satzes 6.15 betrachten wir den Fehler der Projektion. Wegen der üblichen Orthogonalität hat man

$$\begin{aligned} \|f - P_\varphi f\|_{L_2(\mathbb{R})}^2 &= \|f\|_{L_2(\mathbb{R})}^2 - \|P_\varphi f\|_{L_2(\mathbb{R})}^2 \\ &= \|\hat{f}\|_{L_2(\mathbb{R})}^2 - \|(P_\varphi f)^\wedge\|_{L_2(\mathbb{R})}^2 \\ &= \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega - \int_{\mathbb{R}} \frac{|[f,\varphi]|^2(\omega)}{[\varphi,\varphi]^2(\omega)} |\hat{\varphi}(\omega)|^2 d\omega \\ &= \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega - \sum_{k\in\mathbb{R}} \int_{-\pi}^{\pi} \frac{|[f,\varphi]|^2(\omega)}{[\varphi,\varphi]^2(\omega)} |\hat{\varphi}(\omega + 2k\pi)|^2 d\omega \\ &= \int_{\mathbb{R}} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|[f,\varphi]|^2(\omega)}{[\varphi,\varphi](\omega)} d\omega. \end{aligned}$$

Jetzt machen wir wie beim Shannon sampling die Annahme

$$\hat{f}(\omega) = 0 \text{ für alle } |\omega| > \pi. \tag{6.22}$$

Dann folgt für alle $\omega \in [-\pi, \pi]$ die Gleichung

$$\begin{aligned} [f, \varphi](\omega) &= \sum_{k \in \mathbb{R}} \hat{f}(\omega + 2\pi k)\overline{\hat{\varphi}(\omega + 2\pi k)} \\ &= \hat{f}(\omega)\overline{\hat{\varphi}(\omega)}. \end{aligned}$$

Das liefert

$$\begin{aligned} \|f - P_\varphi f\|^2_{L_2(\mathbb{R})} &= \int_{-\pi}^{\pi} |\hat{f}(\omega)|^2 \underbrace{\left(1 - \frac{|\hat{\varphi}(\omega)|^2}{[\varphi, \varphi](\omega)}\right)}_{=: L_\varphi(\omega)} d\omega \\ &= \int_{-\pi}^{\pi} |\hat{f}(\omega)|^2 L_\varphi(\omega) d\omega. \end{aligned} \tag{6.23}$$

Soweit L_φ punktweise definiert ist, gilt

$$0 \le L_\varphi(\omega) \le 1,$$

weil $|\hat{\varphi}(\omega)|^2$ genau der Term mit $k = 0$ aus der Summe der Terme der Form $|\hat{\varphi}(\omega + 2k\pi)|^2$ in $[\varphi, \varphi](\omega)$ ist.

Für den Shannon–Operator gilt sogar $L_{\text{sinc}} = 0$, und wenn wir Theorem 6.19 mit (6.23) vergleichen, liegt nahe, dass wir die verschärfte Voraussetzung

$$0 \le L_\varphi(\omega) \le C_L |\omega|^{2m}, \ |\omega| \le \pi \tag{6.24}$$

machen sollten. Dann wird aus (6.23) genau die Voraussetzung von Theorem 6.21 und wir bekommen unser Hauptergebnis

Theorem 6.25. *Gilt (6.24) mit einem punktweise wohldefinierten L_φ, so hat die duch φ definierte skalierte Projektion $P_{\varphi,h}$ im Sobolevraum $W_2^m(\mathbb{R})$ die Approximationsordnung m.*

6.5 Strang–Fix–Bedingungen

Dies sind Bedingungen an φ, um (6.24) zu erreichen. Wir setzen wie bisher Stabilität von φ und zusätzlich Wohldefiniertheit von L_φ voraus, und dann ist es für (6.24) hinreichend, daß die Summe

$$\sum_{k \neq 0} |\hat{\varphi}(\omega + 2\pi k)|^2$$

eine m–fache Nullstelle in 0 hat, denn dieser Ausdruck ist der Zähler von L_φ, während der Nenner gleichmäßig von Null weg beschränkt und positiv ist.

Sehen wir uns das für kleine Argumente $|\omega| << \pi$ an. Dann sind alle Terme voneinander im Verhalten bei Null unabhängig, und alle Terme müssen gleichzeitig eine m-fache Nullstelle in Null haben.

Theorem 6.26. *Ist $\hat{\varphi}$ mindestens m–mal stetig differenzierbar und gelten die* **Strang–Fix–Bedingungen**

$$(\hat{\varphi})^{(j)}(2\pi k) = 0,\ k \in \mathbb{Z},\ k \neq 0,\ 0 \leq j < m,$$

so hat $P_{\varphi,h}$ im Sobolevraum $W_2^m(\mathbb{R})$ die Approximationsordnung m. □

Hier ist ein Programm, das eine leichte Modifikation des Programms für die Auswertung der Shannon–Reihe ist. Wir wenden es für die sinc–Funktion und die Hutfunktion als Generatoren auf einen Sinus an, was zu den Abbildungen 21 und 22 führt.

```
 % testing PSI by convolution
function testPSIconv()
clear all;
close all;
nc=11; % bound for data or coefficient range
xc=(-nc:nc)'; % points for data sampling
freq=1; % frequency for the following
c=sin(freq*xc*2*pi/nc);
n=15;  % divisor for sampling at Z + i/n
ind=(-nc:nc)'; % the basic points for the p vector
h=1/n; % stepsize for upsampling
y=zeros((4*nc+1)*n,1); % the resulting vector
for i=0:n-1
    q=i/n;
    p=phi(ind+q); % this is where we use phi
    cv=conv(c,p); % we convolve
    y(1+i:n:i+n*length(cv))=cv;  % and store the result
end
x=((-nc):h:(nc))'; % this is where we want to plot
dx=sin(freq*x*2*pi/nc);  % data there
dy=y(n*(nc)+1:end-n+1-n*(nc)); % result there
figure
plot(xc,c,'o',x,dy,x,dx);
```

```
legend('Data','Expansion','True function')
title('Evaluation of PSI expansion')
figure
plot(x,dx-dy);
title('Error')

function y=phi(x)
% Hutfunktion
ind=find(abs(x)<1);
y=zeros(size(x));
y(ind)=1-abs(x(ind));
% alternativ sinc
% y=sinc(x);
```

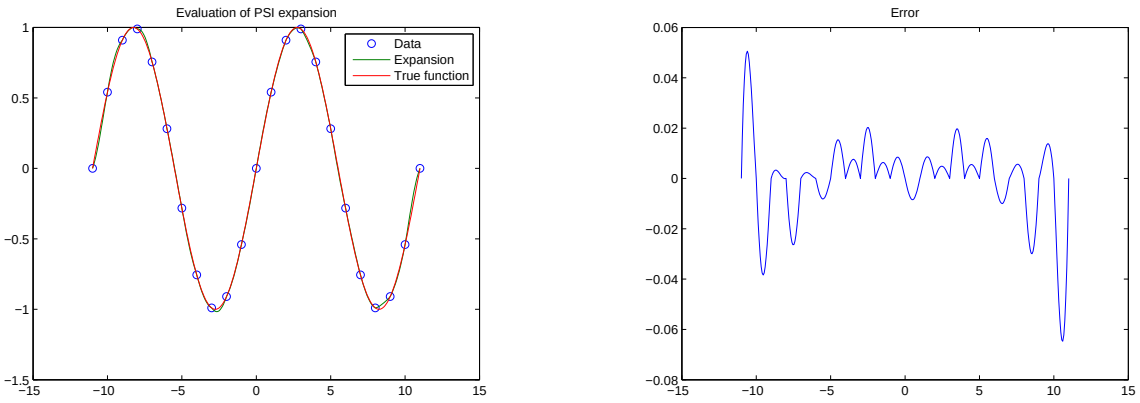

Figure 21: Approximation und Fehler bei sinc–Approximation

6.6 B–Spline–Generatoren

Wir definieren

$$\varphi_1(x) := \chi_{[0,1]}(x) = \begin{cases} 1 & 0 \le x \le 1 \\ 0 & |x| > 1 \end{cases}$$

als die Haarsche Skalierungsfunktion, aber wir falten sie rekursiv zu

$$\varphi_n(x) := (\varphi_{n-1} * \varphi_1)(x) := \int_0^1 \varphi_{n-1}(x-t)dt,\ x \in \mathbb{R},\ n > 1.$$

Man sieht schnell, daß dies stückweise Polynome der Ordnung n ergibt, die "breaks" in $0, 1, \ldots, n$ und einen Träger in $[0, n]$ haben und noch stetige

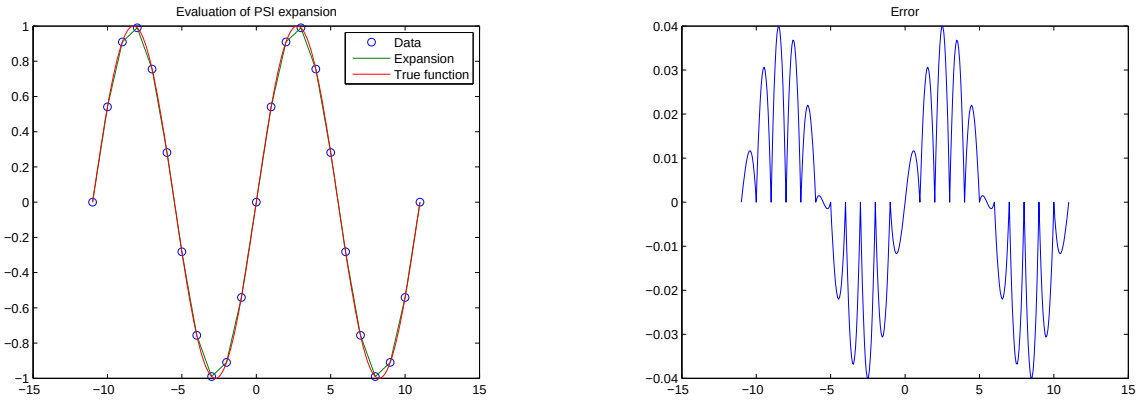

Figure 22: Approximation und Fehler bei Hut–Approximation

Ableitungen bis zur Ordnung $n-1$ haben. Weil ihr Träger im Verhältnis zur Ordnung minimal ist, kann man zeigen, daß sie bis auf die Normierung mit den B–Splines $\Delta_t^n(0,\ldots,n)(x-t)_+^{n-1}$ übereinstimmen.

Ihre Fouriertransformierten sind $\hat{\varphi}_n = \hat{\varphi}_1^n$, und wir wollen zuerst $\hat{\varphi}^1$ ausrechnen:

$$\begin{aligned}
\hat{\varphi}^1(\omega) &= \frac{1}{\sqrt{2\pi}} \int_0^1 e^{-i\omega x} dx \\
&= \frac{1}{\sqrt{2\pi}} \frac{1}{-i\omega} (e^{-i\omega} - 1) \\
&= \frac{1}{\sqrt{2\pi}} \frac{2i}{i\omega} \frac{e^{i\omega/2} - e^{-i\omega/2}}{2i} e^{-i\omega/2} \\
&= \frac{1}{\sqrt{2\pi}} \frac{\sin(\omega/2)}{\frac{\omega}{2}} e^{-i\omega/2} \\
&= \frac{1}{\sqrt{2\pi}} \operatorname{sinc}\left(\frac{\omega}{2\pi}\right) e^{-i\omega/2}.
\end{aligned}$$

Bis auf Faktoren ist die Fouriertransformierte eines Faltungsprodukts das Produkt der Fouriertransformierten. Bei unserer Normierung gilt für

$$(f * g)(x) := \int_{\mathbb{R}} f(x-t) g(t) dt$$

die Fouriertransformation

$$\begin{aligned}(\widehat{f*g})(\omega) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}(f*g)(x)e^{-i\omega x}dx\\ &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\int_{\mathbb{R}}f(x-t)g(t)dt\,e^{-i\omega x}dx\\ &= \sqrt{2\pi}\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}g(t)e^{-i\omega t}dt\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}f(y)e^{-i\omega y}dy\\ &= \sqrt{2\pi}\hat{f}(\omega)\hat{g}(\omega)\end{aligned}$$

Also gilt, weil wir eine n–te Potenz bilden und $(n-1)$–mal falten, die Gleichung

$$\hat{\varphi}_n(\omega) = (2\pi)^{+(n-1)/2}(2\pi)^{-n/2}\mathrm{sinc}^n\left(\frac{\omega}{2\pi}\right)e^{-in\omega/2} = \frac{1}{\sqrt{2\pi}}\mathrm{sinc}^n\left(\frac{\omega}{2\pi}\right)e^{-in\omega/2}.$$

Wir sollten nachprüfen, ob die Translate stabil sind. Dazu müssen wir das Klammerprodukt

$$\begin{aligned}[\varphi_n,\varphi_n](\omega) &= \sum_{k\in\mathbb{Z}}|\hat{\varphi}_n(\omega+2\pi k)|^2\\ &= (2\pi)^{-n}\sum_{k\in\mathbb{Z}}\mathrm{sinc}^{2n}\left(\frac{\omega+2\pi k}{2\pi}\right)\\ &= (2\pi)^{-n}\sum_{k\in\mathbb{Z}}\frac{\sin^{2n}\left(\frac{\omega}{2}+k\pi\right)}{\left(\frac{\omega}{2}+k\pi\right)^{2n}}\\ &= (2\pi)^{-n}\sin^{2n}\left(\frac{\omega}{2}\right)\sum_{k\in\mathbb{Z}}\frac{1}{\left(\frac{\omega}{2}+k\pi\right)^{2n}}\\ &= (2\pi)^{-n}\frac{\sin^{2n}\left(\frac{\omega}{2}\right)}{\left(\frac{\omega}{2}\right)^{2n}}\sum_{k\in\mathbb{Z}}\frac{\left(\frac{\omega}{2}\right)^{2n}}{\left(\frac{\omega}{2}+k\pi\right)^{2n}}\end{aligned}$$

untersuchen. Weil $|\mathrm{sinc}(x)|\le 1$ global gilt, ist der Faktor vor der Summe durch $(2\pi)^{-n}$ abschätzbar. Die Summe läßt sich für $|\omega|\le\pi$ zerlegen und abschätzen durch

$$\begin{aligned}&\sum_{k\in\mathbb{Z}}\frac{\left(\frac{\omega}{2}\right)^{2n}}{\left(\frac{\omega}{2}+k\pi\right)^{2n}}\\ \le\ & 1+\sum_{k>0}\frac{\left(\frac{\pi}{2}\right)^{2n}}{\left(-\frac{\pi}{2}+k\pi\right)^{2n}}+\sum_{k<0}\frac{\left(\frac{\pi}{2}\right)^{2n}}{\left(+\frac{\pi}{2}+k\pi\right)^{2n}}\\ \le\ & 1+\frac{2}{2^{2n}}\sum_{k>0}\frac{1}{(k-1/2)^{2n}}<\infty\end{aligned}$$

für $n \geq 1$. Daraus folgt Beschränktheit von $[\varphi_n, \varphi_n](\omega)$ nach oben. Weil man ω auf $|\omega| \leq \pi$ einschränken kann, folgt auch

$$[\varphi_n, \varphi_n](\omega) \geq (2\pi)^{-n} \min_{|\omega|\leq\pi} \frac{\sin^{2n}\left(\frac{\omega}{2}\right)}{\left(\frac{\omega}{2}\right)^{2n}} = (2\pi)^{-n} \frac{\sin^{2n}\left(\frac{\pi}{2}\right)}{\left(\frac{\pi}{2}\right)^{2n}} = \frac{2^n}{\pi^n} > 0$$

durch Weglassen der Summenterme mit $k \neq 0$ und wir haben Stabilität.

Jetzt sehen wir uns die Strang–Fix–Bedingungen an. Die sinc-Funktion hat einfache Nullstellen an den Punkten $k \neq 0,\ k \in \mathbb{Z}$. Also hat $\hat{\varphi}_n$ in den Stellen $2k\pi$ mit $k \neq 0$ noch n–fache Nullstellen. Es folgt

Theorem 6.27. *Die Approximation in durch B–Splines φ_n erzeugten translationsinvarianten Räumen ist stabil und hat Approximationsordnung m in den Sobolevräumen $W_2^m(\mathbb{R})$ für alle $m \leq n$.* □

7 Das Haar–Wavelet

In diesem Kapitel fügen wir den translationsinvarianten Räumen eine Skalierung hinzu. Das haben wir schon im vorigen Kapitel bei der stationären Skalierung (6.16) des Projektionsoperators gemacht, aber hier sind unsere Skalierungen immer dyadisch, d.h. wir benutzen nur die $h = 2^{-j}$ für $j \geq 0$ mit einem "Levelindex" j. Aber erstmal gibt es eine sehr elementare Einführung, die sich als reine lineare Algebra schreiben läßt und wesentliche algorithmische Dinge vorbereitet.

7.1 Summen und Differenzen

Wir wollen den Gedanken der effizienten Speicherung von zwei Zahlen benutzen, um das Haar-Wavelet[43] herzuleiten. Nehmen wir einmal an, es seien zwei Zahlen a und b gegeben. Natürlich können die zwei Zahlen separat gespeichert werden. Gilt aber $a \approx b$, so erscheint dies nicht sehr effizient. Statt dessen bietet es sich an, den Mittelwert s und die Differenz d zu speichern:

$$s = \frac{a+b}{2}, \qquad d = a - b.$$

Der Vorteil hier ist, dass s von derselben Größenordnung wie a und b ist und dementsprechend genausoviel Speicherplatz benötigt, die Differenz d dagegen mit weniger Speicherplatz auskommen sollte. Man kann sie sogar ganz

[43] `http://de.wikipedia.org/wiki/Haar-Wavelet`

weglassen und erreicht so eine Speicherplatzersparnis auf Kosten eines zu analysierenden Fehlers.

Die Rekonstruktion der Orginalwerte ist gegeben durch

$$a = s + \frac{d}{2}, \qquad b = s - \frac{d}{2}.$$

Es geht auch mit anderen Faktoren. Nimmt man

$$c = a + b,\ d = a - b \tag{7.1}$$

so folgt

$$a = \frac{c+d}{2},\ b = \frac{c-d}{2}.$$

und für

$$c = \frac{a+b}{\sqrt{2}},\ d = \frac{a-b}{\sqrt{2}} \tag{7.2}$$

folgt

$$a = \frac{c+d}{\sqrt{2}},\ b = \frac{c-d}{\sqrt{2}}.$$

Im zweiten Fall wird die Transformation

$$\begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}$$

sogar orthogonal. Rechentechnisch am einfachsten ist der Fall (7.1), aber theoretisch ist (7.2) wegen seiner Symmetrie und Orthogonalität vorteilhafter. Für den Rest dieses Abschnitts ist es gleichgültig, welche Variante wir nehmen, aber wir schreiben alles nur für (7.1) hin.

Nehmen wir an, dass wir nicht nur zwei Zahlen, sondern ein Signal $f^{(n)}$ bestehend aus 2^n Werten gegeben haben, d.h. $f^{(n)} = \{f_k^{(n)} : 0 \le k < 2^n\}$. Ein Signal ist also nichts anderes als ein Vektor von rellen Zahlen. Wir können uns diesen Vektor z.B. als Funktionswerte einer Funktion an den "dyadischen" Stützstellen $2^{-n}k \in [0,1)$ vorstellen, d.h. $f_k^{(n)} = f(k2^{-n})$, $0 \le k < 2^n$. Wenn wir nun die Summen– und Differenzbildung auf jedes der Paare $a = f_{2k}^{(n)}$ und $b = f_{2k+1}^{(n)}$, $0 \le k < 2^{n-1}$, anwenden, erhalten wir zwei neue Vektoren $f^{(n-1)}$ und $r^{(n-1)}$ vermöge

$$f_k^{(n-1)} = f_{2k}^{(n)} + f_{2k+1}^{(n)}, \qquad r_k^{(n-1)} = f_{2k}^{(n)} - f_{2k+1}^{(n)},\ 0 \le k < 2^{n-1}.$$

Das Ausgangssignal $f^{(n)}$, bestehend aus 2^n Samples, wurde also aufgesplittet in zwei Signale mit jeweils 2^{n-1} Samples. Natürlich lässt sich das Ausgangssignal aus den zwei neuen Signalen wieder rekonstruieren, und zwar mit

$$f_{2k}^{(n)} = \frac{f_k^{(n-1)} + r_k^{(n-1)}}{2}, \qquad f_{2k+1}^{(n)} = \frac{f_k^{(n-1)} - r_k^{(n-1)}}{2}, \qquad 0 \leq k < 2^{n-1}.$$

Wendet man den eben beschriebenen Schritt nun rekursiv auf die Signale $f^{(n-1)}, f^{(n-2)}, \ldots, f^{(1)}$ an, so erhält man einen einzelnen Wert $f^{(0)}$ und eine Folge von Signalen $r^{(n-j)}$, $1 \leq j \leq n$, mit jeweils 2^{n-j} Samples. Eine Veranschaulichung ist

$$\begin{array}{ccccccccc} f^{(n)} & \rightarrow & f^{(n-1)} & \rightarrow & f^{(n-2)} & \ldots & f^{(1)} & \rightarrow & f^{(0)} \\ & \searrow & & \searrow & & & & \searrow & \\ & & r^{(n-1)} & & r^{(n-2)} & \ldots & r^{(1)} & & r^{(0)} \end{array}$$

wobei man nur die untere Reihe und $f^{(0)}$ speichert um rückwärts folgendermaßen zu rechnen:

$$\begin{array}{ccccccccc} f^{(n)} & \leftarrow & f^{(n-1)} & \leftarrow & f^{(n-2)} & \ldots & f^{(1)} & \leftarrow & f^{(0)} \\ & \nwarrow & & \nwarrow & & & & \nwarrow & \\ & & r^{(n-1)} & & r^{(n-2)} & \ldots & r^{(1)} & & r^{(0)} \end{array}$$

Die Transformation ist umkehrbar und verlustfrei. Sie bekommt aber einen großen praktischen Nutzen, wenn das Ausgangssignal nur geringfügig variiert und deshalb die Differenzen $r^{(j)}$ klein sind. Man speichert dann nur die betragsgrößten von ihnen und rekonstruiert das Signal aus diesen durch Anwendung der inversen Transformation. Das ist nicht mehr verlustfrei, führt aber oft zu sehr guten Kompressionsraten bei kleinen Reproduktionsfehlern. Wir werden dieses Phänomen genauer zu untersuchen haben.

Der Aufwand, um die Zerlegung zu berechnen, beträgt im j-ten Schritt $\mathcal{O}(2^{n-j})$, $1 \leq j \leq n$, sodass er sich zu $\mathcal{O}(2^n)$ aufsummiert, d.h. linear ist. Dies ist im Vergleich zur FFT, die $\mathcal{O}(n2^n)$ braucht, ausgesprochen günstig. Desweiteren kann die gesamte Transformation *in situ* ausgeführt werden, d.h. es fällt kein weiterer benötigter Speicherplatz an. Die Speicherung erfolgt immer in Vektoren der Länge 2^n, in die man nacheinander

$$\begin{array}{ccccc} & & (f^{(n)}) & & \\ & (f^{(n-1)} & & , r^{(n-1)}) & \\ (f^{(n-2)} & & , r^{(n-2)} & , r^{(n-1)}) & \\ (f^{(n-3)} & , r^{(n-3)} & , r^{(n-2)} & , r^{(n-1)}) & \end{array}$$

bei Zeilenvektorschreibweise einträgt.

7.2 Haarsche Skalierungsfunktion

Bis hierhin ist das Ganze als Matrixtransformation aus der linearen Algebra zu verstehen. Für die weitere Analyse brauchen wir aber Funktionen, und wir wollen ausnutzen, was wir über translationsinvariante Räume wissen. Wir interpretieren deshalb die obigen Werte $f_k^{(j)}$ für $0 \le k < 2^j$ als Funktionswerte einer Funktion $v^{(j)}$ an der Stelle $k2^{-j} \in [0,1)$ und setzen deshalb diese Funktion mit einer zunächst beliebigen Kardinalfunktion φ an als

$$v^{(j)}(x) := \sum_{k=0}^{2^j-1} f_k^{(j)} \varphi(2^j x - k). \tag{7.3}$$

Das ist im Sinne des vorigen Kapitels ein Ansatz aus einem translationsinvarianten Raum

$$V_j := \overline{\text{ span } \{\varphi(2^j x - k),\ k \in \mathbb{Z}\}}. \tag{7.4}$$

Hat man einen dyadischen Punkt $x = m2^{-j} \in [0,1)$ mit $0 \le m < 2^j$, so folgt

$$\begin{aligned} v^{(j)}(m2^{-j}) &= \sum_{k=0}^{2^j-1} f_k^{(j)} \underbrace{\varphi(m-k)}_{=\delta_{mk}} \\ &= f_m^{(j)},\ 0 \le m < 2^j, \end{aligned}$$

d.h. die Funktion $v^{(j)}$ interpoliert die Werte $f_k^{(j)}$ auf dem Gitter in $[0,1)$ mit Schrittweite 2^{-j}, und das gilt für $0 \le j \le n$, wobei wir so tun können, als kämen unsere Anfangsdaten von der stückweise konstanten Funktion $v^{(n)}$, denn diese interpoliert f auf dem "feinsten" Level $j = n$.

Man sieht an der obigen Diskussion, daß man durch $j \ge 0$ indizierte "Levels" von translationsinvarianten Räumen V_j hat. Auf Level j ist es natürlich, sich ein Sampling mit Schrittweite 2^{-j} vorzustellen, und deshalb ist Level $j+1$ reichhaltiger als Level j.

Aber wir wollen alles noch etwas eleganter machen. Dazu nehmen wir einen kardinalen Generator φ, dessen ganzzahlige Translate orthonormal sind. Dann sind die skalierten Translate

$$\varphi_{j,k}(x) := 2^{j/2}\varphi(2^{-j}x - k)$$

bei beliebigem $j \ge 0$ und variierendem $k \in \mathbb{Z}$ auch orthonormal wegen

$$\begin{aligned} & \int_{\mathbb{R}} \varphi_{j,k}(x)\varphi_{j,m}(x)dx \\ = \ & 2^j \int_{\mathbb{R}} \varphi(2^{-j}x - k)\varphi(2^{-j}x - m)dx \\ = \ & \int_{\mathbb{R}} \varphi(y-k)\varphi(y-m)dy = \delta_{mk},\ m,k \in \mathbb{Z}. \end{aligned}$$

Ein wesentlicher Clou der wavelet–Theorie ist nun, zwischen den Levels Verbindungen zu postulieren, und zwar so, daß $V_j \subseteq V_{j+1}$ gilt, weil, wie eben bemerkt, Level $j+1$ reichhaltiger ist als Level j. Das bedeutet aber, daß der Generator $\varphi(2^j x)$ des translationsinvarianten Raums V_j des Levels j sich als Linearkombination von Funktionen aus V_{j+1} schreiben lassen muß, d.h. es muß eine Gleichung

$$\varphi(2^j x) = \sum_{k\in\mathbb{Z}} p_k^{(j)} \varphi(2^{j+1}x - k),\ x \in \mathbb{R}$$

mit gewissen gutartigen Gewichten $p_k^{(j)}$ gelten. Setzt man $y = 2^j x$, so sieht man, daß die Gewichte gar nicht von j abhängen sollten und man eine **Verfeinerungsgleichung**

$$\varphi(y) = \sum_{k\in\mathbb{Z}} p_k \varphi(2y - k),\ y \in \mathbb{R}$$

postulieren sollte. Wir betrachten den einfachsten Fall:

Definition 7.5. *Die Haar'sche* **Skalierungsfunktion** *ist definiert durch*

$$\phi(x) = \begin{cases} 1, & \textit{falls } 0 \le x < 1, \\ 0, & \textit{sonst.} \end{cases}$$

Deren ganzzahlige Translate sind natürlich orthonormal in $L_2(\mathbb{R})$, und die skalierten und orthonormalisierten Translate sind

$$2^{j/2}\phi(2^j x - k) = \left\{ \begin{array}{ll} 2^{j/2}, & \text{falls } k \le 2^j x < k+1, \\ 0, & \text{sonst.} \end{array} \right\} = 2^{j/2}\chi_{[2^{-j}k, 2^{-j}(k+1))}.$$

Sie leben als stückweise konstante Funktionen auf den dyadischen Gitter mit Schrittweite 2^{-j}. Auch die Verfeinerungsgleichung der Haarschen Skalierungsfunktion ist einfach:

$$\begin{aligned} \phi(x) &= \chi_{[0,1)}(x) \\ &= \chi_{[0,1/2)}(x) + \chi_{[1/2,1)}(x) \\ &= \phi(2x) + \phi(2x-1). \end{aligned} \tag{7.6}$$

Damit folgt sofort $V_j \subset V_{j+1}$, aber das reicht nicht, um unsere obige Rechentechnik allgemeiner zu erklären, und wir haben auch noch keine wavelets definiert.

Diese ergeben sich aus der Orthogonalprojektion vom größeren Raum V_{j+1} auf seinen Unterraum V_j und dessen orthogonales Komplement

$$W_j := \{w \in V_{j+1}\ :\ w \perp V_j\}$$

in V_{j+1}. Um W_j in der Form eines translationsinvarianten Raums

$$W_j = \overline{\text{ span } \{\psi(x-k) \ : \ k \in \mathbb{Z}\}} \subset V_{j+1}$$

zu erzeugen, brauchen wir ein "wavelet" ψ, dessen ganzzahlige Translate in V_{j+1} liegen und zu V_j orthogonal sind. Das ist nicht schwierig, weil man folgendes einführen kann:

Definition 7.7. *Das* **Haar wavelet** *ist die Funktion*

$$\psi(x) = \phi(2x) - \phi(2x-1) = \begin{cases} 1, & \textit{falls } 0 \le x < 1/2, \\ -1, & \textit{falls } 1/2 \le x < 1, \\ 0 & \textit{sonst.} \end{cases}$$

Ausgedrückt durch die Skalierungsfunktion hat das Haar wavelet die Form

$$\psi(x) = \phi(2x) - \phi(2x-1) = \phi(x) - 2\phi(2x-1) = 2\phi(2x) - \phi(x),$$

wobei die beiden zweiten Formen sich aus der ersten z.B. durch Anwendung von (7.6) ergeben.

