Approximationsverfahren I

R. Schaback

16. April 2007

Vorwort

Dieser Text ist zusammengestoppelt aus älteren Bestandteilen und neuen Zutaten, und er dient als Hintergrundtext zur Vorlesung "Approximationsverfahren I" an der Universität Göttingen im WS 2006/2007. Er beschränkt sich auf Approximation von und mit **univariaten** Funktionen, denn die multivariaten Funktionen sollen in "Approximatinsverfahren II" drankommen.

Die Beamer–Folien habe ich einfach in den Text hineinkopiert, und sie stehen in der Regel vor den zugehörigen Texten, was die Numerierung etwas hakelig erscheinen läßt. An vielen Stellen fehlt zusätzlicher Standardtext, der aus der Literatur zu nehmen ist, und dazu gibt es am Ende ein Literaturverzeichnis. Zwar fehlen die Standardtexte, aber die nicht ganz so standardmäßigen Dinge habe ich entweder auf Deutsch oder auf Englisch in dieses Skript aufgenommen.

Wegen der katastrophalen Unterausstattung der Fakultät mit Mittelbaustellen gibt es leider keine Möglichkeit, diesen Text durch fachkundige Hilfe in vernünftige Form zu bringen. Immerhin ist er vermutlich auch in dieser rudimentären Form für die Studierenden nützlich.

R. Schaback Göttingen, Frühjahr 2007.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einführung

(Folie zur Vorlesung)

Kapitel 1

Einführung

(Folie zur Vorlesung)

Inhalt dieses Kapitels

- Was sind Approximationsverfahren?
- Welche Anwendungsfelder gibt es?
- Gliederung der Vorlesung
- Einige typische Beispiele

1.1 Was ist Approximation?

(Folie zur Vorlesung)

Approximationsverfahren

- *Approximation* =
  Konstruktion von Funktionen aus Daten
- Was heißt "Daten"?
- Was heißt "Konstruktion von Funktionen"?
- Beispiel: Univariate Polynominterpolation
- Beispiel: Poisson-Gleichung
  - ∞ viele Daten
  - Auch Ableitungen als Daten

(Folie zur Vorlesung)

Daten

- *Daten von Funktionen*

- *Funktionswerte an Punkten*
- *Ableitungen an Punkten*
- *Lokale Integrale*
- *Integrale gegen Testfunktionen*
- Lineare Funktionale auf Funktionenräumen

1.2 Anwendungsfelder

(Folie zur Vorlesung)

Typische Anwendungsfelder

- Fitting von Meßwerten (Physik...)
- Konstruktion von Flächen (CAD)
- Lösen von Differentialgleichungen
- Maschinelles Lernen

Gemeinsam: Konstruktion von Funktionen aus Daten

1.3 Fragestellungen

(Folie zur Vorlesung)

Typische Fragestellungen

- Interpolation
- Fehlerabschätzungen
- Beste Approximation
- Dichte Approximation (z.B. Weierstraß)
- Quasi-Interpolation (z.B. Bernstein)
- Asymptotik (z.B. Konvergenzgeschwindigkeiten)

Kommentare und Beispiele dazu (Tafel)

1.4 Gliederung

(Folie zur Vorlesung)

Gliederung der Vorlesung

- I Univariate Approximation
- II Multivariate Approximation

(Folie zur Vorlesung)

Unterschied univariat–multivariat

univariat	multivariat
Ordnung	–
Mittelwertsatz	–
Satz von Rolle	–
Triviale Gebiete (Intervalle)	Nichttriviale Gebiete

(Folie zur Vorlesung)

Univariate Funktionenräume

- Polynome (algebraische)
- Polynome (trigonometrische)
- Splines
- Wavelets

Kommentare: (Tafel)

- Splines sind nötig, wenn viele Daten von nicht glatten Funktionen vorliegen
- wavelets sind Multiskalenverfahren, die sich aus dem Shannon-Ansatz motivieren lassen

(Folie zur Vorlesung)

Multivariate Funktionenräume

- (Tensor-) Produkte von univariaten Funktionen:
- Polynome, Splines, Wavelets

- multivariate Splines:
  - Finite Elemente
  - Box Splines (hier ignoriert)
  - Simplex Splines (hier ignoriert)
- Kernbasierte Methoden

Kommentare: (Tafel)

- Geometrie der Daten ist wichtig
- Kernbasierte Methoden als Verallgemeinerung des Shannon-Ansatzes
- und bei Lernverfahren

2 Polynome

(Folie zur Vorlesung)

Kapitel 2

Polynome

(Folie zur Vorlesung)

Inhalt dieses Kapitels (Vorschau)

- Polynomräume:
  algebraisch, trigonometrisch, reell und komplex
- Wiederholung: Polynom-Interpolation
- Basen
- Stabilitätsfragen
- Weierstraß-Sätze
- Fouriertransformation
- Verallgemeinerte Funktionen
- Fehlerabschätzungen (Jackson-Sätze)
- Umkehrsätze (Bernstein-Sätze)
- Schnelle Fourier-und Cosinustransformation

2.1 Polynomräume

(Folie zur Vorlesung)

Univariate Polynome

- algebraische im Reellen
- algebraische im Komplexen
- trigonometrische im Reellen
- Beziehungen:
  - trigonometrische $\Leftrightarrow$ spezielle rationale im Komplexen
  - Transformation: $z = e^{i\varphi}$
  - gerade trigonometrische $\Leftrightarrow$ algebraische im Reellen
  - Transformation: $x = \cos\varphi$
- Wie gehen Daten ineinander über?
- Was heißt "äquidistant"?

2.2 Wiederholung Interpolation

(Folie zur Vorlesung)

Wiederholung Interpolation

- algebraische Polynome:
  - Existenz, Eindeutigkeit
  - Lagrange
  - Newton
  - Fehlerabschätzung
- Übertragung auf den komplexen Fall ?
- Übertragung auf den trig. Fall ?
- Übertragung auf den multivariaten Fall ?

(Folie zur Vorlesung)

Satz von Mairhuber

- *Satz*
- *Sei G ein Gebiet im IR^d*
  mit nichtleerem Inneren und mit $d \geq 2$.
  Es seien $n \geq 2$ stetige Funktionen $p_1, \ldots, p_n$ auf G vorgegeben.
  Dann gibt es n Punkte $x_1, \ldots, x_n \in G$, so daß
  die Matrix der Werte $p_j(x_k)$, $1 \leq j, k \leq n$ singulär ist.
- Beweisidee: Rangierbahnhofargument
  Determinante als stetige Funktion der Punkte
- *Konsequenz: Bei multivariater Interpolation*
  müssen Ansatzräume datenabhängig sein

2.3 Basen

(Folie zur Vorlesung)

Diverse Basen

- Monome
- Bernstein-Polynome
- Chebyshev-Polynome
- Legendre-Polynome
- Allgemeine Orthogonalpolynome, Bessel und Laguerre

(Folie zur Vorlesung)

Bernstein-Polynome

- Bernstein-Operator auf $C[0,1]$

$$f \mapsto \sum_{j=0}^{n} f\left(\frac{j}{n}\right) \underbrace{\binom{n}{j} x^j (1-x)^{n-j}}_{=:B_{j,n}(x)}$$

- Eigenschaften:
  - konserviert lineare Funktionen
  - liefert Satz von Weierstraß
  - hat Norm Eins in $\|.\|_\infty$
- Kondition ist dennoch schlecht

(Folie zur Vorlesung)

Bernstein-Bézier-Kurven

- Vektorwertige Funktion:

$$\begin{array}{rcl} p(x) & := & \sum_{j=0}^{n} b_j \binom{n}{j} x^j (1-x)^{n-j} \\ b_j & : & \text{Kontrollpunkte } \in IR^d \\ p & : & [0,1] \to R^d, \text{ Kurve} \end{array}$$

- Eigenschaften:
  - Partition der Eins
  - Bild in konvexer Hülle der Kontrollpunkte
  - Casteljau-Verfahren (Tafelskizze)

(Folie zur Vorlesung)

Orthogonalität

- Raum mit Skalarprodukt
- Def. von orthogonalen und orthonormalen Funktionen
- Def. Vollständigkeit
- Beispiele: trig. Pol. und alg. Orthogonalpol.
- *Charakterisierungssatz: Beste Appproximationen p^* zu f bezüglich eines Unterraums P sind eindeutig bestimmt durch die Orthogonalitätsrelationen*

$$(f - p^*, p) = 0, \text{ für alle } p \in P$$

- *Orthogonalprojektoren*

$$P \; : \; f \mapsto \sum_{j=0}^{n} (f, p_j) p_j$$

  realisieren die beste Approximation.

- *Beweis der 3-Term-Rekurrenz bei alg. Orthopolynomen*

(Folie zur Vorlesung)

Chebyshev-Polynome

- Definition $T_n(x) = \cos(n \cdot \text{arc cos } (x)),\ n \geq 0$
- Eigenschaften:
  - Rekursion $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x),\ n \geq 2$
  - $|T_n(x)| \leq 1$ auf $[-1, 1]$
  - orthogonal auf $[-1, 1]$ bei $w(x) = (1 - x^2)^{-1/2}$
  - wichtige Nullstellen und Extremstellen
- Dort ist Interpolation in T-Basis sehr stabil
- Grund: Diskrete Orthogonalität
- Querverbindung zur diskreten Cosinustransformation
- Es gibt schnelle Algorithmen

2.4 Stabilität

(Folie zur Vorlesung)

Grundproblem Stabilität: Kurzfassung

- Basenwahl
- Datenwahl
- Abbildung Koeffizienten $\Leftrightarrow$ Daten
- Umkehrung
- Parametrisierungen
- Normen
- Kondition
- Numerische Beispiele zur Stabilität
- Numerische Beispiele zur Monombasis
- Chebyshev-Polynome auf Chebyshev-Punkten
- Lebesgue-Funktionen und Lebesgue-Konstanten

(Folie zur Vorlesung)

Auswertungsstabilität

- Basis $P(x) := (p_0(x), \ldots, p_n(x))^T$,
- Koeffizienten $\alpha := (\alpha_0, \ldots, \alpha_n)^T$
- Abbildung $A_x \ : \ \alpha \mapsto p_\alpha(x) = \sum_{j=0}^n \alpha_j p_j(x) = \alpha^T P(x)$
- $\|A_x\| \leq \|\alpha\|_\infty \|P(x)\|_1 = \|\alpha\|_\infty \sum_{j=0}^n |p_j(x)| = \|\alpha\|_\infty L_n(x)$
- Lebesgue Funktion $L_n(x)$
- Lebesgue Konstante $\|L_n\|_\infty$
- Spezialfall Lagrange-Basis in $x_0 < \ldots < x_n$
- Abbildung
$$P_n \ : \ f \ \mapsto \sum_{j=0}^n f(x_j) p_j(x)$$
- Eigenschaften: linear, $P_n^2 = P_n$
$\|P_n\| = L_n$
- Numerische Beispiele für Lebesgue-Funktionen
und -Konstanten

(Folie zur Vorlesung)

Projektoren: Allgemeines

- P Projektor, wenn linear und idempotent, d.h. $P^2 = P$
- Beispiele: Interpolation, L_2-Approximation
- *Satz Sei $P : U \to V$ ein Projektor*
$V \subset U$ linearer Unterraum, U normiert. Dann gilt:
$$\|u - Pu\| \leq (1 + \|P\|) \inf_{v \in V} \|u - v\|$$
Der Verschlechterungsfaktor ist maximal $(1 + \|P\|)$
gegenüber der besten Approximation
- Satz von Kharshiladze-Lozinski
P_n Projektor von $C[a, b]$ auf P_n in $\|.\|_\infty$
Dann:
$$\|P\| > \frac{2}{\pi^2} \log(n + 1) + \frac{1}{2}$$

(Folie zur Vorlesung)

Interpolations-Projektoren

- Lebesgue Funktion $L_n(x)$ für Interpolation in beliebiger Lagrange Basis:
  Dann: (Erdös, Brutman)

$$L_n > \frac{2}{\pi}\log(n+1) + 0.53$$

- Für Interpolation in T_{n+1}-Nullstellen:

$$\frac{2}{\pi}\log(n+1) + 0.53 < L_n \leq \frac{2}{\pi}\log(n+1) + 1$$

2.5 Weierstraß-Sätze

(Folie zur Vorlesung)

Korovkin-Operatoren und Weierstraß-Sätze

- Korovkin-Operatoren K_n im algebraischen Fall:
- linear, monoton, Werte in $\mathbb{P}_n$ und

$$\lim_{n\to\infty} \|f - K_n(f)\|_\infty = 0 \text{ für } f = 1, x, x^2$$

- Beispiel: $K_n = B_n$ Bernstein-Operatoren
- *Satz Dann*

$$\lim_{n\to\infty} \|f - K_n(f)\|_\infty = 0 \textit{ für alle } f \in C[a,b]$$

- *Beweisidee: f lokal zwischen 2 Parabeln quetschen, dann die K_n anwenden*
- *Erweiterung auf trig. Polynome*
- *Erweiterung auf L_2-Normen*

2.6 Fouriertransformation

(Folie zur Vorlesung)

Fouriertransformation: Überblick

- Orthogonalbasen in $[-\pi, \pi]$ und auf S^1
- Zusammenhang derselben, Skalarprodukte
- Definition der Fouriertransformation (FT) als Orthogonalprojektion
- Vollständigkeit (nicht bewiesen)

- Besselsche Ungleichung, Parsevalsche Gleichung
- Fehlerabschätzung
- Vorsicht mit den Konvergenzbegriffen!
- Kompression durch Wegwerfen kleiner Transformierter
- Schreibweise der FT als Integral, mit Dirichlet-Kern
- Lebesgue-Konstante dazu
- Abschätzungen der Lebesgue-Konstanten (nur Skizze)
- Ausblick auf Beweistechnik des Satzes von Kharsiladze-Lozinski

(Folie zur Vorlesung)

Transformationen (Transforms)

- Sei F ein Raum mit einem Skalarprodukt und einem abzählbaren vollständigen Orthonormalsystem $\{\varphi_j\}_{j\in\mathbb{N}_0}$.
- Dann ist die *Transform-Abbildung*

$$f \stackrel{\hat{}}{\mapsto} \hat{f} := \{(f,\varphi_j)_F\}_{j\in\mathbb{N}_0}$$

eine Isometrie zwischen $(F, \|.\|_F)$ und dem Folgenraum

$$\ell_2(\mathbb{N}_0) := \{\{c_j\}_{j\in\mathbb{N}_0} \ : \ \sum_{j\in\mathbb{N}_0} |c_j|^2 < \infty\}$$

- Parsevalsche Gleichung:

$$\|f\|_F^2 = \|\hat{f}\|_{\ell_2(\mathbb{N}_0)}^2 = \sum_{j=0}^{\infty} |(f,\varphi_j)_F|^2$$

- Orthogonalprojektor:

$$P_n(f) := \sum_{j=0}^{n} (f,\varphi_j)_F \cdot \varphi_j = \sum_{j=0}^{n} \hat{f}(j)\cdot\varphi_j$$

(Folie zur Vorlesung)

Transformationen (Transforms) II

- Orthogonalprojektor:

$$P_n(f) := \sum_{j=0}^{n} \hat{f}(j)\cdot\varphi_j$$

- Orthogonalität:
$$(f - P_n(f), \varphi_j)_F = 0,\ 0 \leq j \leq n$$
- Satz des Pythagoras:
$$\|f\|_F^2 = \|f - P_n(f)\|_F^2 + \|P_n(f)\|_F^2$$
- Normkonvergenz
$$\|f - P_n(f)\|_F \to 0,\ n \to \infty$$
Vorsicht: Andere Konvergenzen sind unklar (punktweise oder gleichmäßige Konvergenz)

(Folie zur Vorlesung)

Transformationen (Transforms) III

- Orthogonalprojektor:
$$P_n(f) := \sum_{j=0}^{n} \hat{f}(j) \cdot \varphi_j$$
- Besselsche Ungleichung
$$\|P_n(f)\|_F^2 = \sum_{j=0}^{n} |\hat{f}(j)|^2 \leq \|f\|_F^2$$
gilt auch ohne Vollständigkeit
- Bei Vollständigkeit: (aus Satz des Pythagoras)
$$\|f - P_n(f)\|_F^2 = \sum_{j=n+1}^{\infty} |\hat{f}(j)|^2$$

(Folie zur Vorlesung)

Transformationen (Transforms) IV

- Gute Approximation durch Weglassen kleiner Koeffizienten
$$\tilde{P}_n(f) := \sum_{j \in M_n(f)} \hat{f}(j) \cdot \varphi_j$$
mit $M_n(f)$ als Menge der n Indizes j der größten $|\hat{f}(j)|$.
- Fehlerabschätzung:
$$\|f - \tilde{P}_n(f)\|_F^2 = \sum_{j \notin M_n(f)}^{\infty} |\hat{f}(j)|^2$$
- *Vorsicht: $\tilde{P}$ ist nicht linear*

- Guter Kompressionseffekt

(Folie zur Vorlesung)

Transformationen (Transforms) V

- Spezialfall komplexe Fouriertransformation

$$\hat{f}(j) := \frac{1}{2\pi}\int_{S^1} f(z)\overline{z^j}dz = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\varphi)\exp(-ij\varphi)d\varphi$$

- Spezialfall reelle Fouriertransformation

$$\begin{array}{rcl} P_n(f)(\varphi) & := & \frac{a_0(f)}{2} + \sum_{j=1}^{n}\left(a_j(f)\cos(j\varphi) + b_j(f)\sin(j\varphi)\right) \\ a_0(f) & := & \frac{1}{\pi}\int_{-\pi}^{\pi} f(\varphi)\frac{1}{\sqrt{2}}d\varphi \\ a_j(f) & := & \frac{1}{\pi}\int_{-\pi}^{\pi} f(\varphi)\cos(j\varphi)d\varphi \\ b_j(f) & := & \frac{1}{\pi}\int_{-\pi}^{\pi} f(\varphi)\sin(j\varphi)d\varphi \end{array}$$

- Zusammenhang mit dem Komplexen:

$$\hat{f}(j) := \frac{1}{2\pi}\int_{-\pi}^{\pi} f(\varphi)\exp(-ij\varphi)d\varphi = \frac{a_j - ib_j}{2},\ j \geq 1$$

(Folie zur Vorlesung)

Konvergenzgeschwindigkeit von Fourier-Partialsummen: Überblick

- Bezug zur gewichteten Summierbarkeit der Transformierten
- Dadurch wird die Konvergenzgeschwindigkeit der Fourier-Partialsummen zu f durch die Glätte von f ausgedrückt
- Es gilt auch die Umkehrung!
- Dasselbe gilt auch für Approximationen mit Orthogonalpolynomen
- Ausblick auf abstrakte harmonische Analyse
- Sobolevräume bei Vorliegen von Transformierten (über gewichtete L_2-Normen der Transformierten)
- Ausblick auf Jackson- und Bernstein-Sätze
- Ausblick auf Féjer- und Jackson-Kerne

2.7 Verallgemeinerte Funktionen

(Folie zur Vorlesung)

Verallgemeinerte Funktionen

- $a_j(f) := \frac{1}{\pi} \int_{-\pi}^{\pi} f(\varphi) \cos(j\varphi) d\varphi,\ j \geq 1$

  $f(\varphi) := \frac{a_0(f)}{2} + \sum_{j=0}^{\infty} (a_j(f) \cos(j\varphi) + b_j(f) \sin(j\varphi))$
- $f \Leftrightarrow \{(a_j, b_j)\}_{j \geq 0}$ Isometrie durch "Transform"
- definiert verallgemeinerte Funktion f
- $L_2^{2\pi}(IR) := \{f \ : \ \sum_{j=0}^{n} (a_j^2(f) + b_j^2(f)) < \infty\}$
- $L_2^{2\pi}(S^1) := \{f \ : \ \|\hat{f}\|_2^2 < \infty\}$
- $L_2^{2\pi}(IR) \supset \{f \in C_{2\pi} \ : \ \int_{-\pi}^{\pi} f^2(t) dt < \infty\}$
- Vervollständigung
- Hilbertraum L_2 mit verallg. Funktionen
- $f \mapsto f(x)$ ist auf L_2 *nicht stetig!*

(Folie zur Vorlesung)

Ableitungen

- $\cos(j\varphi)' = -j \sin(j\varphi),\ \sin(j\varphi)' = +j \cos(j\varphi)$
- $$\|P_n(f)\|_2^2 = \sum_{j=0}^{n} (a_j(f)^2 + b_j(f)^2)$$
  $$\Rightarrow \|P_n(f)'\|_2^2 = \sum_{j=1}^{n} (j^2 a_j(f)^2 + j^2 b_j(f)^2)$$
- $P_n(f') = P_n(f)'$
- $f \Leftrightarrow \{(a_j, b_j)\}_{j \geq 0}$ Isometrie durch "Transform"
- Verallgemeinerte Ableitung:
- $Df \Leftrightarrow \{j(-b_j, a_j)\}_{j \geq 0}$
- $Df \in L_2 \Leftrightarrow \sum j^2 (a_j^2 + b_j^2) < \infty$
- Differenzierbarkeit $\Leftrightarrow$ Konvergenzgeschwindigkeit

(Folie zur Vorlesung)

Sobolewräume, simpelster Fall

- $H^k_{2,2\pi}(IR) = \{f \ : \ D^k f \in L_{2,2\pi}\}$
- $H^k_{2,2\pi}(IR) = \{f \ : \ f \Leftrightarrow \{(a_j, b_j)\}_{j\geq 0}, \ \sum(j^{2k}(a_j^2 + b_j^2)) < \infty\}$
- $H^0 = L_2 \supset H^1 \supset H^2 \supset \cdots$
- *Satz Verallgemeinerte Differenzierbarkeit*
  $\Leftrightarrow$ Konvergenzgeschwindigkeit der Fourier-Partialsummen
- *Wenn man das klassisch ausdrücken will,*
  kommt man in Schwierigkeiten, aber es geht irgendwie
- Prinzip: je glatter eine Funktion ist,
  desto besser läßt sie sich approximieren (Jackson)
- *und umgekehrt!* *(Bernstein)*
- *Man kann $k \in IR$ nehmen*
- *Numerische Experimente dazu*

(Folie zur Vorlesung)

Sätze vom Jackson-Bernstein-Typ in Sobolewräumen

- *Satz Für alle $f \in L_{2,2\pi}$ und alle $k > 0$ gilt*

$$\|f - P_n f\|_2 \leq \frac{C}{(n+1)^k}, \ \forall \ n \geq 0 \Leftrightarrow f \in H^k_{2\pi}$$

- $\overset{Bernstein}{\Rightarrow}$
- $\overset{Jackson}{\Leftarrow}$
- *Die* richtigen Jackson-Bernstein-Sätze sind komplizierter
- Konvergenzgeschwindigkeit $\Leftrightarrow$ Glätte
- Beweis an der Tafel
- Numerische Demonstration dazu

2.8 Schnelle Transformationen

(Folie zur Vorlesung)

Diskrete und schnelle Fouriertransformation

- Äquidistante komplexe Interpolation auf S^1 durch Monome
- Diskrete Fouriertransformation (DFT) im Komplexen
- Inversion dazu, mit Beweis
- Schnelle Fouriertransformation (FFT) im geraden Fall, mit Beweis
- *Vorsicht! Die DFT und die DCT liefern* periodische Werte.
  Ungeeignet zur direkten Berechnung exakter Fourierkoeffizienten!

(Folie zur Vorlesung)

Diskrete Cosinustransformation

- Rückgang auf Tschebyscheff-Interpolation
  (siehe Zusatztext auf der website)
- Zusammenhang mit diskreter Cosinustransformation (DCT)
- Zusammenhang mit `dct, idct` von MATLAB
- Zusammenhang mit DCT II und DCT III
- Reduktion der DCT(n) auf eine DFT(4n) bzw. FFT(4n)
- Konsequenz: schnelle Algorithmen für DCT
  und Chebyshev-Interpolation
- Stabilitätsfragen dazu
- Kompression durch Wegwerfen kleiner Transformierter
- Numerische Demonstration dazu
- Ausblick auf JPEG

2.9 Chebyshev interpolation

Hier beginnt ein Zusatztext, der Dinge enthält, die leider nicht in der klassischen Literatur über Interpolation udn Approximation mit Polynomen vorkommen, obwohl sie dort dringend gebracht werden müßten, denn sie sind für die technischen Anwendungen extrem wichtig.

Recall the definition of the Chebyshev polynomials:

$$\begin{aligned} T_n(x) &= \cos(n \cdot \arccos(x)),\ n \geq 0,\ x \in [-1,1] \\ T_0(x) &= 1, \\ T_1(x) &= x, \\ T_n(x) &= 2xT_{n-1}(x) - T_{n-2}(x),\ n \geq 2,\ x \in I\!R. \end{aligned}$$

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:

$$T_j(x_k) = \cos\left(\frac{j(2k+1)\pi}{2n+2}\right),\ 0 \leq j,k \leq n. \tag{2.1}$$

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 2.1 *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 2.2 *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

$$\begin{aligned} Ta &= y \\ \sum_{j=0}^{n} a_j \cos\left(\frac{j(2k+1)\pi}{2n+2}\right) &= y_k, \ 0 \leq k \leq n \end{aligned} \tag{2.2}$$

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

$$\begin{aligned} a_j &= \frac{2}{n+1} \sum_{k=0}^{n} y_k \cos\left(\frac{j(2k+1)\pi}{2n+2}\right), \ 1 \leq j \leq n \\ a_0 &= \frac{1}{n+1} \sum_{k=0}^{n} y_k. \end{aligned}$$

2.9.1 Discrete Cosine Transform

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 documentation)

$$\begin{aligned}
y(k) &= w(k)\sum_{n=1}^{N} x(n)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \quad 1 \le k \le N \\
x(n) &= \sum_{k=1}^{N} w(k)y(k)\cos\left(\frac{\pi(2n-1)(k-1)}{2N}\right), \quad 1 \le n \le N \\
w(1) &= \tfrac{1}{\sqrt{N}} \\
w(n) &= \tfrac{\sqrt{2}}{\sqrt{N}}, \qquad\qquad 2 \le n \le N
\end{aligned}$$

which, if transformed back from MATLAB $1:N$ notation to standard $0:n-1$ notation of the discrete Fourier transform DFT, gives

$$\begin{aligned}
Y(k) &= w(k)\sum_{j=0}^{n-1} X(j)\cos\left(\frac{\pi(2j+1)k}{2n}\right), \quad 0 \le k < n \\
X(j) &= \sum_{k=0}^{n-1} w(k)Y(k)\cos\left(\frac{\pi(2j+1)k}{2n}\right), \quad 0 \le j < n \qquad (2.3) \\
w(0) &= \tfrac{1}{\sqrt{n}} \\
w(j) &= \tfrac{\sqrt{2}}{\sqrt{n}}, \qquad\qquad 1 \le j < n.
\end{aligned}$$

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 (2.2). 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 2.3 *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.

2.9.2 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

$$\begin{aligned} z_j &= \sum_{k=0}^{n-1} x_k \cos\left(\frac{\pi(2k+1)j}{2n}\right), \quad 0 \le j < n \\ Z_j &= \sum_{k=0}^{4n-1} X_k \exp\left(\frac{2\pi i j k}{4n}\right), \qquad 0 \le j < 4n. \end{aligned} \tag{2.4}$$

If we start with the first (and this will yield a DFT implementation of the DCT), we go over to the second by setting

$$\begin{aligned} X_{2k} &= 0, \quad 0 \le k < 2n \\ X_{2k+1} &= x_k, \quad 0 \le k < n \\ X_{4n-(2k+1)} &= x_k, \quad 0 \le k < n. \end{aligned} \tag{2.5}$$

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

$$2z_0, \ldots, 2z_{n-1}, 0, -2z_{n-1}, \ldots, -2z_1, -2z_0, -2z_1, \ldots, -2z_{n-1}, 0, 2z_{n-1}, \ldots, 2z_1. \tag{2.6}$$

If we have given data $x_0, \ldots, x_{n-1}$ for our cosine transform of length n in (2.4), we apply (2.5) first to get a vector of $4n$ values X_j. These are plugged into an FFT program implementing the second formula of (2.4), and the result will be (2.6), 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 (2.6), do the inverse DCT transform, and get the X_j and the x_j related by (2.5).