Jetzt sehen wir uns an, was bei dieser Skalierungsfunktion und bei diesem wavelet passiert, wenn wir die Orthogonalzerlegung

$$V_{j+1} = V_j + W_j$$

verwenden. Dann sollten wir aber, um die Orthogonalitäten richtig auszunutzen, die Formeln (7.2) in der Form

$$f_k^{(j)} := \frac{f_{2k}^{(j+1)} + f_{2k+1}^{(j+1)}}{\sqrt{2}}, \ r_k^{(j)} := \frac{f_{2k}^{(j+1)} - f_{2k+1}^{(j+1)}}{\sqrt{2}},$$

verwenden und statt (7.3) die Darstellung

$$v^{(j)}(x) := \sum_{k=0}^{2^j-1} f_k^{(j)} 2^{j/2} \varphi(2^j x - k)$$

als Orthonormalentwicklung einsetzen. Es folgt

$$
\begin{aligned}
& v^{(j+1)}(x) - v^{(j)}(x) \\
= & \sum_{k=0}^{2^{j+1}-1} f_k^{(j+1)} 2^{(j+1)/2} \phi(2^{j+1}x - k) - \sum_{k=0}^{2^j-1} f_k^{(j)} 2^{j/2} \phi(2^j x - k) \\
= & \sum_{k=0}^{2^{j+1}-1} f_k^{(j+1)} 2^{(j+1)/2} \phi(2^{j+1}x - k) - \sum_{k=0}^{2^j-1} \frac{f_{2k}^{(j+1)} + f_{2k+1}^{(j+1)}}{\sqrt{2}} 2^{j/2} \phi(2^j x - k) \\
= & \sum_{k=0}^{2^j-1} f_{2k}^{(j+1)} (2^{(j+1)/2} \phi(2^{j+1}x - 2k) - 2^{(j-1)/2} \phi(2^j x - k)) \\
& + \sum_{k=0}^{2^j-1} f_{2k+1}^{(j+1)} (2^{(j+1)/2} \phi(2^{j+1}x - 2k - 1) - 2^{(j-1)/2} \phi(2^j x - k)) \\
= & \sum_{k=0}^{2^j-1} f_{2k}^{(j+1)} 2^{(j-1)/2} \underbrace{(2\phi(2(2^j x - k)) - \phi(2^j x - k))}_{=\psi(2^j x - k)} \\
& + \sum_{k=0}^{2^j-1} f_{2k+1}^{(j+1)} 2^{(j-1)/2} \underbrace{(2\phi(2(2^j x - k) - 1) - \phi(2^j x - k))}_{=-\psi(2^j x - k)} \\
= & \sum_{k=0}^{2^j-1} \frac{f_{2k}^{(j+1)} - f_{2k+1}^{(j+1)}}{\sqrt{2}} 2^{j/2} \psi(2^j x - k) =: w^{(j)} \in W_j. \\
= & \sum_{k=0}^{2^j-1} \frac{f_{2k}^{(j+1)} - f_{2k+1}^{(j+1)}}{\sqrt{2}} 2^{j/2} \psi(2^j x - k) \\
= & \sum_{k=0}^{2^j-1} r_k^{(j)} 2^{j/2} \psi(2^j x - k) =: w^{(j)}(x) \in W_j.
\end{aligned}
$$

Mit dem Anteil

$$w^{(j)}(x) := \sum_{k=0}^{2^j-1} r_k^{(j)} 2^{j/2} \psi(2^j x - k)$$

hat man also hinter unserer ganzen Rechnerei eine Orthogonalzerlegung

$$v^{(j+1)}(x) = v^{(j)}(x) + w^{(j)}(x) \in V_{j+1} = V_j + W_j,\ 0 \le j < n.$$

Die "feinen" Koeffizienten $f_k^{(j+1)},\ 0 \le k < 2^{j+1}$ von $v^{(j+1)}$ werden also in die "groben" Koeffizienten $f_k^{(j)},\ 0 \le k < 2^j$ bzw. $r_k^{(j)},\ 0 \le k < 2^j$ von v^j bzw. $w^{(j)}$ aufgeteilt.

Die Haarsche Skalierungsfunktion wird uns im Folgenden immer als Musterbeispiel dienen. Ebenso wird der Index j immer die *Skalierung* und der Index k die *Verschiebung* oder auch *Translation* bezeichnen.

7.3 Multi-Skalen-Analyse

Die Räume V_j aus (7.4) haben einige nützliche Eigenschaften, die wir nun zusammenstellen wollen.

Theorem 7.8. *Die V_j sind abgeschlossene Unterräume von $L_2(\mathbb{R})$ mit den folgenden Eigenschaften:*

1. $V_j \subseteq V_{j+1}$,
2. $v \in V_j$ *genau dann wenn* $v(2\cdot) \in V_{j+1}$,
3. $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L_2(\mathbb{R})$,
4. $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$,
5. $\{\phi(\cdot - k) : k \in \mathbb{Z}\}$ *ist eine orthonormale Basis von* V_0.

Proof. Die Eigenschaften (1) und (2) sind offensichtlich erfüllt, (3) folgt aus der Tatsache, dass sich jede $L_2(\mathbb{R})$-Funktion durch Treppenfunktionen beliebig gut approximieren lässt. Für (4) reicht es zu bemerken, dass eine Funktion aus V_j auf Intervallen der Länge 2^{-j} konstant ist. Bei $j \to -\infty$ bleibt nur die Nullfunktion als Funktion in $L_2(\mathbb{R})$ über. Schließlich folgt (5) aus der Tatsache, dass je zwei verschiedene Funktionen nie zusammen von Null verschieden sind. □ □

Aus (2) und (5) (und natürlich sofort aus der Definition) folgt, dass $\{\phi_{j,k} : k \in \mathbb{Z}\}$ eine orthonormale Basis für V_j bildet. Allerdings bildet $\{\phi_{j,k} : j, k \in \mathbb{Z}\}$ keine Basis für $L_2(\mathbb{R})$, da Redundanzen auftreten.

Die in Satz 7.8 hergeleiteten Eigenschaften sind in der Wavelet-Theorie enorm wichtig und geben Anlass zu folgender Definition.

Definition 7.9. *Sei $\{V_j\}_{j\in\mathbb{Z}}$ eine Familie von abgeschlossenen Unterräumen, zu der es eine Funktion $\phi \in L_2(\mathbb{R})$ gibt, sodass die Eigenschaften (1)-(5) aus Satz 7.8 gelten. Dann heißt $\{V_j\}$ eine* Multi-Skalen-Analyse (Multiresolution Analysis)[44] *mit* Skalierungsfunktion ϕ.

Die letzte Bedingung, dass die Shifts von ϕ eine Orthonormalbasis bilden, wird oft abgeschwächt zu einer *Riesz-Basis*, worauf wir hier aber nicht eingehen wollen.

Da $\{\phi(\cdot - k) : k \in \mathbb{Z}\}$ eine Orthonormalbasis von V_0 ist, folgt aus dem klassischen Projektionssatz, dass jede Funktion $f \in V_0$ eine Darstellung $f = \sum_{k\in\mathbb{Z}} p_k \phi(\cdot - k)$ mit $p = (p_k) \in \ell_2$, d.h. $\sum p_k^2 < \infty$, besitzt. Entsprechendes

[44] `http://de.wikipedia.org/wiki/Multiskalenanalyse`

gilt natürlich für alle V_j. Betrachten wir insbesondere die Relation $V_0 \subseteq V_1$, so folgt, dass eine Folge von Zahlen $\{p_k\}_{k\in\mathbb{Z}} \in \ell_2$ exisitiert mit

$$\phi(x) = \sum_{k\in\mathbb{Z}} p_k \phi(2x-k), \tag{7.10}$$

oder

$$\phi = \frac{1}{\sqrt{2}} \sum_{k\in\mathbb{Z}} p_k \phi_{1,k}.$$

Diese Beziehung nennt man *two-scale relation* oder auch Verfeinerungsgleichung. Im Falle der Haarschen Skalierungsfunktion ist die Gleichung einfach gegeben durch

$$\phi(x) = \phi(2x) + \phi(2x-1), \tag{7.11}$$

was sich überträgt auf die skalierten und verschobenen Funktionen zu

$$\phi_{j,k} = \frac{1}{\sqrt{2}} \left(\phi_{j+1,2k} + \phi_{j+1,2k+1}\right).$$

Aus der Tatsache, dass V_j abgeschlossener Unterraum von V_{j+1} ist, folgt die Existenz eines abgeschlossenen Raumes $W_j \subseteq V_{j+1}$, sodass

$$V_{j+1} = V_j \oplus W_j.$$

Die dabei auftretende Summe ist sogar orthogonal. Das Erstaunliche dabei ist, dass diese Räume W_j wieder von den Verschiebungen *einer* skalierten Funktion ψ aufgespannt werden. Diese Funktion ψ heißt dann auch *Wavelet*.

7.4 Das Haarsche Wavelet

Wir kennen scon ein wavelet, n"amlich das für den Spezialfall der Haarschen Skalierungsfunktion, und beweisen jetzt etwas mehr:

Theorem 7.12. *Sei ψ das Haar-Wavelet. Dann ist die Familie $\{\psi_{j,k} : k \in \mathbb{Z}\}$ mit*

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$$

eine orthonormale Basis für W_j und $\{\psi_{j,k}, \phi_{j,\ell} : k, \ell \in \mathbb{Z}\}$ eine orthonormale Basis für V_{j+1}. Insbesondere gilt

$$L_2(\mathbb{R}) = \bigoplus_{j\in\mathbb{Z}} W_j.$$

Die $\{\psi_{j,k} : j, k \in \mathbb{Z}\}$ bilden eine orthonormale Basis für $L_2(\mathbb{R})$.

Beweis: Da die V_j über die Skalierung zusammenhängen, reicht es, die ersten beiden Behauptungen für $j = 0$ zu beweisen. Offensichtlich ist $\psi(\cdot - k)$ ein Element von V_1 aber nicht von V_0. Ferner ist

$$\int_{-\infty}^{\infty} \psi(x-k)\phi(x-\ell)dx = 0,$$

da im Fall $\ell \neq k$ die Träger wieder im wesentlichen verschieden sind, im Fall $\ell = k$ die Behauptung aber offensichtlich gilt. Dies bedeutet, dass der von den $\psi(\cdot - k)$, $k \in \mathbb{Z}$, aufgespannte Raum orthogonal zu V_0 ist. Es reicht also zu zeigen, dass sich jedes $f \in V_1$ als Linearkombination der Shifts von ϕ und ψ schreiben lässt. Aus

$$\phi(x) + \psi(x) = 2\phi(2x), \qquad \phi(x) - \psi(x) = 2\phi(2x-1),$$

folgt

$$\phi_{1,2k} = \frac{1}{\sqrt{2}}(\phi_{0,k} + \psi_{0,k}), \qquad \phi_{1,2k+1} = \frac{1}{\sqrt{2}}(\phi_{0,k} - \psi_{0,k}).$$

Daher lässt sich $f = \sum_{k\in\mathbb{Z}} c_k^{(1)}(f)\phi_{1,k} \in V_1$. schreiben als

$$\begin{aligned} f &= \sum_{k\in\mathbb{Z}} c_{2k}^{(1)}(f)\phi_{1,2k} + \sum_{k\in\mathbb{Z}} c_{2k+1}^{(1)}(f)\phi_{1,2k+1} \\ &= \sum_{k\in\mathbb{Z}} \frac{c_{2k}^{(1)}(f)}{\sqrt{2}}(\phi_{0,k} + \psi_{0,k}) + \sum_{k\in\mathbb{Z}} \frac{c_{2k+1}^{(1)}(f)}{\sqrt{2}}(\phi_{0,k} - \psi_{0,k}) \\ &= \sum_{k\in\mathbb{Z}} \frac{c_{2k}^{(1)}(f) + c_{2k+1}^{(1)}(f)}{\sqrt{2}}\phi_{0,k} + \sum_{k\in\mathbb{Z}} \frac{c_{2k}^{(1)}(f) - c_{2k+1}^{(1)}(f)}{\sqrt{2}}\psi_{0,k}, \end{aligned}$$

sodass W_0 in der Tat von $\{\psi_{j,k} : k \in \mathbb{Z}\}$ aufgespannt wird. Die Funktionen sind auch orthonormal, da je zwei verschiedene im wesentlichen disjunkte Träger haben. Für den nächsten Teil wendet man die Definition der W_j sukzessive an:

$$V_{j+1} = W_j \oplus V_j = W_j \oplus W_{j-1} \oplus V_{j-1} = \ldots = \bigoplus_{\ell \leq j} W_i.$$

Der Grenzwert liefert dann die Behauptung. Schließlich bilden die $\{\psi_{j,k} : j, k \in \mathbb{Z}\}$ tatsächlich eine orthonormale Basis für $L_2(\mathbb{R})$. Für zwei Elemente auf dem gleichen j-Level wissen wir dies bereits. Für zwei unterschiedliche Skalierungslevel j und $i < j$, muss man nur Elemente betrachten, deren Träger sich wesentlich überschneiden. In diesem Fall liegt der Träger des i-Elementes aber in einer Region, wo das j-Element das Vorzeichen nicht wechselt. Daher ist auch Skalarprodukt dieser Elemente Null. □

Die Existenz eines Wavelets bei beliebiger gegebener Multi-Skalen-Analyse folgt aus folgendem Satz, der im Folgekapitel bewiesen wird. Wir werden aber im Rahmen der schnellen Wavelet-Transformation im nächsten Abschnitt zumindest zeigen, dass die Shifts von ϕ und ψ den vollen Raum V_0 ergeben. Man beachte, dass die im Satz angegebene Konstruktion bei der Haarschen Skalierungsfunktion bis auf das Vorzeichen zu obigem Haar-Wavelet führt.

Theorem 7.13. *Sei (V_j) eine MRA mit orthogonaler Skalierungsfunktion $\phi \in V_0$. Seien $\{c_k\} \in \ell_2$ die Koeffizienten der Verfeinerungsgleichung (7.10). Setzt man*

$$\psi = \sum_{k\in\mathbb{Z}} (-1)^k p_{1-k} \phi(2x-k), \tag{7.14}$$

so bekommt man ein wavelet, und es ist $\{\psi_{0,k} : k \in \mathbb{Z}\}$ eine Orthonormalbasis für W_0 und $\{\psi_{j,k} : j,k \in \mathbb{Z}\}$ eine Orthonormalbasis für $L_2(\mathbb{R})$.

Das Haar-Wavelet und die Haarsche Skalierungsfunktion haben einige numerisch sehr wertvolle Eigenschaften. Sie haben beide kompakten Träger und die Verfeinerungsgleichung ist endlich, d.h. nur endlich viele (nämlich zwei) Koeffizienten sind von Null verschieden. Ein gravierender Nachteil ist allerdings die fehlende Glätte. Die Konstruktion glatterer Funktionen benötigt allerdings Mittel, die über die Ziele dieses Kapitels hinausgehen, aber wir werden das im nächsten Kapitel sehr gründlich nachholen. Interessanterweise ist für die konkrete Rechnung die Kenntnis des Wavelets nicht nötig. Es reicht völlig aus, die Verfeinerungsgleichung zu kennen, wie wir gleich sehen werden.

7.5 Die schnelle Wavelet-Transformation

Bisher haben wir mit einem Rechenverfahren begonnen und dann festgestellt, daß es sich als eine Multiresolutionsanalyse mit der Haarschen Skalierungsfunktion und dem Haarschen wavelet schreiben läßt. Wir gehen jetzt umgekehrt und etwas allgemeiner vor: wir wollen aus der Orthonormalität und der Verfeinerung einer Skalierungsfunktion und eines wavelets auf ein Rechenverfahren schließen.

Seien also Generatoren φ und ψ aus $L_2(\mathbb{R})$ gegeben, die orthonormale Translate

$$\varphi_{j,k}(x) := 2^{j/2}\varphi(2^j x - k),\ \psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k),\ k \in \mathbb{Z},\ j \geq 0$$

haben, die die translationsinvarianten Räume V_j und W_j erzeugen, und zwar so, daß die orthogonale Zerlegung $V_{j+1} = V_j \oplus W_j$ gilt. Sie mögen die Ver-

feinerungsgleichungen

$$\varphi(x) = \sum_{k\in\mathbb{Z}} p_k \varphi(2x-k),\ \psi(x) = \sum_{k\in\mathbb{Z}} q_k \varphi(2x-k)$$

haben, wobei wir uns nicht darum kümmern, welche Bedingungen man stellen muß, um diese eventuell unendlichen Summen auszuwerten. Man kann das wie im vorigen Kapitel analysieren, aber das wollen wir hier (noch) nicht machen.

Geht man zu beliebigen Levels j über, so folgt

$$\begin{aligned}
\varphi_{j,m}(x) &= 2^{j/2}\varphi(2^j x - m) \\
&= 2^{j/2} \sum_{k\in\mathbb{Z}} p_k \varphi(2(2^j x - m)x - k) \\
&= \frac{1}{\sqrt{2}} \sum_{k\in\mathbb{Z}} p_k 2^{(j+1)/2} \varphi(2^{j+1}x - 2m - k) \\
&= \frac{1}{\sqrt{2}} \sum_{k\in\mathbb{Z}} p_k\, \varphi_{j+1,2m-k}(x),\ m\in\mathbb{Z},\ j\geq 0.
\end{aligned}$$

und analog

$$\psi_{j,m}(x) = \frac{1}{\sqrt{2}} \sum_{k\in\mathbb{Z}} q_k\, \varphi_{j+1,2m-k}(x),\ m\in\mathbb{Z},\ j\geq 0.$$

Für spätere Zwecke brauchen wir auch noch, wie sich $\varphi_{j+1,\ell} \in V_{j+1}$ aus den $\varphi_{j,m} \in V_j$ und den $\psi_{j,m} \in W_j$ zusammensetzen läßt:

$$\begin{aligned}
\varphi_{j+1,\ell} &= \sum_{m\in\mathbb{Z}} ((\varphi_{j+1,\ell}, \varphi_{j,m})_2 \varphi_{j,m} + (\varphi_{j+1,\ell}, \psi_{j,m})_2 \psi_{j,m}) \\
&= \frac{1}{\sqrt{2}} \sum_{m\in\mathbb{Z}} \left((\varphi_{j+1,\ell}, \sum_{k\in\mathbb{Z}} p_k\, \varphi_{j+1,2m-k})_2 \varphi_{j,m} + (\varphi_{j+1,\ell}, \sum_{k\in\mathbb{Z}} q_k\, \varphi_{j+1,2m-k})_2 \psi_{j,m} \right) \\
&= \frac{1}{\sqrt{2}} \sum_{m\in\mathbb{Z}} \left(p_{2m-\ell}\, \varphi_{j,m} + q_{2m-\ell}\, \psi_{j,m} \right),\ \ell\in\mathbb{Z},\ j\geq 0.
\end{aligned} \tag{7.15}$$

Im Gegensatz zur bisherigen Theorie sind wir jetzt noch etwas radikaler und definieren formal die Projektoren

$$P_j\ :\ L_2(\mathbb{R}) \to V_j,\ Q_j\ :\ L_2(\mathbb{R}) \to W_j$$

mit

$$\begin{aligned}
v^{(j)} &:= P_j(f) &&= \sum_{k\in\mathbb{Z}} (f, \varphi_{j,k})_2 \varphi_{j,k} \\
w^{(j)} &:= Q_j(f) &&= \sum_{k\in\mathbb{Z}} (f, \psi_{j,k})_2 \psi_{j,k}
\end{aligned}$$

für Funktionen $f \in L_2(\mathbb{R})$ und alle $j \geq 0$. Dann wollen wir natürlich rekursiv die Gleichungen

$$v^{(j+1)} = v^{(j)} + w^{(j)},\ j \geq 0$$

haben und über geeignete Rekursionsformeln die Koeffizienten berechnen. Der rechentechnische Ausgangspunkt ist dabei $v^{(n)} = P_n(f)$ mit den Koeffizienten $(f, \varphi_{n,k})_2,\ k \in \mathbb{Z}$. Beim Haar-Wavelet lassen sich diese Koeffizienten auf dem höchsten Level leicht (wenigstens näherungsweise) berechnen. Da die $\varphi_{n,k},\ k \in \mathbb{Z}$, eine orthonormale Basis bilden, gilt

$$c_k^{(n)}(f) = (f, \varphi_{n,k}) = \int_{-\infty}^{\infty} f(x)\varphi_{n,k}(x)dx = 2^{n/2} \int_{2^{-n}k}^{2^{-n}(k+1)} f(x)dx,$$

und der letzte Ausdruck kann z.B. durch eine Quadraturformel genähert werden. Er ist das Integralmittel über je ein Intervall auf dem feinsten Level.

Wir definieren zwecks Notationsverkürzung

$$c_k^{(j)} := (f, \varphi_{j,k})_2,\ d_k^{(j)} := (f, \psi_{j,k})_2,\ k \in \mathbb{Z},\ j \geq 0$$

und berechnen die Koeffizienten für die **wavelet–Zerlegung** über

$$\begin{aligned} c_m^{(j)} &= (f, \varphi_{j,m})_2 \\ &= \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} p_k\, (f, \varphi_{j+1,2m-k})_2 \\ &= \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} p_k\, c_{2m-k}^{(j+1)},\ m \in \mathbb{Z},\ j \geq 0. \end{aligned}$$

Ganz analog folgt

$$d_m^{(j)} = \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} q_k\, c_{2m-k}^{(j+1)},\ m \in \mathbb{Z},\ j \geq 0,$$

und damit ist klar, wie man vom feineren Level $j+1$ zum gröberen Level j übergeht. Wir bekommen wieder eine Rechnung der Art

$$\begin{array}{ccc} c^{(j+1)} & \rightarrow & c^{(j)} \\ & \searrow & \\ & & d^{(j)} \end{array}$$

die allerdings je nach Länge der Masken $\{p_k\}$ und $\{q_k\}$ recht kompliziert ausfallen kann. Wir werden uns später darum kümmern müssen, verfeinerbare Generatoren φ und ψ mit möglichst kleiner Maske und möglichst hoher Konvergenzordnung, d.h. mit möglichst "guten" Strang–Fix–Bedingungen

zu bekommen. Von der Orthonormalitätsforderung werden wir dann aber wieder abgehen.

Wie sieht dann die **wavelet–Rekonstruktion**

$$\begin{array}{ccc} c^{(j+1)} & \leftarrow & c^{(j)} \\ & \nwarrow & \\ & & d^{(j)} \end{array}$$

aus? Dazu rechnen wir folgendermaßen:

$$\begin{aligned} c_\ell^{(j+1)} &= (f, \varphi_{j+1,\ell})_2 \\ &= \frac{1}{\sqrt{2}} \sum_{m\in\mathbb{Z}} \left(p_{2m-\ell}\,(f, \varphi_{j,m})_2 + q_{2m-\ell}\,(f, \psi_{j,m})_2\right) \\ &= \frac{1}{\sqrt{2}} \sum_{m\in\mathbb{Z}} \left(p_{2m-\ell}\, c_m^{(j)} + q_{2m-\ell} d_m^{(j)}\right),\ \ell \in \mathbb{Z},\ j \geq 0 \end{aligned}$$

und haben damit auch die Umkehrung der Zerlegung. Rekonstruktion und Zerlegung kann man zusammen als **diskrete wavelet–Transformation** bezeichnen. Ob sie auch "schnell" ist, hängt von den Masken ab.

Die bei der Zerlegung und der Rekonstruktion auftretenden Summen sind diskrete Faltungen. Bei der "schnellen Wavelet-Transformation"[45] geht es also darum, die feinere Darstellung auf V_{j+1} in der gröberen Darstellung auf V_j plus der Detail-Differenz aus W_j darzustellen und umgekehrt. Speichern muss man dabei nur die Koeffizienten auf auf dem gröbsten Level und sämtliche Details. Eine Datenkompression bekommt man, wenn man "kleine" Details aus den wavelet–Anteilen $w^{(j)}$ wegläßt und rücktransformiert.

8 Allgemeine Wavelets

Jetzt werden wir die spezielle Situation der Haarschen Skalierungsfunktion und des Haarschen wavelets verlassen und sehr viel allgemeiner an wavelets[46] herangehen. Unser Ziel ist, aufbauend auf die Theorie der translationsinvarianten Räume eine wavelet–Theorie zu entwickeln, die mit allgemeineren Generatoren (weder kardinal noch mit orthogonalen Translaten) arbeitet. Sie sollte eine Multiskalenanalyse und eine "schnelle" wavelet–Transformation liefern, die möglichst hohe Konvergenzordnung garantiert.

[45] `http://de.wikipedia.org/wiki/Schnelle_Wavelet-Transformation`
[46] `http://de.wikipedia.org/wiki/Wavelet`

8.1 Verfeinerbare Funktionen

Es gelte die Verfeinerungsgleichung

$$\varphi(x) = \sum_{k\in\mathbb{Z}} p_k \varphi(2x-k)$$

für eine **Skalierungsfunktion** $\varphi \in L_2(\mathbb{R})$ unter geeigneten Voraussetzungen an φ bzw. die Koeffizienten $p_k \in \mathbb{R}$ der "Maske" $p := \{p_k\}_{k\in\mathbb{Z}}$. Beispielsweise kann man voraussetzen, dass entweder die Folge der p_k endlich ist oder φ kompakten Träger hat, aber es sind auch andere Voraussetzungen denkbar, z.B. rasches Abklingen der p_k. Wir werden im Folgenden immer voraussetzen, dass φ im Sinne von (6.14) stabil ist.

Lemma 8.1. *Die Fouriertransformierte einer verfeinerbaren Funktion φ mit*

$$\sum_{k\in\mathbb{Z}} |p_k| < \infty \tag{8.2}$$

erfüllt

$$\hat{\varphi}(\omega) = \hat{\varphi}\left(\frac{\omega}{2}\right) P\left(e^{-i\omega/2}\right) \tag{8.3}$$

mit der Laurentreihe

$$P(z) := \frac{1}{2}\sum_{k\in\mathbb{Z}} p_k z^k,\ z = e^{-i\omega/2}.$$

Beweis: Wir berechnen zuerst

$$\begin{aligned}(\hat{\varphi}(2\cdot - k))(\omega) &= \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \varphi(2x-k)e^{-ix\omega}dx \\ &= \frac{1}{2}\frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}} \varphi(y)e^{-iy\omega/2}e^{-ik\omega/2}dx \\ &= \frac{1}{2}\hat{\varphi}\left(\frac{\omega}{2}\right)e^{-ik\omega/2}\end{aligned} \tag{8.4}$$

und das ergibt

$$\begin{aligned}\hat{\varphi}(\omega) &= \hat{\varphi}\left(\frac{\omega}{2}\right)\frac{1}{2}\sum_{k\in\mathbb{Z}} p_k e^{-ik\omega/2} \\ &= \hat{\varphi}\left(\frac{\omega}{2}\right) P\left(e^{-i\omega/2}\right).\end{aligned}$$

□

Biinfinite Reihen der Art von $P(z)$werden wir uns nur auf dem Einheitskreisrand ansehen und in den Anwendungen erwarten, daß die Koeffizienten für

$|k| \to \infty$ schnell genug abklingen, um (8.2) zu garantieren. Insbesondere setzen wir ab hier immer (8.2) voraus.

Iteriert man die Beziehung (8.3) formell, so kann man das infinite Produkt

$$\prod_{j\geq 1} P\left(e^{-i2^{-j}\omega}\right)$$

bilden, um damit die Fouriertransformierte von φ aus den Koeffizienten p_k der Maske zu berechnen, bis auf einen multiplikativen Faktor. Aber das wollen wir hier nicht ausführen, denn man kann besser die Verfeinerungsgleichung selber benutzen, um φ näherungsweise aus der Maske auszurechnen. Das machen wir im nächsten Abschnitt. Aber wir folgern aus (8.3) noch, dass aus der Gleichung im Nullpunkt folgt, dass $P(1) = 1$ und damit

$$\sum_{k\in\mathbb{Z}} p_k = 2 \tag{8.5}$$

gelten sollte, wenn man sich an die Konstruktion wagt.

Wir rechnen jetzt mal das Klammerprodukt einer verfeinerbaren Funktion aus:

$$\begin{aligned}
[\varphi,\varphi](\omega) &= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega+2\pi k)|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi k)P(e^{-i(\omega/2+\pi k)})|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi 2k)P(e^{-i(\omega/2+\pi 2k)})|^2 \\
&\quad + \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi(2k+1))P(e^{-i(\omega/2+\pi(2k+1))})|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2\pi k)P(e^{-i\omega/2})|^2 \\
&\quad + \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2k\pi+\pi)P(e^{-i(\omega/2+\pi)})|^2 \\
&= |P(e^{-i\omega/2})|^2 \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2\pi k)|^2 \\
&\quad + |P(e^{-i(\omega/2+\pi)})|^2 \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2k\pi+\pi)|^2 \\
&= |P(e^{-i\omega/2})|^2[\varphi,\varphi](\omega/2) + |P(e^{-i(\omega/2+\pi)})|^2[\varphi,\varphi](\omega/2+\pi) \\
&= |P(z)|^2[\varphi,\varphi](\omega/2) + |P(-z)|^2[\varphi,\varphi](\omega/2+\pi),
\end{aligned} \tag{8.6}$$

wieder mit $z = e^{-i\omega/2}$. Das ergibt eine 4π–periodische Funktion.

Die Translate von φ sind genau dann orthogonal, wenn das Klammerprodukt konstant ist. Das ist bei gegebenem und verfeinerbarem φ nicht garantiert.

Ein wichtiges Beispiel sind die B–Splines. Sie sind verfeinerbar, haben aber keine orthogonalen Translate.

Letzteres haben wir schon im Kapitel über translationsinvariante Räume gesehen, und die Verfeinerbarkeit der Haarschen Funktion $\varphi_1(x) := \chi_{[0,1]}(x)$ gilt mit $P(z) = (1+z)/2$ wegen

$$\varphi_1(x) = \varphi_1(2x) + \varphi_1(2x-1).$$

Somit hat man

$$\hat{\varphi}_1(\omega) = \hat{\varphi}_1(\omega/2)\frac{1}{2}(1+z)^1,\ z = e^{-i\omega/2}.$$

Durch Potenzieren folgt

$$\hat{\varphi}_n(\omega) = \hat{\varphi}_1^n(\omega/2)\frac{1}{2}\frac{1}{2^{n-1}}(1+z)^n,\ z = e^{-i\omega/2},$$

und das beweist die Verfeinerbarkeit aller φ_n mit dem Polynom

$$P_n(z) = \frac{1}{2}\frac{1}{2^{n-1}}(1+z)^n$$

und den Maskenkoeffizienten

$$p_k^{(n)} := \frac{1}{2^{n-1}}\binom{n}{k},\ 0 \le k \le n.$$

An dieser Stelle könnte man diskutieren, ob die Orthogonalisierung einer verfeinerbaren Funktion wieder verfeinerbar ist, aber das lassen wir mal als Übungsaufgabe.

Lemma 8.7. *Im Falle orthogonaler Translate von φ gilt*

$$1 = |P(z)|^2 + |P(-z)|^2 \tag{8.8}$$

auf dem Einheitskreis. Diese Beziehung ist äquivalent zu

$$2\delta_{j0} = \sum_{k\in\mathbb{Z}} p_k p_{k-2j},\ j \in \mathbb{Z},$$

und das sind unendlich viele quadratische Gleichungen mit unendlich vielen Unbekannten.

Beweis; Der erste Teil folgt aus der obigen Rechnung sofort. Den zweiten Teil rechnet man folgendermaßen herbei:

$$\begin{aligned}
1 &= |P(z)|^2 + |P(-z)|^2 \\
4 &= \left|\sum_{k\in\mathbb{Z}} p_k z^k\right|^2 + \left|\sum_{k\in\mathbb{Z}} p_k (-1)^k z^k\right|^2 \\
&= \sum_{k,m\in\mathbb{Z}} p_k p_m z^{k-m} + \sum_{k,m\in\mathbb{Z}} p_k p_m (-1)^{k+m} z^{k-m} \\
&= \sum_{n\in\mathbb{Z}} z^n \left(\sum_{k\in\mathbb{Z}} p_k p_{k-n} + \sum_{k\in\mathbb{Z}} p_k p_{k-n} (-1)^{k+k-n}\right) \\
&= \sum_{n\in\mathbb{Z}} z^n \left(1 + (-1)^n\right) \sum_{k\in\mathbb{Z}} p_k p_{k-n} \\
&= 2 \sum_{2j\in\mathbb{Z}} z^{2j} \sum_{k\in\mathbb{Z}} p_k p_{k-2j}
\end{aligned}$$

und weil diese Potenzreihe konstant sein muß, folgt die Behauptung. □

8.2 Strang–Fix–Bedingungen

Jetzt wollen wir untersuchen, wann verfeinerbare Skalierungsfunktionen die Strang–Fix–Bedingungen erfüllen. Wir setzen die Verfeinerungsgleichung in der Form

$$\hat{\varphi}(\omega) = \hat{\varphi}(\omega/2) P(z),\ z = e^{-i\omega/2}$$

voraus und nehmen an, daß $\hat{\varphi}$ glatt ist. Dann wollen wir die Gleichung j–mal differenzieren. Dazu brauchen wir die Ableitungen von $P(e^{-i\omega/2})$ nach ω, die wir per Induktion schreiben können als

$$\frac{d^j}{d\omega^j} P(e^{-i\omega/2}) = \sum_{k=0}^{j} P^{(k)}(e^{-i\omega/2}) p_{k,j}(e^{-i\omega/2})$$

mit gewissen Polynomen $p_{k.j}$ vom Grad $\le k$. Wir bekommen

$$\hat{\varphi}^{(j)}(\omega) = \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(\omega/2) 2^{-m} \sum_{\ell=0}^{j-m} P^{(\ell)}(z) p_{\ell,j-m}(z),\ z = e^{-i\omega/2}.$$

Das werten wir mal in $\omega_k = 2\pi k$ aus und benutzen $z = e^{-i\omega/2} = (-1)^k$. Es folgt

$$\hat{\varphi}^{(j)}(2\pi k) = \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(\pi k) 2^{-m} \sum_{\ell=0}^{j-m} P^{(\ell)}((-1)^k) p_{\ell,j-m}((-1)^k).$$

Ist k in obiger Gleichung gerade, so folgt für alle $k \in \mathbb{Z}$

$$\hat{\varphi}^{(j)}(4\pi k) = \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(2\pi k) 2^{-m} \sum_{\ell=0}^{j-m} P^{(\ell)}(1) p_{\ell,j-m}(1),$$

d.h. das eventuelle Verschwinden von Ableitungen von $\hat{\varphi}$ in den Punkten $2\pi k$ vererbt sich auf die Punkte $4\pi k$. Im ungeraden Fall folgt für alle $k \in \mathbb{Z}$

$$\hat{\varphi}^{(j)}(4\pi k + 2\pi) = \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(2\pi k + \pi) 2^{-m} \sum_{\ell=0}^{j-m} P^{(\ell)}(-1) p_{\ell,j-m}(-1).$$

Wenn wir nun voraussetzen, daß

$$P(z) = (1+z)^n R(z) \tag{8.9}$$

gilt, d.h. daß P in -1 eine n–fache Nullstelle hat, so verschwindet die linke Seite der obigen Gleichung für alle dort auftretenden k und alle $j \leq n-1$. Die Nullstellen sind dann genau die ungeraden Vielfachen von 2π. Durch Anwenden dieser Beziehungen kann man alle $\omega = 2\pi k$ mit $k \neq 0$ erreichen, denn nach endlich vielen Anwendungen der vorletzten Formel reduziert sich alles auf die ungeraden Vielfachen von 2π. Somit gilt

Theorem 8.10. *Ist $\hat{\varphi}$ mindestens n–mal stetig differenzierbar und verfeinerbar mit (8.3) und (8.9), so gelten die Strang–Fix–Bedingungen bis zur Ordnung n und die stationär skalierte Projektion auf Translate von φ hat die Approximationsordnung n im Sobolevraum $W_2^n(\mathbb{R})$.* □

8.3 Wavelets

Wenn man von einer verfeinerbaren Funktion φ ausgeht, bekommt man erst einmal den shift–invarianten Raum

$$V_0 := S_\varphi := \operatorname{clos}_{L_2(\mathbb{R})} \operatorname{span} \{\varphi(\cdot - k) \ : \ k \in \mathbb{Z}\}.$$

Die Verfeinerbarkeit sichert die Inklusion $V_0 \subset V_1$ mit

$$V_1 := S_{\varphi(2\cdot)} := \operatorname{clos}_{L_2(\mathbb{R})} \operatorname{span} \{\varphi(2 \cdot -k) \ : \ k \in \mathbb{Z}\}.$$

Wir wollen nun die Orthogonalzerlegung

$$V_1 = V_0 + W_0$$

durchführen und W_0 durch ein **wavelet** ψ erzeugen als

$$W_0 := S_\psi := \operatorname{clos}_{L_2(\mathbb{R})} \operatorname{span} \{\psi(\cdot - k) \ : \ k \in \mathbb{Z}\}.$$

Weil es in $W_0 \subset V_1$ liegt, müßte es dann auch eine Gleichung der Form

$$\psi(x) := \sum_{k\in\mathbb{Z}} q_k \varphi(2x-k) \tag{8.11}$$

erfüllen, d.h. es gilt

$$\hat{\psi}(\omega) = \hat{\varphi}(\omega/2)Q(z),\ z = e^{-i\omega/2},\ Q(z) := \frac{1}{2}\sum_{k\in\mathbb{Z}} q_k z^k. \tag{8.12}$$

Wir suchen zuerst nach notwendigen Bedingungen für ψ oder Q. Wegen $\psi \in W_0$ brauchen wir $\psi \perp V_0$, d.h. nach dem vorigen Kapitel muß gelten

$$\begin{aligned} 0 &= [\psi,\varphi](\omega) \\ &= \sum_{k\in\mathbb{Z}} \hat{\psi}(\omega+2\pi k)\overline{\hat{\varphi}(\omega+2\pi k)} \\ &= \sum_{k\in\mathbb{Z}} \hat{\varphi}\left(\frac{\omega}{2}+\pi k\right) Q\left(e^{-i\left(\frac{\omega}{2}+\pi k\right)}\right) \overline{\hat{\varphi}\left(\frac{\omega}{2}+\pi k\right) P\left(e^{-i\left(\frac{\omega}{2}+\pi k\right)}\right)} \end{aligned}$$

und wenn wir das wie in (8.6) in gerade und ungerade k aufspalten, folgt

Theorem 8.13. *Für $\psi \perp V_0$ ist notwendig und hinreichend, daß gilt*

$$0 = [\varphi,\varphi]\left(\frac{\omega}{2}\right) Q(z)\overline{P(z)} + [\varphi,\varphi]\left(\frac{\omega}{2}+\pi\right) Q(-z)\overline{P(-z)}. \tag{8.14}$$

Sind die Translate von φ orthogonal, so vereinfacht sich diese Bedingung zu

$$0 = Q(z)\overline{P(z)} + Q(-z)\overline{P(-z)}.$$

Eine weitere wichtige Bedingung ist, dass aus

$$f \in V_1,\ f \perp V_0,\ f \perp W_0 \tag{8.15}$$

folgt, daß f verschwindet, denn dann hat man die direkte orthogonale Summe $V_1 = V_0 + W_0$. Jedes $f \in V_1$ hat die Form

$$f(x) = \sum_{k\in\mathbb{Z}} c_k(f)\varphi(2x-k)$$

und deshalb gilt

$$\hat{f}(\omega) = \hat{\varphi}(\omega/2)F(z),\ F(z) = \frac{1}{2}\sum_{k\in\mathbb{Z}} c_k(f) z^k.$$

Wenn wir die Bedingung $[f, \varphi] = 0$ auswerten, bekommen wir wie in (8.14) das Ergebnis

$$0 = [\varphi, \varphi] \left(\frac{\omega}{2}\right) F(z)\overline{P(z)} + [\varphi, \varphi] \left(\frac{\omega}{2} + \pi\right) F(-z)\overline{P(-z)}$$

und $[f, \psi] = 0$ wird zu

$$0 = [\varphi, \varphi] \left(\frac{\omega}{2}\right) F(z)\overline{Q(z)} + [\varphi, \varphi] \left(\frac{\omega}{2} + \pi\right) F(-z)\overline{Q(-z)}.$$

Wenn wir diese zwei homogenen Gleichungen ins konjugiert Komplexe überführen, sind die Funktionen $[\varphi, \varphi] \left(\frac{\omega}{2}\right) F(z)$ und $[\varphi, \varphi] \left(\frac{\omega}{2} + \pi\right) F(-z)$ Lösungen eines homogenen Gleichungssystems mit der Koeffizientenmatrix

$$\begin{pmatrix} P(z) & P(-z) \\ Q(z) & Q(-z) \end{pmatrix}.$$

Weil wir immer Stabilität von φ voraussetzen, gilt

Theorem 8.16. *Wenn die Determinante*

$$P(z)Q(-z) - Q(z)P(-z) \tag{8.17}$$

auf dem Einheitskreis nirgends verschwindet, folgt aus (8.15), daß $f = 0$ gilt.

Wir fassen zusammen:

Theorem 8.18. *Wenn es gelingt, eine Funktion ψ mit (8.12) zu finden, so daß (8.14) und*

$$P(z)Q(-z) - Q(z)P(-z) \quad \neq \quad 0 \text{ für alle } |z| = 1$$

gelten, so hat man ein verfeinerbares wavelet, das W_0 erzeugt und garantiert, daß $W_0 \perp V_0$ und $V_1 = V_0 + W_0$ gelten.

Um ein wavelet zu konstruieren, das Theorem 8.18 erfüllt, gehen wir schrittweise vor. Wir definieren

$$\eta(x) := \varphi(2x), \ x \in \mathbb{R}$$

und berechnen das Ergebnis $\psi_0 := \eta - P_\varphi \eta$ des Fehlers des Projektors P_φ auf V_0. Es ist klar orthogonal zu V_0 nach Konstruktion, und es könnte ein

guter Kandidat für ein wavelet sein. Rechnen wir die Fouriertransformierte von $\psi_0 := \eta - P_\varphi \eta$ aus:

$$\begin{aligned}\hat{\psi}_0(\omega) &= \hat{\eta}(\omega) - (P_\varphi \eta)^\wedge(\omega) \\ &= \frac{1}{2}\hat{\varphi}(\omega/2) - \frac{[\eta,\varphi](\omega)}{[\varphi,\varphi](\omega)}\hat{\varphi}(\omega).\end{aligned}$$

Wir machen uns das Leben etwas leichter, wenn wir den Nenner heraufmultiplizieren und das Ergebnis als Fouriertransformierte einer anderen Funktion ψ_1 auffassen. Das liefert

$$\hat{\psi}_1(\omega) := \frac{1}{2}\hat{\varphi}\left(\omega/2\right)[\varphi,\varphi](\omega) - [\eta,\varphi](\omega)\hat{\varphi}(\omega)$$

und wir sehen uns die Teile an. Mit Einsetzen von (8.3) folgt zuerst

$$\begin{aligned}\hat{\varphi}(\omega/2)[\varphi,\varphi](\omega) &= \hat{\varphi}(\omega/2)\sum_{k\in\mathbb{Z}}|\hat{\varphi}(\omega+2\pi k)|^2 \\ &= \hat{\varphi}(\omega/2)\sum_{k\in\mathbb{Z}}|\hat{\varphi}(\omega/2+\pi k)|^2|P(e^{-i(\omega+2\pi k)/2})|^2 \\ &= \hat{\varphi}(\omega/2)\sum_{k\in\mathbb{Z}}|\hat{\varphi}(\omega/2+\pi k)|^2|P((-1)^k z)|^2 \\ &= \hat{\varphi}(\omega/2)|P(z)|^2[\varphi,\varphi](\omega/2) \\ &\quad +\hat{\varphi}(\omega/2)|P(-z)|^2[\varphi,\varphi](\omega/2+\pi)\end{aligned}$$

nach Splitten der Summe in gerade und ungerade $k \in \mathbb{Z}$. Genauso

$$\begin{aligned}[\eta,\varphi](\omega)\hat{\varphi}(\omega) &= \hat{\varphi}(\omega/2)P(z)\sum_{k\in\mathbb{Z}}\hat{\eta}(\omega+2\pi k)\overline{\hat{\varphi}(\omega+2\pi k)} \\ &= \frac{1}{2}\hat{\varphi}(\omega/2)P(z)\sum_{k\in\mathbb{Z}}\hat{\varphi}(\omega/2+\pi k)\overline{\hat{\varphi}(\omega/2+\pi k)P(e^{-i(\omega+2\pi k)/2})} \\ &= \frac{1}{2}\hat{\varphi}(\omega/2)P(z)\sum_{k\in\mathbb{Z}}|\hat{\varphi}(\omega/2+\pi k)|^2\overline{P(z(-1)^k)} \\ &= \frac{1}{2}\hat{\varphi}(\omega/2)P(z)\Big([\varphi,\varphi](\omega/2)\overline{P(z)} + [\varphi,\varphi](\omega/2+\pi)\overline{P(-z)}\Big).\end{aligned}$$

Insgesamt ist das

$$\begin{aligned}2\hat{\psi}_1(\omega) &= \hat{\varphi}(\omega/2)[\varphi,\varphi](\omega/2+\pi)\Big(|P(-z)|^2 - P(z)\overline{P(-z)}\Big) \\ &= \hat{\varphi}(\omega/2)[\varphi,\varphi](\omega/2+\pi)\overline{P(-z)}z\underbrace{z^{-1}\left(P(-z)-P(z)\right)}_{=:A(\omega)}\end{aligned}$$

mit dem 2π–periodischen Teil

$$\begin{aligned}A(\omega) &= z^{-1}\left(P(-z)-P(z)\right) \\ &= e^{i\omega/2}\Big(P(-e^{-i\omega/2}) - P(e^{-i\omega/2})\Big) \\ A(\omega+2\pi) &= e^{i(\omega+2\pi)/2}\Big(P(-e^{-i(\omega+2\pi)/2}) - P(e^{-i(\omega+2\pi)/2})\Big) \\ &= (-z^{-1})\left(P(z)-P(-z)\right) \\ &= A(\omega).\end{aligned}$$

Wir dividieren diesen ab, weil er 2π–periodisch ist und vereinfachen unseren Ansatz zu

$$\hat{\psi}_2(\omega) := \hat{\varphi}(\omega/2)[\varphi,\varphi](\omega/2+\pi)z\overline{P(-z)}.$$

Der Anteil $[\varphi,\varphi](\omega/2+\pi)z\overline{P(-z)}$ ist wegen $z = e^{-i\omega/2}$ auf jeden Fall 4π–periodisch und hat deshalb eine Fourierreihe in $\omega/2$. Dann kann man schreiben

$$\hat{\psi}_2(\omega) = \hat{\varphi}(\omega/2)Q\left(e^{-i\omega/2}\right)$$

mit einer formalen Laurentreihe

$$Q(z) := \frac{1}{2}\sum_{k\in\mathbb{Z}} q_k z^k = [\varphi,\varphi](\omega/2+\pi)z\overline{P(-z)}. \tag{8.19}$$

Das liefert die Existenz einer Verfeinerungsleichung (8.11) und wir haben ψ_2 als Kandidaten für ein wavelet. Diese Konstruktionstechnik führt zum gewünschten Ergebnis: man bekommt ein wavelet, das ein Generator von W_0 ist. Sind insbesondere die Translate von φ orthogonal, so folgt sofort, daß

$$Q(z) = z\overline{P(-z)}$$

eine gute Wahl ist. Das sehen wir uns später genauer an.

Wir haben die Bedingung (8.14) für (8.19) nachzuprüfen. Dazu brauchen wir $Q(-z)$, und wenn $z = e^{-i\omega/2}$ gilt, folgt $-z = e^{-i(\omega/2+\pi)}$ und

$$Q(-z) = -[\varphi,\varphi](\omega/2)z\overline{P(z)}.$$

Damit folgt

$$\begin{aligned}
& [\varphi,\varphi]\left(\tfrac{\omega}{2}\right)Q(z)\overline{P(z)} + [\varphi,\varphi]\left(\tfrac{\omega}{2}+\pi\right)Q(-z)\overline{P(-z)} \\
= \;& [\varphi,\varphi]\left(\tfrac{\omega}{2}\right)[\varphi,\varphi](\omega/2+\pi)z\overline{P(-z)P(z)} \\
& -[\varphi,\varphi]\left(\tfrac{\omega}{2}+\pi\right)[\varphi,\varphi](\omega/2)z\overline{P(z)P(-z)} \\
= \;& 0.
\end{aligned}$$

Die Determinante (8.35) wird mit (8.6) zu

$$\begin{aligned}
& P(z)Q(-z) - Q(z)P(-z) \\
= \;& -P(z)[\varphi,\varphi](\omega/2)z\overline{P(z)} - [\varphi,\varphi](\omega/2+\pi)z\overline{P(-z)}P(-z) \\
= \;& -z[\varphi,\varphi](\omega) \neq 0.
\end{aligned}$$

Insgesamt haben wir also

Theorem 8.20. *Die Translate von ψ_2 sind orthogonal zu denen von φ, und sie spannen zusammen mit diesen den Raum V_1 auf.* □

Deshalb leistet ψ_2 das Verlangte, hat aber nicht notwendig orthogonale Translate, ebensowenig wie φ. □

Immerhin gilt

Theorem 8.21. *Hat φ stabile shifts, so auch ψ_2. Hat φ orthogonale Translate, so auch ψ_2.*

Beweis: Wir sehen uns das Klammerprodukt von ψ_2 an und bekommen

$$\begin{aligned}
[\psi_2,\psi_2](\omega) &= \sum_{k\in\mathbb{Z}} |\hat{\psi}_2(\omega+2\pi k)|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi k)|^2 [\varphi,\varphi]^2(\omega/2+\pi k+\pi)|P(-e^{-i(\omega+2\pi k)/2})|^2 \\
&= [\varphi,\varphi]^2(\omega/2+\pi)[\varphi,\varphi](\omega/2)|P(-z)|^2 \\
&\quad +[\varphi,\varphi]^2(\omega/2)[\varphi,\varphi](\omega/2+\pi)|P(z)|^2 \\
&= [\varphi,\varphi](\omega/2+\pi)[\varphi,\varphi](\omega/2)\cdot \\
&\quad \cdot([\varphi,\varphi](\omega/2+\pi)|P(-z)|^2+[\varphi,\varphi](\omega/2)|P(z)|^2) \\
&= [\varphi,\varphi](\omega/2+\pi)[\varphi,\varphi](\omega/2)[\varphi,\varphi](\omega).\square
\end{aligned}$$

8.4 B–Spline wavelets

Wir gehen noch einmal auf die verfeinerbaren B–Splines φ_n aus dem Text über translationsinvariante Räume zurück, und wir wissen auch schon, dass wir zugehörige wavelets nicht so einfach wie im orthogonalen Fall berechnen können.

Wir hatten schon das Klammerprodukt ausgerechnet als

$$\begin{aligned}
[\varphi_n,\varphi_n](\omega) &= \sum_{m\in Z} |\hat{\varphi}_n(\omega+2\pi m)|^2 \\
&= \sum_{m\in Z} |\hat{\varphi}_{n-1}(\omega+2\pi m)|^2|\hat{\varphi}_1(\omega+2\pi m)|^2 \\
&= \sum_{m\in Z} |\hat{\varphi}_1(\omega+2\pi m)|^{2n} \\
&= (2\pi)^{-n}\sin^{2n}(\omega/2)\sum_{m\in Z}(\omega/2+\pi m)^{-2n} \\
&= (2\pi)^{-n}\frac{\sin^{2n}(\omega/2)}{(\omega/2)^{2n}}\left(1+(\omega/2)^{2n}\sum_{m\in Z\setminus\{0\}}(\omega/2+\pi m)^{-2n}\right)
\end{aligned}$$

mit der üblichen Vorsicht bei Null, und diese Funktion ist strikt positiv, beschränkt, 2π–periodisch und unendlich oft differenzierbar. Man kann sie also in das obige Kalkül einsetzen und dann dazu ein wavelet ausrechnen. Leider bekommt es eine infinite Maske, die aber immerhin exponentiell abfällt. Details lassen wir aber hier weg.

8.5 Orthogonale Wavelets

Wir wollen uns das Leben etwas leichter machen und rechnen ab sofort nur noch mit den Maskenkoeffizienten q_k von Q und der Gleichung

$$\hat{\psi}(\omega) = \hat{\varphi}\left(\frac{\omega}{2}\right) Q\left(e^{-i\omega/2}\right).$$

Zuerst wollen wir die q_k so bestimmen, daß die Translate von ψ zu denen von φ orthogonal sind.

Theorem 8.22. *Die Translate von ψ sind genau dann zu denen von φ orthogonal, wenn gilt*

$$P(z)\overline{Q(z)}[\varphi,\varphi](\omega/2) + P(-z)\overline{Q(-z)}[\varphi,\varphi](\omega/2+\pi) = 0.$$

Beweis: Wir wissen schon, daß $[\varphi,\psi] = 0$ genau dann gilt, wenn ψ zu allen Translaten von φ orthogonal ist. Also rechnen wir das mal etwas genauer aus:

$$\begin{aligned}
0 &= [\varphi,\psi](\omega) \\
&= \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi k)\overline{\hat{\psi}(\omega+2\pi k)} \\
&= \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi k)P(e^{-i(\omega/2+\pi k)})\overline{\hat{\varphi}(\omega/2+\pi k)Q(e^{-i(\omega/2+\pi k)})} \\
&= \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi 2k)P(e^{-i(\omega/2+\pi 2k)})\overline{\hat{\varphi}(\omega/2+\pi 2k)Q(e^{-i(\omega/2+\pi 2k)})} \\
&\quad + \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi 2k+\pi)P(e^{-i(\omega/2+\pi 2k+\pi)})\overline{\hat{\varphi}(\omega/2+\pi 2k+\pi)Q(e^{-i(\omega/2+\pi 2k+\pi)})} \\
&= P(e^{-i\omega/2})\overline{Q(e^{-i\omega/2})} \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi 2k)\overline{\hat{\varphi}(\omega/2+\pi 2k)} \\
&\quad + P(e^{-i(\omega/2+\pi)})\overline{Q(e^{-i(\omega/2+\pi)})} \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi 2k+\pi)\overline{\hat{\varphi}(\omega/2+\pi 2k+\pi)} \\
&= P(e^{-i\omega/2})\overline{Q(e^{-i\omega/2})}[\varphi,\varphi](\omega/2) + P(e^{-i(\omega/2+\pi)})\overline{Q(e^{-i(\omega/2+\pi)})}[\varphi,\varphi](\omega/2+\pi) \\
&= P(z)\overline{Q(z)}[\varphi,\varphi](\omega/2) + P(-z)\overline{Q(-z)}[\varphi,\varphi](\omega/2+\pi).
\end{aligned}$$

□

Wenn die Translate von φ orthogonal sind, hat man die Gleichung

$$\begin{aligned} 0 &= P(z)\overline{Q(z)} + P(-z)\overline{Q(-z)} \\ &= \sum_{k\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} p_k q_m z^{k-m} + \sum_{k\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} p_k q_m z^{k-m}(-1)^{k+m} \\ &= \sum_{n\in\mathbb{Z}} z^n \left(\sum_{k\in\mathbb{Z}} p_k q_{k-n} + (-1)^n \sum_{k\in\mathbb{Z}} p_k q_{k-n} \right) \end{aligned}$$

und durch Koeffizientenvergleich

$$0 = \sum_{k\in\mathbb{Z}} p_k q_{k-2j},\ j \in \mathbb{Z}. \tag{8.23}$$

Wenn wir zusätzlich die Translate von ψ orthonormal haben wollen, muss $[\psi,\psi]$ konstant gleich $1/2\pi$ sein. Das bedeutet

$$[\psi,\psi](\omega) = |Q(z)|^2[\varphi,\varphi](\omega/2) + |Q(-z)|^2[\varphi,\varphi](\omega/2+\pi) = \frac{1}{2\pi}.$$

Wenn wir wieder Orthonormalität der Translate von φ voraussetzen, folgt daraus

$$1 = |Q(z)|^2 + |Q(-z)|^2,$$

und wir wissen schon, dass dann

$$2\delta_{j0} = \sum_{k\in\mathbb{Z}} q_k q_{k-2j},\ j \in \mathbb{Z}$$

folgt.

Theorem 8.24. *Wenn φ orthonormale Translate hat, und wenn man ψ über (8.11) definiert, so folgt aus den simultanen Gleichungen*

$$\begin{aligned} 1 &= |P(z)|^2 + |P(-z)|^2 \\ 1 &= |Q(z)|^2 + |Q(-z)|^2 \\ 0 &= P(z)\overline{Q(z)} + P(-z)\overline{Q(-z)} \end{aligned} \tag{8.25}$$

dass auch ψ orthonormale Translate hat, die auf denen von φ senkrecht stehen. Ferner wird der gesamte Raum V_1, der nach Definition von den Translaten von $\varphi(2\cdot)$ aufgespannt wird, schon von den Translaten von φ und ψ aufgespannt.

Zu gegebenem P ist

$$Q(z) := -z\overline{P(-z)}$$

eine Lösung dieser Gleichungen. Man kann die obigen Aussagen auch durch die Koeffizienten als

$$\begin{aligned} 2\delta_{j0} &= \sum_{k\in\mathbb{Z}} p_k p_{k-2j},\ j\in\mathbb{Z} \\ 2\delta_{j0} &= \sum_{k\in\mathbb{Z}} q_k q_{k-2j},\ j\in\mathbb{Z} \\ 0 &= \sum_{k\in\mathbb{Z}} p_k q_{k-2j},\ j\in\mathbb{Z} \\ q_k &= (-1)^k p_{1-k},\ k\in\mathbb{Z} \end{aligned}$$

ausdrücken.

Beweis: Man rechnet leicht nach, daß $Q(z) := -z\overline{P(-z)}$ die Gleichungen erfüllt, und das bedeutet

$$q_k := (-1)^k p_{1-k},\ k\in\mathbb{Z},$$

denn es gilt

$$\begin{aligned} Q(z) &= -z\sum_{k\in\mathbb{Z}} p_k(-\overline{z})^k \\ &= -\sum_{k\in\mathbb{Z}} p_k(-1)^k z^{-k+1} \\ &= -\sum_{n\in\mathbb{Z}} p_{1-n}(-1)^{1-n} z^n \text{ (mit } n=-k+1) \\ &= \sum_{n\in\mathbb{Z}} p_{1-n}(-1)^n z^n. \end{aligned}$$

Man kann dann (8.23) auch direkt ausrechnen:

$$\begin{aligned} \alpha_j &:= \sum_{k\in\mathbb{Z}} p_k(-1)^{k-2j} p_{1-(k-2j)} \\ &= \sum_{k\in\mathbb{Z}} p_k(-1)^k p_{2j+1-k} \\ &= \sum_{k\in\mathbb{Z}} p_k(-1)^k p_{2j+1-k} \\ &= \sum_{m\in\mathbb{Z}} p_{2j+1-m}(-1)^{2j+1-m} p_m \\ &= -\sum_{m\in\mathbb{Z}} p_m(-1)^m p_{2j+1-m} \\ &= -\alpha_j, \text{ also} \\ \alpha_j &= 0 \end{aligned}$$

für alle $j\in\mathbb{Z}$.

Wir prüfen im orthogonalen Fall noch nach, ob sich V_1 aus den Translaten von φ und ψ komplett aufspannen läßt. Dazu wollen wir die Funktionen

$f_\ell(x) := \varphi(2x-\ell)$ für $\ell \in \mathbb{Z}$ auf den Span der Translate von φ und ψ projizieren und dann nachweisen, daß das Ergebnis g_ℓ mit f_ℓ übereinstimmt. Wir haben

$$g_\ell(x) := \sum_{m\in\mathbb{Z}} (f_\ell, \varphi(\cdot - m))_2 \varphi(x-m) + \sum_{m\in\mathbb{Z}} (f_\ell, \psi(\cdot - m))_2 \psi(x-m).$$

Wir benutzen unsere Verfeinerungsgleichungen und die Skalarprodukte in der Form

$$\begin{aligned}
\varphi &= \sum_{k\in\mathbb{Z}} p_k f_k \\
\psi &= \sum_{k\in\mathbb{Z}} q_k f_k \\
(f_k, f_\ell)_2 &= \int_{\mathbb{R}} \varphi(2x-k)\varphi(2x-\ell)dx \\
&= \frac{1}{2}\int_{\mathbb{R}} \varphi(y-k)\varphi(y-\ell)d4 \\
&= \frac{1}{2}\delta_{k\ell} \\
f_k(x-m) &= \varphi(2(x-m)-k) \\
&= \varphi(2x-2m+k) \\
&= f_{2m+k}(x)
\end{aligned}$$

.