Theorem 2.4 *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
```

2.9.3 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 (2.3) 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

$$a_j(g) - a_j(P_n(g)) \ = \ \frac{1}{\pi}\int_{-\pi}^{\pi}(g(\varphi) - P_n(g)(\varphi))\cos j\varphi d\varphi$$

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.

3 Beste Approximation

(Folie zur Vorlesung)

Kapitel 3

Beste Approximation

(Folie zur Vorlesung)

Inhalt dieses Kapitels (Vorschau)

- Existenz
- Eindeutigkeit

- Charakterisierung
- Diskrete beste Approximation
- Chebyshev-Approximation

3.1 Existenz

(Folie zur Vorlesung)

Existenz

- Def. best App. in normierten Räumen
- Existenz im endlichdim. Fall
- Existenz im unendlichdim. Fall

3.2 Eindeutigkeit

(Folie zur Vorlesung)

Eindeutigkeit

- Beispiele
- Strikt konvexe Normen
- Eindeutigkeit im strikt konvexen Fall

3.3 Charakterisierung

(Folie zur Vorlesung)

Charakterisierung

- Einfacher Fall: L_2
- Gateaux-Ableitung der Norm
- Verallgemeinertes Kolmogoroff-Kriterium
- Spezialfall L_∞

3.4 Diskrete beste Approximation

(Folie zur Vorlesung)

Diskrete beste Approximation

- Problemstellung in endlichdim. Räumen
  als überbestimmtes lineares Gleichungssystem
- Einfacher Fall: L_2: Ausgleichsrechnung
- Wiederholung dazu
- Spezialfall L_∞:
  Reduktion auf lineare Optimierung
- Spezialfall L_1:
  Reduktion auf lineare Optimierung

(Folie zur Vorlesung)

Diskrete beste Chebyshev–Approximation

- Haarsche Bedingung
- Wiederholung: Satz von Mairhuber
- Referenzen
- Approximation auf Referenzen
- Satz von de la Vallée–Poussin
- Das Funktional $D(X)f$

3.5 Remes-Algorithmus

(Folie zur Vorlesung)

Remes-Algorithmus

- Beste Chebyshev-Approximation
  mit Haarscher Bedingung
- Schrittweise Verbesserung von Referenzen:
  Remes-Algorithmus
- Lemma: Stetige Fortsetzung von $D(X)f$
- Satz: Lineare Konvergenz des Remes-Algorithmus
- Demonstration mit MATLAB

3.6 Anwendungen der linearen Optimierung

Dies ist ein Zusatztext zur Vorlesung "Optimierung", der in der Vorlesung "Approximationsverfahren I" nur sehr auszugsweise benutzt wurde. Insbesondere wurden behandelt:

- die Formulierung von Approximationsaufgaben als Optimierungsprobleme,
- die Anwendung linearer Optimierung im diskreten Fall bei L_∞ und L_1–Approximation,
- die Gâteaux–Ableitung, und das als Überleitung zu den
- Spline–Funktionen.

.

3.6.1 Minimaxaufgaben

Problemstellung Gegeben sei ein überbestimmtes lineares Gleichumgssystem

$$By = z,\ B \in I\!R^{m\times k}.$$

Die Grundidee bei der Lösung solcher Probleme ist, stattdessen eine Fehlerminimierung zu versuchen. Das ist, nebenbei, ein Standardtrick bei allen Arten von "unlösbaren" Problemen. Man wähle also eine Norm $\|.\|$ auf $I\!R^m$ und minimiere

$$\min_{y\in I\!R^k} \|z - By\|.$$

Das Ergebnis hängt von der gewählten Norm ab. Im Falle $\|.\| = \|.\|_2$ bekommt man die klassische Ausgleichsrechnung (Methode der kleinsten Quadrate von Gauß). Sie führt (in der Theorie!) auf das Gaußsche Normalgleichungssystem $B^TBy = B^Tz$, das man aber aus Stabilitatsgründen besser gar nicht erst aufstellt. Stattdessen verwendet man geeignete Orthogonaltransformationen, aber dieses Thema gehört in die Numerikvorlesung und nicht hierher. Man kann das Ganze zwar auch als quadratische Optimierungsaufgabe sehen, aber das werden wir erst später tun.

Im Falle $\|.\| = \|.\|_\infty$ bekommt man ein **Minimaxproblem**

$$\min_{y\in I\!R^k} \max_{1\le i\le m} |z_i - \sum_{j=1}^{k} b_{ij}y_j| \tag{3.1}$$

und im Falle $\|.\| = \|.\|_1$ das L_1**-Problem**

$$\min_{y\in I\!R^k} \sum_{i=1}^{m} |z_i - \sum_{j=1}^{k} b_{ij}y_j|.$$

Das riecht nach nichtlinearer Optimierung, aber läßt sich als lineare Optimierung schreiben, denn es gibt ein paar

Standardtricks Es seien $f, f_1, f_2, \ldots$ affin-lineare Ausdrücke.

Trick 1
Kommt irgendwo $|f|$ vor, so setzt man eine Gleichung $f = u - v$ mit neuen Variablen $u, v \geq 0$ an und ersetzt $|f|$ durch $u + v$.

Trick 2
Kommt irgendwo $\max(f_1, f_2, \ldots)$ vor, so führt man neue Ungleichungen $f_j \leq u$ mit einer neuen Variablen u ein.

Trick 3
Kommt irgendwo $\min(f_1, f_2, \ldots)$ vor, so führt man neue Ungleichungen $f_j \geq v$ mit einer neuen Variablen v ein.

Achtung: Die beiden letzten Tricks helfen nur, wenn man u klein und v gross halten kann (siehe Minimaxproblem). So etwas muß man in der Regel irgendwie in die Zielfunktion einbauen, wenn es nicht schon ohnehin drin ist.

Anwendung der Standardtricks auf Minimaxprobleme Standardtrick Nummer 2 bei Minimaxproblemen ergibt eine Umformulierung als lineares Optimierungsproblem:

> Minimiere $\epsilon \geq 0$ unter den Nebenbedingungen
>
> $$-\epsilon \leq z_i - \sum_{j=1}^{k} b_{ij} y_j \leq \epsilon, \ 1 \leq i \leq m$$
>
> und den $k+1$ Variablen $\epsilon, y_1, \ldots, y_k$,

denn dann hat man

$$\max_{1 \leq j \leq m} |z_i - \sum_{j=1}^{k} b_{ij} y_j| \leq \epsilon \to Min.$$

Das bedeutet bei vektorieller Ausformulierung gerade

$$-\epsilon \mathbf{1} \leq z - By \leq \epsilon \mathbf{1}$$

oder

$$\begin{array}{rcrcr} By & - & \epsilon \mathbf{1} & \leq & z \\ -By & - & \epsilon \mathbf{1} & \leq & -z \end{array}$$

und läßt sich als "Dual"problem

$$\begin{array}{rcl} \underbrace{\begin{pmatrix} B & -\mathbf{1} \\ -B & -\mathbf{1} \end{pmatrix}}_{=:A^T} \underbrace{\begin{pmatrix} y \\ \epsilon \end{pmatrix}}_{=:w} & \leq & \underbrace{\begin{pmatrix} z \\ -z \end{pmatrix}}_{=:p} \\ A^T w & \leq & p \\ b^T w := (\mathbf{0}_k^T, -1)^T w & = & -\epsilon \to Max! \end{array}$$

schreiben.

Dualisierung bei Minimaxproblemen Das zugehörige Dualproblem zum Minimaxproblem ist also das Primalproblem

$$\begin{aligned} Ax &= b \\ x &\geq 0 \\ p^T x &= Min! \end{aligned}$$

zu obigem "Dual"problem, d.h.

$$\begin{aligned} \underbrace{\begin{pmatrix} B^T & -B^T \\ -\mathbf{1}^T & -\mathbf{1}^T \end{pmatrix}}_{=A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{:=x} &= \underbrace{\begin{pmatrix} \mathbf{0}_k \\ -1 \end{pmatrix}}_{=b} \\ u &\geq 0 \\ v &\geq 0 \\ p^T x = z^T(u-v) &= Min! \end{aligned} \tag{3.2}$$

Es ist klar, daß das Ausgangs-Minimax-Problem (als Minimierungsproblem für ϵ) eine nach unten beschränkte Zielfunktion und eine nichtleere zulässige Menge hat. Deshalb ist es lösbar, ebenso das obige Dualproblem. Im Folgenden werden wir zwecks Ausschaltung gewisser seltener Sonderfälle annehmen, daß der Wert ϵ^* im Optimalpunkt positiv ist.

Die Komplementarität der Optimallösungen $\epsilon^*,\ y^*,\ u^*,\ v^*$ liefert die Gleichungen

$$\begin{aligned} (x^*)^T(p - A^T w^*) &= 0, \quad \text{d.h.} \\ u_j^*(z_j - (By^*)_j + \epsilon^*) &= 0, \quad 1 \leq j \leq m \\ v_j^*(-z_j + (By^*)_j + \epsilon^*) &= 0, \quad 1 \leq j \leq m. \end{aligned}$$

Ferner kann die zur Optimallösung $(x^*)^T = ((u^*)^T, (v^*)^T)$ gehörige Ecken-Indexmenge nicht mehr als $k+1$ Elemente enthalten, denn das ist die Zeilenzahl von A. Man kann dann die zwei Indexmengen $I_+ := I_{u^*}$ und $I_- := I_{v^*}$ mit zusammen nicht mehr als $k+1$ Elementen hernehmen und feststellen, daß

$$\begin{aligned} (By^*)_j - z_j &= +\epsilon^* \quad \text{für alle } j \in I_+ \\ (By^*)_j - z_j &= -\epsilon^* \quad \text{für alle } j \in I_- \end{aligned} \tag{3.3}$$

gilt. Im Falle $\epsilon^* > 0$ sind die beiden Indexmengen disjunkt. Der Fehler "alterniert" also im Vorzeichen an den Komponenten mit Indizes $j \in I_+ \cup I_-$ und nimmt dort betragsmäßig seinen Extremwert ϵ^* an. In allen anderen Komponenten gilt wegen der Optimalität der Minimaxlösung noch

$$|(By^*)_j - z_j| \leq \epsilon^*,\ 1 \leq j \leq m.$$

Man spricht dann von einer "Alternante".

Satz 1 *Ein Minimaxproblem der Form* (3.1) *hat immer eine Lösung, die in einer gewissen Anzahl von Komponenten des m-dimensionalen Bildraums alterniert, d.h. betragsmässig den Optimalfehler ϵ^* annimmt. Im Falle $\epsilon^* > 0$ gibt es eine maximal $(k+1)$-elementige Teilmenge $I := I_+ \cup I_-$ von $\{1, \ldots, m\}$ mit* (3.3)*. Sie hat die Eigenschaft, daß das auf die Komponenten mit Indizes aus I eingeschränkte Minimaxproblem dieselbe Lösung hat, d.h. die übrigen Komponenten hätte man gar nicht betrachten müssen, wenn man sie vorab gekannt hätte.*

Wir müssen nur noch den Nachsatz beweisen. Das machen wir allgemeiner:

Satz 2 *Es sei ein lösbares Normalformproblem*

$$Ax = b,\ x \geq 0,\ p^T x = \min_x,\ A \in I\!R^{m\times n},\ x, p \in I\!R^n,\ b \in I\!R^m$$

mit Optimallösung x^* und zugehöriger Indexmange X^* gegeben. Dann löst x_{X^*} das Problem

$$A_{X^*} z = b,\ z \geq 0,\ p_{X^*}^T z = \min_z,\ A_{X^*} \in I\!R^{m\times|X^*|},\ z, p_{X^*} \in I\!R^{|X^*|},\ b \in I\!R^m$$

und läßt sich ohne alle Optimierung als Lösung des Gleichungssystems $A_{X^*} z = b$ ausrechnen. Die Optimallösung w^* des Dualproblems des Ausgangsproblems ist als Lösung des Systems $A_{X^*}^T w^* = p_{X^*}$ direkt ausrechenbar, und sie löst das zum obigen eingeschränkten Problem duale Problem.

Die Optimalität ist klar, weil x_{X^*} für das zweite Problem zulässig ist, die Zielfunktionswerte $p^T x^* = p_{X^*}^T x_{X^*}$ gleich sind, und das zweite Problem eine Einschränkung des ersten ist, d.h. keinen kleineren optimalen Zielfunktionswert haben kann. Das System $A_{X^*} z = b$ ist lösbar und hat maximalen Spaltenrang, also ist x_{X^*} dadurch eindeutig bestimmt. Die Berechenbarkeitsaussage über w^* gilt immer, und dieser Vektor ist zulässig und optimal für das Dualproblem des eingeschränkten Problems. □

Das eingeschränkte Problem des obigen Satzes ist nur formell ein Optimierungsproblem, denn es gilt $|X^*| \leq m$ und somit ist das primale Ergebnis nicht verwunderlich. Die interessante Ausage ist die zum Dualproblem, weil sie besagt, dass man bei Vorab-Kenntnis der optimalen "aktiven" Restriktionen in $A^T w \leq p$ sich das Leben leicht machen könnte, indem man $A_{X^*}^T w^* = p_{X^*}$ löst.

Die Anwendung dieses Satzes auf Minimaxprobleme mit Alternante entnimmt die optimale Indexmenge aus dem Normalformproblem als Duales zum Minimaxproblem und wendet den obigen Satz an. Dabei ist eine Spaltenselektion von A eine Zeilenselektion von B, und das im Satz gemeinte Dualproblem ist genau ein Minimaxproblem mit Einschränkung der betrachteten Komponenten aus $\{1, \ldots, m\}$ auf die Komponenten mit Indizes aus der Alternante.

Programmbeispiel zu Minimaxproblemen In MATLAB kann man Minimaxaufgaben einfach (und ineffizient) durch einen passenden Aufruf von `linprog` bewerkstelligen, obwohl ein duales Simplexverfahren sicher besser wäre:

```
function [x, fval]=myminimax(A,b)
[m n]=size(A);
B=[A -ones(m,1); -A -ones(m,1)];
p=[b; -b];
z=[zeros(n,1) ; 1];
options = optimset('LargeScale','off')
[y fval]=linprog(z,B,p,[],[],[],[],[],options);
x=y(1:n);
```

Das Kommando `options = optimset('LargeScale','off')` dient zur exakteren Ausrechnung der Ecke, denn das ansonsten verwendete Innere-Punkte-Verfahren liefert Ergebnisse, die manchmal ziemlich neben der Theorie liegen, weil sie keiner exakten Ecke entsprechen.

Ein passender Treiber ist

```
clear all;
t=-1:0.15:1;
% Punktesatz
f=t.^2-0.2*t.^3+0.02*(2*rand(size(t))-1);
% verrauschte Daten
ft=t.^2-0.2*t.^3; % Originaldaten
A=[ones(size(t))' t' t.^2' t.^3' t.^4']
% Approximationsmatrix, Gread <=4
[x fval]=myminimax(A,f')  % Minimaxrechnung
g=A*x % Ergebnis in Funktionswerten
xset=find(abs(f'-g)>fval-100*length(t)*eps)
% hole Extremalpunktindizes
plot(t,ft,t,f,'.',t,g,'+',t(xset),f(xset),'o')
% Plotten Funktion, Daten, Reproduktion
figure(2)
plot(t,ft'-g,t,f'-g,'.',t(xset),f(xset)'-g(xset),'o')
% Plotten Fehlerfunktion
```

und in der zugehörigen Plotausgabe sieht man die Alternationspunkte an den Stellen, wo die kleinen Punkte (verrauschte Daten, Komponenten von z) von den zugehörigen Kreisen (Komponenten von By^*) am weitesten, nämlich um ϵ^* entfernt liegen. Im Beispiel ist $k = 5$ und es gibt $k + 1 = 6$ Alternationspunkte.

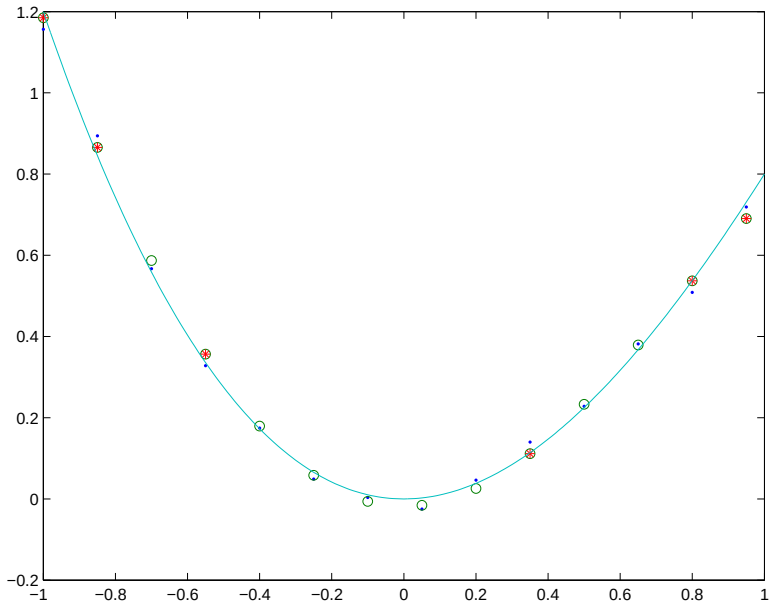

Abbildung 1: Ausgabe zum Minimaxproblem

Noch etwas zur Dualität Das Optimierungsproblem (3.2) kann man mit $s := u - v,\ u, v \geq 0$ noch etwas umformulieren in

$$\begin{array}{rcl} B^T s & = & \mathbf{0}_k \\ \sum_j (u_j + v_j) = \mathbf{1}^T (u + v) = \|s\|_1 & = & 1 \\ z^T s & = & Min! \end{array} \tag{3.4}$$

was wieder einmal täuschend nichtlinear aussieht.

Die Zielfunktion des obigen Problems wird wegen unserer Annahme $\epsilon^* > 0$ sicher negativ, nämlich im Optimalfall gleich $-\epsilon^*$, so daß man auch $\|s\|_1 \leq 1$ zulassen kann, ohne die Lösungsmenge zu verändern. Ist nämlich $s^* \neq 0$ eine Lösung des erweiterten Problems mit $\|s\|_1 < 1$ und $z^T s^* < 0$, so erfüllt $s^*/\|s^*\|_1$ das auf $\|s\|_1 = 1$ eingeschränkte Problem mit kleinerem Zielfunktionswert, was nicht möglich ist.

Gleichung (3.4) zeigt also, dass die Dualitätstheorie des Minimaxproblems für $B \in I\!R^{m\times k}$, $z \in I\!R^m$ die Ausage

$$\min_{y\in I\!R^k} \|By - z\|_\infty = \max_{s\in I\!R^m, B^T s=0, \|s\|_1\leq 1} |z^T s|.$$

liefert. Analog gilt aber auch

$$\min_{y\in I\!R^k} \|By - z\|_1 = \max_{u\in I\!R^m, B^T u=0, \|u\|_\infty\leq 1} |z^T u|$$

wobei die Normen $\|.\|_1$ und $\|.\|_\infty$ vertauscht sind. Der Beweis war als Übungsaufgabe gestellt und wird hier kurz skizziert. Das L_1-Problem ist mit unseren Standardtricks als

$$B(y^+ - y^-) - z = u^+ - u^-, \mathbf{1}^T(u^+ + u^-) = min!$$

zu schreiben, und es wird dualisiert zu

$$B^T u = 0, -\mathbf{1} \leq u \leq \mathbf{1}, z^T u = Max,$$

was zu beweisen war.

Eigenartigerweise transformiert das Dualisieren also die $\|.\|_1$-Norm in die $\|.\|_\infty$-Norm und umgekehrt. Das ist kein Zufall, sondert lehrt, dass **der Dualitätsbegriff der Optimierung zusammenfällt mit dem der normierten Vektorräume**. Um das zu erklären, nehmen wir einen normierten Vektorraum V mit (primaler) Norm $\|.\|_V$ und bilden seinen (topologischen) Dualraum

$$V^* := \{\lambda \ : \ V \to I\!R \ : \ \text{linear und beschränkt}\}$$

wobei Beschränktheit eines Funktionals λ meint, daß eine Konstante c_λ existiert mit

$$|\lambda(v)| \leq c_\lambda \|v\|_V \text{ für alle } v \in V,$$

und diese Eigenschaft ist äquivalent zur Stetigkeit von λ als reellwertige Abbildung auf einem normierten Vektorraum. Dann kann man eine (duale) Norm auf dem topologischen Dualraum V^* definieren als

$$\|\lambda\|_{V^*} := \sup_{v\neq 0} \frac{\lambda(v)}{\|v\|_V} \leq c_\lambda.$$

Im Sonderfall $V = I\!R^n$ ist V^* nicht nur algebraisch isomorph zu V, sondern auch *topologisch*, d.h. es gibt einen *stetigen* Isomorphismus zwischen V und V^*. Deshalb ist auf $V = I\!R^n$ die zu einer Norm $\|.\|_V$ duale Norm definiert als

$$\|z\|_{V^*} := \sup_{v\neq 0} \frac{z^T v}{\|v\|_V},$$

wobei wir benutzt haben, wie Funktionale des IR^n als Dualraum auf den IR^n als "Primalraum" wirken. Als Konsequenz bekommt man

$$z^T v \leq \|z\|_{V^*} \|v\|_V \text{ für alle } z, v \in IR^n.$$

Bei dieser Dualitätsbeziehung in normierten Vektorräumen erweisen sich die Normen $\|.\|_p$ und $\|.\|_q$ als dual zueinander, sobald $\frac{1}{p} + \frac{1}{q} = 1$ gilt, und dabei kann man $1 \leq p, q \leq \infty$ zulassen. Die p-Norm $\|.\|_p$ für $1 \leq p < \infty$ wird dabei definiert über

$$\|x\|_p^p := \sum_{j=1}^{n} |x_j|^p \text{ für alle } x \in IR^n,$$

und der zugehörige Beweis verwendet die **Minkowskische Ungleichung**

$$z^T v \leq \|z\|_p \|v\|_q \text{ für alle } v \in IR^n,\ 1 \leq p \leq \infty,\ \frac{1}{p} + \frac{1}{q} = 1.$$

Die drei wichtigsten Fälle sind der "selbstduale" euklidische Fall $p = q = 2$ und die oben schon bemerkten Situationen $p = 1,\ q = \infty$ und umgekehrt.

3.6.2 Lernen mit Kernen

Problemstellung Eine wichtige heutige Anwendung der Optimierung ist das "maschinelle Lernen". Das wurde in früheren Jahren bevorzugt mit neuronalen Netzen durchgeführt, aber es hat sich gezeigt, dass "kernbasierte" Lernverfahren leistungsfähiger sind, weil sie nicht an die biologische Modellbildung gebunden sind.

Gesucht ist ein System, das auf Reize x Reaktionen y produziert, also (mathematisch) eine Abbildung $f : X \to Y$ darstellt. Ein System, das Eingaben x in zwei Kategorien (gut $\Leftrightarrow$ schlecht, spam $\Leftrightarrow$ kein spam) klassifiziert, benutzt die Wertemenge $Y = \{-1, +1\}$. In anderen Fällen werden die Reaktionen $y \in Y$ reellwertig sein, etwa wenn Grundstückspreise aus diversen Informationen geschätzt werden sollen (Regression, $Y = IR$). Im allgemeinen trägt die Menge X der Reize oder Eingaben keine mathematische Struktur, denn sie kann z.B. auch aus Bildern oder Texten bestehen.

Neben anderen Formen des maschinellen Lernens ist das **supervidierte** Lernen (supervised learning) besonders wichtig. Es benutzt vorgegebene *Trainingsdaten*, die als Paare $(x_j, y_j) \in X \times Y,\ 1 \leq j \leq m$ vorliegen und von einem Supervisor, Trainer oder *master mind* als Soll-Reaktionen $y_j = f(x_j)$ anerkannt sind. Unter *Training* versteht man dann die Berechnung einer Abbildung g, die einigermaßen gut die Trainingsdaten reproduziert, d.h. es sollte gelten

$$y_j \approx g(x_j),\ 1 \leq j \leq m.$$

Nach dem Training wird dann die "gelernte" Abbildung g (es sollte besser "gelehrte" heißen) auf die reale Welt losgelassen und muß ihren Wert beweisen, indem sie zu ganz neuen Eingaben x eigene Ausgaben $g(x)$ macht. Deshalb verwendet man zusätzliche *Testdaten*, die man nach dem Lernen einsetzt, um die Qualität des Gelernten zu überprüfen. Gewisse Ähnlichkeiten mit dem mathematischen Übungbetrieb liegen auf der Hand: die Vorlesungen und die Übungsaufgaben sind die Trainingsdaten, und die abschließenden Klausuraufgaben machen einen Praxistest an bisher unbekannten Aufgaben.

***Feature Maps* und Kerne** Auf einer unstrukturierten Menge kann man keine brauchbare Mathematik treiben. Also muß eine Struktur her. Das geschieht dadurch, daß man zu jeder denkbaren Eingabe $x \in X$ eine möglichst lange Liste von quantifizierbaren Eigenschaften assoziiert. Man beschreibt also x durch einen *feature vector* $\phi(x)$, der möglichst viel Typisches über x aussagt.

> Beispiel: Will Aschenputtel die guten von den schlechten Erbsen unterscheiden, so sollte sie vielleicht Farbe, Größe, Gewicht und Form der Erbsen in den *feature vector* aufnehmen.

Mathematisch wird das durch eine Abbildung (*feature map*)

$$\phi \ : \ X \to \mathcal{F}$$

mit Werten in einem *feature space* $\mathcal{F}$ beschrieben, und dieser Raum sollte ein Vektorraum über IR sein, der ein Skalarprodukt $\langle . \rangle$ trägt, damit man dort "euklidisch messen" kann.

Ab sofort wird dann fast nur noch mit den *feature vectors* $\phi(x) \in \mathcal{F}$ statt mit den Eingaben $x \in X$ gearbeitet. Das hat zur Folge, daß Eingaben x und y mit $\phi(x) = \phi(y)$ nicht mehr unterscheidbar werden, d.h. man arbeitet praktisch "modulo gleicher features". Deshalb sollte man sicher gehen, dass die *feature map* so reichhaltig ist, dass sie alle wichtigen Unterschiede zwischen möglichen Eingaben auch berücksichtigt.

Ein zugehöriger **Kern** ist dann

$$K \ : \ X \times X \to IR, \ \ K(x,y) := \langle \phi(x), \phi(y) \rangle \text{ für alle } x, y \in X.$$

Er erzeugt eine "schöne" mathematische Struktur auf X, z.B einen (schwachen) Abstandsbegriff

$$d^2(x,y) := \|\phi(x) - \phi(y)\|^2_{\mathcal{F}} := K(x,x) - 2K(x,y) + K(y,y) \text{ für alle } x, y \in X,$$

was man durch Ausmultiplizieren von

$$\|\phi(x) - \phi(y)\|^2_{\mathcal{F}} := \langle \phi(x) - \phi(y), \phi(x) - \phi(y) \rangle$$

sieht. Obendrein hat man jetzt auch plötzlich einen Vorrat von Funktionen auf der unstrukturierten Menge X, nämlich zu jedem $y \in X$ die Funktion

$$x \mapsto K(x,y) = \langle \phi(x), \phi(y) \rangle \text{ für alle } x \in X.$$

Lernen mit Kernen Hat man Trainingsdaten $(x_j, y_j) \in X \times IR, \ 1 \leq j \leq m$, so liegt es nahe, einen Ansatz der Form

$$g(x) := \sum_{i=1}^{m} \alpha_i K(x, x_i) = \sum_{i=1}^{m} \alpha_i \langle \phi(x), \phi(x_i) \rangle, \ \alpha_i \in IR$$

zu machen und das "Lernen" von g als Berechnung geeigneter Koeffizienten $\alpha_1, \ldots, \alpha_m$ zu verstehen. Dieser Ansatz läßt sich sogar durch ein Optimierungsargument in unendlichdimensionalen Räumen begründen (siehe unten Satz 17), aber das kann hier noch nicht dargestellt werden. Im Idealfall würde man also das lineare $m \times m$ Gleichungssystem

$$y_j = g(x_j) = \sum_{i=1}^{m} \alpha_i K(x_j, x_i) = \sum_{i=1}^{m} \alpha_i \langle \phi(x_j), \phi(x_i) \rangle, \ 1 \leq j \leq m \tag{3.5}$$

ansetzen, dessen Koeffizientenmatrix mit den Einträgen

$$K(x_j, x_i) = \langle \phi(x_j), \phi(x_i) \rangle,\ 1 \leq i, j \leq m$$

als **Kernmatrix** bezeichnet wird. Diese ist immer symmetrisch und positiv semidefinit (weil sie eine **Gramsche** Matrix ist), aber sie kann riesig und singulär sein. Obendrein darf die Lösung nicht dramatisch von einzelnen der Trainingsdaten abhängen, wenn sie einigermaßen "stabile" Resultate produzieren soll. Denn sobald sich Zufall und Fehler in die Eingabedaten einschleichen, wäre der Ausgang vollkommen ungewiss. Deshalb verwendet man diverse, meist durch einen stochastischen Hintergrund motivierte Tricks, die eine exakte Lösung des Systems (3.5) gar nicht erst versuchen, sondern ein simpleres Modell einsetzen, das nicht alle Trainingsdaten exakt reproduziert und weniger "anfällig" ist. Man hat immer eine Abwägung zwischen Reproduktionsgenauigkeit der Trainingsdaten und Stabilitat des Modells zu treffen.

Wir behandeln hier als Einführung nur den simplen Spezialfall, daß wir weniger Ansatzfunktionen als Daten benutzen und dann ein Minimaxproblem aufstellen. Das bekommt die Form

$$\epsilon = Min!,\ -\epsilon \leq y_j - \sum_{i=1}^{k} \alpha_i K(x_j, y_i) \leq \epsilon,\ 1 \leq j \leq m \tag{3.6}$$

mit $k < m$ und gewissen $y_1 \ldots, y_k \in X$, die wir eventuell als Teilmenge der Trainingsdaten $x_1, \ldots, x_m$ wählen. Dieses Problem läßt sich mit den Methoden des vorigen Abschnitts behandeln, und wir bekommen im Allgemeinen gewisse Alternanten als Auswahl von maximal $k+1$ Punkten aus den Trainingspunkten $x_1, \ldots, x_k$. Raffiniertere Techniken folgen später.

Beispiel: Klassifikation als Minimaxaufgabe Hier ist ein halbwegs kommentiertes Beispiel, in dem ein nichtsahnendes Programm lernen soll, Punkte innerhalb und außerhalb des Kreises

$$(x - 0.5)^2 + (y - 0.5)^2 = 0.1$$

sauber zu unterscheiden. Als Trainingsdaten werden 50 zufällige Punkte x_j aus $[0,1]^2$ genommen und die Werte y_j auf 1 für draußen liegende und auf -1 für innen liegende Punkte gesetzt. Die *feature map* wird so gebaut, daß ein Gitter aus Punkten $z_k \in [0,1]^2$ vorgegeben wird, und dann besteht $\phi(x)$ für festes $x \in I\!R^2$ aus dem Vektor aller $\|x - z_k\|_\infty$, wobei die z_k über das Gitter laufen. Die "features" von x sind also die Abstände zu den Gitterpunkten; sie haben nichts mit der zu lernenden Figur zu tun. Durch Verfeinerung des Gitters kann man das Auflösungsvermögen des Lernprogramms leicht steigern, egal was da zu lernen ist.

Die Wahl der Ansatzpunkte y_i aus dem obigen Text wird sehr grob so gemacht, dass je 5 Trainingsdaten drinnen und draußen ausgewählt werden. Weil die Trainingsdaten ohnehin zufällig sind, kann man die ersten 5 drinnen und die ersten 5 draussen nehmen. Der obere Plot zeigt die Testdaten (+ und o), den exakten Kreis (affin verzerrt, also als Ellipse) und die Ansatzpunkte (x). Man sieht, welche Testpunkte als Ansatzpunkte ausgesucht wurden.

Der Rest ist dann ziemlich klar: man setzt ein Minimaxproblem auf und löst es. Danach werden 250 zufällige Testdaten in $[0,1]^2$ generiert und getestet, ob sie das Programm richtig klassifiziert. Dazu wertet man g an jeder Teststelle aus, und deklariert einen Testpunkt als "drinnen", wenn g negativ ist, sonst als "draußen". Schließlich haben wir ja die Trainingswerte y_j auf 1 für draußen liegende und auf -1 für innen liegende Punkte gesetzt. Das Ergebnis zeigt dann der zweite Plot.

Der dritte zeigt die Alternationspunkte, d.h. diejenigen Trainingspunkte, an denen der Fehler extremal war. Man könnte mit diesen als Ansatzpunkten das Verfahren wiederholen, denn in der Regel gibt es genau einen Alternationspunkt mehr als Ansatzpunkte. Hier ist reichlich Platz zum Experimentieren. Noch etwas: Der Zufallsgenerator wurde nicht rückgesetzt, so daß alle neuen Rechnungen verschieden ausfallen. Es ist ziemlich einfach, andere Parameter durchzuspielen und das Programm andere Formen lernen zu lassen. Man wird immer sehen, dass die Klassifizierung von neuen Testdaten dort besonders schlecht ausfällt, wo keine oder nur wenige Trainingsdaten vorhanden sind. Im Beispiel sieht man, das das Programm den linken Rand nicht genau festlegen kann, weil ihm nicht "klar" ist, ob die Ellipse nicht "links" etwas kleiner ist. Im Prinzip benutzt das Programm eine kleinere Figur um die als "innen" vorgegebenen Trainingsdaten. Das kann man ihm nicht übelnehmen.

Fazit: *Was nicht geübt wird, kann auch nicht gelernt werden* (alte Grundregel des Mathematik- und Klavierstudiums).