Das ergibt

$$\begin{aligned}
(f_\ell, \varphi(\cdot-m))_2 &= (f_\ell, \sum_{k\in\mathbb{Z}} p_k f_k(\cdot - m))_2 \\
&= \sum_{k\in\mathbb{Z}} p_k (f_\ell, f_{2m+k})_2 \\
&= \frac{1}{2} p_{2m-\ell}, \\
(f_\ell, \psi(\cdot-m))_2 &= (f_\ell, \sum_{k\in\mathbb{Z}} q_k f_k(\cdot - m))_2 \\
&= \sum_{k\in\mathbb{Z}} q_k (f_\ell, f_{2m+k})_2 \\
&= \frac{1}{2} q_{2m-\ell}
\end{aligned}$$

und insgesamt

$$g_\ell(x) := \frac{1}{2}\sum_{m\in\mathbb{Z}} p_{2m-\ell}\varphi(x-m) + \frac{1}{2}\sum_{m\in\mathbb{Z}} q_{2m-\ell}\psi(x-m).$$

Wir benutzen jetzt, daß g_ℓ die Orthogonalprojektion von f_ℓ auf $V_0 + W_0$ ist. Deshalb steht $f_\ell - g_\ell =: h_\ell$ auf g_ℓ senkrecht, und es folgt nach dem Satz des Pythagoras

$$\|h_\ell\|_2^2 = \|f_\ell\|_2^2 - \|g_\ell\|_2^2 = \frac{1}{2} - \|g_\ell\|_2^2.$$

Wir müssen noch zeigen, daß $\|g_\ell\|_2^2 = \frac{1}{2}$ gilt. Dazu benutzen wir die Parsevalsche Gleichung in der Form

$$\begin{aligned} 4\|g_\ell\|_2^2 &= \sum_{m\in\mathbb{Z}} (p_{2m-\ell}^2 + q_{2m-\ell}^2) \\ &= \sum_{m\in\mathbb{Z}} (p_{2m-\ell}^2 + p_{1-2m+\ell}^2) \\ &= \sum_{k\in\mathbb{Z}} p_k^2. \end{aligned}$$

Aus der Verfeinerungsgleichung, gesehen als eine Projektion in V_1 mit Koeffizienten $p_k/\sqrt{2}$ und einer Orthonormalbasis $\sqrt{2}\varphi(2\cdot -k)$, folgt aber auch

$$1 = \sum_{k\in\mathbb{Z}} \frac{p_k^2}{2}$$

und das ergibt die Behauptung. □

8.6 Skalierungsfunktionen aus Masken

Dieser und die nächsten Abschnitte einschließlich der Bilder können übersprungen wereden, wenn man sich nur für die Theorie interessiert. Hier ist etwas auszurechnen.

Gegeben sei eine **endliche** Maske $\{p_k\}_k$ mit der man eine Verfeinerungsgleichung

$$\varphi(x) := \sum_k p_k \varphi(2x-k)$$

aufstellen und lösen will. Das macht man durch ein iteratives Verfahren, bei dem die Funktion φ auf immer feineren Gittern ausgerechnet wird.

Man interpretiert die Gleichung als ein Upsampling, indem man $x = 2^{-(m+1)}\ell$ einsetzt:

$$\begin{aligned} \varphi(2^{-(m+1)}\ell) &= \sum_k p_k \varphi(2\cdot 2^{-(m+1)}\ell - k) \\ &= \sum_k p_k \varphi(2^{-m}\ell - k) \\ &= \sum_k p_k \varphi(2^{-m}(\ell - 2^m k). \end{aligned}$$

Das iteriert man, indem man setzt

$$\begin{aligned} c_\ell^{(m+1)} &:= \varphi(2^{-(m+1)}\ell) \\ &= \sum_k p_k \varphi(2 \cdot 2^{-(m+1)}\ell - k) \\ &= \sum_k p_k \varphi(2^{-m}\ell - k) \\ &= \sum_k p_k \underbrace{\varphi(2^{-m}(\ell - 2^m k))}_{=: c_{\ell - 2^m k}^{(m)}} \\ &= \sum_k p_k c_{\ell - 2^m k}^{(m)}. \end{aligned}$$

Dieses Verfahren erlaubt das Ausrechnen neuer Werte auf einem Gitter mit Punktabstand $2^{-(m+1)}$, wenn Werte auf einem halb so feinen Gitter vorliegen. Man startet mit $x = 0$, wo in dem man den Wert $1 = \varphi(0)$ annimmt, und dann rechnet man die anderen Werte einfach aus. Insofern sind die Ergebnisse immer korrekt, wenn auch manchmal überraschend. Weiter unten folgen Bilder und ein MATLAB–Programm.

So weit, so gut, aber das kann man die obige Gleichung so nicht in MATLAB programmieren. Zuerst behandeln wir die Masken. Sie seien mathematisch als p_k mit $k^- \le k \le k^+$ beschrieben, wobei k^- durchaus negativ sein kann. In MATLAB nimmt man dann einen Vektor mit Komponenten P_j mit den Indizes $1 \le j \le k^+ - k^- + 1$ und definiert $P_j = p_{k^- + j - 1}$ oder $p_k = P_{k - k^- + 1}$.

Jetzt die Indizierung der c-Vektoren. Wir überlegen uns das erst einmal mathematisch, dann MATLABig. Der Start sei so, daß wir mit $m = 0$ und $c_k^{(0)} = \delta_{0k}$ anfangen. Der Laufindex ℓ geht also von $L_0^- := 1$ bis $L_0^+ := 1$, wobei die restlichen $c_k^{(0)}$ eben Null sind.

Induktiv seien die $c_\ell^{(m)}$ nur ungleich Null, wenn $L_m^- \le \ell \le L_m^+$ gilt. Wann ist dann $c_\ell^{(m+1)} = 0$? Nach der obigen Gleichung sicher dann, wenn

$$\begin{aligned} \ell - 2^m k^- &< L_m^- \\ \ell - 2^m k^+ &> L_m^+ \end{aligned}$$

gilt. Man braucht also nur die ℓ mit

$$L_m^- + 2^m k^- \le \ell \le L_m^+ + 2^m k^+$$

auszurechnen, d.h. man setzt

$$L_{m+1}^- := L_m^- + 2^m k^-, \quad L_{m+1}^+ := L_m^+ + 2^m k^+.$$

Die Gesamtzahl der Komponenten im Schritt m ist $L_m^+ - L_m^- + 1$ mit der Rekursion

$$\begin{aligned} L_{m+1}^+ - L_{m+1}^- + 1 &= L_m^+ + 2^m k^+ - (L_m^- + 2^m k^-) + 1 \\ &= L_m^+ - L_m^- + 1 + 2^m (k^+ - k^-). \end{aligned}$$

Der Wert $L_m^+ - L_m^- + 1$ ist also genau die obere Grenze der Rechnung in MATLAB auf Stufe m mit einem MATLAB-Feld $C^{(m)}$. Die Indexumrechnung ist dann

$$C_i^{(m)} = c^{(m)}_{L_m^- + i - 1},\ 1 \le i \le L_m^+ - L_m^- + 1,$$
$$c_r^{(m)} = C^{(m)}_{r - L_m^- + 1} =,\ L_m^- \le r \le L_m^+.$$

Die Indexumrechnung der linken Seite ist dieselbe, aber mit $m+1$ anstelle von m. Es folgt

$$\begin{aligned} c_\ell^{(m+1)} &= \sum_k p_k c^{(m)}_{\ell - 2^m k} \\ C^{(m+1)}_{\ell - L_{m+1}^- + 1} &= \sum_{k=k^-}^{k^+} P_{k-k^-+1} C^{(m)}_{\ell - 2^m k - L_m^- + 1} \\ C_j^{(m+1)} &= \sum_{s=1}^{k^+ - k^- + 1} P_s C^{(m)}_{j + L_{m+1}^- - 1 - 2^m (s + k^- - 1) - L_m^- + 1} \\ &= \sum_{s=1}^{k^+ - k^- + 1} P_s C^{(m)}_{j - 2^m (s-1)} \end{aligned}$$

mit Summationstransformationen $k = s + k^- - 1$ und $\ell = j + L_{m+1}^- - 1$ wegen

$$\begin{aligned} & j + L_{m+1}^- - 1 - 2^m (s + k^- - 1) - L_m^- + 1 \\ = \ & j + L_{m+1}^- - 2^m (s + k^- - 1) - L_m^- \\ = \ & j + L_m^- + 2^m k^- - 2^m (s + k^- - 1) - L_m^- \\ = \ & j - 2^m (s - 1). \end{aligned}$$

Mit der Formel

$$C_j^{(m+1)} = \sum_{s=1}^{k^+ - k^- + 1} P_s C^{(m)}_{j - 2^m (s-1)},\ 1 \le j \le L_{m+1}^+ - L_{m+1}^- + 1$$

kann man dann in MATLAB arbeiten, aber man muss aufpassen, bei der Programmierung in den Indizes von $C^{(m)}$ keine Bereichsüberschreitung zu bekommen. Das geschieht, indem man die entsprechenden Terme wegläßt, denn sie sind ohnehin Null.

Die zu den $c_\ell^{(m+1)}$ gehörigen Werte sind als $\varphi(2^{-(m+1)}\ell)$ zu verstehen. Das bedeutet, dass wir φ näherungsweise auf den Punkten

$$2^{-(m+1)} L_{m+1}^- \le x \le 2^{-(m+1)} L_{m+1}^+$$

ausgerechnet haben, und ansonsten ist φ gleich Null. Per Induktion findet man aber

$$\begin{aligned} L^+_{m+1} &= L^+_m + 2^m k^+ \\ &= L^+_{m-1} + 2^m k^+ + 2^{m-1} k^+ \\ &= L^+_0 + 2^m k^+ + 2^{m-1} k^+ \ldots + 2k^+ + k^+ \\ &= 1 + k^+ \frac{2^{m+1}-1}{2-1} \\ &= 1 + k^+ (2^{m+1} - 1) \\ L^-_{m+1} &= 1 + k^- (2^{m+1} - 1) \end{aligned}$$

und deshalb

$$\begin{array}{rcccl} 2^{-(m+1)}(1 + k^-(2^{m+1}-1)) & \leq & x & \leq & 2^{-(m+1)}(1 + k^+(2^{m+1}-1)) \\ 2^{-(m+1)} + k^-(1 - 2^{-(m+1)}) & \leq & x & \leq & 2^{-(m+1)} + k^+(1 - 2^{-(m+1)}) \\ k^- + 2^{-(m+1)}(1 - k^-) & \leq & x & \leq & k^+ + 2^{-(m+1)}(1 - k^+) \end{array}$$

mi der Schrittweite $2^{-(m+1)}$. Es entsteht also ein Gebilde, dessen Träger im Limes das Intervall $[k^-, k^+]$ ist.

Man kann die Berechnung der Laufgrenzen rekursiv vereinfachen. Mit

$$\begin{aligned} x^-_{m+1} &:= k^- + 2^{-(m+1)}(1 - k^-) \\ x^+_{m+1} &:= k^+ + 2^{-(m+1)}(1 - k^+) \end{aligned}$$

folgt

$$\begin{aligned} x^\pm_{m+1} - k^\pm &= 2^{-(m+1)}(1 - k^\pm) \\ &= \tfrac{1}{2} 2^{-m}(1 - k^\pm) \\ &= \tfrac{1}{2}(x^\pm_m - k^\pm) \\ x^\pm_{m+1} &= \tfrac{1}{2}(x^\pm_m + k^\pm). \end{aligned}$$

Man startet die Rekursion mit $x^-_0 = x^+_0 = 0$, aber für $m = 0$ plottet man nicht.

8.7 Wavelets aus Masken

Gegeben sei eine Maske $\{p_k\}_k$ wie oben, und dazu die Maske $\{q_k\}_k$ mit der man das wavelet ψ als

$$\psi(x) := \sum_k q_k \varphi(2x - k)$$

berechnen will. Das kann man näherungsweise durch einen einzigen weiteren Schritt des obigen Verfahrens machen, wobei man nur klammheimlich die

Maske ändert. Die im orthogonalen Falle übliche Maske ist (bis auf das Vorzeichen)

$$q_k := (-1)^{-k-1}\overline{p_{-k-1}}$$

und sie hat im reellen Fall die Form

$$(-1)^{-k^+-1}p_{-k^+-1}, \ldots, (-1)^{-k^--1}p_{-k^--1}.$$

Das werden wir weiter unten vorrechnen. Die neuen Indexgrenzen n^+ und n^- sind also

$$n^- := -k^+ - 1,\ n^+ := -k^- - 1.$$

Sie übernehmen die Rolle von k^- und k^+.

Jetzt funktioniert alles genau wie bisher, er wird lediglich mit einer neuen Maske und anderen Indexgrenzen gearbeitet. Der Definitionsbereich wird mit der Formel

$$x^{\pm}_{neu} = \frac{1}{2}(x^{\pm}_{alt} + n^{\pm})$$

angepaßt.

Hier ist ein Programm dazu.

```
% Programm zum Berechnen von Skalierungsfunktionen
% und wavelets aus endlichen Masken.
% Siehe den obigen Text.
clear all;
% Hier werden Maske und Definitionsbereich angegeben.
% Wenn die Maske N Terme hat, sollte kplus-kminus=N-1 gelten.
% Die Summe der Maskenkoeffizienten sollte 2 sein.
wavcase=7;
switch wavcase
    case 1    %% Haar
        kminus=0;
        kplus=1;
        p=[1 1];
    case 2    %%% ?????
        p=[1/3 2/3 2/3 1/3];
        kminus=-2;
        kplus=1;
    case 3    % Daubechies N=2
        p=[(1+sqrt(3))/4 (3+sqrt(3))/4 ...
            (3-sqrt(3))/4  (1-sqrt(3))/4  ]% /sqrt(2)
```

```
        kminus=0;
        kplus=3;
    case 4   % Daubechies N=3
        p=[0.4704672080 1.141116916 .6503650005 ...
            -.190934416  -.1208322083  0.049817499];
        kminus=0;
        kplus=5;
    case 5
        p=[1 4 6 4 1]/8;
        kminus=-2;
        kplus=2;     %% kubischer Spline
    case 6
        p=[1/16 1 15/16];
        kminus=-1;
        kplus=1;                        %%  ????
    case 7
        p=[1 0 2 6 2 0 1 ]/6;
        kminus=-3;
        kplus=3;
    case 8
        p=[1 21 5 0 15 1 1]/32;     %% ?????
        kminus=-3;
        kplus=3;
    case 9
        p=[1 6 15 20 15 6 1]/32;  % B-Spline 5. Grades
        kminus=-3;
        kplus=3;
    otherwise       %% Hut
        p=[1/2 1 1/2];
        kminus=-1;
        kplus=1;
end
m=0;
c=ones(1,1);
zm=1;  % 2 hoch m
oldupper=1;
xmin=0;
xmax=0;
subplot(4,1,1)
plot(kminus:kplus,p,'*')
title('Maske')
```

```
for m=1:12
    zm2=2*zm; % 2 hoch m, aber hier gilt das NEUE m schon,
              % d.h. m+1 in der Vorlesung
    newupper=1+(zm2-1)*(kplus-kminus);
    cnew=zeros(1,newupper);
    for s=1:newupper
        for i=1:kplus-kminus+1
            if s+zm*(1-i)<=0
                break;
            end
            if s+zm*(1-i)<=oldupper
                cnew(1,s)=cnew(1,s)+p(1,i)*c(1,s+zm*(1-i));
            end
        end
    end
    xmin=(xmin+kminus)/2;
    xmax=(xmax+kplus)/2;
    xnew=xmin:1/zm2:xmax;
    c=cnew;
    oldupper=newupper;
    zm=zm2;
end
subplot(4,1,2)
plot(xnew,cnew);
title('Skalierungsfunktion')
% jetzt das wavelet

q=-p(length(p):-1:1).*(-1).^(1:length(p))
qminus=-kplus-1;
qplus=-kminus-1;
subplot(4,1,3)
plot(qminus:qplus,q,'*')
title('Maske')

% Wie gut, wenn man abschreiben kann! Also:

zm2=2*zm; % 2 hoch m, aber hier gilt das NEUE m schon,
          % d.h. m+1 in der Vorlesung
newupper=1+(zm2-1)*(qplus-qminus);
dnew=zeros(1,newupper);
for s=1:newupper
```

```
    for i=1:qplus-qminus+1
        if s+zm*(1-i)<=0
            break;
        end
        if s+zm*(1-i)<=oldupper
            dnew(1,s)=dnew(1,s)+q(1,i)*c(1,s+zm*(1-i));
        end
    end
end
subplot(4,1,4)
xmin=(xmin+qminus)/2;
xmax=(xmax+qplus)/2;
xnew=xmin:1/zm2:xmax;

plot(xnew,dnew);
title('Wavelet dazu');
```

Man kann Skalierungsfunktionen und wavelets aus B–Splines machen. Die Maske besteht bei B–Splines der Ordnung n aus den n Binomialkoeffizienten mit Renormierung auf Gesamtsumme 2, wie wir schon wissen, aber die Maskenkoeffizienten des wavelets sind nicht über die Formel $q_k = (-1)^k p_{1-k}$ gegeben, weil man keine Orthogonalität der Translate hat. Abbildung 23 zeigt den kubischen Fall, aber beim wavelet haben wir gemogelt, weil wir die Formel fest einprogrammiert haben.

Ein orthogonales wavelet vom Daubechies–Typ ist in Abbildung 24 zu sehen.

Wählt man irgendwelche wilden Masken, so bekommt man oft fraktale Gebilde, siehe Abbildung 25.

8.8 Die wavelets von Ingrid Daubechies

Die Gleichungen (8.25) enthalten im orthogonalen Fall lediglich Bedingungen an P, weil man Q immer durch $Q(z) = z\overline{P(-z)}$ ausrechnen kann. Gesucht sind aber "gute" P mit endlichen Masken. Dazu gibt es eine mathematisch sehr originelle Konstruktion[47] von Ingrid Daubechies.

[47] `http://de.wikipedia.org/wiki/Daubechies-Wavelets`

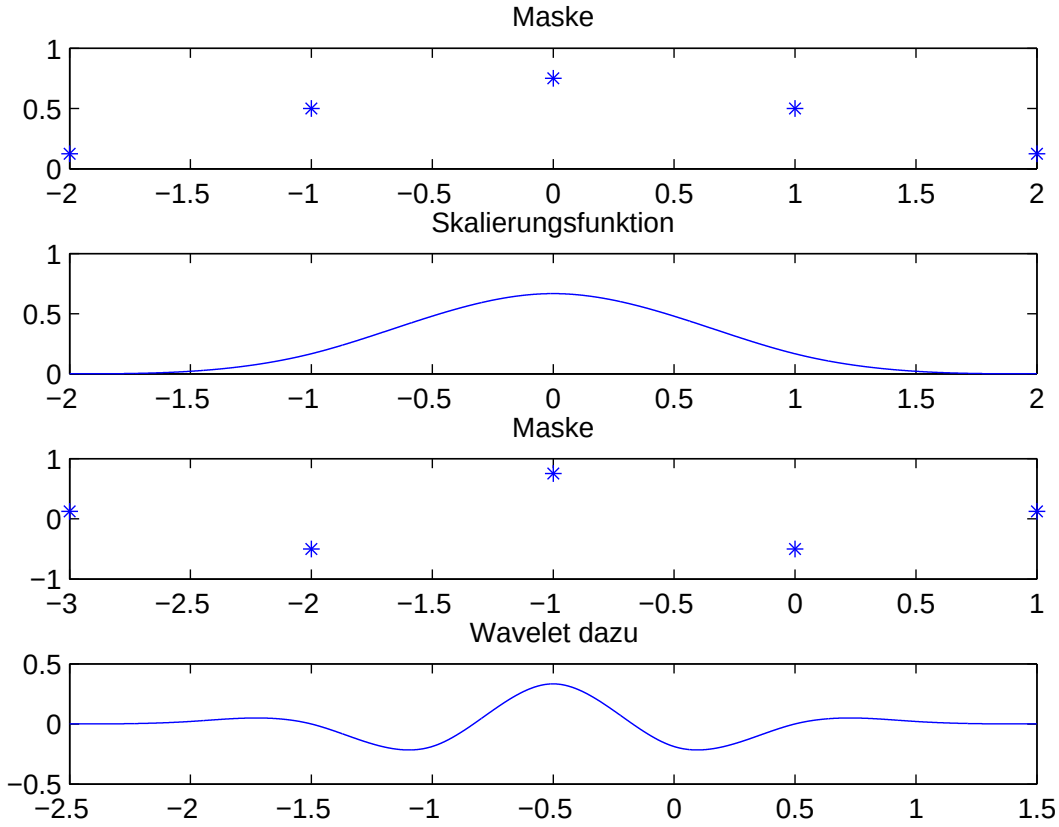

Figure 23: Kubisches B–Spline wavelet

Aus (8.3) folgte $P(1) = 1$ und damit auch (8.5). Wenn wir Orthogonalität haben wollen, muss (8.8) gelten, und es folgt auch

$$P(-1) = 0, \text{ d.h. } \sum_{k\in\mathbb{Z}} p_k(-1)^k = 0 = Q(1).$$

Entscheidend ist nun, daß die Ordnung der Nullstelle von P in -1 die Glätte der verfeinerbaren Funktion und ihre Approximationseigenschaften bestimmt. Letzteres wissen wir aus dem Abschnitt über die Strang–Fix–Bedingungen, aber die Glätte der verfeinerbaren Funktion in Abhängigkeit von Eigenschaften ihrer Maske untersuchen wir hier nicht.

Man macht also den Ansatz

$$P(z) \quad = \quad (1+z)^n R_1(z)$$

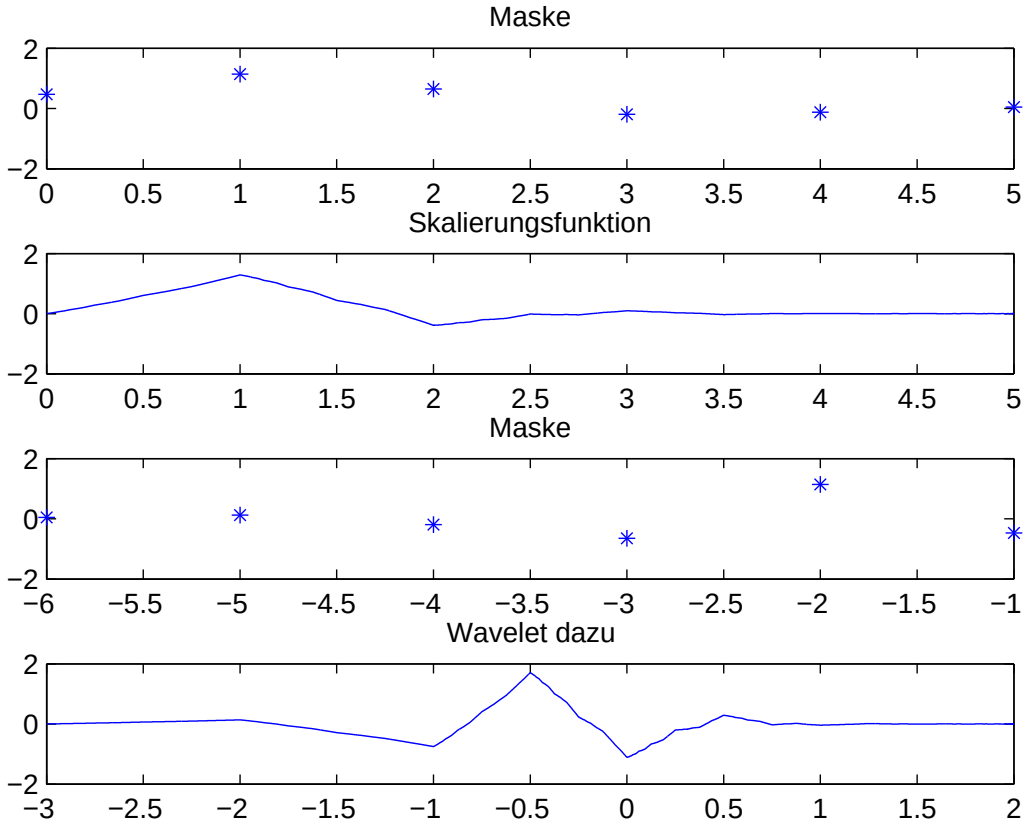

Figure 24: Daubechies wavelet

mit einem möglichst großen $n \in \mathbb{N}$, wobei man

$$R_1(z^2) = R_1(e^{-i\omega}) =: r(\omega)$$

als ein trigonometrisches Polynom r in ω mit reellen Koeffizienten ansetzt. Dann ist

$$|r(\omega)|^2 = r(\omega)\overline{r(\omega)} = r(\omega)r(-\omega)$$

ein gerades trigonometrisches Polynom und es folgt mit $\cos\alpha = 1-2\sin^2(\alpha/2)$ auch

$$\begin{aligned} |r(\omega)|^2 &= |R_1(z^2)|^2 \\ &= T(\cos\omega) \\ &= T(1-2\sin^2(\omega)/2)) \\ &=: R(\sin^2(\omega)/2)) \end{aligned}$$

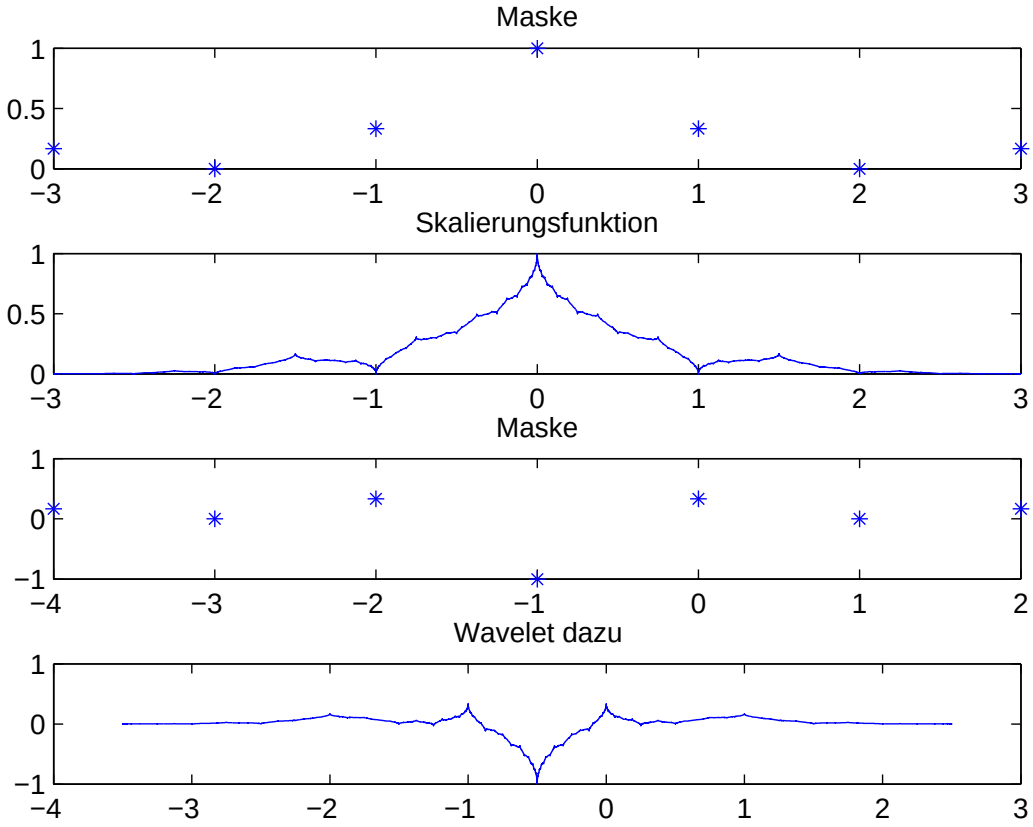

Figure 25: Irgendein fraktales wavelet

mit passenden algebraischen Polynomen T und R. Wir halten an dieser Stelle fest, daß R auf $[0,1]$ nichtnegativ sein muss.

Ferner gilt

$$\begin{aligned} R_1(-z^2) &= R_1(-e^{-i\omega}) \\ &= R_1(e^{-i(\omega+\pi)}) \\ &= r(\omega+\pi), \\ |R_1(-z^2)|^2 &= |r(\omega+\pi)|^2 \\ &= R(\sin^2(\omega+\pi)/2)) \\ &= R(\cos^2(\omega)/2)). \end{aligned}$$

Eine weitere simple Rechnung ist

$$\begin{aligned}
1 \pm z^2 &= 1 \pm e^{-i\omega} \\
\overline{1 \pm z^2} &= 1 \pm e^{+i\omega} \\
|(1 \pm z^2)|^2 &= 2 \pm (e^{+i\omega} + e^{-i\omega}) \\
&= 2 \pm 2\cos\omega \\
1 + \cos\alpha &= 2\cos^2(\alpha/2) \\
1 - \cos\alpha &= 2\sin^2(\alpha/2).
\end{aligned}$$

Deshalb bekommt man

$$\begin{aligned}
1 &= |P(z^2)|^2 + |P(-z^2)|^2 \\
&= |(1+z^2)^n R_1(z^2)|^2 + |(1-z^2)^n R_1(-z^2)|^2 \\
&= 4^n \cos^{2n}(\omega/2) R(\sin^2(\omega/2)) + 4^n \sin^{2n}(\omega/2) R(\cos^2(\omega/2))
\end{aligned}$$

und bei Setzung $t := \sin^2(\omega/2)$ ergibt sich schließlich die Gleichung

$$4^{-n} = (1-t)^n R(t) + t^n R(1-t) \tag{8.26}$$

für ein zu bestimmendes reelles algebraisches Polynom R, das auf $[0,1]$ nichtnegativ sein sollte. In der obigen Gleichung müssen sich also alle Terme bis auf den konstanten Term wegheben.

Wegen der Positivitätsforderung in $[0,1]$ setzt man R am besten in der Bernsteinbasis an, und zwar als

$$R(t) = \sum_{j=0}^{n-1} \rho_j \binom{n-1}{j} t^j (1-t)^{n-1-j}$$

mit hoffentlich positiven Koeffizienten ρ_j. Es folgt

$$\begin{aligned}
4^{-n} &= (1-t)^n \sum_{j=0}^{n-1} \rho_j \binom{n-1}{j} t^j (1-t)^{n-1-j} \\
&\quad + t^n \sum_{j=0}^{n-1} \rho_j \binom{n-1}{j} (1-t)^j t^{n-1-j} \\
&= \sum_{j=0}^{n-1} \rho_j \binom{n-1}{j} t^j (1-t)^{2n-1-j} \\
&\quad + \sum_{j=0}^{n-1} \rho_j \binom{n-1}{j} (1-t)^j t^{2n-1-j} \\
&= \sum_{k=0}^{n-1} \rho_k \binom{n-1}{k} t^k (1-t)^{2n-1-k} \\
&\quad + \sum_{k=n}^{2n-1} \rho_{2n-1-k} \binom{n-1}{2n-1-k} (1-t)^{2n-1-k} t^k
\end{aligned}$$

und man macht einen Koeffizientenvergleich in der Bernsteinbasis mit

$$\begin{aligned} 4^{-n} &= 4^{-n}(1-t+t)^{2n-1} \\ &= 4^{-n}\sum_{k=0}^{2n-1}\binom{2n-1}{k}t^k(1-t)^{2n-1-k}. \end{aligned}$$

Das erfordert

$$\begin{aligned} \rho_k &= 4^{-n}\frac{\binom{2n-1}{k}}{\binom{n-1}{k}},\ 0\le k\le n-1 \\ \rho_{2n-1-k} &= 4^{-n}\frac{\binom{2n-1}{k}}{\binom{n-1}{2n-1-k}},\ n\le k\le 2n-1 \end{aligned}$$

was leider alle Koeffizienten doppelt definiert. Wenn wir aber in der zweiten Gleichung $j := 2n-1-k$ setzen, folgt

$$\begin{aligned} \rho_j &= 4^{-n}\frac{\binom{2n-1}{2n-1-j}}{\binom{n-1}{j}},\ 0\le j\le n-1 \\ &= 4^{-n}\frac{\binom{2n-1}{j}}{\binom{n-1}{j}},\ 0\le j\le n-1 \end{aligned}$$

und die beiden Fälle stimmen überein! Deshalb können wir ein in $[0,1]$ strikt positives Polynom R vom Grade $n-1$ finden, das unseren Forderungen genügt.

Aber jetzt müssen wir zurückrudern. Die Gleichung (8.26) ist erfüllt, aber wir brauchen ein trigonometrisches Polynom r mit

$$|r(\omega)|^2 = R(\sin^2(\omega/2)). \tag{8.27}$$

Das ist mit einem "Wurzelziehen" aus einem positiven Polynom vergleichbar, und nach einem Satz von Féjer und Riesz geht das immer, wobei R nur nichtnegativ auf $[0,1]$ sein muß und r automatisch denselben Grad wie R hat. Allerdings ist das Lösen der obigen Gleichung unangenehm, weil man ein System quadratischer Gleichungen für die Koeffizienten von r bekommt, wenn die von R bekannt sind. Wenn man r hat, bekommt man R_1 und P, und damit auch Q.

Sehen wir uns einfache Fälle an. Für $n=1$ kann man (8.26) durch die Konstante $R=\frac{1}{4}$ lösen und (8.27) wird durch die Konstante $r=\frac{1}{2}=R_1$ erfüllt. Man bekommt

$$P(z) = (1+z)/2,\ \text{d.h. } p_0 = p_1 = 1$$

und damit die Haarsche Verfeinerungsfunktion sowie im weiteren Verlauf das Haarsche wavelet.

Jetzt untersuchen wir $n = 2$. Durch direktes Ansetzen der Gleichung (8.26) mit einer linearen Funktion bekommt man zunächst

$$\begin{aligned} \tfrac{1}{16} &= (1-t)^2(a+bt) + t^2(a+b(1-t)) \\ &= a + t(-2a+b) + t^2(2a-b) \end{aligned}$$

und daraus

$$R(t) = \frac{1}{16} + \frac{1}{8}t.$$

Dann muss man auch r als trigonometrisches Polynom vom Grade 1 ansetzen als

$$R_1(e^{-i\omega}) =: r_0 + r_1 e^{-i\omega} =: r(\omega)$$

mit reellen Koeffizienten. Jetzt bekommt (8.27) die Form

$$\begin{aligned} |r(\omega)|^2 &= r(\omega)\overline{r(\omega)} \\ &= (r_0 + r_1 e^{-i\omega})(\overline{r_0 + r_1 e^{-i\omega}}) \\ &= r_0^2 + r_1^2 + 2r_0 r_1 \cos(\omega) \\ &= R(\sin^2(\omega/2)) \\ &= R((1-\cos(\omega))/2) \\ &= \tfrac{1}{16} + \tfrac{1}{8}(1-\cos(\omega))/2 \\ &= \tfrac{1}{8} - \tfrac{1}{16}\cos(\omega) \end{aligned}$$

und somit hat man die quadratischen Gleichungen

$$\begin{aligned} r_0^2 + r_1^2 &= \tfrac{1}{8} \\ 2r_0 r_1 &= -\tfrac{1}{16}. \end{aligned}$$

Das ist der Schnitt eines Kreises mit einer Hyperbel, und man bekommt die Lösung

$$\begin{aligned} r_0 &= \frac{1+\sqrt{3}}{8}, \\ r_1 &= \frac{1-\sqrt{3}}{8}. \end{aligned}$$

Dann müssen wir noch $P(z) = (1+z)^2 R_1(z)$ ausrechnen. Das ist

$$\begin{aligned} P(z) &= (1+z)^2 R_1(z) \\ &= (1+2z+z^2)(r_0 + r_1 z) \\ &= r_0 + z(2r_0 + r_1) + z^2(r_0 + 2r_1) + z^3 r_1 \end{aligned}$$

und schließlich ergeben sich die Maskenkoeffizienten

$$\begin{aligned}\tfrac{1}{2}(p_0,\ldots,p_3) &= (r_0, 2r_0+r_1, r_0+2r_1, r_1)\\ &= \tfrac{1}{8}(1+\sqrt{3}, 3+\sqrt{3}, 3-\sqrt{3}, 1-\sqrt{3}).\end{aligned}$$

Es resultiert eine verfeinerbare Funktion mit kompaktem Träger in $[0,3]$, und diese können wir mit unserem Programm leicht ausrechnen. Das zugehörige wavelet hat dann einen kompakten Träger in $[-2,1]$, wie wir uns im Umfeld unseres Programms überlegt haben.

Man kann sich vorstellen, dass größere n ziemlich unangenehm werden, weil man damit rechnen muß, n quadratische Gleichungen in n Unbekannten zu lösen. Das kann man aber mit entsprechendem numerischem Aufwand sehr genau erledigen, und aus der Theorie weiß man die Lösbarkeit.

Die obigen orthogonalen wavelets und Skalierungsfunktionen sehen außer im Haarschen Fall immer eigenartig unsymmetrisch aus. Das ist nicht anders machbar, denn man kann beweisen, daß bei endlichen Masken und voller Orthonormalität nur im Haarschen Fall Symmetrien vorliegen können. Wir kommen auf Symmetriefragen im nächsten Abschnitt zurück.

8.9 Die allgemeine Wavelet–Transformation

Wir wollen noch einmal auf die Formeln der wavelet–Analyse und –Synthese eingehen. Um die allgemeine Form dieser Formeln herzuleiten, setzen wir Verfeinerbarkeit von φ und ψ voraus, aber keine Orthogonalität. Das bedeutet

$$\varphi(x) = \sum_{k\in\mathbb{Z}} p_k \varphi(2x-k),\ \psi(x) = \sum_{k\in\mathbb{Z}} q_k \varphi(2x-k),$$

mit den entsprechenden formalen Laurentreihen P und Q. Um auch $V_1 = V_0 + W_0$ zu haben, brauchen wir auch Formeln der Form

$$\varphi(2x-\ell) = \frac{1}{2}\sum_{k\in\mathbb{Z}} \left(g_{2k-\ell}\varphi(x-k) + h_{2k-\ell}\psi(x-k)\right), \tag{8.28}$$

die wir aus (7.15) im orthogonalen Fall abgeschaut haben. Dazu bilden wir die entsprechenden Laurentreihen

$$G(z) = \frac{1}{2}\sum_{k\in\mathbb{Z}} g_k z^k,\ H(z) = \frac{1}{2}\sum_{k\in\mathbb{Z}} h_k z^k.$$

In diesem Abschnitt (und dem folgenden) reicht es aus, wenn alle Masken endlich sind. Deshalb kümmern wir uns nicht um Summierbarkeitsbedingungen.

Wir machen jetzt den formalen Ansatz

$$\begin{aligned} v^j(x) &:= \sum_{k\in\mathbb{Z}} c_k^j \varphi(2^j x - k) \\ w^j(x) &:= \sum_{k\in\mathbb{Z}} d_k^j \psi(2^j x - k) \end{aligned}$$

und wollen die Gleichung $v^{j+1} = v^j + w^j$ aufstellen und dafür sorgen, daß die biinfiniten Koeffizientensätze

$$\{c_k^{j+1}\} \Leftrightarrow \begin{matrix} \{c_k^j\} \\ \{d_k^j\} \end{matrix} \tag{8.29}$$

bijektiv und linear aufeinander abgebildet werden. Das läuft auf die Invertierbarkeit biinfiniter Matrizen hinaus. Wir werden die Invertierbarkeit nicht abstrakt erschließen, sondern konkret realisieren, weil wir an wohldefinierten Rechenverfahren interessiert sind, die die Darstellungen $v^{j+1} = v^j + w^j$ verlustfrei auf den verschiedenen Levels umrechnen können.

Zur Herleitung der entsprechenden allgemeinen Formeln beginnen wir mit

$$\begin{aligned} v^j(x) &= \sum_{k\in\mathbb{Z}} c_k^j \varphi(2^j x - k) \\ &= \sum_{k\in\mathbb{Z}} c_k^j \sum_{\ell\in\mathbb{Z}} p_\ell \varphi(2(2^j x - k) - \ell) \\ &= \sum_{m\in\mathbb{Z}} \varphi(2^{j+1} x - m) \sum_{k\in\mathbb{Z}} c_k^j p_{m-2k}, \\ w^j(x) &= \sum_{k\in\mathbb{Z}} d_k^j \psi(2^j x - k) \\ &= \sum_{k\in\mathbb{Z}} d_k^j \sum_{\ell\in\mathbb{Z}} q_\ell \varphi(2(2^j x - k) - \ell) \\ &= \sum_{m\in\mathbb{Z}} \varphi(2^{j+1} x - m) \sum_{k\in\mathbb{Z}} d_k^j q_{m-2k}, \end{aligned}$$

und bekommen f¨ur $v^{j+1} = v^j + w^j$ durch Summation die hinreichenden Gleichungen

$$c_m^{j+1} = \sum_{k\in\mathbb{Z}} \Big(c_k^j p_{m-2k} + d_k^j q_{m-2k} \Big). \tag{8.30}$$

für eine Richtung der Abbildung (8.29). Für die andere Richtung benutzen wir

$$\begin{aligned} v^{j+1}(x) &= \sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} \varphi(2^{j+1} x - \ell) \\ &= \frac{1}{2} \sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} \sum_{k\in\mathbb{Z}} \left(g_{2k-\ell} \varphi(x-k) + h_{2k-\ell} \psi(x-k) \right) \\ &= \frac{1}{2} \sum_{k\in\mathbb{Z}} \varphi(x-k) \sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} g_{2k-\ell} + \sum_{k\in\mathbb{Z}} \psi(x-k) \sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} h_{2k-\ell} \end{aligned}$$

und bekommen die hinreichenden Bedingungen

$$\begin{aligned} c_k^j &= \frac{1}{2}\sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} g_{2k-\ell}, \\ d_k^j &= \frac{1}{2}\sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} h_{2k-\ell}. \end{aligned} \tag{8.31}$$

Die Formeln aus (8.30) und (8.31) bilden die allgemeine Form der **Wavelet–Transformation**, aber wir wissen nicht, ob sie bijektiv sind. Immerhin können wir vergessen, daß die Formeln aus Skalierungsfunktionen und wavelets stammen, und die Formeln rein rechnerisch anwenden und auf Bijektivität untersuchen.

Weil wir unendlich viele Gleichungen mit unendlich vielen Variablen haben, müssen wir beide Richtungen der Abbildung invertieren. Wir beginnen mit

$$\begin{aligned} c_m^{j+1} &= \sum_{k\in\mathbb{Z}} \left(c_k^j p_{m-2k} + d_k^j q_{m-2k}\right) \\ &= \frac{1}{2}\sum_{k\in\mathbb{Z}} \left(\sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} g_{2k-\ell} p_{m-2k} + \sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} h_{2k-\ell} q_{m-2k}\right) \\ &= \frac{1}{2}\sum_{\ell\in\mathbb{Z}} c_\ell^{j+1} \sum_{k\in\mathbb{Z}} \left(g_{2k-\ell} p_{m-2k} + h_{2k-\ell} q_{m-2k}\right) \end{aligned}$$

und postulieren deshalb die Gleichungen

$$2\delta_{m\ell} = \sum_{k\in\mathbb{Z}} \left(g_{2k-\ell} p_{m-2k} + h_{2k-\ell} q_{m-2k}\right).$$

Für die Umkehrung untersuchen wir

$$\begin{aligned} c_\ell^j &= \frac{1}{2}\sum_{m\in\mathbb{Z}} c_m^{j+1} g_{2\ell-m}, \\ &= \frac{1}{2}\sum_{m\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} \left(c_k^j p_{m-2k} + d_k^j q_{m-2k}\right) g_{2\ell-m}, \\ &= \frac{1}{2}\sum_{k\in\mathbb{Z}} \left(c_k^j \sum_{m\in\mathbb{Z}} p_{m-2k} g_{2\ell-m} + d_k^j \sum_{m\in\mathbb{Z}} q_{m-2k} g_{2\ell-m}\right) \end{aligned}$$

und brauchen

$$\begin{aligned} 2\delta_{k\ell} &= \sum_{m\in\mathbb{Z}} p_{m-2k} g_{2\ell-m}, \\ 0 &= \sum_{m\in\mathbb{Z}} q_{m-2k} g_{2\ell-m}. \end{aligned}$$

Analog folgt

$$\begin{aligned} d_\ell^j &= \frac{1}{2}\sum_{m\in\mathbb{Z}} c_m^{j+1} h_{2\ell-m}, \\ &= \frac{1}{2}\sum_{m\in\mathbb{Z}}\sum_{k\in\mathbb{Z}} \left(c_k^j p_{m-2k} + d_k^j q_{m-2k}\right) h_{2\ell-m}, \\ &= \frac{1}{2}\sum_{k\in\mathbb{Z}} \left(c_k^j \sum_{m\in\mathbb{Z}} p_{m-2k} h_{2\ell-m} + d_k^j \sum_{m\in\mathbb{Z}} q_{m-2k} h_{2\ell-m} \right) \end{aligned}$$

und erfordert

$$\begin{aligned} 0 &= \sum_{m\in\mathbb{Z}} p_{m-2k} h_{2\ell-m}, \\ 2\delta_{k\ell} &= \sum_{m\in\mathbb{Z}} q_{m-2k} h_{2\ell-m}. \end{aligned}$$

Diese Gleichungen haben die allgemeine Form

$$\sum_{m\in\mathbb{Z}} r_{m-2k} s_{2\ell-m} = \sum_{n\in\mathbb{Z}} r_n s_{2\ell-2k-n} =: t_{2\ell-2k}$$

und wir untersuchen sie mit den formalen Laurentreihen

$$R(z) = \frac{1}{2}\sum_{k\in\mathbb{Z}} r_k z^k,\ S(z) = \frac{1}{2}\sum_{k\in\mathbb{Z}} s_k z^k.$$

Zunächst folgt

$$\begin{aligned} R(z)S(z) &= \frac{1}{4}\sum_{k\in\mathbb{Z}} r_k z^k \sum_{\ell\in\mathbb{Z}} s_\ell z^\ell \\ &= \frac{1}{4}\sum_{n\in\mathbb{Z}} z^n \sum_{k\in\mathbb{Z}} r_k s_{n-k} \end{aligned}$$

und dann

$$\begin{aligned} R(z)S(z) + R(-z)S(-z) &= \frac{1}{2}\sum_{n\in\mathbb{Z}} z^{2n} \sum_{k\in\mathbb{Z}} r_k s_{2n-k} \\ &= \frac{1}{2}\sum_{n\in\mathbb{Z}} z^{2n} t_{2n}. \end{aligned}$$

Damit bekommen wir durch Koeffizientenvergleich die vier Gleichungen

$$\begin{array}{rcccl} 1 &=& P(z)G(z) &+& P(-z)G(-z), \\ 0 &=& Q(z)G(z) &+& Q(-z)G(-z), \\ 0 &=& P(z)H(z) &+& P(-z)H(-z), \\ 1 &=& Q(z)H(z) &+& Q(-z)H(-z), \end{array} \tag{8.32}$$

und als Übung lassen wir die Herleitung der für (8.28) hinreichenden Bedingungen

$$\begin{array}{rcccl} 1 &=& P(z)G(z) &+& Q(z)H(z), \\ 0 &=& P(z)G(-z) &+& Q(z)H(-z). \end{array} \tag{8.33}$$

Theorem 8.34. *Die obigen 6 Gleichungen für Laurentreihen sind hinreichend dafür, daß die allgemeine wavelet–Transformation in beiden Richtungen bijektiv ist. Dabei ist vorausgesetzt, daß die Reihen absolut konvergieren.*

Man kann diese Gleichungen auf verschiedene Weise weiter verarbeiten. Wir stellen uns zuerst auf den Standpunkt, daß wir P und Q schon kennen und daß die schon in Satz 8.18 aufgetretene Determinante

$$P(z)Q(-z) - Q(z)P(-z) =: D(z) \tag{8.35}$$

nicht Null ist, wobei wir offenlassen, ob (8.14) gilt. Die Gleichungen (8.32) sind dann

$$\begin{pmatrix} P(z) & P(-z) \\ Q(z) & Q(-z) \end{pmatrix} \cdot \begin{pmatrix} G(z) & H(z) \\ G(-z) & H(-z) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

und es folgt

$$\begin{aligned} G(z) &= \frac{Q(-z)}{D(z)}, \\ H(z) &= \frac{-P(-z)}{D(z)} \end{aligned} \tag{8.36}$$

als eindeutig bestimmte Lösung, die dann die Koeffizienten von (8.28) und der wavelet–Zerlegung (8.31) liefert. Wir prüfen auch (8.33) nach mit

$$\begin{aligned} & P(z)G(z) + Q(z)H(z) \\ = \; & P(z)\frac{Q(-z)}{D(z)} - Q(z)\frac{P(-z)}{D(z)} \\ = \; & 1, \\ & P(z)G(-z) + Q(z)H(-z) \\ = \; & P(z)\frac{Q(z)}{D(-z)} - Q(z)\frac{P(z)}{D(-z)} \\ = \; & 0 \end{aligned}$$

und sehen, daß diese Gleichungen automatisch erfüllt sind.

Theorem 8.37. *Verschwindet für gegebene P und Q die Determinante (8.35) auf dem Einheitskreis nicht, so ist die wavelet–Transformation bijektiv und die Laurentreihen G und H sind eindeutig durch (8.36) bestimmt. Insbesondere gilt dies für die wavelet–Standardkonstruktion mit $Q(z) = -z\overline{P(-z)}$ unter der Voraussetzung der Stabilität der Translate von φ.*

8.10 Biorthogonale Wavelets

Aber wir können die Gleichungen (8.32) und (8.33) auch anders anwenden, nämlich wenn P und G als bekannt vorausgesetzt werden. Da die Determinante $D(z)$ eine ungerade Funktion ist, ersetzen wir sie formell durch $z^{-1}K(z^2)$ mit einer beliebigen, auf dem Einheitskreis nicht verschwindenden formalen Laurentreihe K und definieren ohne Rücksicht auf die bisherige Herleitung

$$\begin{aligned} Q(z) &:= -\frac{G(-z)K(z^2)}{z}, \\ H(z) &:= \frac{-zP(-z)}{K(z^2)} \end{aligned} \tag{8.38}$$

und müssen prüfen, ob dann die Gleichungen (8.32) und (8.33) gelten. Die erste Gleichung von (8.32) ist eine weiterhin bestehende Forderung an P und G, um die wir uns noch kümmern müssen. Aber alle anderen Gleichungen sind dann erfüllt, wie man leicht nachrechnet, und auch die Determinante ist, rückwärts gerechnet, wegen

$$\begin{aligned} & P(z)Q(-z) - P(-z)Q(z) \\ = \; & P(z)\frac{G(z)K(z^2)}{z} + P(-z)\frac{G(-z)K(z^2)}{z} \\ = \; & \frac{K(z^2)}{z}\left(P(z)G(z) + P(-z)G(-z)\right) \\ = \; & \frac{K(z^2)}{z} \end{aligned}$$

in Ordnung, wenn nur die Gleichung

$$P(z)G(z) + P(-z)G(-z) = 1 \tag{8.39}$$

aus (8.32) gilt.

Theorem 8.40. *Hat man zwei absolut konvergente formale Laurentreihen P und G mit (8.39), und definiert man Q und H durch (8.38) mit einer auf dem Einheitskreisrand nicht verschwindenden komplexen Funktion K, so ist die resultierende wavelet–Transformation bijektiv.*

Die Gleichung (8.39) hat viele mögliche Lösungen, und durch Wahl von K bekommt man zu festen P und G auch beliebig viele Q und H. Aber wir wollen gute Approximationseigenschaften, und deshalb machen wir mit Blick auf die Strang–Fix–Bedingungen bei im folgenden festgehaltenem n den Ansatz

$$P(z) = (1+z)^n R(z),\ G(z) = (1+z)^n S(z)$$

und versuchen, geeignete R und S zu finden. Für $R = S = 2^{-n}$ bekommen wir natürlich die zu B–Splines gehörenden P und G.

Die Gleichung (8.39) wird

$$(1+z)^{2n}R(z)S(z) + (1-z)^{2n}R(-z)S(-z) = 1$$

und es kommt offenbar nur auf $T(z) := R(z)S(z)$ und

$$(1+z)^{2n}T(z) + (1-z)^{2n}T(-z) = 1$$

an, was bei den noch herzuleitenden T zu verschiedenen Wahlmöglichkeiten für R und S führt, sofern T fest ist.

Wie bei den Daubechies–wavelets wird jetzt ziemlich getrickst, und zwar mit

$$\begin{aligned} z &= e^{-i\omega/2}, \\ \cos(\omega/2) &= \frac{e^{i\omega/2} + e^{-i\omega/2}}{2} \\ &= \frac{\frac{1}{z} + z}{2} \\ &= \frac{1+z^2}{2z} \end{aligned}$$

und

$$\begin{aligned} \cos^2(\omega/2) &= \frac{(1+z^2)^2}{4z^2} \\ &=: x, \\ \sin^2(\omega/2) &= 1 - x \\ &= -\frac{(1-z^2)^2}{4z^2}. \end{aligned}$$

Zuerst gehen wir zu

$$(1+z^2)^{2n}T(z^2) + (1-z^2)^{2n}T(-z^2) = 1$$

über und setzen die trigonometrischen Ausdrücke ein:

$$1 = x^n(4z^2)^nT(z^2) + (1-x)^n(-4z^2)^nT(-z^2).$$

An dieser Stelle definieren wir

$$U(x) := (-4z^2)^nT(-z^2) = U\left(\frac{(1+z^2)^2}{4z^2}\right),$$

was wegen der bijektiven Beziehungen zwischen $x \in [0,1]$ und z^2 auf dem Einheitskreisrand und $\omega \in [0, 2\pi]$ kein Problem ist. Es folgt dann

$$U(1-x) \;=\; U\left(-\frac{(1-z^2)^2}{4z^2}\right)$$

und weil das die Ersetzung $z^2 \to -z^2$ in der Definition von U ist, folgt

$$U(1-x) = (4z^2)^n T(z^2)$$

und wir landen bei

$$1 = x^n U(1-x) + (1-x)^n U(x).$$

Das ist bis auf die Normierung schon aus (8.26) bekannt, und wir können $U(x) = 4^n R(x)$ mit dem dortigen $R(x)$ nehmen, aber weil wir die Bedingung (8.27) nicht brauchen, ist es hier einfacher. Wir setzen

$$\begin{aligned} U(x) &= 4^n \sum_{j=0}^{n-1} 4^{-n} \binom{2n-1}{j} x^j (1-x)^{n-1-j} \\ &= \sum_{j=0}^{n-1} \binom{2n-1}{j} x^j (1-x)^{n-1-j} \end{aligned} \tag{8.41}$$

und sind mit unserer Konstruktion fertig.

Der Fall $n = 1$ führt zu

$$\begin{aligned} U(x) &= 1 \\ T(z) &= \frac{1}{4z} \end{aligned}$$

was man zum Beispiel mit

$$R(z) = \frac{1}{2},\; S(z) = \frac{1}{2z}$$

erfüllen kann, und das ergibt die Masken

$$P(z) = \frac{1+z}{2},\; p_0 = p_1 = 1,\; G(z) = \frac{1+z}{2z},\; g_0 = g_{-1} = 1,$$

d.h. zwei verschobene Haarsche Skalierungsfunktionen. Mit $K(z^2) = z^2$ bekommt man dann

$$Q(z) = \frac{1-z}{2},\; q_0 = -q_1 = 1,\; H(z) = -\frac{1-z}{2z},\; h_0 = -h_{-1} = 1,$$

d.h. zwei verschobene Haarsche wavelets.

Gehen wir in den Fall $n = 2$. Es folgt

$$\begin{aligned} U(x) &= 1 + 2x \\ &= 1 + 2\frac{(1+z^2)^2}{4z^2} \\ &= \frac{2z^2 + (1+z^2)^2}{2z^2} \\ &= \frac{1 + 4z^2 + z^4}{2z^2} \\ &= (-4z^2)^2 T(-z^2) \\ T(z) &= -\frac{1 - 4z + z^2}{32z^3} \\ &= -\frac{(z - z_1)(z - z_2)}{32z^3} \end{aligned}$$

mit $z_1 = 2 + \sqrt{3},\ z_2 = 2 - \sqrt{3}$. Wenn wir diese beiden Wurzeln gerecht auf P und G verteilen, bekommen die Zähler von $P(z)$ und $G(z)$ bis auf Konstanten die Form

$$(1+z)^2(z - z_j) = -z_j + z(1 - 2z_j) + z^2(2 - z_j) + z^3$$

mit der Koeffizientensumme $4 - 4z_j$. Man kann dann insgesamt mit

$$\begin{aligned} P(z) &= -\frac{(1+z)^2(z - z_1)(1 - z_2)}{8z} \\ G(z) &= \frac{(1+z)^2(z - z_2)}{4z^2(1 - z_2)} \end{aligned}$$

arbeiten, denn die Koeffizientensummen sind dann

$$\begin{aligned} \frac{-4(1 - z_1)(1 - z_2)}{8} &= \frac{-4(1 - (2 - \sqrt{3}))(1 - (2 + \sqrt{3}))}{8} \\ &= \frac{-4(-1 + \sqrt{3})(-1 - \sqrt{3})}{8} \\ &= 1, \\ \frac{4(1 - z_2)}{4(1 - z_2)} &= 1. \end{aligned}$$

Die beiden Masken haben die Länge 4, und bis auf Renormierung auf Maskensumme 2 bekommt man

$$p_{-1} = -z_1,\ p_0 = (1 - 2z_1),\ p_1 = 2 - z_1,\ p_2 = 1$$

sowie

$$g_{-2} = z_2,\ g_{-1} = -(1 - 2z_2),\ g_0 = -2 + z_2,\ g_1 = -1$$

wobei man darüber streiten kann, welche der Wurzeln man nimmt und wie man die Maskenverschiebung um -1 bzw. -2 verteilt. Ebenfalls bis auf Normierung ergibt sich mit $K(z^2) = z^2$

$$q_{-1} = z_2,\ q_0 = (1 - 2z_2),\ q_1 = z_2 - 2,\ q_2 = 1$$

sowie

$$h_{-2} = z_1,\ h_{-1} = 1 - 2z_1,\ h_0 = 2 - z_1,\ h_1 = -1.$$

Das sind zwei unsymmetrische Masken der Länge 4.

Aber wir haben bisher nicht auf Symmetrie geachtet. Eine zu Null symmetrische Maske $\{p_k\}$ hat $p_k = p_{-k}$ und deshalb gilt dann

$$\begin{aligned} P(z) &= \frac{1}{2}\sum_{k\in\mathbb{Z}} p_k z^k \\ &= \frac{p_0}{2} + \sum_{k\geq 1} p_k \frac{z^k + z^{-k}}{2} \\ &= \frac{p_0}{2} + \sum_{k\geq 1} p_k \cos(k\omega/2) \\ &=: p(\cos(\omega/2)) \end{aligned}$$

und

$$P(z^2) = p(\cos(\omega)).$$

Die Funktion P sollte in $z- = 1$ eine n–fache Nullstelle haben, und das ist an der Stelle $\omega = 2\pi$ für $P(z)$ und $\omega = \pi$ für $P(z^2)$. Weil der entsprechende Linearfaktor die Form

$$1 + \cos(\omega) = 2\cos^2(\omega/2)$$

hat, sollten wir deshalb unseren Ansatz abändern zu

$$P(z^2) = \cos^{2\ell}(\omega/2)p(\cos(\omega)).$$

Mit

$$\cos(\omega/2) = \frac{z + 1/z}{2} = \frac{1 + z^2}{2z},\ \cos(\omega) = \frac{z^2 + z^{-2}}{2} = \frac{1 + z^4}{2z^2}$$

bedeutet das

$$\begin{aligned} P(z^2) &= \left(\frac{1 + z^2}{2z}\right)^{2\ell} p\left(\frac{1 + z^4}{2z^2}\right), \\ P(z) &= \left(\frac{1 + z}{2}\right)^{2\ell} z^{-\ell} p\left(\frac{1 + z^2}{2z}\right). \end{aligned} \tag{8.42}$$

und wir bekommen die $n = 2\ell$–fache Nullstelle bei -1, und man kann leicht noch einmal nachrechnen, daß dieser Ansatz symmetrisch ist. Wegen

$$-z^2 = -e^{-i\omega} = e^{-i(\omega+\pi)}$$

und

$$1 - \cos(\omega) = 2\sin^2(\omega/2)$$

sowie

$$1 + \cos(\omega + \pi) = 1 - \cos(\omega) = 2\cos^2((\omega + \pi)/2) = 2\sin^2(\omega/2)$$

folgt dann

$$\begin{aligned} P(-z^2) &= \cos^{2\ell}((\omega + \pi)/2)p(\cos(\omega + \pi)) \\ &= \sin^{2\ell}(\omega/2)p(-\cos(\omega)). \end{aligned}$$

Wir machen das genauso auch für

$$G(z^2) = \cos^{2\ell}(\omega/2)g(\cos(\omega))$$

und erhalten bei Einsetzen von z^2 in (8.39) die Bedingung

$$\cos^{4\ell}(\omega/2)p(\cos(\omega))g(\cos(\omega)) + \sin^{4\ell}(\omega/2)p(-\cos(\omega))g(-\cos(\omega)) = 1.$$

Setzen wir wieder $x = \cos^2(\omega/2)$, und diesmal

$$U(x) := p(-\cos(\omega))g(-\cos(\omega)),$$

so folgt

$$x^{2\ell}U(1 - x) + (1 - x)^{2\ell}U(x) = 1. \tag{8.43}$$

Die Lösung ist (8.41), und wir machen uns das Leben leichter durch die Setzung $g = 1$, so daß wir $U(x) = p(-\cos(\omega))$ und

$$\begin{aligned} P(z^2) &= \cos^{2\ell}(\omega/2)p(\cos(\omega)) \\ &= \cos^{2\ell}(\omega/2)U(1 - x) \\ &= x^\ell \textstyle\sum_{j=0}^{2\ell-1} \binom{4\ell-1}{j}(1 - x)^j x^{2\ell-1-j} \end{aligned}$$

bekommen. Wenn wir alles einbauen, was wir wissen, erhalten wir

$$P(z) = \left(\frac{1 + z}{2}\right)^{2\ell} z^{-\ell} U_{2\ell}\left(\frac{-z^{-1} + 4 - z}{4}\right),$$

wenn $U_{2\ell}$ die Lösung von (8.43) ist.