```
clear all;
np=50; % Anzahl der Trainingsdaten
% hier die Trainingsdaten, zufällig in [0,1]
randx=rand(np,1);
randy=rand(np,1);
radsq=0.1; % Radius zum Quadrat
testval=(randx-0.5).^2+(randy-0.5).^2;
% denn wir wollen einen Kreis lernen
kreisx=0.5+sqrt(radsq)*cos(2*pi*[0:0.01:1]);
% exakter Kreis, feine Plotdaten
kreisy=0.5+sqrt(radsq)*sin(2*pi*[0:0.01:1]);
xset=find(testval<=radsq); % holt Indizes der inneren Punkte
val=ones(np,1);  % und wir setzen die Trainingswerte
val(xset,1)=-1; % drinnen -1, draussen +1
posset=find(val>0);  % zum Plotten splitten wir die Daten
negset=find(val<0);
% Wir müssen jetzt die feature vectors wählen
[X Y]=meshgrid(0:0.1:1);
% ein gleichmäßiges Gitter zwecks feature vectors
XX=X(:); % die x- Gitterwerte als Liste
YY=Y(:); % dito y
nd=length(XX); % das wird dann die Länge der feature vectors
fv=zeros(np,nd);     % Matrix der feature vectors aufbauen
for i=1:nd    % wir nehmen die Distanzwerte zum Gitter
    fv(:,i)=max(abs(randx(:,1)-XX(i)),abs(randy(:,1)-YY(i))); %
    %  das war die  Maximumsnorm - Distanz
end
% Jetzt wählen wir die Ansatzpunkte
nq=5; % halbe Anzahl der Ansatzdaten
% Wir nehmen je die ersten nq
% aus den inneren und äußeren Punkten
% Ziemlich wahllos, das geht besser.....
```

```
Xset=[posset(1:nq) negset(1:nq)]
% und das war schon unsere Selektion
subplot(3,1,1) % und plotten sie
% als ersten Plot in einer 3x1 Konfiguration
plot(randx(posset),randy(posset),'+',kreisx,kreisy)
hold on  % das friert die Skalierung ein
plot(randx(negset),randy(negset),'o')
plot(randx(Xset),randy(Xset),'x')
axis([0 1 0 1])
title('Trainings- und Ansatzdaten (+,o und x)')
% Das ergibt eine nichtquadratische Kernmatrix
Kmat=fv*fv(Xset,:)';
[x fval]=myminimax(Kmat,val); % und rein ins Minimaxproblem
% Ab hier wird getestet
neval=250; % Anzahl der Testpunkte
npx=rand(neval,1); % und zufällige Auswahl
npy=rand(neval,1);
fp=zeros(neval,nd);
% deren feature vectors ausrechnen, wie oben
for i=1:nd
    fp(:,i)=max(abs(npx(:,1)-XX(i)),abs(npy(:,1)-YY(i)));
end
zp=fp*fv(Xset,:)'*x;
% das ist der Vorhersagewert des gelernten Modells
% Zum Plotten brauchen wir die Entscheidungen, wer
% drin ist und wer draussen
posfset=find(zp>0);
negfset=find(zp<0);
subplot(3,1,2)
plot(npx(posfset),npy(posfset),'+',kreisx,kreisy)
hold on
plot(npx(negfset),npy(negfset),'o')
axis([0 1 0 1])
title('Testdaten')
% und jetzt plotten wir noch Alternationspunkte
resid=abs(Kmat*x-val);
yset=find(resid>fval-0.0001);
posyset=find(val(yset)>0);
negyset=find(val(yset)<0);
% und plotten sie hier
subplot(3,1,3)
plot(randx(yset(posyset)),randy(yset(posyset)),'+',kreisx,kreisy)
hold on
plot(randx(yset(negyset)),randy(yset(negyset)),'o')
axis([0 1 0 1])
title('Alternationspunkte')
```

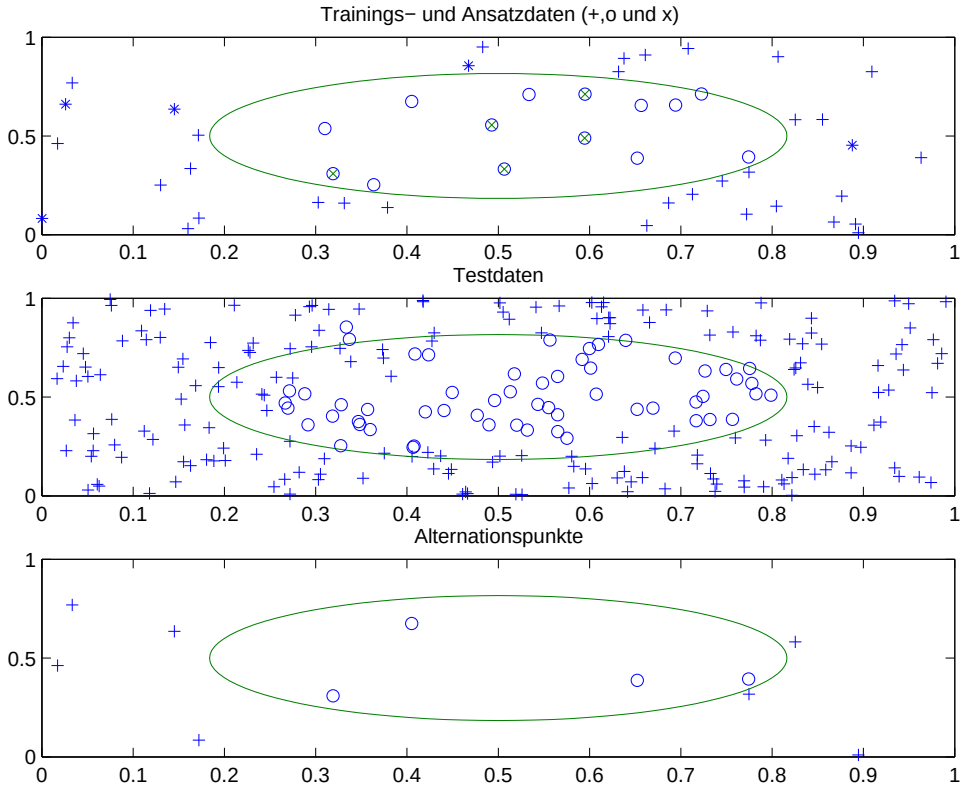

Abbildung 2: Ausgabe zum Lernproblem

Beispiel: Klassifikation als Trennungsaufgabe .
(oder auch *Aschenputtel's support vector machine*).

Aschenputtel muß lernen, Erbsen in gute und schlechte zu klassifizieren. Sie erhebt jeweils m reellwertige Merkmale von ihren Erbsen, z.B. Durchmesser in mm, Gewicht in Gramm, etc. Sie hat von der bösen Stiefmutter einen Lernsatz mit n_+ guten und n_- schlechten Erbsen bekommen. Die Merkmale dieser Erbsen ergeben je eine $n_+ \times m$- und $n_- \times m$-Matrix, die Aschenputtel M^+ und M^- nennt. Allerdings sind n_+ und n_- viel größer als $m \geq 2$, so daß Aschenputtel, die sich im IR^m gut auskennt, schnell sieht, daß die Zeilen von M^+ und von M^- als Vektoren des IR^m durch eine Hyperebene im IR^m trennbar sind. Es gibt also einen Vektor $x \in IR^m \setminus \{0\}$ und eine relle Zahl β, so daß

$$M^+x + \beta\mathbf{1} \geq 0, 0 \geq M^-x + \beta\mathbf{1}$$

gilt. Wer sich nicht so gut im IR^m auskennt wie Aschenputtel, möge sich mal für ein paar "trennbare" Punkte des IR^2 klarmachen, wieso dies "Trennung" bedeutet.

Auf Grund dieser Trennbarkeit kommt Aschenputtel auf die gute Idee, zu jeder Erbse e den zugehörigen Merkmalsvektor $\phi(e) \in IR^m$ zu bilden, dann $f(e) := \phi(e)^T x + \beta$ auszurechnen, und Erbsen e mit $f(e) \geq 0$ als "gut" und solche mit $f(e) < 0$ als "schlecht" zu klassifizieren. Denn diese Regel würde auf allen Testerbsen richtige Ergebnisse bringen.

Sie merkt aber auch, dass es bei ihrem Testsatz unendlich viele solche trennende Hyperebenen gibt, und sie will eine optimale Hyperebene finden, die eine möglichst sichere Unterscheidung

ermöglicht. Also "verbreitert" sie die Hyperebene $\{z \in I\!R^m \ : \ z^T x+\beta = 0\}$ auf einen "Streifen" $\{z \in I\!R^m \ : \ |z^T x + \beta| \leq \epsilon\}$ (der "Breite" $2\epsilon/\|x\|_2$, aber das ist hier nicht wichtig). Damit will sie einen möglichst breiten Streifen zwischen die Merkmalsvektoren der guten und schlechten Testerbsen legen. Sie will also ein maximales ϵ suchen, so daß

$$M^+ x + \beta \mathbf{1} \geq \epsilon \mathbf{1} > 0 \geq -\epsilon \mathbf{1} \geq M^- x + \beta \mathbf{1} \tag{3.7}$$

gilt. Weil man diese Ungleichungskette aber mit beliebig großen positiven Zahlen multiplizieren könnte, um ϵ hochzutreiben, muß Aschenputtel den Vektor x in Schach halten. Weil Aschenputtel (noch) nichts von quadratischer Optimierung weiss, fügt sie die Nebenbedingung $\|x\|_\infty \leq 1$ hinzu, von der sie weiss, dass sie sich "linearisieren" läßt. Jetzt hat sie ein wunderbares lineares Optimierungsproblem, und kann ihre Erbsen bis zum Beginn des Balls sehr zur Zufriedenheit der bösen Stiefmutter klassifizieren.

Als Übungsaufgabe wurde folgendes gestellt:

1. Wie sieht das komplette Optimierungsproblem von Aschenputtel aus, und was ist das Dualproblem?

2. Warum hätte Aschenputtel alle ihre Testerbsen bis auf höchstens $m+2$ wichtige wegwerfen können, ohne ein anderes Ergebnis zu bekommen?

3. Wodurch sind diese wichtigen "Stütz"erbsen bestimmt?

Man verwende dazu den Satz 2, der auch beim Beweis des Alternantensatzes wichtig war.

Hier ist eine Lösungsskizze. Das Problem ist

$$\begin{pmatrix} -M^+ & \mathbf{1} & -\mathbf{1} \\ M^- & \mathbf{1} & \mathbf{1} \\ I & 0 & 0 \\ -I & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ \epsilon \\ \beta \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ \mathbf{1} \\ \mathbf{1} \\ = Max! \end{pmatrix}$$

und das Duale ist

$$\begin{pmatrix} -(M^+)^T & (M^-)^T & I & -I \\ \mathbf{1}^T & \mathbf{1}^T & 0^T & 0^T \\ -\mathbf{1}^T & \mathbf{1}^T & 0^T & 0^T \\ 0^T & 0^T & \mathbf{1}^T & \mathbf{1}^T \end{pmatrix} \begin{pmatrix} u \\ v \\ r \\ s \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ Min! \end{pmatrix}$$

Das Problem ist sicher lösbar, weil $\epsilon = 0$ wegen der vorausgesetzten Trennbarkeit erlaubt ist, aber beliebig große ϵ nicht mehr trennen würden. Die zulässige Menge ist also nicht leer, und die Zielfunktion ist nach oben beschränkt, also ist das Problem lösbar. Das Dualproblem ist ein Normalformproblem mit $m + 2$ Zeilen, und deshalb haben Ecken maximal $m + 2$ von Null verschiedene Komponenten. Wir haben also eine Indexmenge zu einer Optimallösung mit maximal $m + 2$ Einträgen. Komplementarität liefert dann im Ausgangsproblem, dass die entsprechenden Zeilen des Ausgangsproblems exakt erfüllt sind, d.h. es gibt eine Anzahl von Indizes j und k mit $e_j^T M^+ x^* + \beta = \epsilon^*$ und $e_k^T M^+ x^* + \beta = -\epsilon^*$. Diese bestimmen die wichtigen "Testerbsen" nach dem Satz 2, und das löst Teile 2 und 3. Man nennt diese Vektoren "support vectors". Sie liegen auf dem "margin" des trennenden Streifens.

Aschenputtel's Programm und Ergebnis

```
clear all;
np=25 % Anzahl der guten Punkte
nn=25 % Anzahl der bösen Punkte
r=[0.2 0.5]; % Richtungsvektor der idealen Hyperebene
nor=[-0.5 0.2] % Normale dazu
bs=[0 0]; % Aufpunkt für Strahl auf Hyperebene
% wir gehen zufällig vor und berechnen Punkte
% entlang der Geraden und gleichzeitig links und rechts
for ip=1:np
    Mp(ip,:)=bs+rand(1,1)*r+0.2*rand(1,1)*nor;
    Mn(ip,:)=bs+rand(1,1)*r-0.2*rand(1,1)*nor;
end
% So, jetzt bauen wir das Aschenputtel-Problem auf
A=[-Mp ones(np,1) -ones(np,1);...
    Mn ones(nn,1) ones(nn,1);...
  eye(2) zeros(2,2); -eye(2) zeros(2,2)];
b=[zeros(np+nn,1) ;ones(4,1)];
p=zeros(4,1);
p(3,1)=-1;
% und lösen es
[x,fval]=linprog(p,A,b);
% Wir wollen die trennende Ebene malen
tt=-0:0.01:0.2; % das werden die x-Werte
% und es kommen die umgerechneten y-Werte
% dreier paralleler Geraden
y0=(       -x(4,1)-x(1,1)*tt)/x(2,1);
yp=( x(3,1)-x(4,1)-x(1,1)*tt)/x(2,1);
yn=(-x(3,1)-x(4,1)-x(1,1)*tt)/x(2,1);
% und die malen wir
plot(tt,y0,tt,yp,tt,yn)
hold on
% mit den gegebenen Daten
plot(Mp(:,1),Mp(:,2),'+',Mn(:,1),Mn(:,2),'o')
% Achtung, die Geometrie ist nicht euklidisch!
```

Es sollten 4 Testerbsen ausreichen, um sauber zu klassifizieren, und das sind 4 Datenpunkte, die auf dem Rand des kritischen Streifens liegen.

Wie man sich von der Voraussetzung der Trennbarkeit befreit, wird später behandelt.

3.6.3 Konvexe Optimierung

Gâteaux-Differential Es sei f eine konvexe Funktion auf einer nichtleeren konvexen ("zulässigen") Menge $\mathcal{M}$ in einem **nicht notwendig endlichdimensionalen** Vektorraum V über IR gegeben.

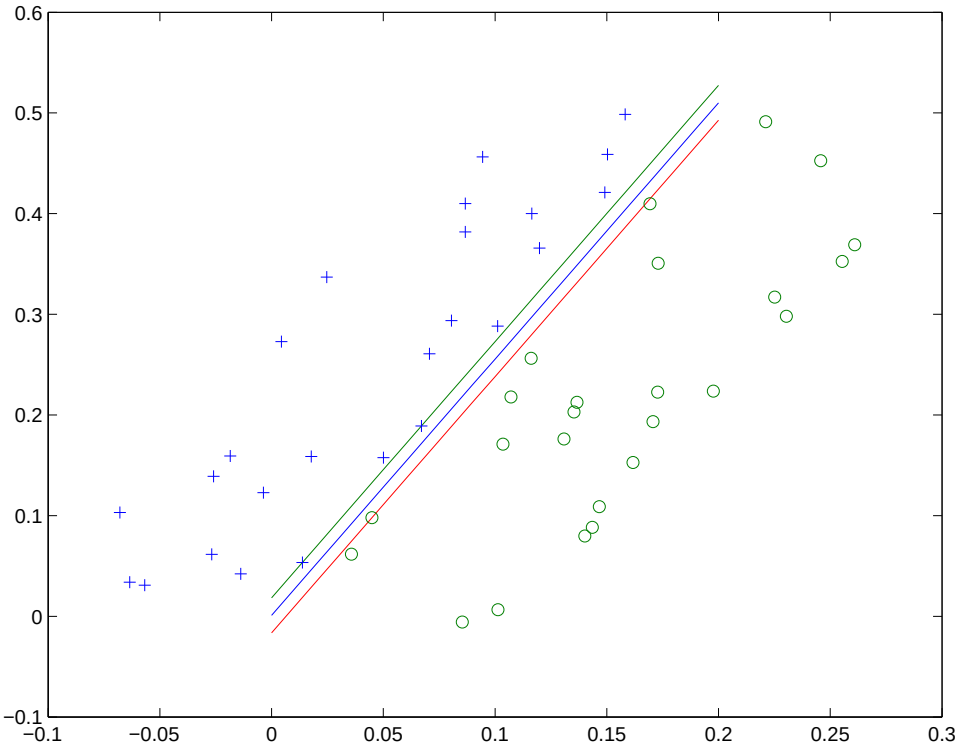

Abbildung 3: Ausgabe zum Aschenputtelproblem

Lemma 3 *Es sei $x \in \mathcal{M}$ gegeben, und es sei $y \in V$ eine zulässige Richtung, d.h. $x + hy \in \mathcal{M}$ für $h \in [0, h_0]$ mit einem $h_0 > 0$. Dann sind die Differenzenquotienten*

$$\frac{f(x+hy) - f(x)}{h}, \ h \in (0, h_0]$$

schwach monoton steigend als Funktion von h.
(Veranschaulichung durch Zeichnung!)

Beweisidee: man wählt $0 < s \leq t \leq h_0$ und schreibt $x + sy$ als Konvexkombination von x und $x+ty$. Darauf wendet man die Konvexitätsvoraussetzung von f an und rechnet die Behauptung herbei.

Lemma 4 *Es sei $x \in \mathcal{M}$ gegeben, und es seien $y, \ -y \in V$ zulässige Richtungen. Dann gilt*

$$\frac{f(x) - f(x-sy)}{s} \leq \frac{f(x+ty) - f(x)}{t}$$

und die linke Seite ist schwach monoton fallend als Funktion von s für kleine s.
(Veranschaulichung durch Zeichnung!)

Beweisidee: man schreibt x als Konvexkombination von $x - sy$ und $x + ty$. Darauf wendet man die Konvexitätsvoraussetzung von f an und rechnet die erste Behauptung herbei. Die zweite ergibt sich wie im Lemma 3.

Lemma 5 *Es sei $x \in \mathcal{M}$ gegeben, und es seien $y, \ -y \in V$ zulässige Richtungen. Dann ist f auf einer Umgebung von x auf der Strecke $[x - y, x + y]$ stetig.*

Beweisidee: Im vorigen Lemma müssen die Zähler der beiden Seiten gegen Null gehen, wenn s und t gegen Null gehen.

Lemma 6 *Es sei V endlichdimensional, und es sei $x \in \mathcal{M}$ ein innerer Punkt von $\mathcal{M}$, d.h. alle $y \in V$ sind zulässige Richtungen. Dann ist f in x stetig.*

Beweisidee: Man kann das vorige Lemma "gleichmäßig" für alle Richtungen anwenden, denn bei endlichdimensionalem V kann man die Richtungen auf die kompakte Einheitskugel einschränken.

Definition 7 *Es sei $x \in \mathcal{M}$ gegeben, und es seien y, $-y \in V$ zulässige Richtungen. Dann existieren (nach Lemma 4) die Limiten*

$$\lim_{s \searrow 0} \frac{f(x) - f(x - sy)}{s} =: -f'_+(x, -y) \leq f'_+(x, y) := \lim_{t \searrow 0} \frac{f(x + ty) - f(x)}{t}$$

und werden **Gâteaux-Richtungsableitungen** *im Punkt x in Richtung y und $-y$ genannt. Ist $f'_+(x, y)$ eine lineare Abbildung als Funktion von y, so spricht man vom* **Gâteaux-Differential**.

Allgemeiner:

Definition 8 *Es sei $x \in \mathcal{M}$ gegeben, und es sei $y \in V$ eine zulässige Richtung bezüglich $\mathcal{M}$ in x, aber er werde* **nicht** *vorausgesetzt, dass f oder $\mathcal{M}$ konvex seien. Wenn der Limes*

$$f'_+(x, y) := \lim_{t \searrow 0} \frac{f(x + ty) - f(x)}{t}$$

existiert, wird er **Gâteaux-Richtungsableitung** *im Punkt x in Richtung y genannt. Ist $f'_+(x, y)$ eine lineare Abbildung als Funktion von y, so spricht man vom* **Gâteaux-Differential**.

Lemma 9 *Die Gâteaux-Richtungsableitungen haben einige Eigenschaften:*

1. $f'_+(x, \alpha y) = \alpha f'_+(x, y)$, *für alle* $\alpha \geq 0$
2. *Ist f konvex, so ist $f'_+(x, y)$ konvex als Funktion von y auf dem Kegel der zulässigen Richtungen in x bezüglich $\mathcal{M}$. Deshalb kann man in beliebiger Weise Gâteaux-Richtungsableitungen von Gâteaux-Richtungsableitungen usw. bilden, sofern Konvexität vorliegt.*
3. *Ist f im klassischen oder Fréchet-Sinne in x differenzierbar mit der Ableitung $\nabla f(x)$, so gilt*
$$(\nabla f(x))(y) = f'_+(x, y)$$
*und ist als Funktion von y linear. Das erklärt den Begriff des Gâteaux-***Differentials**.

Hier kommt eine sehr einfache Verallgemeinerung dessen, was man von der Schule her kennt:

Satz 10 *Es sei f eine konvexe Funktion auf einer nichtleeren konvexen Menge $\mathcal{M}$ in einem nicht notwendig endlichdimensionalen Vektorraum V. Ferner sei $x \in \mathcal{M}$ ein zulässiger Punkt, in dem die Gâteaux-Richtungsableitungen in alle zulässigen Richtungen existieren. Dann gilt: x ist genau dann ein Minimum von f auf $\mathcal{M}$, wenn $f'_+(x, y) \geq 0$ für alle zulässigen Richtungen y in x gilt.*

Beweisskizze: Für beide Richtungen wendet man Lemma 3 und die Definition der Gâteaux-Richtungsableitung an.

Ein Problem bei der Anwendung des obigen Satzes entsteht, weil x normalerweise "am Rand" von $\mathcal{M}$ liegt, und dann ist die Existenz von Gâteaux-Richtungsableitungen in alle zulässigen Richtungen nicht automatisch garantiert (Übungsaufgabe), sondern muß gesondert nachgewiesen werden. In vielen Fällen hilft aber eine allgemeine Differenzierbarkeit von f über diese Hürde hinweg.

Satz 11 *Es sei f eine* **nicht notwendig konvexe** *Funktion auf einer nichtleeren* **nicht notwendig konvexen** *Menge $\mathcal{M}$ in einem nicht notwendig endlichdimensionalen Vektorraum V. Ferner sei $x \in \mathcal{M}$ ein zulässiger Punkt, in dem die Gâteaux-Richtungsableitungen in alle zulässigen Richtungen existieren. Dann gilt: Ist x ein* **lokales** *Minimum von f auf $\mathcal{M}$, so folgt $f'_+(x,y) \geq 0$ für alle zulässigen Richtungen y in x.*

Beweisskizze: Das folgt aus der Definition der Gâteaux-Richtungsableitung.

Achtung:
Die Konvexität in Satz 10 liefert eine **notwendige und hinreichende** bedingung für ein **globales** Minimum, während Satz 11 zwar ohne Konvexität auskommt, aber dann nur eine **notwendige** Bedingung für ein **lokales** Minimum liefert.

Beide Sätze liefern **keine** Existenzaussage. Stattdessen liefern sie sogenannte **Variationsungleichungen** der Form

$$f'_+(x,y) \geq 0 \text{ für alle zulässigen Richtungen } y \text{ in } x$$

als notwendige und im konvexen Fall auch hinreichende Bedingungen für Optimallösungen. In vielen Fällen muß man damit zufrieden sein, insbesondere bei heiklen Optimierungsproblemen in unendlichdimensionalen Räumen.

Ist die Gâteaux-Ableitung $f'_+(x,y)$ in y linear und bilden die zulässigen Richtungen y einen linearen Raum V, so sind die obigen Variations**un**gleichungen äquivalent zu Variations**gleichungen**

$$f'_+(x,y) = 0 \text{ für alle zulässigen Richtungen } y \text{ in } x,$$

was sich in diversen Fällen sehr schön auswerten läßt, wie wir gleich sehen werden.

Lagrange-Multiplikatoren In allen Texten über Optimierung treten gewisse "Lagrange-Multiplikatoren" mit gewissen Vorzeichenbedingungen auf. Sie ergeben sich formal immer über Funktionale, die gewisse konvexe Mengen "trennen", aber wir wollen sie hier durch etwas naheliegendere Argumente motivierend einführen.

Wir gehen der Einfachheit halber erst von einem konvexen Problem $f(x) = Min!$ auf dem IR^n mit differenzierbarer Zielfunktion f und $m < n$ affin-linearen Gleichungs-Nebenbedingungen $h(x) := Ax - b = 0$ aus. Satz 10 und die Bemerkung am Ende des vorigen Abschnitts besagen dann, daß die Variationsgleichung

$$f'_+(x,y) = (\nabla^T f(x))y = 0 \text{ für alle } y \text{ mit } Ay = 0$$

notwendig und hinreichend für eine Optimallösung x ist. Führen wir für einen Moment bei festem x die lineare Abbildung

$$B \ : \ I\!R^n \to I\!R, \ y \mapsto (\nabla^T f(x))y$$

ein, so haben wir die formale Situation

$$By = 0 \text{ für alle } y \text{ mit } Ay = 0 \tag{3.8}$$

für zwei lineare Abbildungen $A \ : \ U \to A(U) =: V, \ B \ : \ U \to W$ zwischen gewissen Vektorräumen $U, \ V, \ W$. So etwas tritt in der Mathematik sehr oft auf, wird aber in den Anfängervorlesungen nicht mit dem notwendigen Nachdruck behandelt.

Unter schwachen Zusatzvoraussetzungen neben (3.8) **faktorisiert** nämlich B über das **Bild** von A, d.h. es gibt eine lineare Abbildung $C \ : \ V = A(U) \to W$ mit

$$B = C \circ A.$$

Bevor wir die genauen Voraussetzungen für die Faktorisierung klären, stellen wir in unserem Fall fest, daß es dann einen Vektor $v \in I\!R^m$ geben muß, so daß

$$\nabla^T f(x) = v^T A$$

gilt, und das ist der einfachste Fall eines Vektors von "Lagrange-Multiplikatoren".

Im Falle endlichdimensionaler Vektorräume (d.h. also auch in unserem Fall) ist die Faktorisierung eine einfache Folgerung aus dem bekannten Isomorphiesatz

$$A(U) = V \simeq U/ker \ A,$$

denn man kann C auf $A(U) = V \simeq U/ker \ A$ durch

$$C(A(u)) := Bu$$

vertreterinvariant als lineare Abbildung definieren. Man kann sie auf jeden endlichdimensionalen Vektorraum T, der $V = A(U) \subseteq T$ als Untervektorraum hat, problemlos fortsetzen, so daß wir keine Rangvoraussetzung an unsere Matrix A brauchen und unsere reellwertige lineare Abbildung C als Funktional auf dem ganzen $I\!R^m$ wählen können. Im unendlichdimensionalen Fall muß man etwas aufpassen und Zusatzforderungen (Stetigkeit, und Fortsetzbarkeit mit dem Satz von Hahn-Banach) stellen, aber das wollen wir hier nicht vertiefen. Bestenfalls ist noch darauf hinzuweisen, daß (bei trivialem Beweis analog wie oben) der Faktorisierungssatz bei Verzicht auf Linearität auch in der folgenden abstrakten Form gilt:

Satz 12 *Sind $A \ : \ U \to V := A(U)$ und $B \ : \ U \to W$ Abbildungen mit der Eigenschaft*

$$B(x) = B(y) \text{ für alle } x, y \in U \text{ mit } A(x) = A(y),$$

so gibt es eine Abbildung $C \ : \ V \to W$ mit $B = C \circ A$.

Wir sollten aber noch den Fall von Ungleichungsnebenbedingungen der Form $g_j(x) \leq 0,\ 1 \leq j \leq \ell$ mit konvexen und differenzierbaren Funktionen g_j auf IR^n ansehen, wobei wir aber der Einfachheit halber jetzt die affin-linearen Gleichungsnebenbedingungen weglassen. Wann ist ein Vektor $y \in IR^n$ eine zulässige Richtung? Es sollte

$$g_j(x + hy) \leq 0 \text{ für alle } j,\ 1 \leq j \leq \ell,\ h \in [0, h_0] \tag{3.9}$$

mit einem $h_0 > 0$ gelten. Für die j mit $g_j(x) < 0$ stellt das keine Bedingung an y, weil unter unseren Voraussetzungen die g_j stetig sind. Für die j mit $g_j(x) = 0$, die "aktiven" Restriktionen, muß dann

$$\lim_{h \searrow 0} \frac{1}{h} \left(g_j(x + hy) - g_j(x)\right) = g'_{j+}(x, y) = (\nabla g_j(x))y \leq 0$$

gefordert werden, aber das ist nur notwendig, nicht hinreichend für (3.9). Dieses Problem wird uns noch beschäftigen.

Wenn wir erst einmal nur mit den notwendigen Bedingungen für zulässige Richtungen y weitermachen, bekommen wir die notwendigen Variationsungleichungen $(\nabla f(x))y \geq 0$ für alle $y \in IR^n$ mit $(\nabla g_j(x))y \leq 0$ für alle $j,\ 1 \leq j \leq \ell$ mit $g_j(x) = 0$. Das kann man analog zu unserem obigen Vorgehen formalisieren zu einer Aussage der Form

$$By \geq 0 \text{ für alle } y \in IR^n \text{ mit } Gy \leq 0 \tag{3.10}$$

mit linearen Abbildungen

$$B\ :\ IR^n \to IR,\ G\ :\ IR^n \to IR^k,\ k \leq \ell.$$

Betrachtet man erst einmal den Teilraum $U = \ker\, G$ der y mit $Gy = 0$, so folgt aus der vorausgesetzten Linearität sofort

$$By = 0 \text{ für alle } y \in IR^n \text{ mit } Gy = 0$$

und es faktorisiert B über das Bild von G im Raum IR^k, wie wir oben schon gesehen haben. Es gibt also einen Vektor $u \in IR^k$ von "Lagrange-Multiplikatoren" mit

$$By = u^T Gy \text{ für alle } y \in IR^n.$$

Setzt man das in (3.10) ein, so folgt

$$By = u^T Gy \geq 0 \text{ für alle } y \in IR^n \text{ mit } \ Gy \leq 0.$$

Das ist sicher erfüllt, wenn wir zusätzlich $u \leq 0$ fordern, aber $u \leq 0$ ist nicht ohne weiteres als notwendige Bedingung an u zu erschließen. Obendrein kann man leider nicht erwarten, dass jedes u, das sich durch das Faktorisierungsargument ergibt, zwingend nichtpositive Komponenten hat.

Man kann aber durch nichttriviale Zusatzüberlegungen die Existenz eines nichtpositiven u mit der obigen Eigenschaft erschließen. Die obige Bedingung besagt nämlich, daß es keine zulässigen y gibt mit $-Gy \geq 0$ und $(u^T G)y < 0$. Das Farkas-Lemma (siehe Werner-Skript, S. 23, Lemma 1.6) liefert dann die Existenz eines $x \geq 0$ mit $-G^T x = G^T u$, und wir können unser u durch $-x \leq 0$ ersetzen.

Wir erweitern unser $u \leq 0$ noch durch Nullen auf die Komponenten j mit $g_j(x) < 0$ und erhalten die **Komplementaritätsbedingungen**

$$u_j g_j(x) = 0,\ 1 \leq j \leq \ell.$$

Wir könne das Ganze zu den notwendigen Optimalitätsbedingungen

$$\begin{aligned} (\nabla f)(x) + u^T(\nabla g)(x) + v^T(\nabla h)(x) &= 0 \\ h(x) := Ax - b &= 0 \\ g(x) &\leq 0 \\ u &\geq 0 \\ u_j g_j(x) &= 0 \end{aligned}$$

zusammenfassen, wenn wir das Vorzeichen von u bei der Umsetzung auf die linke Seite berücksichtigen und (ohne Beweis) annehmen, daß sich Ungleichungsbedingungen und Gleichungsbedingungen additiv zusammenpacken lassen.

Bei diesem Zugang ist einigermaßen klar, wie die Lagrange-Multiplikatoren zustandekommen, und es verwundert nicht, daß man

$$L(x, u, v) := f(x) + u^T g(x) + v^T h(x)$$

die "Lagrange-Funktion" nennt.

Beispiele

Normen Normen sind global definierte konvexe Funktionen, deshalb haben sie überall Gâteaux-Richtungsableitungen, die wieder konvexe Funktionen sind. Im Nullpunkt sind diese trivial:

Lemma 13 *Ist $\|.\|$ eine Norm auf einem Vektorraum V, so gilt (in naheliegender Notation)*

$$\|'_+(0, y) = \|y\| \text{ für alle } y \in V.$$

Außerhalb des Nullpunktes kann das nicht so simpel sein. Zuerst:

Lemma 14 *Ist $\|.\|$ eine "euklidische" Norm auf einem Vektorraum V, die aus einem Skalarprodukt $(.,.)$ durch $\|x\|^2 := (x, x)$ entsteht, so gilt (in naheliegender Notation)*

$$\|'_+(x, y) = \frac{(x, y)}{\|x\|} \text{ für alle } y \in V, x \in V \setminus \{0\}.$$

Das ist netterweise linear in y. Anders ist es bei

Lemma 15 *Es sei $\|.\| = \|.\|_\infty$ die Maximumsnorm auf $V = IR^n$. Dann gilt*

$$\|'_{\infty,+}(x, y) = \max_{i;:\ |x_i|=\|x\|_\infty} y_i \cdot \operatorname{sgn}(x_i) \text{ für alle } y \in IR^n, x \in IR^n \setminus \{0\}.$$

Machen wir das doch im Unendlichdimensionalen, etwa mit der Norm

$$\|x\|_\infty := \max_{a\le t\le b} |x(t)|$$

auf $V := C[a,b],\ a < b \in I\!R$. Erwartungsgemäß bekommt man

$$\|'_{\infty,+}(x,y) = \max_{t\in[a,b]\,:\,|x(t)|=\|x\|_\infty} y(t)\cdot \text{sgn}\,(x(t))$$

für alle $y \in C[a,b], x \in C[a,b]\setminus\{0\}$, was denn sonst? (Beweise als Tafeldemo oder Übung).

Wenn wir auf $V := C[a,b]$ die euklidische Norm $\|.\|_2$ über das Skalarprodukt

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

definieren, können wir Lemma 14 direkt anwenden und bekommen

$$\|'_{2,+}(x,y) = \frac{\int_a^b x(t)y(t)dt}{\sqrt{\int_a^b x^2(t)dt}}$$

Variationsrechnung Wir stellen uns das Problem, eine Kurve kürzester Bogenlänge im $I\!R^2$ zwischen den Punkten $(0,0)$ und $(1,1)$ zu finden. Eine Gummibandüberlegung zeigt, dass die Verbindungsgerade vermutlich die beste Lösung ist, mit der Bogenlänge $\sqrt{2}$. Allgemeinere und sehr viel interessantere Probleme dieser Art befassen sich mit "Geodätischen" auf Mannigfaltigkeiten. Beispielsweise weiss jeder Pilot und jeder Kapitän, dass die kürzesten Verbindungen auf der Kugel entlang Großkreisen verlaufen. Und Captain Kirk weiß seit Albert Einstein, dass sich Himmelskörper und Raumschiffe entlang von Geodätischen in der Raumzeit der allgemeinen Relativitätstheorie bewegen.

Zu minimieren ist in unserem simplen Fall

$$f(x) := \int_0^1 \sqrt{1+x'^2(t)}dt$$

unter allen stetig differenzierbaren Funktionen x auf $[0,1]$ mit $x(0) = 0,\ x(1) = 1$. Wir haben also den unendlichdimensionalen Raum $V = C^1[0,1]$ und wollen Gâteaux-Richtungsableitungen von f in zulässige Richtungen y berechnen. Diese sind klar: sie sind die $y \in C^1[a,b]$ mit $y(0) = y(1) = 0$, bilden also einen Unterraum von $V = C^1[a,b]$ mit "Kodimension" 2, und sie hängen gar nicht vom "Aufpunkt" x ab.

Bevor wir uns zu Fuß auf den Weg machen, die Ableitungen über die Definition in diesem Spezialfall auszurechnen, sollten wir das Problem verallgemeinern und uns

$$f(x) := \int_a^b F(t,x(t),x'(t))dt$$

mit einer differenzierbaren Funktion $F = F(t,u,v)$ vornehmen. Es folgt

$$\begin{aligned} f(x+hy) &= \int_a^b F(t,x(t)+hy(t),x'(t)+hy'(t))dt \\ &= f(x)+\mathcal{O}(h^2)+ \\ &\quad +\int_a^b \left(hy(t)\frac{\partial F}{\partial u}(t,x(t),x'(t)) + hy'(t)\frac{\partial F}{\partial v}(t,x(t),x'(t))\right) dt \end{aligned}$$

durch Entwicklung, und man bekommt das Gâteaux-Differential

$$f'_+(x,y) = \int_a^b \left(y(t)\frac{\partial F}{\partial u}(t,x(t),x'(t)) + y'(t)\frac{\partial F}{\partial v}(t,x(t),x'(t)) \right) dt.$$

In einem lokalen Optimum x wird dann die Variationsungleichung

$$\int_a^b \left(y(t)\frac{\partial F}{\partial u}(t,x(t),x'(t)) + y'(t)\frac{\partial F}{\partial v}(t,x(t),x'(t)) \right) dt \geq 0$$

für alle zulässigen Richtungen y gelten. Wenn, wie in unserem Spezialfall, die Menge der zulässigen Richtungen der komplette lineare Unterraum der Funktionen y mit $y(a) = y(b) = 0$ ist, und wenn wir die Linearitat der Gâteaux-Ableitung ausnutzen, so wird aus der Variationsungleichung die Variations**gleichung**

$$\int_a^b \left(y(t)\frac{\partial F}{\partial u}(t,x(t),x'(t)) + y'(t)\frac{\partial F}{\partial v}(t,x(t),x'(t)) \right) dt = 0$$

für alle $y \in C^1[a,b]$ mit $y(a) = y(b) = 0$. Unter vorausgesetzter Differenzierbarkeit (die sich mit dem "Fundamentallemma der Variationsrechnung" aber auch erschließen läßt) kann man partiell integrieren und bekommt

$$\int_a^b y(t) \left(\frac{\partial F}{\partial u}(t,x(t),x'(t)) - \frac{d}{dt}\frac{\partial F}{\partial v}(t,x(t),x'(t)) \right) dt = 0$$

unter Ausnutzung der Randbedingungen $y(a) = y(b) = 0$. Ist der Klammerausdruck noch stetig, so kann die obige Gleichung nur dann für alle besagten y Null sein, wenn der Klammerausdruck selber Null ist, denn man kann winzige "Hütchenfunktionen" y dort ansetzen, wo der Klammerausdruck nicht Null ist und sein Vorzeichen nicht wechselt.

Es folgt dann die berühmte **Eulergleichung**

$$\frac{\partial F}{\partial u}(t,x(t),x'(t)) = \frac{d}{dt}\frac{\partial F}{\partial v}(t,x(t),x'(t)),\ F = F(t,u,v)$$

als notwendige Bedingung für ein lokales Optimum. Der Weg von einem Optimierungsproblem über eine Variations**un**gleichung zu einer Variationsgleichung und schließlich zu einer **Differentialgleichung für die Optimallösung** ist typisch für solche Aufgaben aus der klassischen **Variationsrechnung**. Die zulässigen Richtungen y werden von Physikern und Ingenieuren mit phantasievollen Namen wie "infinitesimale Verschiebungen" (in der Elastizitätstheorie und der Mechanik) belegt, sind aber nichts als zulässige Richtungen einer Optimierung. Die Eulergleichung ist eine Konsequenz von Satz 11 unter zusätzlichen Voraussetzungen.

In unserem Spezialfall haben wir $F(t,u,v) = \sqrt{1+v^2}$ und bekommen die Eulergleichung

$$0 = \frac{d}{dt}\frac{x'(t)}{\sqrt{1+x'^2(t)}}.$$

Also muß

$$\frac{x'(t)}{\sqrt{1+x'^2(t)}}$$

und dann nach kurzer Rechnung auch x' konstant sein, und die Randbedingungen $x(0) = 0$ und $x(1) = 1$ lassen dann nur noch die Lösung $x(t) = t$ zu, die sich aus der notwendigen Bedingung für eine Lösung des Optimierungsproblems ergibt. Wir haben aber die Existenz einer Lösung und reichlich Differenzierbarkeit vorausgesetzt, so daß dieses Vorgehen nur zeigt, dass, wenn es eine hinreichend glatte Lösung gibt, diese notwendig die besagte Form hat.

Beispiel: Spline-Funktionen Wir suchen eine mindestens zweimal stetig differenzierbare Funktion u auf $[a, b] \subset I\!R$, die das Integral

$$f(u) := \frac{1}{2} \int_a^b (u'')^2(t)dt$$

minimiert und dabei die Interpolations-Bedingungen

$$u(x_j) = y_j,\ 0 \leq j \leq n$$

mit vorgegebenen $x_j, y_j \in I\!R,\ 0 \leq j \leq n$ erfüllt, wobei die Stützstellen x_j eine Zerlegung

$$a = x_0 < x_1 < \ldots < x_{n-1} < x_n = b$$

des Intervalls $[a, b]$ bilden. Das ist eine konvexe Optimierungsaufgabe im unendlichdimensionalen Raum $C^2[a, b]$ mit affin-linearen Nebenbedingungen. Zulässige Richtungen sind Funktionen $v \in C^2[a, b]$ mit $v(x_j) = 0,\ 0 \leq j \leq n$ und bilden also einen linearen Unterraum V von $C^2[a, b]$. Die Gâteaux-Ableitung von f ergibt sich als

$$f'_+(u, v) = \int_a^b u''(t)v''(t)dt$$

nach einfacher Rechnung. Eine Funktion $u \in C^2[a, b]$ ist genau dann Optimallösung, wenn $f'_+(u, v) \geq 0$ für alle zulässigen Richtungen v gilt. Wegen Linearität in v ist das äquivalent zu $f'_+(u, v) = 0$ für alle zulässigen Richtungen v. Unter vorläufiger Annahme von reichlich Differenzierbarkeit in den Teilintervallen $(x_{j-1}, x_j),\ 1 \leq j \leq n$ kann man das auswerten:

$$\begin{aligned}
0 &= f'_+(u, v) \\
&= \int_a^b u''(t)v''(t)dt \\
&= \sum_{j=1}^n \int_{x_{j-1}}^{x_j} u''(t)v''(t)dt \\
&= \sum_{j=1}^n \left(-\int_{x_{j-1}}^{x_j} u'''(t)v'(t)dt + [u'' \cdot v']_{x_{j-1}}^{x_j} \right) \\
&= [u'' \cdot v']_a^b + \sum_{j=1}^n \left(\int_{x_{j-1}}^{x_j} u''''(t)v(t)dt + [u''' \cdot v]_{x_{j-1}}^{x_j} \right) \\
&= [u'' \cdot v']_a^b + \sum_{j=1}^n \int_{x_{j-1}}^{x_j} u''''(t)v(t)dt
\end{aligned}$$

Wie kann man das erfüllen durch eine geeignete Funktion $u \in C^2[a, b]$?

Wenn man u aus Stücken zusammenbaut, die auf jedem Teilintervall $(x_{j-1}, x_j),\ 1 \leq j \leq n$ ein Polynom dritten Grades sind, verschwinden alle lokalen Integrale, und wenn man auch

noch $u''(a) = u''(b) = 0$ verlangt, ist die obige Gleichung erfüllt. Mit Argumenten, die nicht in eine Optimierungsvorlesung gehören, kann man zeigen, dass es immer genau eine Funktion $u \in C^2[a,b]$ gibt, die allen interpolationsbedingungen genügt, in jedem Teilintervall ein Polynom dritten Grades ist und in den Randintervallen affin-linear ist. Man konstruiert so eine Funktion durch Lösen eines nichtsingulären linearen Gleichungssystems mit einer tridiagonalen Koeffizientenmatrix. Funktionen dieser Art heißen *kubische Splines*, und sie sind in der Numerischen Mathematik sehr wichtig.

Es ist für solche Situationen typisch, dass man die konvexe Optimierungtheorie zunächst nur heuristisch anwendet, um die notwendigen Optimalitätsbedingungen auszuwerten, obwohl man keineswegs weiß, ob eine Lösung existiert. Wenn man dann auf ganz anderem Wege beweisen kann, daß die notwendigen Bedingungen erfüllbar sind, benutzt man, daß diese ja auch hinreichend sind, und ist fertig.

Es ist nach dem obigen Schema relativ einfach zu zeigen, daß ein stetiger stückweise affin-linearer Polygonzug u das Minimum von

$$f(u) := \frac{1}{2}\int_a^b (u')^2(t)dt$$

unter den Interpolations-Bedingungen

$$u(x_j) = y_j,\ 0 \le j \le n$$

realisiert (das ist die "connect-the-dots"-Interpolation). Man kann allerdings dabei nicht auf $C^1[a,b]$ arbeiten, aber findige Leser werden herausbekommen, wie man das Ganze sauber ausführen kann.

3.6.4 Quadratisch optimierende Lernalgorithmen

Optimale Modelle Wir gehen zurück zur Lerntheorie aus Abschnitt 3.6.2 und benutzen die *feature map* $\phi : X \to \mathcal{F}$ und den *Kern* $K : X \times X \to IR$ mit

$$K(x,y) := \langle \phi(x), \phi(y) \rangle \text{ für alle } x, y \in X.$$

Wir definieren den Raum

$$\mathcal{K} := \text{span}\,\{K(\cdot, x) : x \in X\}$$

von Funktionen auf X, weil wir sonst nichts haben, was wir als Funktion auf X benutzen können. Auf diesem Raum gibt es eine Bilinearform, die durch Fortsetzung der Definition

$$(K(\cdot,x), K(\cdot,y))_{\mathcal{K}} := K(x,y) \text{ für alle } x, y \in X$$

auf beliebige Linearkombinationen ensteht:

$$\left(\sum_j \alpha_j K(\cdot, x_j), \sum_k \beta_k K(\cdot, y_k)\right)_{\mathcal{K}} = \sum_j \sum_k \alpha_j \beta_k K(x_j, y_k).$$

Sie ist positiv definit und damit ein Skalarprodukt, wenn der Kern die folgende Eigenschaft hat:

Definition 16 *Ein Kern $K : X \times X \to IR$ ist positiv definit, wenn für beliebige endliche Teilmengen $X_n := \{x_1, \ldots, x_n\}$ von X die $n \times n$ Matrix mit Einträgen $K(x_j, x_k),\ 1 \le j, k \le n$ positiv definit ist.*

Das ist gleichbedeutend damit, dass die Funktionen $K(\cdot, x_j)$ für verschiedene x_j immer linear unabhängig sind.

In einer weiter fortgeschrittenen Veranstaltung würde man jetzt zur Hilbertraum-Vervollständigung von $\mathcal{K}$ übergehen, aber das wollen wir hier unterlassen. Wir spezialisieren aber die obige Gleichung zu

$$\left(\sum_j \alpha_j K(\cdot, x_j), K(\cdot, y) \right)_{\mathcal{K}} = \sum_j \alpha_j K(x_j, y),$$

was dann für beliebige Funktionen $g \in \mathcal{K}$ zur *Reproduktionsgleichung*

$$(g, K(\cdot, y))_{\mathcal{K}} = g(y) \text{ für alle } y \in X,\ g \in \mathcal{K}$$

wird. Kerne mit so einer Eigenschaft nennt man *reproduzierend* für einen Raum $\mathcal{K}$ von Funktionen auf X.

Wir gehen wieder davon aus, daß wir m Trainingsdaten $(x_j, y_j) \in X \times IR$ mit $y_j \approx g(x_j)$ für eine zu "lernende" Funktion g haben. Wir werden jetzt unter allen Funktionen $g \in \mathcal{K}$, die eine exakte Reproduktion $y_j = g(x_j),\ 1 \le j \le m$ leisten, eine optimale heraussuchen, indem wir eine mit minimaler Norm $\|.\|_{\mathcal{K}}$ berechnen. Wir landen dabei punktgenau bei der damals "vom Himmel gefallenen" Gleichung (3.5)

Satz 17 *Es sei K ein positiv definiter Kern auf X. Dann hat das quadratische Minimierungsproblem*

$$\begin{aligned} \|g\|_{\mathcal{K}}^2 &= \min_{g \in \mathcal{K}} \\ g(x_j) &= y_j,\ 1 \le j \le m \end{aligned}$$

eine eindeutige Lösung der Form

$$g^*(x) := \sum_{j=1}^{m} \alpha_j K(x, x_j),\ x \in X,$$

die sich durch Lösen des Gleichungssystems

$$\sum_{j=1}^{m} \alpha_j K(x_k, x_j) = y_k,\ 1 \le k \le m$$

mit symmetrischer und positiv definiter Koeffizientenmatrix berechnen läßt.

Der **Beweis** ist auf verschiedene Weisen möglich. Da der Raum $\mathcal{K}$ nicht notwendig endlichdimensional ist, kann man nicht ohne weiteres die Existenz einer Lösung erschließen. Aber wir haben einen Kandidaten, und wir können Satz 10 anwenden. Die Funktion $f(g) := \|g\|_{\mathcal{K}}^2$ hat die Gâteaux-Ableitung $2(g^*, v)_{\mathcal{K}}$ in g^* in jede zulässige Richtung v, und diese Richtungen bestehen aus den $v \in \mathcal{K}$ mit $v(x_j) = 0,\ 1 \le j \le m$. Der Raum dieser Richtungen ist linear, und so wird aus der notwendigen und hinreichenden Variations**ungleichung** des Satzes 10 die Variations**gleichung**

$$(g^*, v)_{\mathcal{K}} = 0 \text{ für alle } v \in \mathcal{K},\ v(x_j) = 0,\ 1 \le j \le m.$$

Setzen wir unser spezielles g^* ein und verwenden die Reproduktiongleichung, so folgt

$$\left(\sum_{j=1}^{m} \alpha_j K(\cdot, x_j), v\right)_{\mathcal{K}} = \sum_{j=1}^{m} \alpha_j v(x_j) = 0,$$

d.h. g^* erfüllt die notwendige und hinreichende Optimalitätsbedingung. □

Wer die Form der Optimallösung g^* nicht "raten" mag, kann sie auch erschließen. Denn wenn g^* eine Funktion aus $\mathcal{K}$ ist, die der Variationsgleichung genügt, so kann man die *Datenabbildung* $T : \mathcal{K} \to I\!R^m$ mit $T(u) := (u(x_1), \ldots, u(x_m))$ definieren und benutzen, dass

$$(g^*, v)_{\mathcal{K}} = 0 \text{ für alle } v \text{ mit } T(v) = 0$$

gilt. Dann faktorisiert (siehe Abschnitt 3.6.3) unter schwachen, hier erfüllten Voraussetzungen das lineare Funktional $v \mapsto (g^*, v)_{\mathcal{K}}$ über das Bild von T, d.h. es gibt einen Vektor $\alpha \in I\!R^n$ mit

$$(g^*, v)_{\mathcal{K}} = \alpha^T T v = \sum_{j=1}^{m} \alpha_j v(x_j) = \left(\sum_{j=1}^{m} \alpha_j K(\cdot, x_j), v\right)_{\mathcal{K}} \text{ für alle } v \in \mathcal{K},$$

und weil dies eine Variationsgleichung für **alle** $v \in \mathcal{K}$ ist, muss g^* die behauptete Form haben.

Inexakte Reproduktion Es macht wenig Sinn, beim obigen Vorgehen auf exakter Reproduktion aller Trainingsdaten zu bestehen, weil dann für jedes neue Trainingsdatum eine neue Rechnung nötig wäre und das Lernergebnis von **allen** Trainingsdaten sehr sensibel abhängig wäre. Das "Relaxieren" der Bedingungen $y_j = g(x_j)$ kann auf verschiedene Weise geschehen und mit der Zielfunktion $\|g\|_{\mathcal{K}}^2$ verbunden werden. Eine typische Variante ist, die linearen Nebenbedingungen

$$-\epsilon \leq y_k - \sum_{j=1}^{m} \alpha_j K(x_k, x_j), \leq \epsilon, \ 1 \leq k \leq m$$

zu fordern und dann die quadratische Zielfunktion

$$\frac{1}{2}\|g\|_{\mathcal{K}}^2 + C\epsilon = \frac{1}{2}\sum_{j,k=1}^{m} \alpha_j \alpha_k K(x_k, x_j) + C\epsilon \tag{3.11}$$

zu minimieren, wobei die positive Konstante C es gestattet, entweder auf gute Reproduktion der Einzeldaten oder auf gute "Generalisierung" des Modells zu setzen.

Diese quadratische Aufgabe mit affin-linearen Ungleichungsnebenbedingungen wollen wir etwas genauer analysieren. Die Variablen sind ϵ und $\alpha_1, \ldots, \alpha_m$, und die Lagrangefunktion bekommt die Form

$$L(\alpha, \epsilon, \lambda, \mu) = \frac{1}{2}\alpha^T Q \alpha + C\epsilon + \lambda^T(-\epsilon \mathbf{1} + Q\alpha - y) + \mu^T(-\epsilon \mathbf{1} - Q\alpha + y)$$

mit der "Kernmatrix" Q aus den $K(x_j, x_k)$. Die Lagrange-Multiplikatoren-Vektoren λ und μ sind nichtnegativ und aus dem $I\!R^m$ zu nehmen.

Wir gehen direkt auf die Idealsituation der primalen und dualen Lösbarkeit zu. Nach den bekannten Sätzen ist das Problem lösbar, weil es zulässig ist und die Zielfunktion nach unten

beschränkt ist. Ferner ist durch die oben diskutierte exakte Rekonstruktionsfunktion g mit $\epsilon = 0$ auch die Slater-Bedingung erfüllt, so daß der verschärfte starke Dualitätssatz gilt. Also existieren optimale Lösungen $\alpha^*,\ \epsilon^* \geq 0,\ \lambda^* \geq 0,\ \mu^* \geq 0$ mit

$$\begin{aligned}(-\epsilon^*\mathbf{1} + Q\alpha^* - y)_j\lambda_j^* &= 0,\ 1 \leq j \leq m \\ (-\epsilon^*\mathbf{1} - Q\alpha^* + y)_j\mu_j^* &= 0,\ 1 \leq j \leq m\end{aligned}$$

d.h.

$$\begin{aligned}\text{aus } \lambda_j^* > 0 \text{ folgt } (Q\alpha^* - y)_j &= \epsilon^* \\ \text{aus } \mu_j^* > 0 \text{ folgt } (Q\alpha^* - y)_j &= -\epsilon^*\end{aligned}$$

und wir sind wieder bei unserer bekannten Alternationseigenschaft und bei den "support" Vektoren. Differenzieren wir die Lagrangefunktion im Optimalpunkt nach α, so folgt $Q\alpha^* = Q(\mu^* - \lambda^*)$, also $\alpha^* = \mu^* - \lambda^*$. Der optimale Koeffizientenvektor α^* hat also nur so viele von Null verschiedene Komponenten wie es "aktive Restriktionen" gibt, und das Vorzeichen der Komponenten ist durch das Vorzeichen des "Fehlers" bestimmt. Trainingsdaten, die nicht zu aktiven Restriktionen im Optimalpunkt führen, kommen in der Optimallösung nicht vor und sind bei a-posteriori-Betrachtung irrelevant. Das ist der wichtigste Vorteil von Lernalgorithmen dieser Art.

Wir haben aber noch nach ϵ zu differenzieren. Im Falle $\epsilon^* > 0$ kann es keine Indizes j geben, für die λ_j^* und μ_j^* beide positiv sind. Deshalb folgt dann aus $\alpha^* = \mu^* - \lambda^*$ auch $|\alpha_j^*| = \mu_j^* + \lambda_j^*$. Wir bekommen damit

$$C = \mathbf{1}^T(\mu^* + \lambda^*) = \|\alpha^*\|_1.$$

als Ableitung der Lagrangefunktion nach ϵ, was zeigt, daß die Kontrolle von C auch die Kontrolle über die Größe der Koeffizienten im Optimalpunkt erlaubt.

Es ist lehrreich, das Dualproblem auszurechnen, aber das lassen wir als Übungsaufgabe.

Natürlich kann man das weiter vorn stehende Beispiel des "Lernens" eines Kreises oder einer anderen geometrischen Figur mit den Methoden dieses Abschnittes behandeln, indem man die damalige Zielfunktion ϵ durch (3.11) ersetzt und die in (3.6) auftretenden k Punkte y_i durch alle m Punkte x_j ersetzt. Die Selektion einer "aktiven" Teilmenge von "support vectors" geschieht nun automatisch durch die quadratische Optimierung mit linearen Nebenbedingungen. Es sind nur so viele Koeffizienten der Optimallösung von Null verschieden, wie es aktive Restriktionen gibt.

Das folgende MATLAB-Programm setzt diesen Ansatz um. Es ist allerdings nicht identisch mit dem früheren Programm, denn es kann beliebige sternförmige Figuren in $[-1, 1]^2$ lernen.