Im Falle $n = 2$ liefert das

$$\begin{aligned} U(x) &= 1 + 2x \\ &= 1 + 2\cos^2(\omega/2) \\ &= 2 + \cos(\omega) \\ &= p(-\cos(\omega)) \end{aligned}$$

und deshalb $p(y) = 2 - y$. Mit (8.42) folgt dann

$$\begin{aligned} P(z) &= \left(\frac{1+z}{2}\right)^2 z^{-1} \left(2 - \frac{1+z^2}{2z}\right) \\ &= \frac{1}{8z^2}(1+z)^2(-1+4z-z^2) \end{aligned}$$

mit der symmetrischen Maske

$$(p_{-2}, \ldots, p_2) = \frac{1}{4}(-1, 2, 6, 2, -1).$$

Das entspricht unserer vorigen Herleitung für eine unsymmetrische Verteilung des Polynoms $-1 + 4z - z^2$ auf R und S. Hier bleibt für S nur noch

$$S(z) = \frac{1}{4z}$$

und die symmetrische Maske für G wird die eines verschobenen B–Splines zweiten Grades, wegen

$$G(z) = \frac{(1+z)^2}{4z} = \frac{1}{2}(z^{-1} + 2 + z).$$

Man sieht, daß die Symmetrie von P durch eine leicht verlängerte Maske erkauft wurde, und es ist wegen (8.42), auf G angewendet, klar, daß sich bei der symmetrischen Konstruktion für gerades n immer G als die Maske eines symmetrischen B–Splines ergibt.

8.11 Wavelet–Fehlerabschätzungen

Wir setzen jetzt voraus, daß wir eine verfeinerbare Funktion φ haben, die Strang–Fix–Bedingungen der Ordnung m erfüllt und die Konstruktion eines vernünftigen wavelets ψ zuläßt. Daraus wollen wir Fehlerabschätzungen herleiten, die auf den Levels der wavelet–Zerlegung gelten.

Wir nehmen die stationäre Skalierung wie im Text über translationsinvariante Räume. Dort projizierten wir für kleine $h > 0$ auf die Shifts von $\frac{1}{h}\varphi((\cdot - hk)/h)$ indem wir den Projektor

$$P_{\varphi,h}(f)(x) := P_\varphi(f(\cdot h))(x/h)$$

nahmen. Bei wavelets mit einer Multiresolutionsanalyse setzt man $h = 2^{-j}$ im "Level" j und projiziert damit auf den span V_j der Translate $\varphi(2^j \cdot -k) = \varphi((\cdot - hk)/h)$.

Geht man von einer Funktion $f \in W_2^m(\mathbb{R})$ aus, so kann man die Projektionen im Level j als

$$f_j := P_{\varphi,2^{-j}}(f)$$

ansetzen und bekommt unter unseren Voraussetzungen aus der Fehlerabschätzung in translationsinvarianten Räumen die Aussage

$$\|f - f_j\|_{L_2(\mathbb{R})} = \|f - P_{\varphi,2^{-j}}(f)\|_{L_2(\mathbb{R})} \leq C 2^{-jm} \|f\|_{W_2^m(\mathbb{R})}.$$

Das ist nicht nur eine Fehlerabschätzung, sondern auch eine Konvergenzaussage für $j \to \infty$. An dieser Stelle müssen wir allerdings Orthogonalität voraussetzen, denn wir benutzen, daß die üblichen wavelet-Rechenformeln mit denen einer Projektion übereinstimmen.

Wir wollen das noch in eine Aussage über die wavelet–Anteile umformen. Dazu definieren wir

$$g_j := f_{j+1} - f_j \in V_{j+1}$$

und benutzen, daß wegen der Projektionseigenschaft

$$\begin{aligned}
(g_j, v_j)_{L_2(\mathbb{R}^d)} &= (f_{j+1} - f_j, v_j)_{L_2(\mathbb{R}^d)} \\
&= (f_{j+1} - f + f - f_j, v_j)_{L_2(\mathbb{R}^d)} \\
&= (f_{j+1} - f, v_j)_{L_2(\mathbb{R}^d)} + (f - f_j, v_j)_{L_2(\mathbb{R}^d)} \\
&= 0
\end{aligned}$$

für alle $v_j \in V_j$ gilt. Also ist $g_j \in W_j$ der wavelet–Anteil, und wir können die wavelet–Zerlegung bis zum Level n als "Teleskopsumme"

$$\begin{aligned}
f_n &= f_0 + \sum_{j=0}^{n-1} (f_{j+1} - f_j) \\
&= f_0 + \sum_{j=0}^{n-1} g_j
\end{aligned}$$

schreiben. Die ℓ_2–Norm der wavelet–Koeffizienten auf Level j ist bei vorausgesetzter Stabilität direkt proportional zu $\|g_j\|_{L_2(\mathbb{R})}$ und es folgt

$$\begin{aligned}
\|g_j\|_{L_2(\mathbb{R})} &= \|f_{j+1} - f_j\|_{L_2(\mathbb{R})} \\
&\leq \|f - f_{j+1}\|_{L_2(\mathbb{R})} + \|f - f_j\|_{L_2(\mathbb{R})} \\
&\leq 2C 2^{-jm} \|f\|_{W_2^m(\mathbb{R})}.
\end{aligned}$$

Von Schritt zu Schritt verkleinert sich sowohl der Approximationsfehler als auch die Größe der wavelet–Koeffizienten um etwa den Faktor 2^{-m}. Das ist der entscheidende Grund für die guten Approximations– und Kompressionseigenschaften von wavelets.

9 Extras

Hier sind ältere Textbausteine, die einige Aspekte der Vorlesung ergänzen. Zuerst kommt eine nicht ganz komplette Darstellung der multivariaten Fouriertransformation, und dann die Durchrechnung der Interpolation in Tschebyscheff–Nullstellen, zusammen mit der diskreten Cosinustransformation DCT-II und DCT-III. Am Ende folgt noch eine Passage aus der Numerischen Mathematik II, die B–Splines bringt, denn diese werden bei den wavelets manchmal gebraucht.

9.1 Fourier Transforms on $\mathbb{R}^d$

Here, we provide the basics of multivariate Fourier transforms.

9.1.1 Fourier Transforms of Tempered Test Functions

There are two major possibilities to pick a space $\mathcal{S}$ of test functions on $\mathbb{R}^d$ to start with, and we take the **tempered test functions** that are verbally defined as real-valued functions on $\mathbb{R}^d$ whose partial derivatives exist for all orders and decay faster than any polynomial towards infinity.

Definition 9.1. *For a test function $u \in \mathcal{S}$, the* **Fourier transform** *is*

$$\widehat{u}(\omega) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} u(x) e^{-ix\cdot\omega} dx,$$

where ω varies in $\mathbb{R}^d$ and $x \cdot \omega$ is shorthand for the scalar product $x^T\omega = \omega^T x$ to avoid the ^{T} symbol in the exponent. Since the definition even works for general $u \in L_1(\mathbb{R}^d)$, it is well-defined on $\mathcal{S}$ and clearly linear. Note that we use the **symmetric** form of the transform and do not introduce a factor 2π in the exponent of the exponential. This sometimes makes comparisons to other presentations somewhat difficult.

To get used to calculations of Fourier transforms, let us start with the **Gaussian** $u_\gamma(x) = \exp(-\gamma\|x\|_2^2)$ for $\gamma > 0$, which clearly is in the space of test functions, since all derivatives are polynomials multiplied with the Gaussian itself. Fortunately, the Gaussian can be written as a d-th power of the entire analytic function $\exp(-\gamma z^2)$, and we can thus work on $\mathbb{C}^d$ instead of $\mathbb{R}^d$. We simply use substitution in

$$\begin{aligned}
\widehat{u_\gamma}(i\omega) &= (2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} e^{-\gamma\|x\|_2^2} e^{x\cdot\omega} dx \\
&= (2\pi)^{-d/2} e^{\|\omega\|_2^2/4\gamma} \textstyle\int_{\mathbb{R}^d} e^{-\|\sqrt{\gamma}x - \omega/2\sqrt{\gamma}\|_2^2} dx \\
&= (2\pi\gamma)^{-d/2} e^{\|\omega\|_2^2/4\gamma} \textstyle\int_{\mathbb{R}^d} e^{-\|y\|_2^2} dy
\end{aligned}$$

and are done up to the evaluation of the dimension-dependent constant

$$\int_{\mathbb{R}^d} e^{-\|y\|_2^2} dy =: c^d$$

which is a d-th power, because the integrand factorizes nicely. We calculate c^2 by using polar coordinates and get

$$\begin{aligned} c^2 &= \int_{\mathbb{R}^2} e^{-\|y\|_2^2} dy \\ &= \int_0^{2\pi} \int_0^\infty e^{-r^2} r \, dr \, d\varphi \\ &= 2\pi \int_0^\infty e^{-r^2} r \, dr \\ &= -\pi \int_0^\infty (-2r) e^{-r^2} \, dr \\ &= \pi. \end{aligned}$$

This proves the first assertion of

Theorem 9.2. *The Gaussian*

$$u_\gamma(x) = \exp(-\gamma \|x\|_2^2)$$

has Fourier transform

$$\widehat{u_\gamma}(\omega) = (2\gamma)^{-d/2} e^{-\|\omega\|_2^2/4\gamma} \tag{9.3}$$

and is (unconditionally) positive definite on $\mathbb{R}^d$.

To understand the second assertion, we add

Definition 9.4. *A real-valued function*

$$\Phi : \Omega \times \Omega \to \mathbb{R}$$

is a **positive definite function** *on Ω, iff for any choice of finite subsets $X = \{x_1, \ldots, x_M\} \subseteq \Omega$ of M different points the matrix*

$$A_{X,\Phi} = (\Phi(x_k, x_j))_{1 \le j,k \le M}$$

is positive definite.

At first sight it seems to be a miracle that a fixed function Φ should be sufficient to make all matrices of the above form positive definite, no matter which points are chosen and no matter how many. It is even more astonishing

that one can often pick radial functions like $\Phi(x,y) = \exp(\|x-y\|_2^2)$ to do the job, and to work for **any** space dimension.

Proof of the theorem: Let us first invert the Fourier transform by setting $\beta := 1/4\gamma$ in (9.3):

$$\begin{aligned} \exp(-\beta\|\omega\|_2^2) &= (4\pi\beta)^{-d/2} \int_{\mathbb{R}^d} e^{-\|x\|_2^2/4\beta} e^{-ix\cdot\omega} dx \\ &= (2\pi)^{-d/2} \int_{\mathbb{R}^d} (2\beta)^{-d/2} e^{-\|x\|_2^2/4\beta} e^{+ix\cdot\omega} dx. \end{aligned}$$

Then take any set $X = \{x_1, \ldots, x_M\} \subset \mathbb{R}^d$ of M distinct points and any vector $\alpha \in \mathbb{R}^M$ to form

$$\begin{aligned} \alpha^T A_{X,u_\gamma} \alpha &= \sum_{j,k=1}^M \alpha_j \alpha_k \exp(-\gamma\|x_j - x_k\|_2^2) \\ &= \sum_{j,k=1}^M \alpha_j \alpha_k (4\pi\gamma)^{-d/2} \int_{\mathbb{R}^d} e^{-\|x\|_2^2/4\gamma} e^{-ix\cdot(x_j-x_k)} dx \\ &= (4\pi\gamma)^{-d/2} \int_{\mathbb{R}^d} e^{-\|x\|_2^2/4\gamma} \sum_{j,k=1}^M \alpha_j \alpha_k e^{-ix\cdot(x_j-x_k)} dx \\ &= (4\pi\gamma)^{-d/2} \int_{\mathbb{R}^d} e^{-\|x\|_2^2/4\gamma} \left| \sum_{j=1}^M \alpha_j e^{-ix\cdot x_j} \right|^2 dx \geq 0. \end{aligned}$$

This proves positive semidefiniteness of the Gaussian. To prove definiteness, we can assume

$$f(x) := \sum_{j=1}^M \alpha_j e^{-ix\cdot x_j} = 0$$

for all $x \in \mathbb{R}^d$ and have to prove that all coefficients α_j vanish. Taking derivatives at zero, we get

$$0 = D^\beta f(0) = \sum_{j=1}^M \alpha_j (-ix_j)^\beta,$$

and this is a homogeneous system for the coefficients α_j whose coefficient matrix is a generalized Vandermonde matrix, possibly transposed and with scalar multiples for rows or columns. This proves the assertion in one dimension, where the matrix corresponds to the classical Vandermonde matrix. The multivariate case reduces to the univariate case by picking a nonzero vector $y \in \mathbb{R}^d$ that is not orthogonal to any of the finitely many differences $x_j - x_k$ for $j \neq k$. Then the real values $y \cdot x_j$ are all distinct for $j = 1, \ldots, M$ and one can consider the univariate function

$$g(t) := f(ty) = \sum_{j=1}^M \alpha_j e^{-ity\cdot x_j} = 0$$

which does the job in one dimension. □

Note that the Gaussian is mapped to itself by the Fourier transform, if we pick $\gamma = 1/2$. We shall use the Gaussian's Fourier transform in the proof of the fundamental **Fourier Inversion Theorem**:

Theorem 9.5. *The Fourier transform is bijective on $\mathcal{S}$, and its inverse is the transform*

$$\check{u}(x) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} u(\omega) e^{ix\cdot\omega} d\omega.$$

Proof: The multivariate derivative D^α of $\widehat{u}$ can be taken under the integral sign, because u is in $\mathcal{S}$. Then

$$(D^\alpha \widehat{u})(\omega) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} u(x)(-ix)^\alpha e^{-ix\cdot\omega} dx,$$

and we multiply this by ω^β and use integration by parts

$$\begin{aligned}
\omega^\beta (D^\alpha \widehat{u})(\omega) &= (2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} u(x)(-ix)^\alpha (i)^\beta (-i\omega)^\beta e^{-ix\cdot\omega} dx \\
&= (2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} u(x)(-ix)^\alpha (i)^\beta \frac{d^\beta}{dx^\beta} e^{-ix\cdot\omega} dx \\
&= (2\pi)^{-d/2} (-1)^{|\alpha|+|\beta|} i^{\alpha+\beta} \textstyle\int_{\mathbb{R}^d} e^{-ix\cdot\omega} \frac{d^\beta}{dx^\beta} (u(x)x^\alpha) dx
\end{aligned}$$

to prove that $\widehat{u}$ lies in $\mathcal{S}$, because all derivatives decay faster than any polynomial towards infinity. The second assertion follows from the Fourier inversion formula

$$u(x) := (2\pi)^{-d/2} \int_{\mathbb{R}^d} \widehat{u}(\omega) e^{ix\cdot\omega} d\omega$$

that we now prove for all $u \in \mathcal{S}$. This does not work directly if we naively put the definition of $\widehat{u}$ into the right-hand-side, because the resulting multiple integral does not satisfy the assumptions of Fubini's theorem. We have to do a regularization of the integral, and since this is a standard trick, we write it out in some detail:

$$\begin{aligned}
(2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} \widehat{u}(\omega) e^{ix\cdot\omega} d\omega &= (2\pi)^{-d} \textstyle\int_{\mathbb{R}^d} \int_{\mathbb{R}^d} u(y) e^{i(x-y)\cdot\omega} dy d\omega \\
&= \lim_{\epsilon \searrow 0} (2\pi)^{-d} \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} u(y) e^{i(x-y)\cdot\omega - \epsilon\|\omega\|_2^2} dy d\omega \\
&= \lim_{\epsilon \searrow 0} (2\pi)^{-d} \int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} e^{i(x-y)\cdot\omega - \epsilon\|\omega\|_2^2} d\omega \right) u(y) dy \\
&= \lim_{\epsilon \searrow 0} \int_{\mathbb{R}^d} \varphi(\epsilon, x-y) u(y) dy
\end{aligned}$$

with

$$\varphi(\epsilon, z) := (2\pi)^{-d} \int_{\mathbb{R}^d} e^{iz\cdot\omega - \epsilon\|\omega\|_2^2} d\omega. \tag{9.6}$$

The proof is completed by application of the following result that is useful in many contexts: □

Lemma 9.7. *The family of functions $\varphi(\epsilon, z)$ of (9.6) approximates the point evaluation functional in the sense*

$$u(x) = \lim_{\epsilon \searrow 0} \int_{\mathbb{R}^d} \varphi(\epsilon, x - y) u(y) dy \tag{9.8}$$

for all functions u that are in $L_1(\mathbb{R}^d)$ and continuous around x.

Proof: We first remark that φ is a disguised form of the inverse Fourier transform equation of the Gaussian. Thus we get

$$\varphi(\epsilon, x) = (4\pi\epsilon)^{-d/2} e^{-\|x\|_2^2/4\epsilon} \tag{9.9}$$

and

$$\int_{\mathbb{R}^d} \varphi(\epsilon, x) dx = (4\pi\epsilon)^{-d/2} \int_{\mathbb{R}^d} e^{-\|x\|_2^2/4\epsilon} dx = 1.$$

To prove (9.8), we start with some given $\delta > 0$ and first find some ball $B_\rho(x)$ of radius $\rho(\delta)$ around x such that $|u(x) - u(y)| \leq \delta/2$ holds uniformly for all $y \in B_\rho(x)$. Then we split the integral in

$$\begin{array}{rcl} |u(x) - \int_{\mathbb{R}^d} \varphi(\epsilon, x - y) u(y) dy| & = & |\int_{\mathbb{R}^d} \varphi(\epsilon, x - y)(u(x) - u(y)) dy| \\ & \leq & \int_{\|y-x\|_2 \leq \rho} \varphi(\epsilon, x - y)|u(x) - u(y)| dy \\ & & + \int_{\|y-x\| > \rho} \varphi(\epsilon, x - y)|u(x) - u(y)| dy \\ & \leq & \delta/2 + (4\pi\epsilon)^{-d/2} e^{-\rho^2/4\epsilon} 2\|u\|_1 \\ & \leq & \delta \end{array}$$

for all sufficiently small ϵ. □

Due to the Fourier inversion formula, we now know that the Fourier transform is bijective on $\mathcal{S}$.

We now relate the Fourier transform to the L_2 inner product, but we have to use the latter over $\mathbb{C}$ to account for the possibly complex values of the Fourier transform. Furthermore, we have good reasons to define the inner product as

$$(f, g)_{L_2(\mathbb{R}^d)} := (2\pi)^{-d/2} \int_{\mathbb{R}^d} f(x) \overline{g(x)} dx \tag{9.10}$$

with a factor that simplifies some of the subsequent formulae.

Fubini's theorem easily proves the identity

$$(v, \widehat{u})_{L_2(\mathbb{R}^d)} = (2\pi)^{-d} \int_{\mathbb{R}^d} v(x) \int_{\mathbb{R}^d} \overline{u(y)} e^{+ix \cdot y} dy dx = (\check{v}, u)_{L_2(\mathbb{R}^d)}$$

for all test functions $u, v \in \mathcal{S}$. Setting $v = \widehat{w}$ we get Parceval's equation

$$(\widehat{w}, \widehat{u})_{L_2(\mathbb{R}^d)} = (w, u)_{L_2(\mathbb{R}^d)} \tag{9.11}$$

for the Fourier transform on $\mathcal{S}$, proving that the Fourier transform is isometric on $\mathcal{S}$ as a subspace of $L_2(\mathbb{R}^d)$.

9.1.2 Fourier Transforms of Functionals

With Parceval's equation in mind, let us look at the linear functional

$$\lambda_u(v) := (u, v)_{L_2(\mathbb{R}^d)}$$

on $\mathcal{S}$. We see that

$$\lambda_{\widehat{u}}(v) = (\widehat{u}, v)_{L_2(\mathbb{R}^d)} = (u, \check{v})_{L_2(\mathbb{R}^d)} = \lambda_u(\check{v})$$

holds. A proper definition of the Fourier transform for functionals λ_u should be in line with the functions u that represent them, and thus we should define

$$\widehat{\lambda_u} := \lambda_{\widehat{u}}$$

or in more generality

$$\widehat{\lambda}(v) := \lambda(\check{v})$$

for all $v \in \mathcal{S}$. Since the space $\mathcal{S}$ of test functions is quite small, its dual, the space of linear functionals on $\mathcal{S}$, will be quite large.

Definition 9.12. *The Fourier transform of a linear functional λ on $\mathcal{S}$ is the linear functional $\widehat{\lambda}$ on $\mathcal{S}$ defined by*

$$\widehat{\lambda}(v) := \lambda(\check{v})$$

for all $v \in \mathcal{S}$.

If we can represent the functional $\widehat{\lambda}$ as λ_v, we write $v = \widehat{\lambda}$ as a shorthand notation, but keep the original meaning in mind. Let us look at some examples.

Example 9.13. *The functional $\delta_x(v) := v(x)$ has the form*

$$\delta_x(v) = v(x) = (2\pi)^{-d/2} \int_{\mathbb{R}^d} \widehat{v}(\omega) e^{+ix\cdot\omega} d\omega,$$

and its Fourier transform is of the form λ_u with

$$u(\omega) = \widehat{\delta_x}(\omega) = e^{-x\cdot\omega}.$$

Here, the normalization of the L_2 inner product (9.10) pays off. Note that the Fourier transform is not a test function, but rather an at most polynomially growing function from $\mathcal{K}$ and in particular a bounded function. The functional $\delta := \delta_0$ has the Fourier transform 1.

Example 9.14. *A very important class of functionals for our purposes consists of the space $\mathcal{P}_0$ of functionals of the form (**??**) that vanish on $\P_m^d$. Their action on a test function v is*

$$\begin{aligned}\lambda_{X,M\alpha}(v) &= \sum_{j=1}^{M} \alpha_j v(x_j) \\ &= (2\pi)^{-d/2} \int_{\mathbb{R}^d} \widehat{v}(\omega) \sum_{j=1}^{M} \alpha_j e^{ix_j\cdot\omega} d\omega \\ &= \widehat{\lambda}_{X,M\alpha}(\widehat{v})\end{aligned}$$

such that the Fourier transform of the functional $\lambda_{X,M\alpha}$ is the functional generated by the bounded function

$$\widehat{\lambda}_{X,M,\alpha}(\omega) = \sum_{j=1}^{M} \alpha_j e^{-ix_j\cdot\omega}.$$

If we expand the exponential into its power series, we see that

$$\begin{aligned}\widehat{\lambda}_{X,M,\alpha}(\omega) &= \sum_{k=0}^{\infty}\sum_{j=1}^{M} \alpha_j (-ix_j\cdot\omega)^k/k! \\ &= \sum_{k=m}^{\infty}\sum_{j=1}^{M} \alpha_j (-ix_j\cdot\omega)^k/k!\end{aligned}$$

since the functional vanishes on $\P_m^d$. Thus $\widehat{\lambda}_{X,M,\alpha}(\omega)$ has a zero of order at least m in zero. If the functional $\lambda_{X,M\alpha}$ itself were representable by a function u, the function u should be orthogonal to all polynomials from $\P_m^d$. We shall use both of these facts later.

Example 9.15. *The monomials x^α are in the space $\mathcal{K}$, and thus they should at least have generalized Fourier transforms in the sense of functionals. This can easily be verified via*

$$\begin{aligned}\left(-i\tfrac{d}{dx}\right)^\alpha v(x) &= \left(-i\tfrac{d}{dx}\right)^\alpha (2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} \widehat{v}(\omega) e^{+ix\cdot\omega} d\omega \\ &= (2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} \widehat{v}(\omega)(-i\cdot i\omega)^\alpha e^{+ix\cdot\omega} d\omega \\ &= (2\pi)^{-d/2} \textstyle\int_{\mathbb{R}^d} \widehat{v}(\omega)\omega^\alpha e^{+ix\cdot\omega} d\omega\end{aligned}$$

and the associated functional is

$$v \mapsto \left(-i\frac{d}{dx}\right)^\alpha v(x)$$

at $x = 0$.

9.1.3 Fourier Transform in $L_2(\mathbb{R}^d)$

The test functions from $\mathcal{S}$ are dense in $L_2(\mathbb{R}^d)$ (see Lemma 9.19 for details), and thus we have

Theorem 9.16. *The Fourier transform has an L_2-isometric extension from the space $\mathcal{S}$ of tempered test functions to $L_2(\mathbb{R}^d)$. The same holds for the inverse Fourier transform, and both extensions are inverses of each other in $L_2(\mathbb{R}^d)$. Furthermore, Parceval's equation (9.11) holds in $L_2(\mathbb{R}^d)$.* □

Note that this result does not allow to use the Fourier transform formula (or its inverse) in the natural pointwise form. For any $f \in L_2(\mathbb{R}^d)$ one first has to provide a sequence of test functions $v_n \in \mathcal{S}$ that converges to f in the L_2 norm for $n \to \infty$, and then, by continuity, the image $\widehat{f}$ of the Fourier transform is uniquely defined almost everywhere by

$$\lim_{n\to\infty} \|\widehat{f} - \widehat{v_n}\|_{L_2(\mathbb{R}^d)} = 0.$$

This can be done via Friedrich's mollifiers as defined in (9.18), replacing the Gaussian in the representation (9.9) by a compactly supported infinitely differentiable function.

A more useful characterization of $\widehat{f}$ is the variational equation

$$(\widehat{f}, v)_{L_2(\mathbb{R}^d)} = (f, \check{v})_{L_2(\mathbb{R}^d)}$$

for all test functions $v \in \mathcal{S}$, or, by continuity, all functions $v \in L_2(\mathbb{R}^d)$.

9.1.4 Necessary Results from Real Analysis

Here we collect some of the basic material on Lebesgue integration, Sobolev spaces, distributions, pseudodifferential operators, and partial differential equations.

9.1.5 Lebesgue Integration

9.1.6 L_2 spaces

Lemma 9.17. *The shift operator $S_z : f(\cdot) \mapsto f(\cdot - z)$ is a continuous function of z near zero in the following sense: for each given $u \in L_2(\mathbb{R}^d)$ and each given $\epsilon > 0$ there is some $\delta > 0$ such that*

$$\|S_z(u) - u\|_{L_2(\mathbb{R}^d)} \leq \epsilon$$

for all $\|z\|_2 \leq \delta$.

Proof: to be supplied later....

We now want to prove that the space $\mathcal{S}$ of tempered test functions is dense in $L_2(\mathbb{R}^d)$. For this, we have to study functions like (9.6) in some more detail. They are in $\mathcal{S}$ for all positive values of ϵ, and Lemma 9.7 tells us that the operation

$$f \mapsto M_\epsilon(f) := \int_{\mathbb{R}^d} f(y)\varphi(\epsilon, \cdot - y)dy$$

maps each continuous L_1 function f to a "mollified" function $M_\epsilon(f)$ such that

$$\lim_{\epsilon \to 0} M_\epsilon(f)(x) = f(x)$$

uniformly on compact subsets of $\mathbb{R}^d$.

It is common to replace the Gaussian in (9.9) by an infinitely differentiable function with compact support, e.g.

$$\varphi(\epsilon, x) = \left\{ \begin{array}{ll} c(\epsilon)\exp(-1/(\epsilon^2 - \|x\|_2^2)) & \|x\|_2 < \epsilon \\ 0 & \|x\|_2 \geq \epsilon \end{array} \right\} \tag{9.18}$$

where the constant $c(\epsilon)$ is such that

$$\int_{\mathbb{R}^d} \varphi(\epsilon, x)dx = 1$$

holds for all $\epsilon > 0$. This **Friedrich's mollifier** can also be used in the definition of M_ϵ. It has the advantage that Lemma 9.7 holds for more general functions, i.e.: for functions which are in L_1 only locally.

We now want to study the action of M_ϵ on L_2 functions. Let $u \in L_2(\mathbb{R}^d)$ be given, and apply the Cauchy-Schwarz inequality to

$$M_\epsilon(f)(x) = \int_{\mathbb{R}^d} (f(y)\sqrt{\varphi(\epsilon, x-y)})\sqrt{\varphi(\epsilon, x-y)}dy$$

to get

$$\begin{array}{rcl} |M_\epsilon(f)(x)|^2 & \leq & \int_{\mathbb{R}^d} |f(y)|^2\varphi(\epsilon, x-y)dy \int_{\mathbb{R}^d} \varphi(\epsilon, x-y)dy \\ & = & \int_{\mathbb{R}^d} |f(y)|^2\varphi(\epsilon, x-y)dy \end{array}$$

and

$$\int_{\mathbb{R}^d} |M_\epsilon(f)(x)|^2 dx \leq \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} |f(y)|^2\varphi(\epsilon, z)dydz = \int_{\mathbb{R}^d} |f(y)|^2 dy$$

such that M_ϵ has norm less than or equal to one in the L_2 norm. It is even more simple to prove the identity

$$(f, M_\epsilon g)_{L_2(R^d)} = (M_\epsilon f, g)_{L_2(R^d)}$$

for all $f, g \in L_2(\mathbb{R}^d)$ by looking at the integrals. Here, we used the Fubini theorem on $\mathbb{R}^d$ which requires some care, but there are no problems because everything can either be done with a Friedrich's mollifier, or be done on sufficiently large compact sets first, letting the sets tend to $\mathbb{R}^d$ later.

We now use a Friedrich's mollifier to study the L_2 error of the mollification. This is very similar to the arguments we already know. The error is representable pointwise as

$$f(x) - M_\epsilon(f)(x) = \int_{\mathbb{R}^d} (f(x) - f(y))\varphi(\epsilon, x - y)dy$$

and we can use the Cauchy-Schwarz inequality to get

$$|f(x) - M_\epsilon(f)(x)|^2 \leq \int_{\|x-y\|_2<\epsilon} |f(x) - f(y)|^2 \varphi(\epsilon, x - y)dy.$$

This can be integrated to get

$$\int_{\mathbb{R}^d} |f(x) - M_\epsilon(f)(x)|^2 dx \leq \int_{\|z\|_2<\epsilon} \varphi(\epsilon, z) \int_{\mathbb{R}^d} |f(y + z) - f(y)|^2 dydz,$$

and we use the continuity of the shift operator as proven in Lemma 9.17 to make this as small as we want by picking a suitably small ϵ. This shows

$$\lim_{\epsilon \to 0} \|f - M_\epsilon(f)\|_{L_2(\mathbb{R}^d)} = 0$$

and proves

Lemma 9.19. *The space $\mathcal{S}$ of test functions is dense in $L_2(\mathbb{R}^d)$.* □

9.2 Chebyshev Interpolation and DCT

Dieser ältere Textbaustein ist leider noch nicht übersetzt und an die bisherigen Bezeichnungen angepaßt. Er behandelt die Interpolation in den Tschebyscheff–Nullstellen, nicht in den Extremstellen.