```
clear all;
np=75; % Anzahl der Trainingsdaten
[X Y]=meshgrid(-1:0.05:1); % ein Gitter zwecks feature vectors
XX=X(:);
YY=Y(:);
nd=length(XX) % das wird später die Länge der feature vectors
randx=2*rand(np,1)-1; % hier die Trainingsdaten
randy=2*rand(np,1)-1;
testval=randx.^2+randy.^2; % aktuelle Radienquadrate
```

```
[theta rho]=cart2pol(randx,randy);
sollrad=radi(theta);
xset=find(testval<=sollrad.^2);
[kreisx kreisy]=pol2cart(2*pi*[0:0.01:1],radi(2*pi*[0:0.01:1]));
val=ones(np,1);  % und wir setzen die Trainingswerte
val(xset,1)=-1;
posset=find(val>0);  % zum Plotten splitten wir die Daten
negset=find(val<0);
subplot(3,1,1)
plot(randx(posset),randy(posset),'+',kreisx,kreisy) % und plotten sie
axis([-1,1,-1,1])
hold on
plot(randx(negset),randy(negset),'o')
title('Trainingsdaten')
fv=zeros(np,nd);     % Matrix der feature vectors
for i=1:nd    % wir nehmen die Distanzwerte zum Gitter
    fv(:,i)=max(abs(randx(:,1)-XX(i)),abs(randy(:,1)-YY(i))); %
%   Maximumsnorm
    % fv(:,i)=sqrt((randx(:,1)-XX(i)).^2.+(randy(:,1)-YY(i)).^2);
    % oder 2-Norm
end
Kmat=fv*fv'; % das wird die Kernmatrix
c=1
[x fval]=mylearner(Kmat,val,c) % und rein ins Kernelproblem
neval=250; % Anzahl der Testpunkte
npx=2*rand(neval,1)-1;
npy=2*rand(neval,1)-1;
% neval=np;
% npx=randx;
% npy=randy;
fp=zeros(neval,nd); % deren feature vectors
for i=1:nd
    fp(:,i)=max(abs(npx(:,1)-XX(i)),abs(npy(:,1)-YY(i)));
end
zp=fp*fv'*x; % und deren Wert als Vorhersage
posset=find(zp>0); % zum Plotten brauchen wir die Entscheidungen...
negset=find(zp<0);
subplot(3,1,2)
plot(npx(posset),npy(posset),'+',kreisx,kreisy)
axis([-1,1,-1,1])
hold on
plot(npx(negset),npy(negset),'o')
hold on
title('Testdaten')
resid=abs(Kmat*x-val);
xset=find(resid>fval-0.0001);
posxset=find(val(xset)>0);
```

```
negxset=find(val(xset)<0);
subplot(3,1,3)
plot(randx(xset(posxset)),randy(xset(posxset)),'+',kreisx,kreisy) % und plotten sie
axis([-1,1,-1,1])
hold on
plot(randx(xset(negxset)),randy(xset(negxset)),'o')
title('Support-Vektoren')
```

Die zu lernende Figur wird spezifiziert durch eine Polarkoodinatenfunktion wie

```
function val=radi(winkel)
val=sqrt(0.3)*(1-0.5*cos(4.*winkel)).*ones(size(winkel));
```

Ferner wird auf eine Funktion der Form

```
[alpha  wert]=mylearner(Q,y,C)
```

zurückgegriffen, die als Übungsaufgabe gestellt wird (sie wird später hier eingebaut). Diese Funktion arbeitet genau so wie im obigen Text beschrieben. Sie erwartet eine $m \times m$ Kernmatrix `Q`, einen Datenvektor `y` mit m Komponenten und das Gewicht `C`. Dann gibt sie den optimalen Koeffizientenvektor `alpha` und den finalen Zielfunktionswert zurück.

Eine typische Ausgabe ist in Abbildung 4 zu sehen. Es ist erstaunlich, wie wenig support-Vektoren nötig sind.

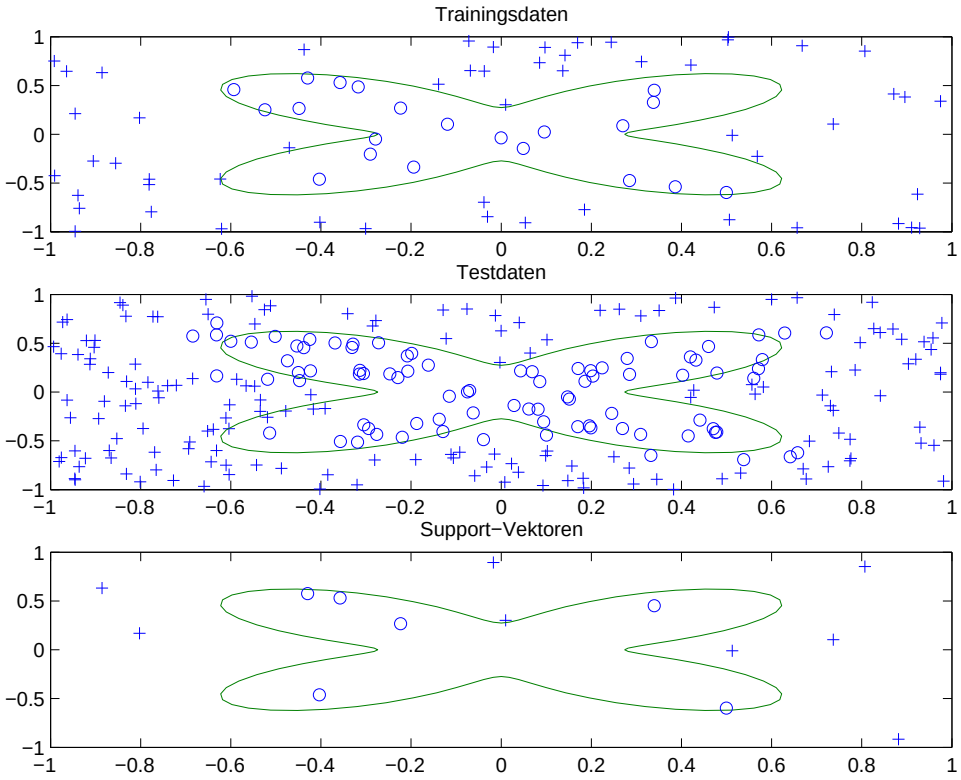

Abbildung 4: Figurenlernen mit Kernen

Klassifikation durch Trennung Wir wollen uns aber auch noch einmal um Aschenputtel kümmern. Inzwischen können wir quadratisch optimieren, und wir wollen uns von der Voraussetzung der Trennbarkeit der gegebenen Trainingsdaten befreien. Wir wollen wieder einen "trennenden Streifen" finden, dessen Breite wir maximieren wollen, aber wir wollen zulassen, dass die Daten gar nicht trennbar sind. Deshalb "bestrafen" wir nicht trennbare Trainingsdaten auf geeignete Weise, und zwar durch Aufnahme in die Zielfunktion. Weil der Rand des trennenden Streifens "aufgeweicht" wird, spricht man von "soft margin classifiers".

Die Bezeichnungen seien wie im Abschnitt 3.6. Statt der Restriktionen (3.7) verwenden wir

$$M^+x + \beta\mathbf{1} + y^+ \geq \epsilon\mathbf{1}, \quad -\epsilon\mathbf{1} + y^- \geq M^-x + \beta\mathbf{1}$$

mit nichtnegativen Vektoren $y^+,\ y^-$ von Schlupfvariablen, die das Nichterfülltsein der ursprünglichen Trennung "messen". Diese Vektoren müssen wir klein halten, und wir wollen gleichzeitig die (nunmehr euklidisch gemessene) Streifenbreite $2\epsilon/\|x\|_2$ maximieren. Dazu renormieren wir die obigen Ungleichungen auf $\epsilon = 1$ zu

$$M^+x + \beta\mathbf{1} + y^+ \geq \mathbf{1}, \quad -\mathbf{1} + y^- \geq M^-x + \beta\mathbf{1}$$

und minimieren $\|x\|_2^2$ stattdessen. Offen bleibt noch, wie wir große $y^+,\ y^-$ bestrafen wollen. Das kann man durch eine gewichtete quadratische Zielfunktion

$$\frac{1}{2}\|x\|_2^2 + C(\|y^+\|_2^2 + \|y^-\|_2^2)$$

erreichen. Das folgende Programm ist eine Adaptation des früheren Aschenputtel-Programms:

```
clear all;
np=25 % Anzahl der guten Punkte
nn=25 % Anzahl der bösen Punkte
r=[0.2 0.5]; % Richtungsvektor der idealen Hyperebene
nor=[-0.5 0.2] % Normale dazu
bs=[0 0]; % Aufpunkt für Strahl auf Hyperebene
% wir gehen zufällig vor und berechnen Punkte
% entlang der Geraden und gleichzeitig links und rechts, mit
overlap=0.2
for ip=1:np
    Mp(ip,:)=bs+rand(1,1)*r+0.2*(rand(1,1)-overlap)*nor;
    Mn(ip,:)=bs+rand(1,1)*r-0.2*(rand(1,1)-overlap)*nor;
end
% plot(Mp(:,1),Mp(:,2),'+',Mn(:,1),Mn(:,2),'o')
% figure(2)
% So, jetzt bauen wir das Aschenputtel-Problem auf
% das wird das Gewicht
c=1.0e5
A=[-Mp  -ones(np,1) -eye(np) zeros(np,np);...
    Mn   ones(np,1) zeros(np,np) -eye(np)];
b=[-ones(np,1);-ones(np,1)];
p=zeros(2*np+3,1);
```

```
Q=c*eye(2*np+3);
Q(1:2,1:2)=eye(2);
Q(3,3)=0.0001;
lb=zeros(2*np+3,1);
ub=[];
lb(1:3,1)=-1.0e12;
[x fval]=quadprog(Q,p,A,b,[],[],lb,ub)
% Wir wollen die trennende Ebene malen
tt=-0:0.01:0.2; % das werden die x-Werte
% und es kommen die umgerechneten y-Werte
y0=(       -x(3,1)-x(1,1)*tt)/x(2,1);
% und die malen wir
plot(tt,y0)
hold on
% mit den gegebenen Daten
plot(Mp(:,1),Mp(:,2),'+',Mn(:,1),Mn(:,2),'o')
```

Eine typische Ausgabe ist in Abbildung 5 zu sehen. Man mache sich klar, dass unsere Pro-

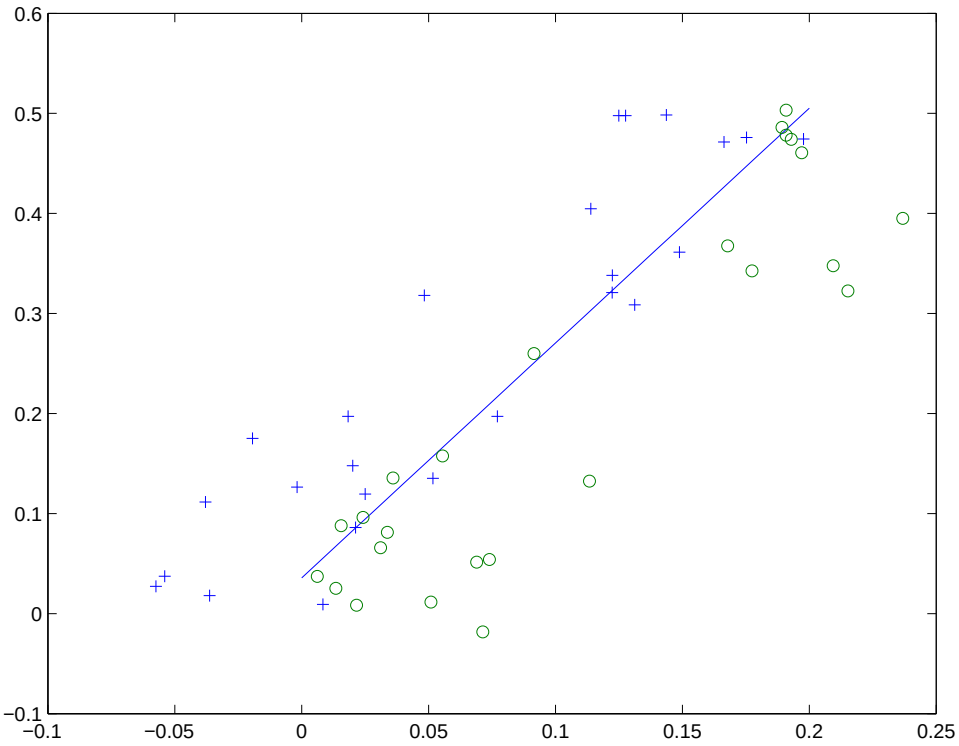

Abbildung 5: Aschenputtelproblem bei nicht trennbaren Daten

grammierung des Aschenputtelproblems ziemlich unrealistisch ist, weil wir einen nur zweidimensionalen feature space benutzen. Die allgemeinere Technik mit Kernen, die durch vernünftige feature maps definiert sind, ist wesentlich leistungsfähiger.

3.6.5 Nichtlineare Optimierung

Rechentechnik Wir fügen hier noch etwas an, was für die Rechenpraxis wichtig ist, aber in den Skripten nicht explizit steht. Wir gehen von einer nichtlinearen Optimierungsaufgabe

$$\begin{array}{rcl} f(x) & = & Min! \\ x & \in & I\!R^n \\ g_i(x) & \leq & 0,\ 1 \leq i \leq \ell \\ h_j(x) & = & 0,\ 1 \leq j \leq m \end{array}$$

mit stetig differenzierbaren reellwertigen Funktionen $f,\ g_i,\ h_j$ auf $I\!R^n$ aus, und schließen den konvexen Fall ein, wobei wir aber auf die zusätzliche konvexe Menge C des Werner-Skripts verzichten.

Die Lagrange-Funktion ist

$$L(x,u,v) := f(x) + u^T g(x) + v^T h(x),\ x \in I\!R^n,\ u \in I\!R^\ell,\ v \in I\!R^m,$$

wenn man wie üblich die Funktionen $g_i,\ h_j$ zu Vektoren zusammenfaßt.

In der Praxis schert man sich wenig um die genauen Voraussetzungen, unter denen der Kuhn-Tucker-Satz gilt. Man wendet bei halbwegs komplizierten Problemen irgendwelche numerischen Standardverfahren an, die am Schluß der Vorlesung skizziert werden. Bei einfachen Problemen, bei denen man eine "Papier-und-Bleistift"-Lösung versuchen kann, setzt man die notwendigen Bedingungen 1. Ordnung als nichtlineares Gleichungssystem an. Das liefert

$$\begin{array}{rcll} \nabla f(x) + u^T \nabla g(x) + v^T \nabla h(x) & = & 0, \quad n & \text{Gleichungen} \\ h(x) & = & 0, \quad m & \text{Gleichungen} \\ u_i g_i(x) & = & 0, \quad \ell & \text{Gleichungen} \\ u_i & \geq & 0, \quad \ell & \text{Ungleichungen} \\ g_i(x) & \leq & 0, \quad \ell & \text{Ungleichungen} \end{array}$$

bei $n + \ell + m$ Unbekannten $x,\ u,\ v$. Mit etwas Glück kann man aus den ersten n Gleichungen x als Funktion von u und v ausrechnen. Das klappt z.B. immer dann, wenn ein quadratisches Optimierungsproblem mit positiv definiten quadratischen Formen vorliegt und die Menge C fehlt. Denn dann ist die Lösung von

$$\min_{x \in I\!R^n} L(x,u,v) = \min_{x \in I\!R^n} f(x) + u^T g(x) + v^T h(x)$$

bei festen u, v eine unrestringierte quadratische Optimierungsaufgabe mit positiv definiter quadratischer Form, die immer eine eindeutige Lösung $x(u,v)$ hat, die man durch Lösen von $\nabla f(x) + u^T \nabla g(x) + v^T \nabla h(x) = 0$ ausrechnen kann. Gleichzeitig liefert das im konvexen Fall die Zielfunktion des dualen Problems als $\Phi(u,v) = L(x(u,v),u,v)$. Wenn man $x(u,v)$ in das zweite System einsetzt, bekommt man $h(x(u,v)) = 0$ und kann mit etwas Glück nach v auflösen, z.B. dann, wenn h affin-linear ist und Vollrang hat (siehe Slater-Bedingung im konvexen Fall). Das liefert v als Funktion von u, und es bleiben die restlichen, leider nichtlinearen und mit Vorzeichenproblemen etwas überfrachteten Bedingungen an u und $g(x(u,v(u)))$, bei denen man nochmal reichlich Glück braucht, um durchzukommen. Natürlich wird man diese Bedingungen aufspalten in "aktive" der Form $g_j(x) = 0,\ u_j \geq 0$ und "inaktive" mit $g_j(x) < 0,\ u_j = 0$. Hat man k aktive und $\ell - k$ inaktive Bedingungen zu erwarten, so reduziert sich das System der ℓ

Gleichungen $u_j g_j(x(u, v(u))) = 0,\ 1 \leq j \leq \ell$ auf k Gleichungen und k Unbekannte, aber es ist nicht immer einfach, die aktiven Restriktionen festzustellen.

Natürlich ist das obige Vorgehen im allgemeinen viel zu hemdsärmelig, um sicher zu funktionieren. Selbst wenn man vorzeichenkorrekte Lösungen des nichtlinearen Gleichungs/Ungleichungssystems finden kann, weiß man nicht, ob sie das ursprüngliche Problem lösen, weil man ja nur die notwendigen Bedingungen hineingesteckt hat. Und in allen Fällen mit vielen lokalen Minima wird das System notwendigerweise viele Lösungen haben, obwohl es aus $n+\ell+m$ Gleichungen (plus 2ℓ Ungleichungen) mit $n+\ell+m$ Unbekannten besteht. Beispielsweise berechnet es im allgemeinen nichtlinearen Fall ohne Ungleichungsnebenbedingungen natürlich auch die lokalen Maxima. Aber zumindestens weiß man, dass, wenn es ein Minimum gibt, dieses unter den Lösungen ist, und man kann bei Vorliegen von nur wenigen Kandidaten einfach die Zielfunktion auswerten, um das Minimum herauszupicken.

Man sollte so etwas auf jeden Fall einmal an Hand einer kleinen Übungsaufgabe durchgerechnet haben.

4 Splines

(Folie zur Vorlesung)

Kapitel 4

Splines

(Folie zur Vorlesung)

Inhalt dieses Kapitels (Vorschau)

- Minimaleigenschaft
- Charakterisierung
- Existenz und Eindeutigkeit
- Symmetrisierung
- Fehlerabschätzungen
- Kubische Splines
- B-Splines

4.1 Minimaleigenschaft

(Folie zur Vorlesung)

Minimaleigenschaft

- Siehe Sondertext
- Bilinearform $(.,.)_k$
- Taylorformel, Reproduktionseigenschaft
- Kern K_k aus der truncated power function
- Problemstellung "Glatteste Interpolante"

4.2 Charakterisierung

(Folie zur Vorlesung)

Charakterisierung

- Siehe Sondertext
- Charakterisierung der "Glattesten Interpolante"
- Natürliche interpolierende Splines

4.3 Existenz und Eindeutigkeit

(Folie zur Vorlesung)

Primitive Konstruktion

- Siehe Sondertext
- Lineares Gleichungssystem für natürliche interpolierende Splines
- Eindeutige Lösbarkeit

4.4 Symmetrisierung

(Folie zur Vorlesung)

Symmetrisierung

- Siehe Sondertext
- Übergang zu einem symmetrischen Kern
- Bedingte positive Definitheit

4.5 Fehlerabschätzung

(Folie zur Vorlesung)

Fehlerabschätzung

- Siehe altes Skript 2001, Teil 1, S. 222
- *Lemma 10.7.19 dort*
- *Siehe Zusatztext:*
- *Verallgemeinerung auf höhere Differenzierbarkeit*
- *Verdoppelung der Konvergenzordnung*

4.6 Kubische Splines

(Folie zur Vorlesung)

Kubische Splines

- Siehe altes Skript 2001
- Siehe Zusatztext
- Interpolation bei beliebigen Randbedingungen
- Konvergenz 4. Ordnung

4.7 B–Splines

(Folie zur Vorlesung)

B–Splines

- Siehe altes Skript 2001, zweiter Teil
- Siehe Zusatztext
- Definition
- Rekursionsformel
- Zerlegung der Eins
- de Boor-Verfahren

The following is a somewhat nonstandard introduction to splines, modeled for later extensions to general multivariate kernel-based function spaces.

4.8 Smoothest Interpolation

First we fix a positive integer k.

4.8.1 Semi-inner product

As a function space, we use the vector space $\mathcal{C}^k[a,b]$ of all real-valued functions f with piecewise continuous k-th derivatives for which

$$|f|_k^2 := \int_a^b \left(\frac{d^k f(t)}{dt^k} \right)^2 dt \tag{4.1}$$

is finite. We leave it to the reader that this defines a reasonable vector space of functions on $[a,b]$.

Equation (4.1) defines a semi-norm, i.e. it has the properties of a norm except for the definiteness, and there is a semi-inner product

$$(f,g)_k := \int_a^b \frac{d^k f(t)}{dt^k} \frac{d^k g(t)}{dt^k} dt.$$

Lemma 4.1 *The seminorm $|f|_k$ is zero if and only if f is a polynomial of degree at most $k-1$.*

Proof: Clearly, the seminorm $|f|_k$ is zero if f is a polynomial of degree at most $k-1$. Conversely, if the seminorm $|f|_k$ is zero for some function $f \in \mathcal{C}^k[a,b]$, then $f^{(k)}$ is zero except for its points of discontinuity. Then f consists of polynomial pieces of degree at most $k-1$ which are glued together in such a way that the $(k-1)$st derivative still is continuous. But then f is a global polynomial of degree at most $k-1$. □

4.8.2 Taylor's Formula

Every function f on $[a,b]$ with k continuous derivatives satisfies

$$f(x) = \sum_{j=0}^{k-1} \frac{f^{(j)}(a)}{j!}(x-a)^j + \int_a^x f^{(k)}(t)\frac{(x-t)^{k-1}}{(k-1)!}dt,\ x \in [a,b]$$

and this generalizes to functions in $\mathcal{C}^k[a,b]$ (without proof here). The upper bound x of the integral can be eliminated by defining the *truncated power* as

$$(z)_+^k := \begin{cases} z^k & z > 0 \\ 0 & z < 0 \\ \frac{1}{2} & z = 0,\ k = 0 \\ 0 & \text{else} \end{cases}$$

to get

$$f(x) = \sum_{j=0}^{k-1} \frac{f^{(j)}(a)}{j!}(x-a)^j + \int_a^b f^{(k)}(t)\frac{(x-t)_+^{k-1}}{(k-1)!}dt,\ x \in [a,b].$$

With the *kernel function*

$$K_k(x,t) := (-1)^k \frac{(x-t)_+^{2k-1}}{(2k-1)!}$$

the above equation takes the form

$$\begin{aligned} f(x) &= \underbrace{\sum_{j=0}^{k-1} \frac{f^{(j)}(a)}{j!}(x-a)^j}_{=:(P_{k-1}f)(x)} + (f, K_k(x,\cdot))_k \\ &= (P_{k-1}f)(x) + (f, K_k(x,\cdot))_k,\ x \in [a,b]. \end{aligned} \tag{4.2}$$

This is a *reproduction formula*, i.e. it allows f to be reproduced from $f^{(k)}$ in $[a,b]$ and the derivatives at a up to order $k-1$.

4.8.3 Taylor's Formula Symmetrized

But note that we have tackled a symmetric problem in an unsymmetric way, which is a mathematical crime. We should also use Taylor's formula at b. This is

$$\begin{aligned} f(x) &= \sum_{j=0}^{k-1} \frac{f^{(j)}(b)}{j!}(x-b)^j + \int_b^x f^{(k)}(t)\frac{(x-t)^{k-1}}{(k-1)!}dt,\ x \in [a,b] \\ &=: (Q_{k-1}f)(x) + \int_x^b f^{(k)}(t)(-1)^k\frac{(t-x)^{k-1}}{(k-1)!}dt \\ &= (Q_{k-1}f)(x) + \int_a^b f^{(k)}(t)(-1)^k\frac{(t-x)_+^{k-1}}{(k-1)!}dt. \end{aligned}$$

To get something symmetric, we take the mean of the two Taylor formulae. This is

$$\begin{aligned} f(x) &= \tfrac{1}{2}(P_{k-1}f)(x) + \tfrac{1}{2}(Q_{k-1}f)(x) \\ &\quad + \tfrac{1}{2}\int_a^b f^{(k)}(t)\left(\frac{(x-t)_+^{k-1}}{(k-1)!} + (-1)^k\frac{(t-x)_+^{k-1}}{(k-1)!}\right)dt \\ &=: (R_{k-1}f)(x) + (f, \Phi_k(x,\cdot))_k \end{aligned} \tag{4.3}$$

with

$$\begin{aligned}(R_{k-1}f)(x) &:= \tfrac{1}{2}(P_{k-1}f)(x) + \tfrac{1}{2}(Q_{k-1}f)(x) \\ &= \frac{1}{2}\sum_{j=0}^{k-1}\frac{f^{(j)}(a)}{j!}(x-a)^j + \frac{1}{2}\sum_{j=0}^{k-1}\frac{f^{(j)}(b)}{j!}(x-b)^j \\ \Phi_k(x,t) &:= \tfrac{1}{2}(-1)^k\frac{|x-t|^{2k-1}}{(2k-1)!}.\end{aligned}$$

To see that the form of the new symmetric kernel Φ_k is correct, we take its k-th derivative with respect to t for the two cases

$$\begin{aligned}\Phi_k(x,t) &= \tfrac{1}{2}(-1)^k\frac{(x-t)^{2k-1}}{(2k-1)!} \qquad x \geq t \\ \Phi_k(x,t) &= \tfrac{1}{2}(-1)^k\frac{(t-x)^{2k-1}}{(2k-1)!} \qquad t \geq x\end{aligned}$$

and get

$$\begin{aligned}\tfrac{d^k}{dt^k}\tfrac{1}{2}(-1)^k\frac{(x-t)^{2k-1}}{(2k-1)!} &= \tfrac{1}{2}\frac{(x-t)^{k-1}}{(k-1)!} \qquad x \geq t \\ \tfrac{d^k}{dt^k}\tfrac{1}{2}(-1)^k\frac{(t-x)^{2k-1}}{(2k-1)!} &= \tfrac{1}{2}(-1)^k\frac{(t-x)^{k-1}}{(k-1)!} \qquad t \geq x\end{aligned}$$

where we can add the $_+$ subscript in both cases in order to arrive at (4.3).

Note that the two reproduction formulae (4.2) and (4.3) can both be used to our convenience. The different kernels are linked to different polynomial projectors.

4.8.4 Smoothest Interpolation

We assume M points $x_1 < x_2 < \cdots < x_M$ in $[a,b] \subset I\!R$ and corresponding real values $y_1, \ldots, y_M$ to be given, and we want to find a function $s^* \in \mathcal{C}^k[a,b]$ which minimizes $|s|_k^2$ under all functions $s \in \mathcal{C}^k[a,b]$ satisfying the *interpolation conditions*

$$s(x_j) = y_j,\ 1 \leq j \leq M.$$

In contrast to standard polynomial interpolation, we keep the smoothness k fixed and allow very large numbers M of data points, asking for the "smoothest" possible interpolant. Note that this is an infinite-dimensional quadratic optimization problem with linear constraints. But we shall not plunge deeply into optimization here and try to solve the problem single-handed.

If the data are values $p(x_j) = y_j$ of a polynomial $p \in I\!P_{k-1}$, the solution obviously is p with $|p|_k = 0$. To assure uniqueness of interpolation even in such a simple case, we need the additional assumption $M \geq k$.

We shall not directly prove the existence of a smoothest interpolant s^*. Instead, we first assume it exists, then derive its necessary form, and finally prove that it can be numerically calculated in its necessary (and simplified) form, proving existence constructively.

If s^* is our "smoothest" interpolant, we now repeat the "parabola argument" used for proving the characterization of best approximants in Euclidean spaces. Take any real number λ and any function $v \in \mathcal{C}^k[a,b]$ with $v(x_j) = 0,\ 1 \leq j \leq M$. Then for all such λ and v we have

$$\begin{aligned}|s^* + \lambda v|_k^2 &= |s^*|_k^2 + 2\lambda(s^*, v)_k + \lambda^2|v|_k^2 \\ &\geq |s^*|_k^2\end{aligned}$$

and this implies

$$(s^*, v)_k = 0 \text{ for all } v \in \mathcal{C}^k[a,b] \text{ with } v(x_j) = 0,\ 1 \le j \le M. \tag{4.4}$$

This argument can be put upside down and proves that any interpolating function s^* with (4.4) must be a smoothest interpolant.

If we define the linear data map $T : \mathcal{C}^k[a,b] \to I\!R^M$ with

$$Tv := (v(x_1), \ldots, v(x_M)),\ v \in \mathcal{C}^k[a,b],$$

and the linear functional $\mu^*(v) := (s^*, v)_k$, the property (4.4) is

$$\mu^*(v) = 0 \text{ for all } v \in \mathcal{C}^k[a,b] \text{ with } T(v) = 0.$$

But then there is a vector $\alpha \in I\!R^M$ with

$$\mu^*(v) = \alpha^T T(v) \text{ for all } v \in \mathcal{C}^k[a,b].$$

This is a standard argument of linear algebra. It follows from the fact that T is surjective and thus the range $I\!R^M = T(\mathcal{C}^k[a,b])$ is isomorphic to the quotient space via $\mathcal{C}^k[a,b]/\ker\ T \xrightarrow{Q} T(\mathcal{C}^k[a,b])$. Since it vanishes on $\ker T$, the functional μ^* can be safely defined on the quotient space and thus be written via the range of T as $\mu^* = Q\alpha = \alpha^T T$. We now know that

$$(s^*, v)_k = \alpha^T T(v) = \sum_{j=1}^{M} \alpha_j v(x_j) \tag{4.5}$$

holds for all $v \in \mathcal{C}^k[a,b]$, and we insert (4.2) to get

$$\begin{aligned}
(s^*, v)_k &= \sum_{j=1}^{M} \alpha_j \left((P_{k-1}v)(x_j) + (v, K_k(x_j, \cdot))_k \right) \\
&= \sum_{j=1}^{M} \alpha_j (P_{k-1}v)(x_j) + (v, \sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot))_k \\
(s^* - \sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot), v)_k &= \sum_{j=1}^{M} \alpha_j (P_{k-1}v)(x_j).
\end{aligned}$$

If we replace v in (4.5) by $P_{k-1}v$, we see that

$$0 = (s^*, P_{k-1}v)_k = \alpha^T T(P_{k-1}v) = \sum_{j=1}^{M} \alpha_j P_{k-1}v(x_j)$$

for all $v \in \mathcal{C}^k[a,b]$. Since P_{k-1} clearly is surjective, this implies

$$\sum_{j=1}^{M} \alpha_j q(x_j) = 0 \text{ for all } q \in I\!P_{k-1}. \tag{4.6}$$

Furthermore,

$$0 = (s^* - \sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot), v)_k$$

for all $v \in \mathcal{C}^k[a,b]$. For the special case

$$v := p := s^* - \sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot)$$

Lemma 4.1 now implies that p is a polynomial in $\mathbb{P}_{k-1}$. This proves the first part of

Theorem 4.1 *The "smoothest" interpolant s^*, if it exists, has the form*

$$s^* = p + \sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot) \tag{4.7}$$

with a polynomial $p \in \mathbb{P}_{k-1}$ and M coefficients $\alpha_1, \ldots, \alpha_M$ satisfying (4.6)*. Conversely, if a function s^* of the form* (4.7) *with* (4.6) *interpolates the data, it is the "smoothest" interpolant.*

Proof of the converse: Just follow the above argument backwards to arrive at the "parabola argument". Details are left to the reader. □

Note that exactly the same argument works when using the symmetric kernel Φ_k instead of K_k.

4.8.5 Primitive Construction

We still have to prove that the "smoothest interpolant" exists. But since we now know what it should look like, we prove existence constructively. But please keep in mind that there are better algorithms to construct the solution. We shall derive these later.

If we introduce a basis $p_1, \ldots, p_k$ for $\mathbb{P}_{k-1}$, we can write the candidate for a smoothest interpolant as

$$s^* := \sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot) + \sum_{\ell=1}^{k} \beta_\ell p_\ell$$

with the additional conditions (4.6) in the form

$$\sum_{j=1}^{M} \alpha_j p_\ell(x_j) = 0, \ 1 \le \ell \le k.$$

Again, the following argument works similarly for the symmetric kernel Φ_k instead of K_k.

Together with the usual interpolation conditions

$$s^*(x_i) = \sum_{j=1}^{M} \alpha_j K_k(x_j, x_i) + \sum_{\ell=1}^{k} \beta_\ell p_\ell(x_i) = y_i, \ 1 \le i \le M$$

we get the $(M+k) \times (M+k)$ block system

$$\begin{pmatrix} A & P \\ P^T & 0_{\ell \times \ell} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} y \\ 0_\ell \end{pmatrix} \tag{4.8}$$

with the matrices and vectors

$$\begin{aligned} A &:= (K_k(x_j, x_i))_{1 \le i,j \le M} \\ P &:= (p_\ell(x_i))_{1 \le i \le M,\ 1 \le \ell \le k} \\ y^T &:= (y_1, \ldots, y_M). \end{aligned}$$

Theorem 4.2 *If $M \geq k$ holds, the system* (4.8) *is uniquely solvable.*

Proof: We show that the homogeneous system has only the trivial solution. Assume that a homogeneous solution is given by vectors $\alpha \in I\!R^M$ and $\beta \in I\!R^k$. We then define s^* and p as in the above argument and see that s^* is the smoothest interpolant to zero data. Since the zero function also does the job, we necessarily have $|s^*|_k = 0$ and $s^* \in I\!P_{k-1}$. But since s^* interpolates zero in $M \geq k$ points, it must be zero everywhere.

Then, for every $v \in \mathcal{C}^k[a,b]$ we have

$$\begin{aligned} 0 &= (s^*, v)_k \\ &= 0 + \left(\sum_{j=1}^{M} \alpha_j K_k(x_j, \cdot), v \right)_k \\ &= \sum_{j=1}^{M} \alpha_j \left(K_k(x_j, \cdot), v \right)_k \\ &= \sum_{j=1}^{M} \alpha_j \left(v(x_j) - (P_{k-1}v)(x_j) \right) \\ &= \sum_{j=1}^{M} \alpha_j v(x_j) \end{aligned}$$

due to (4.6). By picking some useful v, e.g. as Lagrange interpolating polynomials, we get that all α_j must vanish. But the remaining equations then are $P\beta = 0$ and imply that the polynomial

$$p := \sum_{\ell=1}^{k} \beta_\ell p_\ell$$

vanishes at all $M \geq k$ data points. Thus its coefficients must all be zero. □

4.8.6 Properties

From Theorem 4.1 and equation (4.7) we see that the smoothest interpolant is of the form

$$s^*(x) = \sum_{\ell=1}^{k} \beta_\ell p_\ell(x) + \sum_{j=1}^{M} \alpha_j (x_j - x)_+^{2k-1}$$

or, equivalently, but with different coefficients,

$$s^*(x) = \sum_{\ell=1}^{k} \beta_\ell p_\ell(x) + \sum_{j=1}^{M} \alpha_j |x_j - x|_+^{2k-1}$$

with the additional conditions (4.6). Thus it is a piecewise polynomial of degree at most $2k-1$ with "breakpoints" or "knots" at the data locations x_j. It still has $2k-2$ continuous derivatives, which is roughly twice the smoothness originally postulated in the space $\mathcal{C}^k[a,b]$ except for $k = 1$ and $k = 2$.

Furthermore, the first form tells us that it is a polynomial of degree at most $k-1$ in $[a, x_0]$. Since the equivalent second form is symmetric, we conclude in general that s^* is a polynomial of degree at most $k-1$ outside the data locations.

Altogether, the conditions

1. s^* interpolates in $x_0 < \ldots < x_M$ in $[a,b]$ and
2. is a C^{2k-2} function
3. consisting of polynomials of degree at most $2k-1$ in each data interval $[x_j, x_{j+1}]$ and
4. a polynomial of degree at most $k-1$ outside $[x_0, x_M]$

uniquely define the solution to our problem, which is traditionally called the "natural interpolating spline of degree $2k-1$".

4.8.7 Symmetrization

In view of later multivariate methods, we take a closer look at the symmetric kernel Φ_k. In particular,

$$(\Phi_k(x,\cdot),\Phi_k(y,\cdot))_k \;=\; \Phi_k(x,y) - (R_{k-1}\Phi_k(x,\cdot))(y)$$

and due to symmetry of the two other parts,

$$(R_{k-1}\Phi_k(x,\cdot))(y) = (R_{k-1}\Phi_k(y,\cdot))(x).$$

Lemma 4.2 *IF $M \geq k$ holds, anf if formed with Φ_k, the matrix A defines a quadratic form which is positive definite on the subspace of vectors $\alpha \in I\!R^M$ with* (4.6).

Proof: The quadratic form defined by A and taken on the vectors $\alpha \in I\!R^M$ with (4.6) is

$$\begin{aligned}
\alpha^T A\alpha &= \sum_{i=0}^{M}\sum_{j=0}^{M} \alpha_i\alpha_j\Phi_k(x_i,x_j) \\
&= \sum_{i=0}^{M}\sum_{j=0}^{M} \alpha_i\alpha_j(\Phi_k(x_i,\cdot),\Phi_k(x_j,\cdot))_k + 0 \\
&= \left(\sum_{i=0}^{M}\alpha_i\Phi_k(x_i,\cdot), \sum_{j=0}^{M}\alpha_j\Phi_k(x_j,\cdot)\right)_k \\
&= \left|\sum_{i=0}^{M}\alpha_i\Phi_k(x_i,\cdot)\right|_k^2 \\
&\geq 0
\end{aligned}$$

and thus positive semidefinite. If it vanishes, then

$$p(x) := \sum_{i=0}^{M}\alpha_i\Phi_k(x_i,x)$$

must be a polynomial in $I\!P_{k-1}$. With the same argument as in the proof of Theorem 4.2, now taking p instead of s^*, we get that all α_i must vanish if (4.6) holds. □

Definition 4.1 *A kernel with the property described by Lemma 4.2 for all matrices arising on $M \geq k$ points is called* conditionally positive definite of order k.

This property will come up later in multivariate settings. For instance, the *Gaussian* kernel

$$G(x,y) := \exp(-\|x-y\|_2^2),\ x,y \in I\!R^d$$

is positive definite (i.e. conditionally positive definite of order 0) on all spaces $I\!R^d$. In particular, for all sets of M vectors $x_1,\ldots,x_M$ in $I\!R^d$ and arbitrary dimension d, the $M \times M$ matrix with entries $\exp(-\|x_i - x_j\|_2^2)$ is always positive definite. Thus interpolation at the x_j with linear combinations

$$\sum_{i=1}^{M} \alpha_i \exp(-\|x_i - x\|_2^2)$$

will always work, giving us an escape from the Mairhuber theorem, because we have a data-dependent space.

4.9 Convergence

From the 2001 German text cited on the website, we take

Lemma 4.3 *Es sei $g \in C^1(I)$. Ferner sei $\delta > 0$ eine Konstante mit der Eigenschaft, daß für jedes $x \in I$ im Intervall $[x-\delta, x+\delta] \cap I$ mindestens eine Nullstelle $x^*(x)$ von g liege. Dann gelten mit den Normen bzw. Seminormen*

$$\|g\|_\infty := \max_{t\in I} |g(t)|$$

und

$$\|g\|_{(j)} := \left[\int_I (g^{(j)}(t))^2 dt\right]^{1/2} \qquad (j = 0, 1)$$

die Abschätzungen

$$\|g\|_{(0)} \le \frac{\delta}{\sqrt{2}}\, \|g\|_{(1)} \tag{4.9}$$

$$\|g\|_\infty \le \delta\, \|g'\|_\infty \tag{4.10}$$

$$\|g\|_\infty \le \delta^{1/2}\, \|g\|_{(1)} \tag{4.11}$$

Beweis: Für jedes $x \in I$ hat man wegen $g(x^*(x)) = 0$ die Identität

$$g(x) = \int_{x^*(x)}^{x} g'(\tau) d\tau \qquad (x \in I)$$

und (4.10) ergibt sich, wenn man den Integranden durch sein Maximum zwischen x und $x^*(x)$ ersetzt. Aus der Cauchy-Schwarzschen Ungleichung erhält man

$$\begin{aligned} |g(x)| &\le \left|\int_{x^*(x)}^{x} 1^2 d\tau\right|^{1/2} \cdot \left|\int_{x^*(x)}^{x} (g'(\tau))^2 d\tau\right|^{1/2} \qquad (x \in I) \\ &\le |x - x^*(x)|^{1/2} \cdot \left|\int_{x^*(x)}^{x} (g'(\tau))^2 d\tau\right|^{1/2}, \end{aligned}$$

woraus (4.11) folgt. Durch Quadrieren ergibt sich ferner