9.2.1 Chebyshev Interpolation Matrix

Recall the definition of the Chebyshev polynomials:

$$\begin{aligned} T_n(x) &= \cos(n \cdot \arccos(x)),\ n \geq 0,\ x \in [-1,1] \\ T_0(x) &= 1, \\ T_1(x) &= x, \\ T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x),\ n \geq 2,\ x \in \mathbb{R}. \end{aligned}$$

The zeros of T_n are derived via:

$$\begin{aligned} T_n(x_j) &= \cos(n \arccos x_j) = 0 \\ x_j &= \cos\varphi_j \\ n\varphi_j &= (2j-1)\pi/2,\ 1 \leq j \leq n \\ \varphi_j &= \pi\tfrac{2j-1}{2n},\ 1 \leq j \leq n \\ x_j &= \cos\left(\pi\tfrac{2j-1}{2n}\right),\ 1 \leq j \leq n \end{aligned}$$

Extrema of T_n are derived via:

$$\begin{aligned} T_n(y_j) &= \cos(n \arccos y_j) = \pm 1 \\ y_j &= \cos\varphi_j \\ n\varphi_j &= j\pi,\ 0 \leq j \leq n \\ \varphi_j &= \pi\tfrac{j}{n},\ 0 \leq j \leq n \\ y_j &= \cos\left(\pi\tfrac{j}{n}\right),\ 0 \leq j \leq n. \end{aligned}$$

Values of the $T_0, \ldots, T_n$ at the zeros of T_{n+1} are:

$$T_j(x_k) = \cos\left(\frac{j(2k+1)\pi}{2n+2}\right),\ 0 \leq j,k \leq n. \tag{9.20}$$

This is the matrix arising in **Chebyshev interpolation**, i.e. interpolation using the basis $T_0, \ldots, T_n$ and the $n+1$ zeros of T_{n+1} as data points. As in our MATLAB programs, the point index is the row index when we write this as an $(n+1) \times (n+1)$ matrix T. Then we define $C := T^T T$ and consider its entries

$$c_{ij} := \sum_{k=0}^{n} T_i(x_k)T_j(x_k) = \sum_{k=0}^{n} (T_i \cdot T_j)(x_k).$$

We plug this into the Gauss–Chebyshev integration formula

$$\int_{-1}^{+1} \frac{p(t)}{\sqrt{1-t^2}} dt = \frac{\pi}{n+1} \sum_{k=0}^{n} p(x_k)$$

which is exact for all polynomials up to degree $2n+1$. We get

$$c_{ij} = \frac{n+1}{\pi} \int_{-1}^{+1} \frac{T_i(t)T_j(t)}{\sqrt{1-t^2}} dt.$$

We now use the orthogonality relations

$$\int_{-1}^{+1} \frac{T_i(t)T_j(t)}{\sqrt{1-t^2}} dt = \begin{cases} 0 & i \neq j \\ \frac{\pi}{2} & i = j \neq 0 \\ \pi & i = j = 0. \end{cases}$$

If we define D as the $(n+1) \times (n+1)$ diagonal matrix with the diagonal $(1, \frac{1}{2}, \ldots, \frac{1}{2})$ we get $T^T T = C = (n+1)D$.

Theorem 9.21. *Let T be the matrix arising for interpolation by Chebyshev polynomials in Chebyshev zeros. Then the matrix $\frac{1}{\sqrt{n+1}} T D^{-1/2}$ is orthogonal, where $D^{-1/2}$ has the diagonal $(1, \sqrt{2}, \ldots, \sqrt{2})$.*

Now we calculate the spectral condition of T. We have

$$\|T\| = \max\{\sqrt{\lambda} \ : \ \lambda \text{ is eigenvalue of } T^T T\}.$$

But the spectrum of $T^T T = (n+1)D$ is

$$(n+1)(1, \frac{1}{2}, \ldots, \frac{1}{2})$$

such that we get $\|T\| = \sqrt{n+1}$. The same is done for T^{-1}. The spectrum of $(T^{-1})^T T^{-1}$ is the same as of $D^{-1}/(n+1)$, thus it is

$$\frac{1}{n+1}(1, 2, \ldots, 2)$$

and we get $\|T^{-1}\| = \frac{\sqrt{2}}{\sqrt{n+1}}$. Thus

Theorem 9.22. *The spectral condition of the matrix T arising for interpolation by Chebyshev polynomials in Chebyshev zeros is $\sqrt{2}$ independent of the degree.*

We now look at the interpolation problem in the x_k. The linear system is

$$\begin{aligned} Ta &= y \\ \sum_{j=0}^{n} a_j \cos\left(\frac{j(2k+1)\pi}{2n+2}\right) &= y_k,\ 0 \le k \le n \end{aligned} \tag{9.23}$$

for values $y = (y_0, \ldots, y_n)^T$ and coefficients $a = (a_0, \ldots, a_n)^T$. The system can be solved **without inversion** of T via

$$\begin{aligned} T^T T a &= T^T y \\ &= (n+1)Da \\ a &= \tfrac{1}{n+1} D^{-1} T^T y \end{aligned}$$

which means

$$\begin{aligned} a_j &= \frac{2}{n+1} \sum_{k=0}^{n} y_k \cos\left(\frac{j(2k+1)\pi}{2n+2}\right),\ 1 \le j \le n \\ a_0 &= \frac{1}{n+1} \sum_{k=0}^{n} y_k. \end{aligned}$$

9.2.2 Discrete Cosine Transform DCT-II/III

The above transformation is one of the many cases of a **discrete cosine transform** (DCT). Up to slight modifications, we shall show that this is `dct` and `idct` in MATLAB, and there is a close connection to the Fourier transform.

But since there are many cosine transforms on the market, and since the connection to the discrete complex Fourier transform is somewhat unclear, we have to do some additional modifications. First, we go back to standard Fourier transform notation and write

$$\begin{aligned} \sum_{j=0}^{n-1} a_j \cos\left(\frac{j(2k+1)\pi}{2n}\right) &= y_k, \quad 0 \le k < n \\ \frac{2}{n} \sum_{k=0}^{n-1} y_k \cos\left(\frac{j(2k+1)\pi}{2n}\right) &= a_j \quad 1 \le j < n \\ \frac{1}{n} \sum_{k=0}^{n-1} y_k &= a_0. \end{aligned}$$

MATLAB has the `dct` and `idct` transform pair (see the HELP documenta-

tion)

$$\begin{aligned} y(k) &= w(k)\sum_{n=1}^{N} x(n)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \quad 1 \le k \le N \\ x(n) &= \sum_{k=1}^{N} w(k)y(k)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \quad 1 \le n \le N \\ w(1) &= \tfrac{1}{\sqrt{N}} \\ w(n) &= \tfrac{\sqrt{2}}{\sqrt{N}}, \qquad\qquad 2 \le n \le N \end{aligned}$$

which, if transformed back from MATLAB $1:N$ notation to standard $0:n-1$ notation of the discrete Fourier transform DFT, gives

$$\begin{aligned} Y(k) &= w(k)\sum_{j=0}^{n-1} X(j)\cos\left(\frac{\pi(2j+1)k}{2n}\right), \quad 0 \le k < n \\ X(j) &= \sum_{k=0}^{n-1} w(k)Y(k)\cos\left(\frac{\pi(2j+1)k}{2n}\right), \quad 0 \le j < n \\ w(0) &= \tfrac{1}{\sqrt{n}} \\ w(j) &= \tfrac{\sqrt{2}}{\sqrt{n}}, \qquad\qquad 1 \le j < n. \end{aligned} \tag{9.24}$$

To establish the connection to our previous form, we use the diagonal matrix W with the vector w on the diagonal. Then the second transformation above, written as $X = idct(Y)$, takes the form

$$X = idct(Y) = TWY$$

with our transformation matrix T of (9.23). Thus the MATLAB `idct` function acts like TW, while the MATLAB `dct` function is WT^T. Due to $T^{-1} = \frac{1}{n}D^{-1}T^T$ (in new notation $0:n-1$) and $\frac{1}{n}D^{-1} = W^2$ we have

$$\begin{aligned} WT^TTW &= WnDW \\ &= I, \end{aligned}$$

proving that the MATLAB functions `dct, idct` are indeed inverses of each other. Furthermore, we see that these functions agree with ours up to diagonal matrix transformations.

Theorem 9.25. *Interpolation in Chebyshev zeros by Chebyshev polynomials is connected to discrete cosine transforms by certain simple $\mathcal{O}(n)$ transformations by diagonal matrices.*

The discrete cosine transform will turn out to be a special case of the **discrete Fourier transform**, and thus it has a fast implementation via FFT. To see this, and to link our notation with standard DCT notation as in Wikipedia, we now look at the transform pair

$$\begin{aligned} z_j &= \sum_{k=0}^{n-1} x_k \cos\left(\frac{\pi(2k+1)j}{2n}\right),\ 0 \le j < n \\ x_k &= \frac{1}{2} z_0 + \sum_{j=1}^{n-1} z_j \cos\left(\frac{\pi(2k+1)j}{2n}\right),\ 0 \le k < n \end{aligned}$$

which is called DCT II and DCT III, respectively (see the Wikipedia), and which are not exactly inverses of each other, as is to be shown. If we write our first transforms in $0 : n-1$ notation in shorthand as

$$\begin{aligned} Ta &= y \\ T^{-1}y &= a, \end{aligned}$$

the above Wikipedia forms are

$$\begin{aligned} z &= T^T x \\ x &= T \begin{pmatrix} \frac{z_0}{2} \\ z_1 \\ \vdots \\ z_{n-1} \end{pmatrix} \\ &= \tfrac{1}{2} T D^{-1} z. \end{aligned}$$

Multiplication yields

$$\begin{aligned} T^T \tfrac{1}{2} T D^{-1} &= \tfrac{1}{2} T^T T D^{-1} \\ &= \tfrac{1}{2} n D D^{-1} \\ &= \tfrac{n}{2} I, \end{aligned}$$

such that the transformations are inverses of each other up to a scalar factor, as claimed by the Wikipedia. Also, we can now easily relate the Wikipedia forms of DCT II and DCT III to MATLAB functions `dct, idct` and to interpolation in Chebyshev zeros by Chebyshev polynomials.

9.2.3 Connection to the Discrete Fourier Transform

For establishing the connection to the discrete complex Fourier transform DFT (we assume that it is handled elsewhere), we use DCT II for simplicity.

In particular, we shall connect the transforms

$$\begin{aligned} z_j &= \sum_{k=0}^{n-1} x_k \cos\left(\frac{\pi(2k+1)j}{2n}\right), \quad 0 \le j < n \\ Z_j &= \sum_{k=0}^{4n-1} X_k \exp\left(\frac{2\pi ijk}{4n}\right), \qquad 0 \le j < 4n. \end{aligned} \tag{9.26}$$

If we start with the first (and this will yield a DFT implementation of the DCT), we go over to the second by setting

$$\begin{aligned} X_{2k} &= 0, \quad 0 \le k < 2n \\ X_{2k+1} &= x_k, \quad 0 \le k < n \\ X_{4n-(2k+1)} &= x_k, \quad 0 \le k < n. \end{aligned} \tag{9.27}$$

Then

$$\begin{aligned} Z_j &= \sum_{k=0}^{4n-1} X_k \exp\left(\frac{2\pi ijk}{4n}\right) \\ &= \sum_{k=0}^{n-1} X_{2k+1} \exp\left(\frac{2\pi ij(2k+1)}{4n}\right) + \sum_{k=0}^{n-1} X_{4n-(2k+1)} \exp\left(\frac{2\pi ij(4n-(2k+1))}{4n}\right) \\ &= 2\sum_{k=0}^{n-1} x_k \cos\left(\frac{2\pi j(2k+1)}{4n}\right) \\ &= 2\sum_{k=0}^{n-1} x_k \cos\left(\frac{\pi j(2k+1)}{2n}\right), \ 0 \le j < 4n. \end{aligned}$$

Thus $Z_j = 2z_j$ for $0 \le j < n$, but for the other indices we have different relations. Clearly, $Z_{4n-j} = Z_j$ for all $0 \le j < 4n$ and

$$\begin{aligned} Z_{n\pm j} &= 2\sum_{k=0}^{n-1} x_k \cos\left(\frac{2\pi(n \pm j)(2k+1)}{4n}\right) \\ &= 2\sum_{k=0}^{n-1} x_k \cos\left(\frac{2\pi(2kn \pm 2kj + n \pm j)}{4n}\right) \\ &= 2\sum_{k=0}^{n-1} x_k \cos\left(\frac{2\pi(\pm 2kj + n \pm j)}{4n}\right) \\ &= 2\sum_{k=0}^{n-1} x_k \cos\left(\frac{\pi}{2} + \frac{\pi(2k+1)(\pm j)}{2n}\right) \\ &= -2\sum_{k=0}^{n-1} x_k \cos\left(\frac{\pi}{2} - \frac{\pi(2k+1)(\pm j)}{2n}\right) \\ &= -Z_{n\mp j}, \ 0 \le j < n. \end{aligned}$$

This means that the Z_j are a cosine-like extension of the $2z_j$, i.e. $Z_0, \ldots, Z_{4n-1}$ are

$$2z_0, \ldots, 2z_{n-1}, 0, -2z_{n-1}, \ldots, -2z_1, -2z_0, -2z_1, \ldots, -2z_{n-1}, 0, 2z_{n-1}, \ldots, 2z_1. \tag{9.28}$$

If we have given data $x_0, \ldots, x_{n-1}$ for our cosine transform of length n in (9.26), we apply (9.27) first to get a vector of $4n$ values X_j. These are plugged into an FFT program implementing the second formula of (9.26), and the result will be (9.28), providing us with the required values of $z_0, \ldots, z_{n-1}$ with quite some overkill.

For the inverse transformation, we just have to go backwards, i.e. start by extending the $2z_j$ to the Z_j as in (9.28), do the inverse DCT transform, and get the X_j and the x_j related by (9.27).

Theorem 9.29. *The discrete cosine transform and interpolation in Chebyshev zeros by Chebyshev polynomials on n points can be implemented as a discrete Fourier transform of length $4n$. Thus there are FFT algorithms of complexity $n \log n$ for both the DCT and Chebyshev interpolation.*

There are more efficient implementations of the DCT, but we do not want to overdo it here.

But we add a little MATLAB m-file which tests all of the above.

```
% test Chebyshev interpolation, DCT and DFT via FFT
clear all;
close all;
n=5;
tz=cos((pi/(2*n+2):2*pi/(2*n+2):pi))'
T=fliplr(cheby(tz,n))
cond(T)
dv=ones(n+1,1)/2;
dv(1,1)=1;
D=diag(dv)
T'*T-(n+1)*D
Tinv=inv(D)*T'/(n+1)
Tinv*T
nn=n+1
wv=ones(nn,1)*sqrt(2)/sqrt(nn);
wv(1,1)=1/sqrt(nn);
W=diag(wv)
```

```
idct(eye(nn))-T*W
dct(eye(nn))-W*T'
x=rand(nn,1)
z=T'*x
xx=zeros(4*nn,1);
for j=0:nn-1
    xx(2*j+2,1)=x(j+1,1);
    xx(4*nn-2*j,1)=x(j+1,1);
end
xx
ccfull=real(fft(xx))/2
cc=ccfull(1:nn,1)
cc-z
ifft(ccfull)-xx/2
zz=zeros(4*nn,1);
for j=0:nn-1
    zz(j+1,1)=2*z(j+1,1);
    zz(nn+j+2,1)=-2*z(nn-j,1);
end
for j=0:2*nn-1
    zz(4*nn-j,1)=zz(j+2,1);
end
[zz,ccfull*2]
ci=real(ifft(zz))
[xx ci]
```

The function `cheby.m` is much like `polyval`:

```
function V=cheby(z,n)
% generates Chebyshev matrix for points z up to degree n
V(:,n+1) = ones(length(z),1);
V(:,n) = z;
for j = n-1:-1:1
    V(:,j) = 2*z.*V(:,j+1)-V(:,j+2);
end
```

9.2.4 DCT Compression

We have seen that the DCT performs a rescaled version of Chebyshev interpolation. But the connection is somewhat deeper, and we shall see experimentally that chopping the DCT and then doing the inverse DCT is a good

compression algorithm. Thus we now want to work towards understanding the compression effect in the DCT.

We do this in MATLAB style, i.e. we take a sequence $X(0), \ldots, X(n-1)$ interpreted as function values. These are transformed by (9.24) into a sequence $Y(0), \ldots, Y(n-1)$ which have the semantics of coefficients. There, small coefficients may be set to zero, and after backtransformation, the resulting values $\tilde{X}(0), \ldots, \tilde{X}(n-1)$ are interpreted as function values again.

What happens there? If naive users apply the DCT, the numbers $X(j)$ will be values

$$X(j) = f(a + \frac{h}{2} + j \cdot h),\ 0 \leq j < n$$

taken at equidistant data points with spacing $h > 0$ of a function f on $[a, b]$ with

$$b = a + \frac{h}{2} + (n-1) \cdot h + \frac{h}{2} = a + nh.$$

The interval $[a, b]$ can be mapped to $[0, \pi]$ by

$$\varphi = \pi \frac{x - a}{b - a}$$

such that

$$\varphi_j = \pi \frac{a + \frac{h}{2} + jh - a}{nh} = \pi \frac{2j+1}{2n},\ 0 \leq j < n.$$

Thus the equidistant points on $[a, b]$ go into equidistant angles φ_j which are related to the zeros x_j of T_n via

$$x_j = \cos\left(\frac{(2j+1)\pi}{2n}\right) = \cos(\varphi_j).$$

Due to

$$x = a + \varphi \frac{b - a}{\pi}$$

we can define a function

$$g(\varphi) := f\left(a + \varphi \frac{b-a}{\pi}\right)$$

with

$$g(\varphi_j) = f\left(a + \varphi_j \frac{b-a}{\pi}\right) = f(a + \frac{h}{2} + j \cdot h) = X(j),\ 0 \leq j < n.$$

However, in what follows the function g is considered to be even and 2π-periodic, because it is treated as an expansion into cosines. Thus what happens in the DCT is a trigonometric interpolation of an even periodic extension of f. This extension, if renormalized to 2π-periodicity, is exactly g. And since the interpolation preserves even trigonometric polynomials, the result is exactly the representation of $P_n(g)$ in the cosine basis. This fundamental observation controls the approximation and compression properties of the DCT.

If the function g obtained this way is in H^k, the exact Fourier coefficients $a_j(g)$ of g will have a decay like

$$|a_j(g)| \leq C(j+1)^{-k},\ j \geq 0$$

as we have seen when studying Fourier series. If the DCT would calculate the exact $a_j(g)$, this would explain the compression effect completely. Smooth functions g would need only a few large $|a_j(g)|$.

But the algorithm calculates the coefficients of $P_n(g)$ instead of g. Anyway, for $j \geq 1$ we know

$$a_j(g) - a_j(P_n(g)) \quad = \quad \frac{1}{\pi}\int_{-\pi}^{\pi} (g(\varphi) - P_n(g)(\varphi)) \cos j\varphi d\varphi$$

and this implies

$$|a_j(g) - a_j(P_n(g))| \leq \|g - P_n(g)\|_2 \| \cos j\varphi\|_2 = \|g - P_n(g)\|_2 \leq C(n+1)^{-k}$$

if we use the standard scaled L_2 inner product. Thus the decay behavior of the DCT coefficients is well comparable to the one of the exact Fourier coefficients of g, and the accuracy even increases with n.

This is fine, but there will be continuity problems when the even periodic extension of f does not lead to a smooth function g. Derivatives of f of odd order at the artificial symmetry points should be zero for perfect performance of the DCT. Boundary effects due to the even periodic extension can spoil part of the performance.

9.3 Splines

Zur graphischen Interpolation einer Reihe von Datenpunkten $(x_j, f_j),\ 0 \leq j \leq N$, mit einer **Knotenfolge**

$$X: \quad a = x_0 < x_1 < \ldots < x_N = b \ \text{ in } \ I := [a, b] \tag{9.30}$$

benutzten Konstrukteure früher statt eines Kurvenlineals auch häufig einen dünnen biegsamen Stab (**Straklatte**, engl. **spline**), den man durch Festklemmen zwang, auf dem Zeichenpapier die gegebenen Punkte zu verbinden. Anschließend konnte man dann längs des Stabes eine interpolierende Kurve zeichnen. Physikalisch ist die Lage, die der Stab zwischen den Datenpunkten einnimmt, durch ein Minimum der elastischen Energie charakterisiert, d.h. die Gesamtkrümmung, gegeben durch das Integral

$$\int_I \frac{(y''(t))^2}{1+y'^2(t)}dt, \tag{9.31}$$

wird durch die den Stab darstellende Funktion $s(t) \in C^2(I)$ unter allen anderen zweimal stetig differenzierbaren Interpolierenden y minimiert.

Für den Fall kleiner erster Ableitungen kann man das Integral (9.31) näherungsweise durch

$$\int_I y''(t)^2 \, dt \tag{9.32}$$

ersetzen. In der Variationsrechnung wird gezeigt, daß eine dieses Integral minimierende zweimal stetig differenzierbare Funktion s zwischen den Punkten x_j sogar viermal stetig differenzierbar ist und die Gleichung $s^{(4)}(x) = 0$ erfüllt. Daher ist s stückweise ein kubisches Polynom. Dies motiviert die folgende

Definition 9.33. *Die Funktionen aus dem linearen Raum*

$$\mathcal{S}_k(X) := \left\{ s \in C^{k-1}(I) \;\middle|\; s\big|_{[x_{i-1},x_i]} \text{ liegt in } \mathcal{P}_k,\ 1 \le i \le N \right\} \tag{9.34}$$

heißen **polynomiale Spline–Funktionen** *oder* **Splines** *vom Grad $\le k$ auf der Zerlegung X gemäß (9.30).*

Beispiel 9.35. *Im Falle $k = 1$ bestehen die Splines in $\mathcal{S}_1(X)$ aus stetigen, stückweise linearen Funktionen, d.h. aus* **Polygonzügen**. *Bei beliebigem $N \ge 1$ ist jedes* **Lagrange**[48]*-Interpolationsproblem*

$$s(x_i) = f_i,\ 0 \le i \le N, \quad mit \quad s \in \mathcal{S}_1(X)$$

eindeutig lösbar, und die Lösung ist durch die lokale lineare Interpolation von je zwei Datenpunkten einfach konstruierbar. Es besteht hier keine Verknüpfung von Polynomgrad ($k = 1$) und Stützstellenzahl N, und im Gegensatz zur Polynominterpolation läßt sich relativ leicht ein allgemeines Konvergenzresultat beweisen.

[48] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Lagrange.html`

Stammen die Daten $f_i = f(x_i)$ nämlich von einer Funktion $f \in C^2[a,b]$, so gilt nach Satz **??** *die Fehlerabschätzung*

$$|f(x) - s(x)| \leq \frac{1}{8} \, \|f''\|_\infty \cdot h^2$$

für alle $x \in [x_0, x_N]$ und $h := \max_{1 \leq i \leq n}(x_i - x_{i-1})$, weil man zwischen zwei Interpolationspunkten x_i und x_{i+1} stets $|(x - x_i)(x - x_{i+1})| \leq h^2/4$ hat. Für $h \to 0$ folgt also gleichmäßige Konvergenz der Interpolierenden, was nach Beispiel **??** *bei Polynominterpolation mit beliebigen Stützstellen nicht gewährleistet ist. In dieser Hinsicht ist die Spline-Interpolation der Polynom-Interpolation überlegen.*

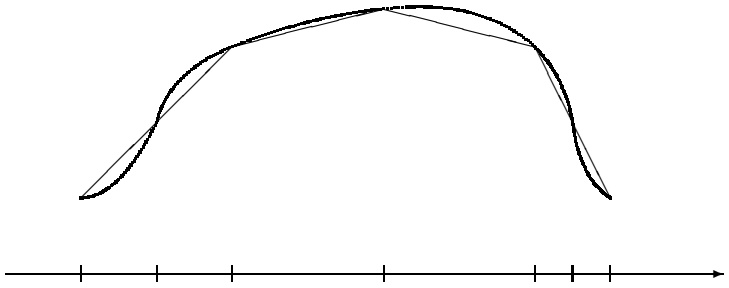

Figure 26: Polygonzug

Für die Praxis werden die im Falle $k = 3$ in Definition 9.33 auftretenden kubischen Splines am häufigsten verwendet; sie entsprechen ja auch dem eingangs dargestellten physikalischen Prinzip der Straklatte. Daher soll in diesem Abschnitt speziell für kubische Splines ein einfaches numerisches Konstruktionsverfahren für die Lösung des Interpolationsproblems im Falle von **Lagrange**[49]-Vorgaben angegeben werden. Allgemeinere Methoden zur Berechnung von Kurven und Flächen mit Spline-Funktionen finden sich in Abschnitt 9.4.

Zu festen Knoten (9.30) seien Interpolationsdaten $f_0 \ldots, f_N \in \mathbb{R}$ vorgegeben. Auf jedem der Teilintervalle $I_j := [x_{j-1}, x_j]$ ist die zweite Ableitung einer Funktion s aus $\mathcal{S}_3(X)$ linear. Mit den Abkürzungen

$$\begin{aligned} h_j &:= x_j - x_{j-1} \qquad (1 \leq j \leq N) \\ M_j &:= s''(x_j) \qquad (0 \leq j \leq N) \end{aligned} \tag{9.36}$$

[49] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Lagrange.html`

gilt also

$$s''(x) = \frac{1}{h_j}\ (M_j(x - x_{j-1}) + M_{j-1}(x_j - x)) \qquad \text{für alle } x \in I_j. \tag{9.37}$$

Daraus folgt für die Restriktion von s auf $[x_{j-1}, x_j]$ durch zweimalige Integration

$$s(x) = \frac{1}{6h_j}\ (M_j(x-x_{j-1})^3 + M_{j-1}(x_j-x)^3) + b_j\ \left(x - \frac{x_j + x_{j-1}}{2}\right) + a_j \tag{9.38}$$

mit gewissen Integrationskonstanten $a_j,\ b_j$. Unter Benutzung der Interpolationsbedingungen soll daraus ein Gleichungssystem für die Parameter $M_j,\ a_j,\ b_j$ hergeleitet werden. Bedient man sich der Identität

$$\begin{aligned}
&(h_j + h_{j+1})\ \Delta^2(x_{j-1}, x_j, x_{j+1})f \\
&= \Delta^1(x_j, x_{j+1})f - \Delta^1(x_j, x_{j-1})f \\
&= (\Delta^1(x_j, x_{j+1})f - \Delta^1(x_j, x_j)f) + (\Delta^1(x_j, x_j)f - \Delta^1(x_{j-1}, x_j)f) \\
&= h_{j+1}\ \Delta^2(x_j, x_j, x_{j+1})f + h_j\ \Delta^2(x_{j-1}, x_j, x_j)f
\end{aligned} \tag{9.39}$$

und berücksichtigt, daß bei der Bildung zweiter Differenzenquotienten lineare Funktionen annulliert werden, so erhält man aufgrund der vorgegebenen Werte f_j einerseits und der Form (9.38) von $s(x)$ andererseits die Gleichungen

$$\begin{aligned}
&(h_j + h_{j+1})\ \Delta^2(x_{j-1}, x_j, x_{j+1})f = (h_j + h_{j+1})\ \Delta^2(x_{j-1}, x_j, x_{j+1})s \\
&= h_{j+1} \cdot \frac{1}{6h_{j+1}}\ (M_{j+1}h_{j+1} + 2M_j h_{j+1}) + h_j \cdot \frac{1}{6h_j}\ (2M_j h_j + M_{j-1}h_j).
\end{aligned} \tag{9.40}$$

Durch Multiplikation mit $3 \cdot (h_j + h_{j+1})^{-1}$ erhält man schließlich das nur noch die M_j als Unbekannte enthaltende lineare Gleichungssystem

$$\mu_j M_{j-1} + M_j + \lambda_j M_{j+1} = 3 \cdot \Delta^2(x_{j-1}, x_j, x_{j+1})f \tag{9.41}$$

für $j = 1, \ldots, N-1$ mit den Größen

$$\mu_j := \frac{h_j}{2(h_j + h_{j+1})}\ , \quad \lambda_j := \frac{h_{j+1}}{2(h_j + h_{j+1})}\ , \quad \lambda_j + \mu_j = \frac{1}{2}. \tag{9.42}$$

In (9.41) sind die Randwerte noch **nicht** berücksichtigt.

Bezüglich der Randvorgaben kann man 3 Fälle unterscheiden:

a) Es seien zusätzlich feste Werte für M_0 und M_N vorgeschrieben. Dann ist durch (9.41) bereits ein System von $N-1$ Gleichungen mit $N-1$ Unbekannten gegeben. Will man eine Straklatte simulieren, die aus physikalischen Gründen außerhalb der Interpolationspunkte immer geradlinig verläuft, wird man einfach $M_0 = M_N = 0$ setzen und erhält dann die sogenannten **natürlichen** Splines.

b) Soll s periodisch sein, so identifiziert man

$$M_0 = M_N, \;\; M_{N+1} = M_1, \;\; f_{N+1} = f_1, \;\; h_{N+1} = h_1$$

und bildet damit (9.41) für die Indizes $j = 1, \ldots, N$ mit den Unbekannten $M_1, \ldots, M_N$. Dies liefert N Gleichungen für N Unbekannte.

c) Sind zusätzlich zwei reelle Zahlen u, v vorgegeben und wird

$$s'(x_0) = u, \;\; s'(x_N) = v$$

gefordert, so folgen mit (9.38) die zusätzlichen Gleichungen

$$\begin{aligned} M_0 + \frac{1}{2} M_1 &= 3\Delta^2(x_0, x_0, x_1)f &&= \frac{3}{x_0 - x_1}\,(u - \Delta^1(x_0, x_1)f), \\ \frac{1}{2} M_{N-1} + M_N &= 3\Delta^2(x_{N-1}, x_N, x_N)f &&= \frac{3}{x_N - x_{N-1}}(\Delta^1(x_{N-1}, x_N)f - v). \end{aligned}$$

Definiert man

$$x_{-1} := x_0, \quad x_{N-1} := x_N, \quad h_0 := h_{N+1} := 0,$$

so hat man in diesem Fall $N+1$ Gleichungen der Form (9.41) für $0 \le j \le N$ zur Bestimmung der $N+1$ Unbekannten $M_0, \ldots, M_N$.

d) Hat man keine Ableitungsrandwerte zur Verfügung, so ist das Erzwingen von $M_0 = M_N = 0$ im Falle natürlicher Splines keineswegs natürlich, sondern sollte durch eine andere, weniger willkürliche Strategie ersetzt werden. Die sogenannte "**not-a-knot**"-Bedingung benutzt die unbestimmten Parameter an den Rändern, um die äußere Sprungstelle der dritten Ableitung zu eliminieren; dann liegt in $[x_0, x_2]$ und $[x_{N-2}, x_N]$ nur je ein kubisches Polynomstück vor.

Aus (9.37) folgt

$$s'''(x) = \frac{1}{h_j}\,(M_j - M_{j-i}) \quad \text{auf } [x_{j-i}, x_j]$$

und man hat $s'''(x_1^-) = s'''(x_1^+)$ genau dann, wenn

$$\frac{1}{h_1}(M_1 - M_0) = \frac{1}{h_2}(M_2 - M_1) \tag{9.43}$$

gilt. Das bedeutet

$$M_0 = M_1 - \frac{h_1}{h_2}(M_2 - M_1) = \frac{1}{h_2}((h_1 + h_2)M_1 - h_1 M_2)$$

und man kann (9.41) für $j = 1$ durch Elimination von M_0 modifizieren oder (9.43) zu (9.41) hinzufügen. Letztere Strategie führt noch zu einer Matrix, die das schwache Zeilensummenkriterium erfüllt und deshalb nichtsingulär ist.

Wegen

$$\lambda_j + \mu_j = \frac{1}{2} \quad \text{und} \quad \lambda_j \geq 0, \;\; \mu_j \geq 0$$

sind die Koeffizientenmatrizen der resultierenden linearen Gleichungssysteme in den Fällen a) – c) diagonaldominant und wegen des Satzes **??** von **Gerschgorin**[50] nicht singulär.

In den Fällen a), c) und d) ist die Koeffizientenmatrix tridiagonal. Dann läßt sich die Lösung des Gleichungssystems nach dem Eliminationsverfahren von **Gauss**[51] mit höchstens $\mathcal{O}(N)$ Punktoperationen durchführen (vgl. Aufgaben **??** und **??**). Stabilitätsprobleme ergeben sich nicht, da die Matrix diagonaldominant ist. Auch der periodische Fall läßt sich mit $\mathcal{O}(N)$ Operationen lösen.

Betrachtet man das Interpolationsproblem

$$\begin{aligned} s(x_j) &= f(x_j) \qquad (0 \leq j \leq N) \\ s''(x_j) &= f''(x_j) \qquad (j = 0, N) \end{aligned}$$

zur Zerlegung (9.30) mit $f \in C^4[a,b]$ und einem kubischen Spline, so ergibt sich aus der Identität (9.39) mit

$$\begin{aligned} A_j := \;& 6\lambda_j \Delta^2(x_j, x_j, x_{j+1})f + 6\mu_j \Delta^2(x_{j-1}, x_j, x_j)f \\ & -\lambda_j(f''(x_{j+1}) + 2f''(x_j)) - \mu_j(2f''(x_j) + f''(x_{j-1})) \end{aligned} \tag{9.44}$$

[50] http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Gerschgorin.html
[51] http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Gauss.html

die Gleichung

$$3\Delta^2(x_{j-1}, x_j, x_{j+1})f - A_j = \mu_j f''(x_{j-1}) + f''(x_j) + \lambda_j f''(x_{j+1})$$

und durch Subtraktion von (9.41) folgt, daß die Werte $\varepsilon_j'' := s''(x_j) - f''(x_j)$ das System

$$\mu_j \varepsilon_{j-1}'' + \varepsilon_j'' + \lambda_j \varepsilon_{j+1}'' = A_j \tag{9.45}$$

erfüllen. Als Anwendung des Satzes von **Peano**[52] in Beispiel **??** liefert (**??**) die Abschätzung

$$|A_j| \leq \frac{1}{8} h^2 \|f^{(4)}\|_\infty$$

mit $h = \max_j(x_{j+1} - x_j)$. Da gleichmäßig in h das Gleichungssystem (9.45) eine diagonaldominante Matrix besitzt von der Form $E + B$ mit $\|B\|_\infty = \frac{1}{2}$, ist die Lösung durch die rechte Seite gleichmäßig abschätzbar, denn es gilt

$$\|(E+B)^{-1}\|_\infty = \left\| \sum_{j=0}^{\infty} (-1)^j B^j \right\|_\infty \leq \sum_{j=0}^{\infty} \|B\|_\infty^j = \frac{1}{1 - \frac{1}{2}} = 2.$$

Man erhält

$$\max_{0 \leq j \leq N} |s''(x_j) - f''(x_j)| \leq \frac{1}{4} h^2 \|f^{(4)}\|_\infty. \tag{9.46}$$

Ist $u(x)$ ein Polygonzug durch die Werte $(x_j, f''(x_j))$, so folgt für $x \in [x_{j-i}, x_j]$ nach der Konvergenzbetrachtung für Polygonzüge in Beispiel 9.35 die Fehlerabschätzung

$$|f''(x) - u''(x)| \leq \frac{h^2}{8} \, \|f^{(4)}\|_\infty.$$

Für den Fehler $s'' - f''$ ergibt sich wegen $u(x_j) = f_j''$

$$\begin{aligned} \|s'' - f''\|_\infty &\leq \|s'' - u\|_\infty + \|u - f''\|_\infty \leq \max_j |s''(x_j) - f''(x_j)| + \frac{h^2}{8} \cdot \|f^{(4)}\|_\infty \\ &\leq \frac{3}{8} \, h^2 \|f^{(4)}\|_\infty. \end{aligned}$$

Damit erhält man einen Teil von

Satz 9.47. *Die kubische Spline-Interpolierende $s \in C^2[a,b]$ der* **Lagrange**[53]*-Daten einer Funktion $f \in C^4[a,b]$ in den Punkten*

$$a = x_0 < x_1 < \ldots < x_N = b$$

[52] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Peano.html`
[53] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Lagrange.html`

mit den Randbedingungen

$$s''(a) = f''(a), \quad s''(b) = f''(b)$$

genügt mit $h := \max_j |x_j - x_{j-i}|$ *den Abschätzungen*

$$\|s^{(j)} - f^{(j)}\|_\infty \leq \frac{3}{8}\, h^{4-j} \|f^{(4)}\|_\infty, \qquad j = 0, 1, 2.$$

Der noch offene Beweis der Fälle $j = 0$ und 1 ergibt sich durch einfache Anwendung des Satzes von **Rolle**[54] und des obigen Lemmas.

9.4 B-Splines

Bei der praktischen Rechnung mit Spline-Funktionen aus dem schon in 9.33 definierten Raum

$$\mathcal{S}_k(X) := \left\{ s \in C^{k-1}(I) \;\mid\; s|_{[x_{i-1},x_i]} \quad \text{ist in } \mathcal{P}_k,\ 1 \leq i \leq N \right\}$$

mit der Zerlegung

$$X \;:\; a \leq x_0 < x_1 < \ldots < x_N \leq b$$

kommt es darauf an, möglichst einfach handzuhabende Basen zu finden. Beispielsweise kann man versuchen, spezielle Spline-Funktionen zu konstruieren, die jeweils nur auf einem möglichst kleinen Teilintervall von Null verschieden sind und eine Zerlegung der Eins bilden.

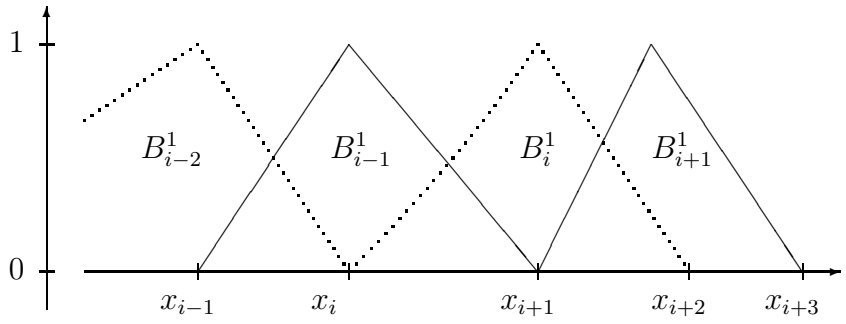

Figure 27: B-Splines ersten Grades

[54] http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Rolle.html

Beispiel 9.48. *Im Falle $k = 1$ der Polygonzüge ist das besonders einfach; bis auf einen Faktor kann man die "Dach-Funktionen"*

$$B_j^1(t) := \left\{ \begin{array}{cc} \frac{t - x_j}{x_{j+1} - x_j} & x_j \leq t \leq x_{j+1} \\ \frac{x_{j+2} - t}{x_{j+2} - x_{j+1}} & x_{j+1} \leq t \leq x_{j+2} \\ 0 & \textit{sonst} \end{array} \right\} \tag{9.49}$$

mit dem in Abb. 27 gezeigten Verlauf nehmen. Mit der schon beim Satz von **Peano**[55] *in (??) verwendeten* **abgeschnittenen Potenzfunktion**

$$(x-t)_+^k := \left\{ \begin{array}{cc} (x-t)^k & x - t > 0, \quad k \geq 0 \\ 1/2 & x - t = 0, \quad k = 0 \\ 0 & \text{sonst} \end{array} \right\}$$

für $x,\ t \in \mathbb{R},\ k \geq 0$ läßt sich durch Einsetzen der Alternativen für t aus (9.49) verifizieren, daß

$$\begin{aligned} & \frac{(x_{j+2} - t)_+^1 - (x_{j+1} - t)_+^1}{x_{j+2} - x_{j+1}} - \frac{(x_{j+1} - t)_+^1 - (x_j - t)_+^1}{x_{j+1} - x_j} \\ = \ & (x_{j+2} - x_j)\Delta_x^2(x_j, x_{j+1}, x_{j+2})(x-t)_+^1 \\ = \ & B_j^1(t) \end{aligned}$$

gilt. Das motiviert

Definition 9.50. *Zu allen $i \in \mathbb{Z}$ seien paarweise verschiedene Punkte $x_i \in \mathbb{R}$ mit $-\infty < \ldots < x_{-1} < x_0 < x_1 \ldots < \infty$ vorgegeben. Dann heißen die Funktionen*

$$B_j^r(t) := (x_{j+r+1} - x_j)\Delta_x^{r+1}(x_j, \ldots, x_{j+r+1})(x-t)_+^r \tag{9.51}$$

*(für $j \in \mathbb{Z},\ r \geq 0$) auch B-***Splines**.

Beispiel 9.52. *Man erhält für $r = 0$ auch*

$$B_j^0(t) := (x_{j+1} - t)_+^0 - (x_j - t)_+^0 = \left\{ \begin{array}{cr} 0 & x_{j+1} < t \\ 1 & x_j < t < x_{j+1} \\ 0 & t < x_j \end{array} \right. \tag{9.53}$$

Diese Funktionen bilden natürliche Basen für Räume von Treppenfunktionen. In den Sprungstellen wird das Mittel des rechts- und linksseitigen Grenzwertes genommen.

[55] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Peano.html`

Beispiel 9.54. *Für dieselbe Knotenverteilung wie in Abb. 27 zeigen die Abbildungen 27 bzw. 28 die quadratischen bzw. kubischen B-Splines.*

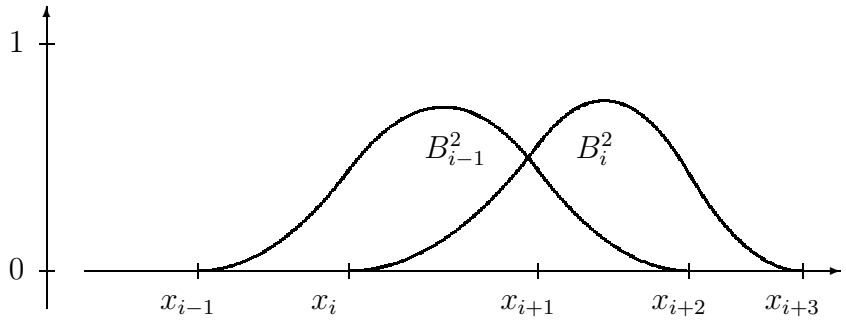

Figure 27: B-Splines zweiten Grades

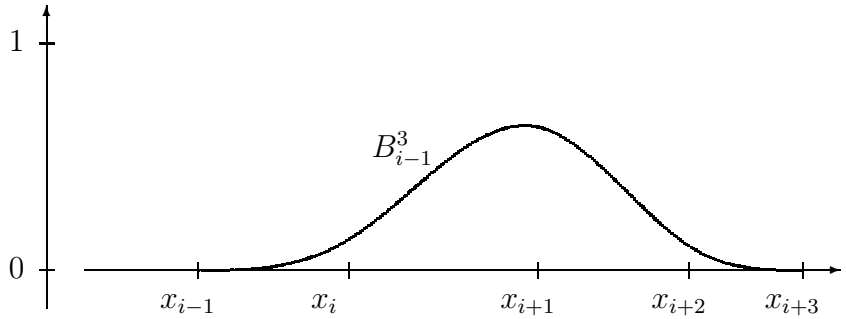

Figure 28: B-Spline dritten Grades

Satz 9.55. *Für $r \geq 1$ haben die B-Splines folgende Eigenschaften:*

$$B_j^r \in C^{r-1}(\mathbb{R}), \quad \text{falls } r \geq 1 \text{ (sonst stückweise stetig)} \tag{9.56}$$

$$B_j^r \in \mathcal{P}_r \text{ in } (x_j, x_{j+r+1}) \tag{9.57}$$

$$B_j^r = 0 \text{ in } (x_{j+r+1}, \infty) \text{ und } (-\infty, x_j) \tag{9.58}$$

$$B_j^r(t) = \frac{x_{j+r+1} - t}{x_{j+r+1} - x_{j+1}} B_{j+1}^{r-1}(t) + \frac{t - x_j}{x_{j+r} - x_j} B_j^{r-1}(t) \tag{9.59}$$

$$B_j^r(t) > 0 \quad \textit{für } t \in (x_j, x_{j+r+1}), \qquad r \geq 0 \tag{9.60}$$

$$\sum_j B_j^{(r)}(t) = 1 \qquad \textit{für alle } t \in \mathbb{R},\ r \geq 0. \tag{9.61}$$

Beweis: Die Aussagen (9.56), (9.57) und (9.58) sind klar. Für $r \geq 1$ folgt

$$\begin{aligned}
& (x_{j+r+1} - x_j)^{-1} B_j^r(t) = \\
= \;& \Delta_x^{r+1}(x_j, \ldots, x_{j+r+1})[(x-t)_+^{r-1}(x_{j+r+1} - t + x - x_{j+r+1})] \\
= \;& \Delta_x^{r+1}(x_j, \ldots, x_{j+r+1})[(x-t)_+^{r-1}(x_{j+r+1} - t)] \\
& + \Delta_x^{r+1}(x_j, \ldots, x_{j+r+1})[(x-t)_+^{r-1}(x - x_{j+r+1})] \\
= \;& (x_{j+r+1} - t)\Delta_z^1(x_j, x_{j+r+1})\Delta_x^r(z, x_{j+1}, \ldots, x_{j+r})(x-t)_+^{r-1} \\
& + \Delta_x^r(x_j, \ldots, x_{j+r})\Delta_z^1(x, x_{j+r+1})[(z-t)_+^{r-1}(z - x_{j+r+1})] \\
= \;& (x_{j+r+1} - t)(x_{j+r+1} - x_j)^{-1}((x_{j+r+1} - x_{j+1})^{-1} B_{j+1}^{r-1}(t) - (x_{j+r} - x_j)^{-1} B_j^{r-1}(t)) \\
& + \Delta_x^r(x_j, \ldots, x_{j+r})(x-t)_+^{r-1}
\end{aligned}$$

mit Aufgabe **??** und

$$\begin{aligned}
\Delta_z^1(x, x_{j+r+1})[(z-t)_+^{r-1}(z - x_{j+r+1})] &= \frac{(x-t)_+^{r-1}(x - x_{j+r+1}) - 0}{x - x_{j+r+1}} \\
&= (x-t)_+^{r-1}.
\end{aligned}$$

Das ergibt

$$\begin{aligned}
B_j^r(t) = \;& \frac{x_{j+r+1} - t}{x_{j+r+1} - x_{j+1}} \, B_{j+1}^{r-1}(t) - \frac{x_{j+r+1} - t}{x_{j+r} - x_j} \, B_j^{r-1}(t) \\
+ \;& \frac{x_{j+r+1} - x_j}{x_{j+r} - x_j} \, B_j^{r-1}(t)
\end{aligned}$$

und daher gilt (9.59). Jetzt ist (9.60) leicht induktiv nachzuweisen; als Induktionsanfang nimmt man (9.53). Gilt (9.60) für B_i^{r-1} und alle $i \in \mathbb{Z}$, so hat B_j^r für alle $t \in (x_j, x_{j+r+1})$ in der Darstellung (9.59) als Linearkombination positive Gewichte und es ist mindestens ein Summand positiv. Ebenso beweist man (9.61) durch Induktion, wobei man mit (9.53) beginnt und (9.59) zum Induktionsschluß heranzieht. Damit ist der Satz bewiesen. □

Ist eine (nur theoretisch infinite) Knotenfolge

$$\ldots x_{-1} < x_0 < x_1 < x_2 \ldots$$

in $\mathbb{R}$ gegeben, so bilden die zugehörigen B-Splines nach Satz 9.55 eine positive Zerlegung der Eins und man kann zu festem Grad $r \geq 1$ allgemeine Linearkombinationen

$$s(t) = \sum_i d_i B_i^r(t) \tag{9.62}$$

von normierten B-Splines betrachten, wobei die Koeffizienten d_i hier vektorwertig aus $\mathbb{R}^d$ sind, als Kontrollpunkte fungieren und **de-Boor**-Punkte genannt werden. Weil $B_i^r(t)$ nur für $t \in (x_i, x_{i+r+1})$ von Null verschieden ist, werden in (9.62) stets nur endlich viele Terme summiert, obwohl die Summe hier und im folgenden stets über alle $i \in \mathbb{Z}$ erstreckt wird. Ferner gilt offensichtlich

Satz 9.63. *Verändert man in einer B-Spline-Kurve (9.62) den* **de-Boor**-*Punkt d_i, so verändert sich die Kurve nur im Bild von (x_i, x_{i+r+1}).* □

Die punktweise Auswertung einer Spline-Kurve

$$s(t) = \sum_j d_j B_t^r(t) \tag{9.64}$$

erfolgt nicht notwendig über die Rekursionsformel der einzelnen B-Splines, sondern über eine zum **de Casteljau**-Verfahren analoge Methode von **de Boor**, die auf der Anwendung der Rekursion (9.59) der B-Splines basiert:

$$\begin{aligned}
s(t) &= \sum_j d_j \left(\frac{x_{j+r+1} - t}{x_{j+r+1} - x_{j+1}} B_{j+1}^{r-1}(t) + \frac{t - x_j}{x_{j+r} - x_j} B_j^{r-1}(t) \right) \\
&= \sum_j B_j^{r-1}(t) \left(\frac{x_{j+r} - t}{x_{j+r} - x_j} d_{j-1} + \frac{t - x_j}{x_{j+r} - x_j} d_j \right) \\
&=: \sum_j B_j^{r-1}(t) d_j^{(1)}(t) = \ldots = \\
&= \sum_j B_j^0(t) d_j^{(r)}(t) \\
&= d_k^{(r)}(t) \text{ falls } x_k < t < x_{k+1}
\end{aligned}$$

bei geeigneter Formulierung einer Rekursionsformel für die $d_j^{(r)}(t)$. Ist t ein fester Punkt aus (x_k, x_{k+1}), so ist die Summe in (9.64) nur über $j = k - r, \ldots, k$ zu erstrecken, weil die übrigen B-Splines in t verschwinden. Es folgt

Satz 9.65. *Es sei $\ldots x_{-1} < x_0 < x_1 < \ldots$ eine Knotenfolge mit einer B-Spline-Linearkombination*

$$s(t) = \sum_j d_j B_j^r(t). \tag{9.66}$$

Ist dann $t \in (x_k, x_{k+1})$ ein fester Punkt, so liefert die Rekursion

$$d_j^{(0)}(t) := d_j, \qquad k-r \le j \le k,$$

$$d_j^{(\ell+1)}(t) := \frac{x_{j+r-\ell} - t}{x_{j+r-\ell} - x_j}\; d_{j-1}^{(\ell)}(t) + \frac{t - x_j}{x_{j+r-\ell} - x_j}\; d_j^{(\ell)}(t),$$

$$k-r+\ell+1 \le j \le k, \quad \ell = 0, 1, \ldots, r-1 \tag{9.67}$$

als $d_k^{(r)}(t)$ den Wert $s(t)$.

Das Verfahren (9.67) von **de Boor** ist einerseits wie das **De-Casteljau**-Verfahren zur Berechnung einzelner Funktionswerte verwendbar; faßt man andererseits die $d_j^{(\ell)}$ als Polynome auf, so ist $d_k^{(r)}$ das Polynom, mit dem s in (x_k, x_{k+1}) übereinstimmt.

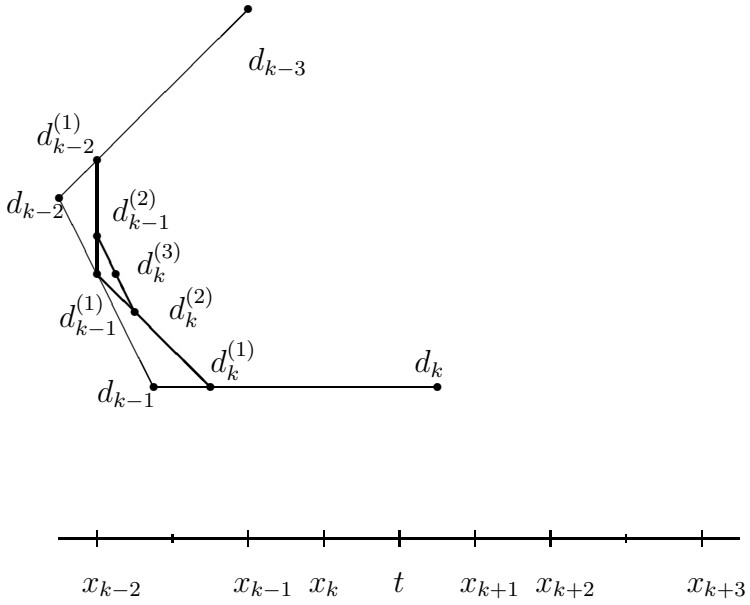

Figure 29: De–Boor–Verfahren

Bemerkung 9.68. *Wegen $x_j \le x_k < x_{k+1} \le x_{j+r-\ell}$ verschwinden die Nenner in*
(9.67) auch dann nicht, wenn mehrfache Knoten zugelassen werden. Die Einschränkung auf das offene Intervall (x_k, x_{k+1}) ist nur für $r = 0$ relevant, im Normalfall $r \ge 1$ ist aus Stetigkeitsgründen $t \in [x_k, x_{k+1}]$ wählbar.

Der Aufwand des Verfahrens ist etwa $\mathcal{O}(r^2 d)$ für jeden festen Punkt t. Die Zahl der insgesamt vorhandenen Terme in (9.66) ist irrelevant, weil immer

nur $r+1$ der B-Splines an einer Stelle t nötig sind. Natürlich ist das bei Auswertung vieler Werte in einem festen Intervall $[x_k, x_{k+1}]$ nicht gegenüber den schon behandelten effizienten Polynomauswertungsverfahren konkurrenzfähig. Normalerweise ist bei Splines der Grad r aber klein gegen die Anzahl der B-Splines, so daß der Mehraufwand des **de-Boor**-*Verfahrens nicht ins Gewicht fällt.*

Die numerische Anwendung von B-Spline-Darstellungen wird erleichtert durch folgende zusätzliche Eigenschaften:

Satz 9.69. *Die B-Splines erfüllen die Gleichungen*

$$\frac{d}{dt}\, B_j^{(r)}(t) = r\left(\frac{B_j^{(r-1)}(t)}{x_{j+r}-x_j} - \frac{B_{j+1}^{(r-1)}(t)}{x_{j+r+1}-x_{j+1}}\right)$$

$$\int_{\mathbb{R}} B_j^{(r)}(t) = 1$$

$$\triangle_t^{r+1}(x_j,\ldots,x_{j+r+1})f(t) = \frac{1}{(r+1)!}\,\frac{1}{x_{j+r+1}-x_j}\,\int_{\mathbb{R}} B_j^{(r)}(t) f^{(r+1)}(t)dt,$$

d.h. der B-Spline ist bis auf die Normierung der **Peano**[56]*-Kern des Differenzenquotienten. Für Linearkombinationen*

$$s(t) = \sum_j d_j\, B_j^{(r)}(t)$$

sind die Formeln

$$\begin{aligned} s'(t) &= \sum_j \frac{r(d_j-d_{j-1})}{x_{j+r}-x_j}\, B_j^{(r-1)}(t) \\ \int_{-\infty}^{x} s'(t)dt &= \sum_j D_j B_j^{(r+1)}(x) \qquad \textit{mit} \\ D_j &= D_{j-1} + d_j(x_{j+r+1}-x_j)/(r+1). \end{aligned}$$

bei Differentiation und Integration nützlich.

Aufgabe 9.70. *Man beweise Satz 9.69.*

Sowohl das Verfahren von **de Casteljau** als auch das Verfahren von **de Boor** bilden neue Kontrollpunkte als Konvexkombinationen alter Kontrollpunkte. Setzt man formal $a = x_{k-r} = \ldots = x_k < b = x_{k+1} = \ldots = x_{k+r}$, so gehen beide Verfahren ineinander über. Die **Bernstein-Bezier**-Darstellung eines Polynoms r-ten Grades über $[a,b]$ ist somit formal identisch zu einer B-Spline-Darstellung mit zwei je r-fachen Knoten.

[56] `http://www-history.mcs.st-andrews.ac.uk/~history/Biographies/Peano.html`

Index

B-Spline, -Kurven, 202
B-Spline, Definition, 199
de Boor, -Punkt, 202
de Boor, -Verfahren, 202

de Boor, 204
de-Boor, 202
De-Casteljau, 203
Lagrange, 192

AD–Wandler, 83
Aliasing, 99
Alternante, 19
Alternation, 16
Analyse, 31
Approximation
    beste, 7
    diskrete, 11
Approximationsfehler, 4
Approximationssatz
    Weierstraß, 46
Ausgleichsrechnung
    linear, 28

Banachraum, 32
baryzentrische Auswertung, 75
Bermsteinoperator, 48
Bernstein-Bezier, 204
Bernsteinpolynome, 49
Besselsche Ungleichung, 30
beste k–Term–Approximation, 31
beste Approximation, 4, 7

Cosinustransformation, 74

DA–Wandler, 83
de Boor, 202, 203, 205, 206
de Casteljau, 202, 204
de-Boor, 202, 204
dicht, 30
Differenzenquotienten, 21
Dirichlet–Kern, 66
diskrete Approximation, 11
diskrete wavelet–Transformation, 130

Einheitskugel, 9
Einheitswurzel, 57
euklidisch, 25
Existenzmenge, 7

Fast Fourier Transform, 59
Folge
    Korovkin-, 46
Fourier
    Koeffizienten, 54
    Transformation, 54
Fourier Inversion Theorem, 175
Fourier transform, 172
Fourier–Partialsumme, 43
Fourier-Koeffizienten, 54
Fourier-Transformation, 54
Fourierreihe, 54
Fouriertransformation, 84
Fouriertransformierte, 84
Frequenzraum, 84
Friedrich's mollifier, 180

Gaußglocke, 85
Gauss, 196
Gaussian, 172
Gerschgorin, 196
Gramsche Matrix, 28

Haar wavelet, 122
harmonischen Analyse, 56
Hilbertraum, 25

Idempotenz, 29

Integralformel von Cauchy, 82
inverse Fouriertransformation, 84
inverse Fouriertransformierte, 84

Kompression, 31
konvex, 9
Konvexkombination, 9
Korovkin-Folge, 46
Kubische Splines, 193
Kurven, B-Spline-, 202

Lagrange, 193, 197
Lebesgue–Konstanten, 65
Legendre-Polynome, 36
lineare Ausgleichsrechnung, 11

Maximumsnorm, 11, 46
Methode der kleinsten Quadrate, 11
Minimalabstand, 7
Minimalfolge, 33
Monombasis, 18

Norm, 7
    Maximums–, 11
    Tschebyscheff–, 11

Operator
    Bernstein-, 48
    monotoner, 46
Optimierung, 4, 11
orthogonal, 25, 28
orthonormal, 28
Orthonormalbasis, 28
Ortsraum, 84

Parallelogrammgleichung, 25
Parseval-Plancherel-sche Gleichung, 86
Parsevalsche Gleichung, 30
Peano, 197, 199, 204
periodisch, 42
Poisson'sche Summenformel, 92
Polynome, 17
    trigonometrische, 42
positive definite function, 173
Prä-Hilbertraum, 25
Projektion, 29
Pythagoras, 25, 27

Referenz, 20
Remez–Algorithmus, 22
Rolle, 198

Sampling Theorem, 91
separabel, 34
Simplexverfahren, 24
Skalarprodukt, 25
Skalierungsfunktion, 121, 131
Sobolevräume, 56
Spline-Funktionen, B-Splines, 198
Spline-Funktionen, kubische, 193
Spline-Funktionen, natürliche, 195
Spline-Funktionen, polygonale, 192
Spline-Funktionen, polynomiale, 192
Stützstellen
    äquidistante, 57
Stone–Weierstraß–Sätze, 53
strikt konvex, 10
Symbol, 105
Synthese, 31

tempered test functions, 172
Testfunktionen, 85
Tschebyscheff, -Polynome, 37
Tschebyscheff–Approximation, 12
Tschebyscheff–Norm, 11
Tschebyscheff-Menge, 7
Tschebyscheff-Norm, 46

Ungleichung
    Besselsche, 30
Upsampling, 94

Vandermondematrix, 19
Verfahren, **de Boor**-, 202
Verfeinerungsgleichung, 121

Vervollständigung, 33
vollständig, 30

wavelet–Rekonstruktion, 130
Wavelet–Transformation, 161
wavelet–Zerlegung, 129
Weierstraß, 46, 49, 52

References

[1] Berrut, J., and Trefethen, L. Barycentric lagrange interpolation. *SIAM Review 46* (2004), 501–517.

[2] Jetter, K., and Plonka, G. A survey on L_2-approximation orders from shift-invariant spaces. In *Multivariate approximation and applications*. Cambridge Univ. Press, Cambridge, 2001, pp. 73–111.

[3] Nevai, P. G. *Orthogonal Polynomials*. Amer. Math. Soc., 1979.

[4] Schaback, R., and Wendland, H. *Numerische Mathematik*. Springer-Lehrbuch. Springer Verlag, 2004. Fifth edition.

[5] Szegoe, G. *Orthogonal Polynomials*, vol. 23 of *Colloquium Publications*. American Mathematical Society, 1959.

[6] Tricomi, F. G. *Vorlesungen uber Orthogonalreihen*, vol. 76 of *Grundlehren der Mathematischen Wissenschaften*. Springer-Verlag, 1955.