$$g^2(x) \le |x - x^*(x)| \cdot \left| \int_{x^*(x)}^{x} (g'(\tau))^2 d\tau \right| \qquad (x \in I),$$

und daher gilt

$$g^2(t) \le |t - x^*(x)| \cdot \left| \int_{x^*(x)}^{x} (g'(\tau))^2 d\tau \right|$$

für alle t zwischen $x^*(x)$ und x. Durch Integration folgt

$$\left| \int_{x^*(x)}^{x} g^2(t) dt \right| \le \frac{1}{2} |x - x^*(x)|^2 \cdot \left| \int_{x^*(x)}^{x} (g'(\tau))^2 d\tau \right|$$

und da sich das Intervall I als Vereinigung endlich vieler Intervalle der Form $[x^*(x), x]$ bzw. $[x, x^*(x)]$ darstellen läßt, kann man die obigen Integrale zusammenfassen zu

$$\|g\|_{(0)}^2 \le \frac{\delta^2}{2} \|g\|_{(1)}^2.$$

Damit ist Lemma 4.3 bewiesen. □

Everbody in the 2006 lecture should get the idea how this works, even if the text is German.

In the notation of the new lecture, we have

Theorem 4.3 *Let $f \in \mathcal{C}^k[a,b]$ be interpolated by s^* in $M \ge k$ data with a* fill distance

$$h := \sup_{x \in [a,b]} \min_{x_j} |x - x_j|.$$

Note that h here is δ above. Then there is a constant c_k depending only on k and $[a,b]$, but not on f or the data or h, such that

$$\begin{array}{rcrcr} \|f - s^*\|_{L_2[a,b]} & \le & c_k h^k |f - s^*|_k & \le & 2c_k h^k |f|_k, \\ \|f - s^*\|_{L_\infty[a,b]} & \le & c_k h^{k-1/2} |f - s^*|_k & \le & 2c_k h^{k-1/2} |f|_k. \end{array}$$

Proof: Note that the zeros of $f - s^*$ have a distance of at most $2h$ between each other and of at most $h \le 2h$ to the boundary. By Rolle's theorem, there are zeros of $(f - s^*)'$ with distance of at most $4h$ between each other and $3h$ to the boundary. This means that we can use the fill distance $4h$ for the zeros of the first derivative. This works up to the derivative of order $k-1$, which has zeros with distance of at most $4^{k-1}h$ between each other and to the boundary. Using induction on the previous Lemma yields

$$\|f - s^*\|_{L_2[a,b]} \le \frac{h \cdot 4h \cdots 4^{k-1}h}{2^{k/2}} |f - s^*|_k =: c_k h^k |f - s^*|_k$$

and the first part of the second assertion follows when taking (4.11) instead of (4.9) once.

For the right-hand parts we use the optimality condition $|s^*|_k \le |f|_k$. □

If some additional boundary conditions are satisfied, the convergence order doubles.

Theorem 4.4 *If, in addition, $f \in \mathcal{C}^{2k}[a,b]$ and if $(f-s^*)^{(j)}$ vanishes at a and b for $j = 0,\ldots,k-1$, then*

$$\begin{aligned} \|f-s^*\|_{L_2[a,b]} &\leq c_k^2 h^{2k} |f|_{2k}. \\ \|f-s^*\|_{L_\infty[a,b]} &\leq \tilde{c}_k^2 h^{2k-1} |f|_{2k}. \end{aligned}$$

Proof: We can use the orthogonality relation

$$(f-s^*, s^*)_k = 0$$

and do integration by parts via

$$\begin{aligned} |f-s^*|_k^2 &= (f-s^*, f-s^*)_k \\ &= (f-s^*, f)_k \\ &= \int_a^b (f-s^*)^{(k)}(t) f^{(k)}(t) dt \\ &= (-1)^k \int_a^b (f-s^*)^{(0)}(t) f^{(2k)}(t) dt \\ &\leq \|f-s^*\|_{L_2[a,b]} |f|_{2k}. \end{aligned}$$

Then

$$\begin{aligned} \|f-s^*\|_{L_2[a,b]}^2 &\leq c_k^2 h^{2k} |f-s^*|_k^2 \\ &\leq c_k^2 h^{2k} \|f-s^*\|_{L_2[a,b]} |f|_{2k} \\ \|f-s^*\|_{L_2[a,b]} &\leq c_k^2 h^{2k} |f|_{2k}. \end{aligned}$$

Similarly,

$$\begin{aligned} \|f-s^*\|_{L_\infty[a,b]}^2 &\leq c_k^2 h^{2k-1} |f-s^*|_k^2 \\ &\leq c_k^2 h^{2k-1} \|f-s^*\|_{L_2[a,b]} |f|_{2k} \\ &\leq \sqrt{b-a}\, c_k^2 h^{2k-1} \|f-s^*\|_{L_\infty[a,b]} |f|_{2k} \\ \|f-s^*\|_{L_\infty[a,b]} &\leq \tilde{c}_k^2 h^{2k-1} |f|_{2k}. \end{aligned}$$

4.10 Cubic Splines

Sorry again, but I have no time to translate this old text into English and to introduce consistent notation.

Zur graphischen Interpolation einer Reihe von Datenpunkten (x_j, f_j), $0 \leq j \leq N$, mit einer **Knotenfolge**

$$X: \quad a = x_0 < x_1 < \ldots < x_N = b \ \text{ in } \ I := [a,b] \tag{4.12}$$

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

$$\int_I \frac{(y''(t))^2}{1+y'^2(t)} dt, \tag{4.13}$$

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 (4.13) näherungsweise durch

$$\int_I y''(t)^2 \, dt \tag{4.14}$$

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 4.2 *Die Funktionen aus dem linearen Raum*

$$\mathcal{S}_k(X) := \left\{ s \in C^{k-1}(I) \;\middle|\; s|_{[x_{i-1},x_i]} \text{ liegt in } \mathcal{P}_k,\ 1 \le i \le N \right\} \tag{4.15}$$

heißen **polynomiale Spline–Funktionen** *oder* **Splines** *vom Grad $\le k$ auf der Zerlegung X gemäß* (4.12)*.*

Beispiel 1 *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-*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.

Stammen die Daten $f_i = f(x_i)$ nämlich von einer Funktion $f \in C^2[a,b]$, so gilt nach Satz **??** *die Fehlerabschätzung*

$$|f(x) - s(x)| \le \frac{1}{8} \, \|f''\|_\infty \cdot h^2$$

für alle $x \in [x_0, x_N]$ und $h := \max_{1 \le i \le n}(x_i - x_{i-1})$, weil man zwischen zwei Interpolationspunkten x_i und x_{i+1} stets $|(x - x_i)(x - x_{i+1})| \le h^2/4$ 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.*

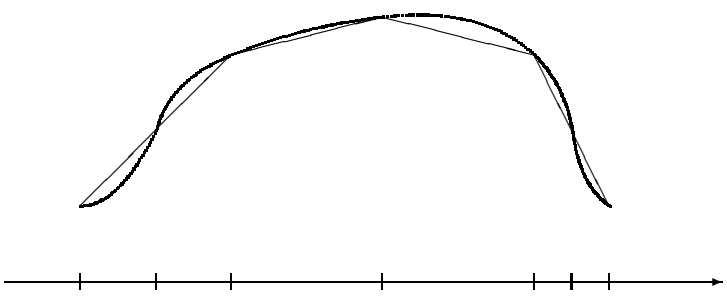

Abbildung 6: Polygonzug

Für die Praxis werden die im Falle $k = 3$ in Definition 4.2 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-Vorgaben angegeben werden. Allgemeinere Methoden zur Berechnung von Kurven und Flächen mit Spline-Funktionen finden sich in Abschnitt 4.11.

Zu festen Knoten (4.12) seien Interpolationsdaten $f_0 \ldots, f_N \in IR$ 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

$$\begin{aligned} h_j &:= x_j - x_{j-1} \qquad (1 \le j \le N) \\ M_j &:= s''(x_j) \qquad\quad (0 \le j \le N) \end{aligned} \tag{4.16}$$

gilt also

$$s''(x) = \frac{1}{h_j}\ (M_j(x - x_{j-1}) + M_{j-1}(x_j - x)) \qquad \text{für alle } x \in I_j. \tag{4.17}$$

Daraus folgt für die Restriktion von s auf $[x_{j-1}, x_j]$ durch zweimalige Integration

$$s(x) = \frac{1}{6h_j}\ (M_j(x - x_{j-1})^3 + M_{j-1}(x_j - x)^3) + b_j\ \left(x - \frac{x_j + x_{j-1}}{2}\right) + a_j \tag{4.18}$$

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

$$\begin{aligned} &(h_j + h_{j+1})\ \Delta^2(x_{j-1}, x_j, x_{j+1})f \\ &= \Delta^1(x_j, x_{j+1})f - \Delta^1(x_j, x_{j-1})f \\ &= (\Delta^1(x_j, x_{j+1})f - \Delta^1(x_j, x_j)f) + (\Delta^1(x_j, x_j)f - \Delta^1(x_{j-1}, x_j)f) \\ &= h_{j+1}\ \Delta^2(x_j, x_j, x_{j+1})f + h_j\ \Delta^2(x_{j-1}, x_j, x_j)f \end{aligned} \tag{4.19}$$

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 (4.18) von $s(x)$ andererseits die Gleichungen

$$\begin{aligned} &(h_j + h_{j+1})\ \Delta^2(x_{j-1}, x_j, x_{j+1})f = (h_j + h_{j+1})\ \Delta^2(x_{j-1}, x_j, x_{j+1})s \\ &= h_{j+1} \cdot \frac{1}{6h_{j+1}}\ (M_{j+1}h_{j+1} + 2M_j h_{j+1}) + h_j \cdot \frac{1}{6h_j}\ (2M_j h_j + M_{j-1}h_j). \end{aligned} \tag{4.20}$$

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

$$\mu_j M_{j-1} + M_j + \lambda_j M_{j+1} = 3 \cdot \Delta^2(x_{j-1}, x_j, x_{j+1})f \tag{4.21}$$

für $j = 1, \ldots, N-1$ mit den Größen

$$\mu_j := \frac{h_j}{2(h_j + h_{j+1})}\ , \quad \lambda_j := \frac{h_{j+1}}{2(h_j + h_{j+1})}\ , \quad \lambda_j + \mu_j = \frac{1}{2}. \tag{4.22}$$

In (4.21) 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 (4.21) 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 (4.21) 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 (4.18) 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 (4.21) 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 (4.17) folgt

$$s'''(x) = \frac{1}{h_j}\,(M_j - M_{j-1}) \quad \text{auf } [x_{j-1}, x_j]$$

und man hat $s'''(x_1^-) = s'''(x_1^+)$ genau dann, wenn

$$\frac{1}{h_1}\,(M_1 - M_0) = \frac{1}{h_2}\,(M_2 - M_1) \tag{4.23}$$

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 (4.21) für $j = 1$ durch Elimination von M_0 modifizieren oder (4.23) zu (4.21) 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 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 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 (4.12) mit $f \in C^4[a,b]$ und einem kubischen Spline, so ergibt sich aus der Identität (4.19) mit

$$\begin{aligned} A_j := \;& 6\lambda_j \Delta^2(x_j, x_j, x_{j+1})f + 6\mu_j \Delta^2(x_{j-1}, x_j, x_j)f \\ & -\lambda_j(f''(x_{j+1}) + 2f''(x_j)) - \mu_j(2f''(x_j) + f''(x_{j-1})) \end{aligned} \tag{4.24}$$

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 (4.21) folgt, daß die Werte $\varepsilon_j'' := s''(x_j) - f''(x_j)$ das System

$$\mu_j \varepsilon_{j-1}'' + \varepsilon_j'' + \lambda_j \varepsilon_{j+1}'' = A_j \tag{4.25}$$

erfüllen. Als Anwendung des Satzes von PEANO 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 (4.25) 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

$$\max_{0 \leq j \leq N} |s''(x_j) - f''(x_j)| \leq \frac{1}{4} h^2 \|f^{(4)}\|_\infty. \tag{4.26}$$

Ist $u(x)$ ein Polygonzug durch die Werte $(x_j, f''(x_j))$, so folgt für $x \in [x_{j-1}, x_j]$ nach der Konvergenzbetrachtung für Polygonzüge in Beispiel 1 die Fehlerabschätzung

$$|f''(x) - u''(x)| \leq \frac{h^2}{8} \, \|f^{(4)}\|_\infty.$$

Für den Fehler $s'' - f''$ ergibt sich wegen $u(x_j) = f_j''$

$$\|s'' - f''\|_\infty \leq \|s'' - u\|_\infty + \|u - f''\|_\infty \leq \max_j |s''(x_j) - f''(x_j)| + \frac{h^2}{8} \cdot \|f^{(4)}\|_\infty$$

$$\leq \frac{3}{8}\, h^2 \|f^{(4)}\|_\infty.$$

Damit erhält man einen Teil von

Satz 1 *Die kubische Spline-Interpolierende $s \in C^2[a,b]$ der* LAGRANGE*-Daten einer Funktion $f \in C^4[a,b]$ in den Punkten*

$$a = x_0 < x_1 < \ldots < x_N = b$$

mit den Randbedingungen

$$s''(a) = f''(a), \quad s''(b) = f''(b)$$

genügt mit $h := \max_j |x_j - x_{j-1}|$ 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 und des obigen Lemmas.

4.11 B-Splines

Bei der praktischen Rechnung mit Spline-Funktionen aus dem schon in 4.2 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.

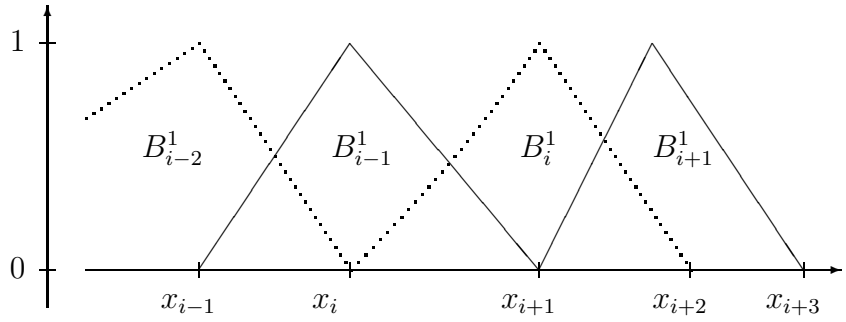

Abbildung 7: B-Splines ersten Grades

Beispiel 2 *Im Falle $k = 1$ der Polygonzüge ist das besonders einfach; bis auf einen Faktor kann man die "Dach-Funktionen"*

$$B_j^1(t) := \left\{ \begin{array}{cc} \frac{t - x_j}{x_{j+1} - x_j} & x_j \leq t \leq x_{j+1} \\ \frac{x_{j+2} - t}{x_{j+2} - x_{j+1}} & x_{j+1} \leq t \leq x_{j+2} \\ 0 & \textit{sonst} \end{array} \right\} \tag{4.27}$$

mit dem in Abb. 7 gezeigten Verlauf nehmen. Mit der schon beim Satz von Peano *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 $x,\ t \in IR,\ k \geq 0$ läßt sich durch Einsetzen der Alternativen für t aus (4.27) *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 4.3 *Zu allen $i \in \mathbb{Z}$ seien paarweise verschiedene Punkte $x_i \in IR$ mit $-\infty < \ldots < x_{-1} < x_0 < x_1 \ldots < \infty$ vorgegeben. Dann heißen die Funktionen*

$$B_j^r(t) := (x_{j+r+1} - x_j)\Delta_x^{r+1}(x_j, \ldots, x_{j+r+1})(x-t)_+^r \tag{4.28}$$

*(für $j \in \mathbb{Z},\ r \geq 0$) auch B-***Splines***.*

Beispiel 3 *Man erhält für $r = 0$ auch*

$$B_j^0(t) := (x_{j+1} - t)_+^0 - (x_j - t)_+^0 = \begin{cases} 0 & x_{j+1} < t \\ 1 & x_j < t < x_{j+1} \\ 0 & t < x_j \end{cases} \tag{4.29}$$

Diese Funktionen bilden natürliche Basen für Räume von Treppenfunktionen. In den Sprungstellen wird das Mittel des rechts- und linksseitigen Grenzwertes genommen.

Beispiel 4 *Für dieselbe Knotenverteilung wie in Abb. 7 zeigen die Abbildungen 27 bzw. 28 die quadratischen bzw. kubischen B-Splines.*

Satz 2 *Für $r \geq 1$ haben die B-Splines folgende Eigenschaften:*

$$B_j^r \in C^{r-1}(IR), \ \text{ falls } \ r \geq 1 \ \text{ (sonst stückweise stetig)} \tag{4.30}$$

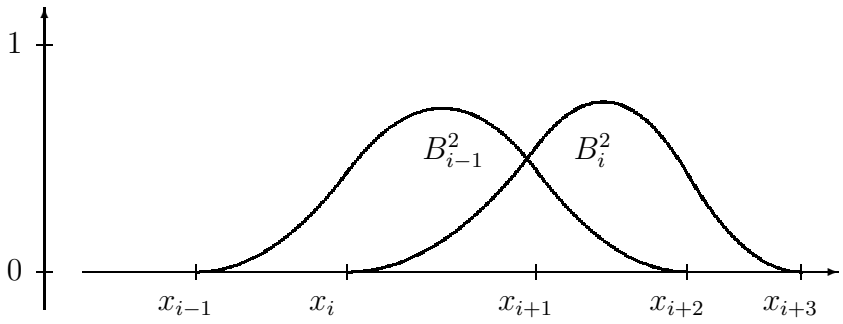

Abbildung 27: B-Splines zweiten Grades

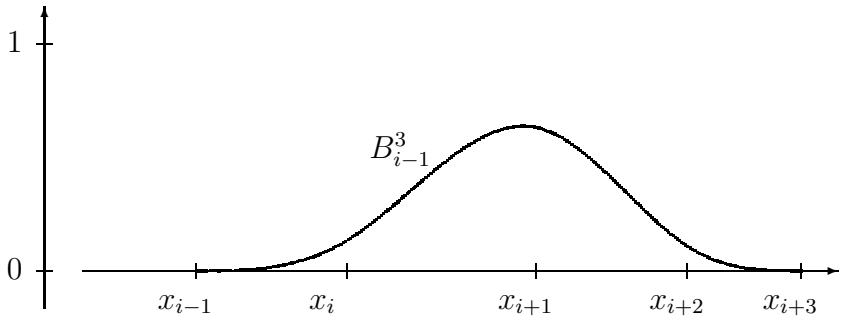

Abbildung 28: B-Spline dritten Grades

$$B_j^r \in \mathcal{P}_r \text{ in } (x_j, x_{j+r+1}) \tag{4.31}$$

$$B_j^r = 0 \text{ in } (x_{j+r+1}, \infty) \text{ und } (-\infty, x_j) \tag{4.32}$$

$$B_j^r(t) = \frac{x_{j+r+1} - t}{x_{j+r+1} - x_{j+1}} B_{j+1}^{r-1}(t) + \frac{t - x_j}{x_{j+r} - x_j} B_j^{r-1}(t) \tag{4.33}$$

$$B_j^r(t) > 0 \quad \textit{für } t \in (x_j, x_{j+r+1}), \qquad r \geq 0 \tag{4.34}$$

$$\sum_j B_j^{(r)}(t) = 1 \qquad \textit{für alle } t \in IR,\ r \geq 0. \tag{4.35}$$

Beweis: Die Aussagen (4.30), (4.31) und (4.32) 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 (4.33). Jetzt ist (4.34) leicht induktiv nachzuweisen; als Induktionsanfang nimmt man (4.29). Gilt (4.34) 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 (4.33) als Linearkombination positive Gewichte und es ist mindestens ein Summand positiv. Ebenso beweist man (4.35) durch Induktion, wobei man mit (4.29) beginnt und (4.33) 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 IR gegeben, so bilden die zugehörigen B-Splines nach Satz 2 eine positive Zerlegung der Eins und man kann zu festem Grad $r \geq 1$ allgemeine Linearkombinationen

$$s(t) = \sum_i d_i B_i^r(t) \tag{4.36}$$

von normierten B-Splines betrachten, wobei die Koeffizienten d_i hier vektorwertig aus IR^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 (4.36) 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 3 *Verändert man in einer B-Spline-Kurve* (4.36) *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

$$s(t) = \sum_j d_j B_t^r(t) \tag{4.37}$$

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 (4.33) 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 (4.37) nur über $j = k-r, \ldots, k$ zu erstrecken, weil die übrigen B-Splines in t verschwinden. Es folgt

Satz 4 *Es sei* $\ldots x_{-1} < x_0 < x_1 < \ldots$ *eine Knotenfolge mit einer B-Spline-Linearkombination*

$$s(t) = \sum_j d_j B_j^r(t). \tag{4.38}$$

Ist dann $t \in (x_k, x_{k+1})$ *ein fester Punkt, so liefert die Rekursion*

$$\begin{aligned} d_j^{(0)}(t) &:= d_j, && k-r \le j \le k, \\ d_j^{(\ell+1)}(t) &:= \frac{x_{j+r-\ell}-t}{x_{j+r-\ell}-x_j} d_{j-1}^{(\ell)}(t) + \frac{t-x_j}{x_{j+r-\ell}-x_j} d_j^{(\ell)}(t), && \\ & && k-r+\ell+1 \le j \le k, \quad \ell = 0, 1, \ldots, r-1 \end{aligned} \tag{4.39}$$

als $d_k^{(r)}(t)$ *den Wert* $s(t)$.

Das Verfahren (4.39) 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.

Bemerkung 1 *Wegen* $x_j \le x_k < x_{k+1} \le x_{j+r-\ell}$ *verschwinden die Nenner in* (4.39) *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* (4.38) *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.*

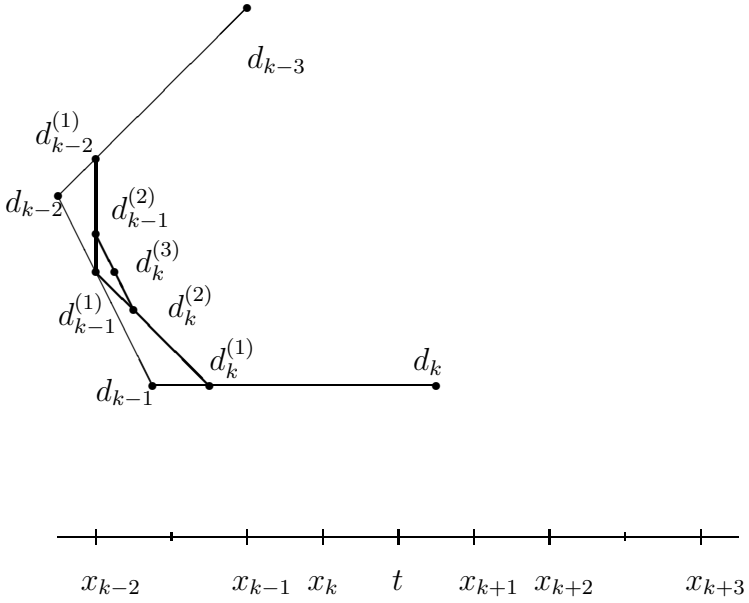

Abbildung 29: De-Boor-Verfahren

Die numerische Anwendung von B-Spline-Darstellungen wird erleichtert durch folgende zusätzliche Eigenschaften:

Satz 5 *Die B-Splines erfüllen die Gleichungen*

$$\frac{d}{dt}\, B_j^{(r)}(t) = r\left(\frac{B_j^{(r-1)}(t)}{x_{j+r} - x_j} - \frac{B_{j+1}^{(r-1)}(t)}{x_{j+r+1} - x_{j+1}}\right)$$

$$\int_{I\!R} B_j^{(r)}(t) = 1$$

$$\triangle_t^{r+1}(x_j, \ldots, x_{j+r+1}) f(t) = \frac{1}{(r+1)!}\, \frac{1}{x_{j+r+1} - x_j} \int_{I\!R} B_j^{(r)}(t) f^{(r+1)}(t) dt,$$

d.h. der B-Spline ist bis auf die Normierung der Peano-*Kern des Differenzenquotienten. Für Linearkombinationen*

$$s(t) = \sum_j d_j\, B_j^{(r)}(t)$$

sind die Formeln

$$s'(t) \;= \sum_j \frac{r(d_j - d_{j-1})}{x_{j+r} - x_j}\, B_j^{(r-1)}(t)$$

$$\int_{-\infty}^{x} s'(t)dt \;= \sum_j D_j B_j^{(r+1)}(x) \qquad \textit{mit}$$

$$D_j \;= D_{j-1} + d_j(x_{j+r+1} - x_j)/(r+1).$$

bei Differentiation und Integration nützlich.

Aufgabe 1 *Man beweise Satz 5.*

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.

$$\begin{aligned} 2(R_{k-1}\Phi_k(x,\cdot))(y) &= \sum_{j=0}^{k-1} \frac{d^j}{dt^j}_{|t=a} (-1)^k \frac{|x-t|^{2k-1}}{(2k-1)!} \frac{(y-a)^j}{j!} \\ &+ \sum_{j=0}^{k-1} \frac{d^j}{dt^j}_{|t=b} (-1)^k \frac{|x-t|^{2k-1}}{(2k-1)!} \frac{(y-b)^j}{j!} \end{aligned}$$

For $x \geq t$ this is

$$\begin{aligned} 2(R_{k-1}\Phi_k(x,\cdot))(y) &= \sum_{j=0}^{k-1} \frac{d^j}{dt^j}_{|t=a} (-1)^k \frac{(x-t)^{2k-1}}{(2k-1)!} \frac{(y-a)^j}{j!} \\ &+ \sum_{j=0}^{k-1} \frac{d^j}{dt^j}_{|t=b} (-1)^k \frac{(x-t)^{2k-1}}{(2k-1)!} \frac{(y-b)^j}{j!} \\ &= \sum_{j=0}^{k-1} (-1)^{k+j} \frac{(x-b)^{2k-1-j}}{(2k-1-j)!} \frac{(y-a)^j}{j!} \\ &+ \sum_{j=0}^{k-1} (-1)^{k+j} \frac{(x-a)^{2k-1-j}}{(2k-1-j)!} \frac{(y-b)^j}{j!} \\ &= \sum_{j=0}^{k-1} (-1)^k \binom{2k-1}{j} (x-b)^{2k-1-j} (a-y)^j \\ &+ \sum_{j=0}^{k-1} (-1)^k \binom{2k-1}{j} (x-a)^{2k-1-j} (b-y)^j \end{aligned}$$

The final form does not look symmetric in x and y, but the definition in the first line is. Note that for each fixed x the difference $\Phi_k(x,y) - K_k(x,y)$ is a polynomial of degree at most $k-1$ in y. Thus we can put this into (4.2) to get

$$f(x) = (P_{k-1}f)(x) + (f, \Phi_k(x,\cdot))_k, \ x \in [a,b] \tag{4.40}$$

and by the same argument also

$$\Phi_k(x,y) = (\Phi_k(x,\cdot), \Phi_k(y,\cdot))_k$$

for all $x, y \in [a, b]$. Therefore all of our considerations remain the same if we replace K_k by Φ_k, but note that the functions $K_k(x_j, x)$ and $\Phi_k(x_j, x)$ differ by a polynomial of degree at most $k-1$. The matrix A of the system (4.8) goes over into the symmetric matrix $\tilde{A}$ with the entries $\Phi_k(x_i, x_j)$, but the solution procedure is the same, just using a slightly different basis.

5 Shannon Sampling

(Folie zur Vorlesung)

Kapitel 5

Sampling

(Folie zur Vorlesung)

Inhalt dieses Kapitels (Vorschau)

- Fouriertransformation
- Shannon's sampling
- Shannon-Whittaker-Kotelnikov Theorem

5.1 Fouriertransformation

(Folie zur Vorlesung)

Fouriertransformation

- Siehe neuen Zusatztext auf der website
- Vorblick auf den Shannon-Operator:
  - Kardinale Interpolation auf $\mathbb{Z}$
  - sinc-Funktion
  - Shannon-Operator
- Definition: Fouriertransformation
- Fouriertransformierte der Gaußglocke
- Fouriertransformierte der sinc–Funktion
- Fouriertransformation auf $L_2(I\!R^d)$
- Parceval'sche Gleichung
- Rechenregeln

5.2 Shannon Sampling

(Folie zur Vorlesung)

Shannon Sampling

- Shannon-Operator
- Orthogonalität der Shifts der sinc-Funktion
- Sichtweise als L_2-Approximation
- Ausrechnen des Spans der Shifts der Sinc-Funktionen
- Bandbreitenbeschränkte Funktionen

5.3 Shannon-Whittaker-Kotelnikov Theorem

(Folie zur Vorlesung)

Shannon-Whittaker-Kotelnikov Theorem

- *Satz: Der Shannon-Operator reproduziert bandbreitenbeschränkte Funktionen*
- *Nyquist-Frequenz*
- *Was passiert bei allgemeinen Funktionen?*

5.4 Kardinale Interpolation

Das Manuskript dieses Kapitels war in 2004 ein Zusatztext zur Vorlesung "Mathematische Methoden der digitalen Signalverarbeitung". Diese Revision von 2006/2007 ist für die Vorlesung "Approximationsverfahren I". Zur Fouriertransformation benutze man einen parallelen Zusatztext, der ebenfalls revidiert wurde. Er ist hinten an dieses Kapitel angefügt.

Wir betrachten Interpolationsaufgaben auf einer biinfiniten Folge äquidistanter Punkte, d.h. auf $\mathbb{Z}$ ofder $h\mathbb{Z}$ 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.

Es geht im folgenden darum, aus einem Sampling die Funktion wieder zu rekonstruieren. Das ist der Normalfall beim Hören einer CD oder eines MP3-komprimierten Signals nach der digitalen Dekompression. In Anlehnung an die Lagrange-Interpolation macht man das am einfachsten durch Verschieben und Skalieren einer **kardinalen** Funktion $K : IR \to IR$ mit

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

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

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

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

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

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

$$\text{sinc}(x) \ := \ \frac{\sin(\pi x)}{\pi x},\ x \in IR.$$

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 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 einem gesonderten Text behandelt wird. Wir verwenden hier die symmetrische Fouriertransformation

$$\begin{aligned} \hat{f}(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{IR} f(t)e^{-it\omega}d\omega \\ f(x) &= \frac{1}{\sqrt{2\pi}} \int_{IR} \hat{f}(\omega)e^{it\omega}dt \end{aligned}$$

und die Parseval-Plancherel-sche Gleichung

$$(f,g)_{L_2(IR)} := \int_{IR} f(t)\overline{g(t)}dt = \int_{IR} \hat{f}(\omega)\overline{\hat{g}(\omega)}d\omega = (\hat{f},\hat{g})_{L_2(IR)}.$$

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

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

5.5 Die sinc–Funktion

Definition 5.1 *Wie schon oben vorweggenommen, wird*

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

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

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 5.1 *Für jedes feste $x \in IR$ gilt*

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

Beweis: Die erste Gleichung folgt aus

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

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

Lemma 5.2 *Die Funktionen* $\mathrm{sinc}\left(\frac{t}{h}-k\right)$ *liegen in* $L_2(IR)$ *und erfüllen die Orthogonalitätsrelation*

$$\left(\mathrm{sinc}\left(\frac{t}{h}-j\right),\mathrm{sinc}\left(\frac{t}{h}-k\right)\right)_{L_2(IR)} = h\delta_{jk},\ j,k\in ZZ,\ h>0.$$

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

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

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

□

5.6 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:

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

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

Definition 5.2 *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 $L_2(I\!R)$. Ihre Fouriertransformierte verschwindet außerhalb des Intervalls $[-\tau, \tau]$.

Lemma 5.3 *Für Funktionen u aus $BLF_{\pi/h}$ und alle $x \in I\!R$ gilt die* **Reproduktionsgleichung**

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

□

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

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

ist, der obendrein die bemerkenswerte Gleichung

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

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

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

können wir beliebige Funktionen $u \in L_2(I\!R)$ 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 $u \in L_2(I\!R)$ und alle $x \in I\!R$ die Gleichung

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

5.7 Beste Approximation in L_2 mit sinc–Funktionen

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

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

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

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

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

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

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

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

und

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

Theorem 5.1 *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 $k \in Z\!\!\!Z$. Die beste Approximation zu einem $u \in L_2(I\!R)$ ist die Shannon–Reihe zu $Trunc_{\pi/h}(u)$.* □

5.8 Shannon-Whittaker-Kotelnikov-Theorem

Aber das alles reicht nicht aus, um das berühmte Shannon-Whittaker-Kotelnikov-Theorem zu beweisen:

Theorem 5.2 *Alle Funktionen $u \in BLF_{\pi/h}$ sind durch ihre Shannon-Reihe im L_2–Sinne exakt reproduzierbar, d.h. es gilt*

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

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 5.3 *(Allgemeine Poisson'sche Summenformel)*
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.

Der Beweis steht im Zusatztext über Fouriertransformation, und dort wird auch

$$h^{d/2} \sum_{k \in \mathbb{Z}^d} u(hk) e^{-ihk^T\omega} = \left(\frac{2\pi}{h}\right)^{d/2} \sum_{j \in \mathbb{Z}^d} \hat{u}\left(\omega + \frac{2\pi j}{h}\right)$$

für den $I\!R^d$ bewiesen. 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 den Satz von Shannon–Whittaker–Kotelnikov 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. □

5.9 Fehlerabschätzung für sinc–Approximation

Aus dem Shannon-Theorem folgt eine ziemlich einfache, aber nützliche Fehlerabschätzung:

Theorem 5.4 *Die beste Approximation $P_h(u)$ einer beliebigen Funktion $u \in L_2(IR^d)$ durch orthonormale sinc–Funktionen $s_{k,h}$ hat den Fehler*

$$\|u - P_h(u)\|^2_{L_2(IR)} = \|u - Trunc_{\pi/h}(u)\|^2_{L_2(IR)} = \int_{|\omega|\geq\pi/h} |\hat{u}(\omega)|^2 d\omega.$$

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

$$u - P_h(u) = u - Trunc_{\pi/h}(u) + Trunc_{\pi/h}(u) - P_h(u) = u - Trunc_{\pi/h}(u).\square$$

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 5.3 *Der Raum*

$$W_2^\tau(IR^d) := \{u \in L_2(IR^d) \ : \ \int_{IR^d} |\hat{u}(\omega)|^2 \|\omega\|^{2\tau} d\omega < \infty\}$$

heißt **Sobolevraum** *der Ordnung τ auf IR^d. Er ist ein Hilbertraum mit dem inneren Produkt*

$$(u,v)_{W_2^\tau(IR^d)} := \int_{IR^d} \hat{u}(\omega)\overline{\hat{v}(\omega)}\|\omega\|^{2\tau} d\omega.$$

Man mache sich klar, daß die Funktionen $u \in W_2^\tau(IR^d)$ die Eigenschaft haben, daß alle Ableitungen bis zur Ordnung τ noch als $L_2(IR)$-Funktionen existieren. Zwar kann man diese Räume auch für nicht–ganzzahlige τ definieren, aber das soll hier nicht vertieft werden.

Theorem 5.5 *Die beste Approximation $P_h(u)$ einer beliebigen Funktion $u \in W_2^\tau(I\!R^d)$ durch orthonormale sinc–Funktionen $s_{k,h}$ hat den Fehler*

$$\|u - P_h(u)\|_{L_2(I\!R)} \leq \frac{h^\tau}{\pi^\tau}\|u\|_{W_2^\tau(I\!R^d)}.$$

Das beweist man durch Einsetzen in

$$\begin{aligned}
& \int_{|\omega|\geq\pi/h} |\hat{u}(\omega)|^2 d\omega \\
= & \int_{|\omega|\geq\pi/h} |\hat{u}(\omega)|^2 \frac{|\omega|^{2\tau}}{|\omega|^{2\tau}} d\omega \\
\leq & \left(\frac{h}{\pi}\right)^{2\tau} \int_{I\!R} |\hat{u}(\omega)|^2 |\omega|^{2\tau} d\omega \\
= & \left(\frac{h}{\pi}\right)^{2\tau} \|u\|^2_{W_2^\tau(I\!R^d)} \Box
\end{aligned}$$

Aber das ist 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 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" f mit $f = 1/h$. Dann hat man

$$f \geq 2F$$

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.

5.10 Direktes Shannon Sampling

Wenn man von einer gegebenen Funktion $u \in L_2(I\!R)$ 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 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 5.1 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

$$\begin{aligned}
S_{h,u}(t) &= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega} g(-h\omega)d\omega \\
&= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} e^{it\omega}\frac{\sqrt{2\pi}}{h}\sum_{j\in\mathbb{Z}}\hat{u}\left(\frac{h\omega-2\pi j}{h}\right)d\omega \\
&= \frac{1}{\sqrt{2\pi}}\int_{-\pi/h}^{\pi/h} e^{it\omega}\sum_{j\in\mathbb{Z}}\hat{u}\left(\omega-\frac{2\pi j}{h}\right)d\omega \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}}\int_{-\pi/h-2\pi j/h}^{\pi/h-2\pi j/h} e^{it\left(\eta+\frac{2\pi j}{h}\right)}\hat{u}\left(\eta\right)d\eta \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}} e^{\frac{2\pi i t j}{h}}\int_{-\pi/h-2\pi j/h}^{\pi/h-2\pi j/h} e^{it\eta}\hat{u}\left(\eta\right)d\eta \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}} e^{-\frac{2\pi i t j}{h}}\int_{\frac{(2j-1)\pi}{h}}^{\frac{(2j+1)\pi}{h}} e^{it\eta}\hat{u}\left(\eta\right)d\eta.
\end{aligned}$$

Zusammen mit der Fouriertransformationsgleichung für u folgt

$$\begin{aligned}
u(t) &= \frac{1}{\sqrt{2\pi}}\int_{I\!R}\hat{u}(\omega)e^{+i\omega t}d\omega \\
&= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}}\int_{\frac{(2j-1)\pi}{h}}^{\frac{(2j+1)\pi}{h}} e^{it\eta}\hat{u}\left(\eta\right)d\eta \\
u(t)-S_{h,u}(t) &= \frac{1}{\sqrt{2\pi}}\sum_{j\in\mathbb{Z}}\left(1-e^{-\frac{2\pi i t j}{h}}\right)\int_{\frac{(2j-1)\pi}{h}}^{\frac{(2j+1)\pi}{h}} e^{it\eta}\hat{u}\left(\eta\right)d\eta.
\end{aligned}$$

Theorem 5.6 *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*

$$|u(t)-S_{h,u}(t)| \le \frac{\sqrt{2}}{\sqrt{\pi}}\int_{|\eta|\ge\pi/h}|\hat{u}\left(\eta\right)|d\eta.$$

Der obige Satz gilt auch allgemeiner, weil man die Gleichung umschreiben kann zu

$$\begin{aligned}
u(t)-S_{h,u}(t) &= \frac{1}{\sqrt{2\pi}}\int_{I\!R}\hat{u}\left(\eta\right)e^{it\eta}\sum_{j\in\mathbb{Z}}\left(1-e^{-\frac{2\pi i t j}{h}}\right)\chi_{[\frac{(2j-1)\pi}{h},\frac{(2j+1)\pi}{h}]}(\eta)d\eta \\
&= \frac{1}{\sqrt{2\pi}}\int_{I\!R}\hat{u}\left(\eta\right)e^{it\eta}\underbrace{\sum_{j\in\mathbb{Z}\setminus\{0\}}\left(1-e^{-\frac{2\pi i t j}{h}}\right)\chi_{[-1/2,+1/2]}\left(\frac{h\eta}{2\pi}-j\right)}_{=:K(h\eta/2\pi,2\pi t/h)}d\eta
\end{aligned}$$

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 5.7 *Wenn man die obige Situation auf Funktionen aus dem Sobolevraum $W_2^\tau(I\!R^d)$ einschränkt, bekommt man ein Konvergenzverhalten wie h^τ für $h\to 0$.*

Das folgt mit der oben schon verwenden Technik zur Abschätzung des Abschneidefehlers. □

5.11 Fourier Transforms on IR^d

Revised version 16. April 2007, the revision concerning Parceval's equation and the Poisson summation formula.

5.11.1 Fourier Transforms of Tempered Test Functions

There are two major possibilities to pick a space $\mathcal{S}$ of test functions on IR^d to start with, and we take the **tempered test functions** forming **Schwartz** space $\mathcal{S}$ that are verbally defined as complex-valued functions on IR^d whose partial derivatives exist for all orders and decay faster than any polynomial towards infinity.

Definition 5.4 *For a test function $u \in \mathcal{S}$, the* **Fourier transform** *is*

$$\widehat{u}(\omega) := (2\pi)^{-d/2} \int_{IR^d} u(x) e^{-ix\cdot\omega} dx,$$

where ω varies in IR^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 $u \in L_1(IR^d)$, 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. As a byproduct we shall get that the Gaussian is positive definite on IR^d. 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 IR^d. We simply use substitution in

$$\begin{aligned}
\widehat{u_\gamma}(i\omega) &= (2\pi)^{-d/2} \textstyle\int_{IR^d} e^{-\gamma\|x\|_2^2} e^{x\cdot\omega} dx \\
&= (2\pi)^{-d/2} e^{\|\omega\|_2^2/4\gamma} \textstyle\int_{IR^d} e^{-\|\sqrt{\gamma}x - \omega/2\sqrt{\gamma}\|_2^2} dx \\
&= (2\pi\gamma)^{-d/2} e^{\|\omega\|_2^2/4\gamma} \textstyle\int_{IR^d} e^{-\|y\|_2^2} dy
\end{aligned}$$

and are done up to the evaluation of the dimension-dependent constant

$$\int_{IR^d} e^{-\|y\|_2^2} dy =: c^d$$

which is a d-th power, because the integrand factorizes nicely. We calculate c^2 by using polar coordinates and get

$$\begin{aligned}
c^2 &= \textstyle\int_{IR^2} e^{-\|y\|_2^2} dy \\
&= \textstyle\int_0^{2\pi} \int_0^\infty e^{-r^2} r \, dr \, d\varphi \\
&= 2\pi \textstyle\int_0^\infty e^{-r^2} r \, dr \\
&= -\pi \textstyle\int_0^\infty (-2r) e^{-r^2} \, dr \\
&= \pi.
\end{aligned}$$

This proves the first assertion of

Theorem 5.8 *The Gaussian*

$$u_\gamma(x) = \exp(-\gamma\|x\|_2^2)$$

has Fourier transform

$$\widehat{u_\gamma}(\omega) = (2\gamma)^{-d/2} e^{-\|\omega\|_2^2/4\gamma} \tag{5.1}$$

and is unconditionally positive definite on IR^d.

To understand the second assertion, we add

Definition 5.5 *A real-valued function*

$$\Phi : \Omega \times \Omega \to IR$$

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 (5.1):

$$\begin{aligned} \exp(-\beta\|\omega\|_2^2) &= (4\pi\beta)^{-d/2} \textstyle\int_{IR^d} e^{-\|x\|_2^2/4\beta} e^{-ix\cdot\omega} dx \\ &= (2\pi)^{-d/2} \textstyle\int_{IR^d} (2\beta)^{-d/2} e^{-\|x\|_2^2/4\beta} e^{+ix\cdot\omega} dx. \end{aligned}$$

Then take any set $X = \{x_1, \ldots, x_M\} \subset IR^d$ of M distinct points and any vector $\alpha \in IR^M$ to form

$$\begin{aligned} \alpha^T A_{X,u_\gamma} \alpha &= \sum_{j,k=1}^M \alpha_j \alpha_k \exp(-\gamma\|x_j - x_k\|_2^2) \\ &= \sum_{j,k=1}^M \alpha_j \alpha_k (4\pi\gamma)^{-d/2} \int_{IR^d} e^{-\|x\|_2^2/4\gamma} e^{-ix\cdot(x_j - x_k)} dx \\ &= (4\pi\gamma)^{-d/2} \int_{IR^d} e^{-\|x\|_2^2/4\gamma} \sum_{j,k=1}^M \alpha_j \alpha_k e^{-ix\cdot(x_j - x_k)} dx \\ &= (4\pi\gamma)^{-d/2} \int_{IR^d} e^{-\|x\|_2^2/4\gamma} \left| \sum_{j=1}^M \alpha_j e^{-ix\cdot x_j} \right|^2 dx \ge 0. \end{aligned}$$

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 $x \in IR^d$ 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 $y \in I\!R^d$ 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 5.9 *The Fourier transform is bijective on $\mathcal{S}$, and its inverse is the transform*

$$\check{u}(x) := (2\pi)^{-d/2} \int_{I\!R^d} u(\omega) e^{ix \cdot \omega} d\omega.$$

Proof: The multivariate derivative D^α of $\widehat{u}$ can be taken under the integral sign, because u is in $\mathcal{S}$. Then

$$(D^\alpha \widehat{u})(\omega) = (2\pi)^{-d/2} \int_{I\!R^d} u(x)(-ix)^\alpha e^{-ix \cdot \omega} dx,$$

and we multiply this by ω^β and use integration by parts

$$\begin{aligned}
\omega^\beta (D^\alpha \widehat{u})(\omega) &= (2\pi)^{-d/2} \textstyle\int_{I\!R^d} u(x)(-ix)^\alpha (i)^\beta (-i\omega)^\beta e^{-ix \cdot \omega} dx \\
&= (2\pi)^{-d/2} \textstyle\int_{I\!R^d} u(x)(-ix)^\alpha (i)^\beta \frac{d^\beta}{dx^\beta} e^{-ix \cdot \omega} dx \\
&= (2\pi)^{-d/2} (-1)^{|\alpha|+|\beta|} i^{\alpha+\beta} \textstyle\int_{I\!R^d} e^{-ix \cdot \omega} \frac{d^\beta}{dx^\beta} (u(x) x^\alpha) dx
\end{aligned}$$

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

$$u(x) := (2\pi)^{-d/2} \int_{I\!R^d} \widehat{u}(\omega) e^{ix \cdot \omega} d\omega$$

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:

$$\begin{aligned}
(2\pi)^{-d/2} \textstyle\int_{I\!R^d} \widehat{u}(\omega) e^{ix \cdot \omega} d\omega &= (2\pi)^{-d} \textstyle\int_{I\!R^d} \int_{I\!R^d} u(y) e^{i(x-y) \cdot \omega} dy d\omega \\
&= \lim_{\epsilon \searrow 0} (2\pi)^{-d} \int_{I\!R^d} \int_{I\!R^d} u(y) e^{i(x-y) \cdot \omega - \epsilon \|\omega\|_2^2} dy d\omega \\
&= \lim_{\epsilon \searrow 0} (2\pi)^{-d} \int_{I\!R^d} \left(\int_{I\!R^d} e^{i(x-y) \cdot \omega - \epsilon \|\omega\|_2^2} d\omega \right) u(y) dy \\
&= \lim_{\epsilon \searrow 0} \int_{I\!R^d} \varphi(\epsilon, x-y) u(y) dy
\end{aligned}$$

with

$$\varphi(\epsilon, z) := (2\pi)^{-d} \int_{I\!R^d} e^{iz\cdot\omega - \epsilon\|\omega\|_2^2} d\omega. \tag{5.2}$$

The proof is completed by application of the following result that is useful in many contexts: □

Lemma 5.4 *The family of functions $\varphi(\epsilon, z)$ of (5.2) approximates the point evaluation functional in the sense*

$$u(x) = \lim_{\epsilon \searrow 0} \int_{I\!R^d} \varphi(\epsilon, x - y) u(y) dy \tag{5.3}$$

for all functions u that are in $L_1(I\!R^d)$ 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

$$\varphi(\epsilon, x) = (4\pi\epsilon)^{-d/2} e^{-\|x\|_2^2/4\epsilon} \tag{5.4}$$

and

$$\int_{I\!R^d} \varphi(\epsilon, x) dx = (4\pi\epsilon)^{-d/2} \int_{I\!R^d} e^{-\|x\|_2^2/4\epsilon} dx = 1.$$

To prove (5.3), 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

$$\begin{array}{rcl} |u(x) - \int_{I\!R^d} \varphi(\epsilon, x - y) u(y) dy| & = & |\int_{I\!R^d} \varphi(\epsilon, x - y)(u(x) - u(y)) dy| \\ & \leq & \int_{\|y-x\|_2 \leq \rho} \varphi(\epsilon, x - y)|u(x) - u(y)| dy \\ & & + \int_{\|y-x\| > \rho} \varphi(\epsilon, x - y)|u(x) - u(y)| dy \\ & \leq & \delta/2 + (4\pi\epsilon)^{-d/2} e^{-\rho^2/4\epsilon} 2\|u\|_1 \\ & \leq & \delta \end{array}$$

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. We define the inner product as

$$(f, g)_{L_2(I\!R^d)} := \int_{I\!R^d} f(x) \overline{g(x)} dx \tag{5.5}$$

without factors that sometimes are used.

Fubini's theorem easily proves the identity

$$(v, \widehat{u})_{L_2(I\!R^d)} = (2\pi)^{-d/2} \int_{I\!R^d} v(x) \int_{I\!R^d} \overline{u(y)} e^{+ix\cdot y} dy dx = (\check{v}, u)_{L_2(I\!R^d)}$$

for all test functions $u, v \in \mathcal{S}$. Setting $v = \widehat{w}$ we get Parceval's equation

$$(\widehat{w}, \widehat{u})_{L_2(I\!R^d)} = (w, u)_{L_2(I\!R^d)} \tag{5.6}$$

for the Fourier transform on $\mathcal{S}$, proving that the Fourier transform is isometric on $\mathcal{S}$ as a subspace of $L_2(I\!R^d)$.

5.11.2 Fourier Transform in $L_2(I\!R^d)$

The test functions from $\mathcal{S}$ are dense in $L_2(I\!R^d)$ (see Lemma **??** for details), and thus we have

Theorem 5.10 *The Fourier transform has an L_2-isometric extension from the space $\mathcal{S}$ of tempered test functions to $L_2(I\!R^d)$. The same holds for the inverse Fourier transform, and both extensions are inverses of each other in $L_2(I\!R^d)$. Furthermore, Parceval's equation* (5.6) *holds in $L_2(I\!R^d)$.* □

Note that this result does not allow to use the Fourier transform formula (or its inverse) in the natural pointwise form. For any $f \in L_2(I\!R^d)$ 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 æby

$$\lim_{n\to\infty} \|\widehat{f} - \widehat{v_n}\|_{L_2(I\!R^d)} = 0.$$

This can be done via Friedrich's mollifiers as defined in (**??**), replacing the Gaussian in the representation (5.4) by a compactly supported infinitely differentiable function.

A more useful characterization of $\widehat{f}$ is the variational equation

$$(\widehat{f}, v)_{L_2(I\!R^d)} = (f, \check{v})_{L_2(I\!R^d)}$$

for all test functions $v \in \mathcal{S}$, or, by continuity, all functions $v \in L_2(I\!R^d)$.

5.11.3 Poisson Summation Formula

This comes in several forms:

$$\begin{aligned}
(2\pi)^{-d/2} \sum_{k\in\mathbb{Z}^d} \hat{u}(k) &= \sum_{j\in\mathbb{Z}^d} u(2\pi j) \\
(2\pi)^{-d/2} \sum_{k\in\mathbb{Z}^d} \hat{u}(k) e^{ik^T x} &= \sum_{j\in\mathbb{Z}^d} u(x + 2\pi j) \\
(2\pi)^{-d/2} \sum_{k\in\mathbb{Z}^d} u(k) e^{-ik^T \omega} &= \sum_{j\in\mathbb{Z}^d} \hat{u}(\omega + 2\pi j) \\
(2\pi)^{-d/2} \sum_{k\in\mathbb{Z}^d} u(hk) e^{-ihk^T \omega} &= h^{-d} \sum_{j\in\mathbb{Z}^d} \hat{u}\left(\omega + \frac{2\pi j}{h}\right)
\end{aligned}$$

but we shall have to assure in which sense and under which assumptions it holds. The first clearly is a consequence of the second, if the second holds pointwise. But we shall not discuss this here. The final two are variations of the second, as follows from standard transformations.

Thus we focus on the second one first and see it as an equation in $L_2(I\!R^d)$. Both sides are 2π-periodic, and the left-hand side can be viewed as the Fourier series representation of the right-hand side. Thus we assume that the right-hand side is a pointwise absolutely convergent series which is also convergent in $L_2[-\pi, \pi]^d$. To make the left-hand side meaningful, we assume that u is in $L_1(I\!R^d)$.

If we write the Fourier analysis of a d-variate 2π-periodic function $f(x)$ as

$$f(x) = \sum_{k\in\mathbb{Z}^d} c_k e^{ik^T x}, \quad c_k = (2\pi)^{-d} \int_{[-\pi,\pi]^d} f(x) e^{-ik^T x} dx,$$

we can apply this to the right-hand side f of the second form of the Poisson summation formula. We get the coefficient

$$\begin{aligned} c_k &= (2\pi)^{-d} \int_{[-\pi,\pi]^d} f(x) e^{-ik^T x} dx \\ &= (2\pi)^{-d} \int_{[-\pi,\pi]^d} \sum_{j \in \mathbb{Z}^d} u(x + 2\pi j) e^{-ik^T x} dx \\ &= (2\pi)^{-d} \int_{[-\pi,\pi]^d} \sum_{j \in \mathbb{Z}^d} u(x + 2\pi j) e^{-ik^T (x+2\pi j)} dx \\ &= (2\pi)^{-d} \int_{I\!R^d} u(t) e^{-ik^T t} dt \\ &= (2\pi)^{-d/2} \hat{u}(k) \end{aligned}$$

under our assumptions. Note that the above argument uses only L_2–continuous transformations. This proves the second equation.

The third form can be deduced exactly like the second one, if we also interchange the role of u and $\hat{u}$ in the assumptions. Formally, we can use the second for $\hat{u}$ instead of u and apply

$$\hat{\hat{u}}(k) = \hat{u}^\vee(-k) = u(-k).$$

The final form takes $v(x) := u(hx)$ and applies the third inequality with

$$\hat{v}(\omega) = h^{-d} \hat{u}\left(\frac{\omega}{h}\right)$$

following from

$$\begin{aligned} \hat{v}(\omega) &= (2\pi)^{-d/2} \int v(x) e^{-ix^T \omega} dx \\ &= (2\pi)^{-d/2} \int u(hx) e^{-ihx^T \omega/h} dx \\ &= h^{-d} (2\pi)^{-d/2} \int u(y) e^{-iy^T \omega/h} dx \\ &= h^{-d} \hat{u}\left(\tfrac{\omega}{h}\right). \end{aligned}$$

This yields

$$\begin{aligned} (2\pi)^{-d/2} \sum_{k \in \mathbb{Z}^d} v(k) e^{-ik^T \eta} &= \sum_{j \in \mathbb{Z}^d} \hat{v}(\eta + 2\pi j) \\ (2\pi)^{-d/2} \sum_{k \in \mathbb{Z}^d} u(hk) e^{-ik^T \eta} &= h^{-d} \sum_{j \in \mathbb{Z}^d} \hat{u}\left(\frac{\eta + 2\pi j}{h}\right) \\ (2\pi)^{-d/2} \sum_{k \in \mathbb{Z}^d} u(hk) e^{-ihk^T \omega} &= h^{-d} \sum_{j \in \mathbb{Z}^d} \hat{u}\left(\omega + \frac{2\pi j}{h}\right) \end{aligned}$$

for $\eta =: h\omega$. But note that the above form is badly scaled. It should read

$$h^{d/2} \sum_{k \in \mathbb{Z}^d} u(hk) e^{-ihk^T \omega} = \left(\frac{2\pi}{h}\right)^{d/2} \sum_{j \in \mathbb{Z}^d} \hat{u}\left(\omega + \frac{2\pi j}{h}\right)$$

in order to represent the fact that the left-hand side is a summation over gridpoints with spacing h, while the right-hand side is a summation over a grid with spacing $\frac{2\pi}{h}$.

6 Translationsinvariante Räume

(Folie zur Vorlesung)

Kapitel 6

Translationsinvariante Räume

(Folie zur Vorlesung)

Inhalt dieses Kapitels (Vorschau)

- Def: Translationsinvariante Räume
- Klammerprodukt
- Projektor
- Stationäre Approximationsschemata
- Strang-Fix-Bedingungen

6.1 Translationsinvariante Räume

(Folie zur Vorlesung)

Translationsinvariante Räume

- Siehe Sondertext auf der website
- Definition
- Klammerprodukt
- Stabiler Fall
- Projektoren
- Fehlerabschätzungen

(Folie zur Vorlesung)

Strang–Fix Theorie

- Strang–Fix Bedingungen
- Fehlerabschätzungen
- Konvergenzsätze

Dieses Manuskript ist 2006/07 ein Zusatztext zur Vorlesung “Approximationsverfahren”. Es setzt Fouriertransformation, B–Splines und das Shannon–sampling voraus.

6.2 Grundlagen

Wir verallgemeinern hier, was wir über das Shannon-Sampling gelernt haben. Statt einer kardinalen Funktion wie

$$\begin{aligned}\varphi(x) &:= \operatorname{sinc}(x)\\ \varphi(x) &:= \begin{cases} 1-|x| & |x|\le 1\\ 0 & \text{sonst}\end{cases}\end{aligned}$$

betrachten wir allgemeine "Generatoren" $\varphi \in L_2(IR)$ und den in $L_2(IR)$ genommenen Abschluß des spans ihrer Translate:

$$S_\varphi := \overline{span\{\varphi(\cdot - k) \ : \ k \in \mathbb{Z}\}} \tag{6.1}$$

Definition 6.1 *Der Raum S_φ aus* (6.1) *ist der von φ erzeugte* **principal shift–invariant space** *(PSI).*

Der Raum ist wohldefiniert, aber wir würden gerne wissen, welche Bedingungen an einen infiniten Koeffizientenvektor $c = \{c_k\}_{k\in\mathbb{Z}}$ man stellen muß, um sicherzustellen, daß die Funktion

$$\varphi_c(x) := (c * \varphi)(x) := \sum_{k\in\mathbb{Z}} c_k \varphi(x-k)$$

punktweise auswertbar ist bzw. noch in $L_2(IR)$ liegt. In der obigen Gleichung wurde die **diskrete Faltung** durch $*$ definiert.

Das ist unter verschiedenen Voraussetzungen machbar, die wir hier teilweise aufzählen, die sich aber nicht gegenseitig ausschließen.

Situation 1: Für endliche Koeffizientenvektoren liegt die Summe immer in $L_2(IR)$. 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

$$x - K \le k \le x + K$$

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 L_2–Norm von φ_c an.

$$\begin{aligned}\|\varphi_c\|_2^2 &= \textstyle\int_{IR} \varphi_c^2(x)dx\\ &= \int_{IR} \left(\sum_{k\in\mathbb{Z}} c_k\varphi(x-k)\right)^2 dx\\ &= \sum_{j\in\mathbb{Z}} \int_j^{j+1} \left(\sum_{k\in\mathbb{Z}} c_k\varphi(x-k)\right)^2 dx\\ &= \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{k\in\mathbb{Z}} c_k\varphi(x-j-k)\right)^2 dx\\ &= \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{m\in\mathbb{Z}} c_{m-j}\varphi(x-m)\right)^2 dx.\end{aligned}$$

Die inneren Indices m können mindestens auf $-K \le m \le K+1$ eingeschränkt werden, weil das Integral über $\varphi(x-m)$ verschwindet, sofern $-m \ge K$ oder $-m+1 \le -K$ 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:

$$\begin{aligned}
\|\varphi_c\|_2^2 &\le \sum_{j\in\mathbb{Z}} \int_0^1 \left(\sum_{m=-K}^{K+1} c_{m-j}^2 \right) \left(\sum_{n=-K}^{K+1} \varphi(x-n)^2 \right) dx \\
&= \sum_{j\in\mathbb{Z}} \left(\sum_{m=-K}^{K+1} c_{m-j}^2 \right) \int_0^1 \sum_{n=-K}^{K+1} \varphi(x-n)^2 dx \\
&\le (2K+2) \left(\sum_{j\in\mathbb{Z}} c_j^2 \right) \int_0^1 \sum_{n=-K}^{K+1} \varphi(x-n)^2 dx \\
&\le (2K+2) \|c\|_{\ell_2}^2 \|\varphi\|_2^2
\end{aligned}$$

weil beim Summieren jedes der c_j^2 maximal $(2K+2)$–mal vorkommen kann.

Theorem 6.1 *Im Falle $\varphi \in L_2(IR)$ mit kompaktem Träger und $c \in \ell_2$ gilt $\varphi_c := c*\varphi \in L_2(IR)$.*

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. 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 6.2 *Gilt $c \in \ell_2$ und ist σ_c eine beschränkte 2π–periodische Funktion, so gilt $\varphi_c \in L_2(IR)$.*

Aber man kann auch folgendermaßen weiterarbeiten:

$$\begin{aligned}
\|\varphi_c\|_2^2 &= \int_{IR} |\hat{\varphi}(\omega)|^2 |\sigma_c(\omega)|^2 d\omega \\
&= \sum_{j\in\mathbb{Z}} \int_{-\pi}^{\pi} |\hat{\varphi}(\omega+2\pi j)|^2 |\sigma_c(\omega+2\pi j)|^2 d\omega \\
&= \int_{-\pi}^{\pi} |\sigma_c(\omega)|^2 \sum_{j\in\mathbb{Z}} |\hat{\varphi}(\omega+2\pi j)|^2 d\omega \\
&=: \int_{-\pi}^{\pi} |\sigma_c(\omega)|^2 [\varphi,\varphi](\omega) d\omega
\end{aligned} \tag{6.2}$$

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 6.3 *Gilt $c \in \ell_2$ und ist das Klammerprodukt $[\varphi,\varphi](\omega)$ punktweise existent, meßbar und gleichmäßig beschränkt, so gilt $\varphi_c \in L_2(IR)$.*

Es sieht zwar nach Spielerei aus, aber wir wollen mal die Fourierkoeffizienten von $[\varphi,\psi]$ ausrechnen:

$$\begin{aligned}
& \int_{-\pi}^{\pi} [\varphi,\psi](\omega)e^{-ik\omega}d\omega \\
= & \int_{-\pi}^{\pi} \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\overline{\hat{\psi}(\omega+2\pi j)}e^{-ik(\omega+2\pi j)}d\omega \\
= & \int_{IR} \hat{\varphi}(\omega)\overline{\hat{\psi}(\omega)}e^{-ik\omega}d\omega \\
= & \int_{IR} \hat{\varphi}(\omega)\overline{\hat{\psi}(\omega)e^{ik\omega}}d\omega \\
= & \int_{IR} \varphi(x)\psi(x-k)dx \\
= & \int_{IR} \varphi(x+k)\psi(x)dx.
\end{aligned}$$

Rückwärts gerechnet folgt daraus, daß alle Fourierkoeffizienten des Klammerprodukts $[\varphi,\psi]$ immer berechenbar sind, wenn ψ und φ in $L_2(IR)$ liegen. Wir machen neben

$$(\phi,\psi)_{L_2} = \int_{-\pi}^{\pi} [\varphi,\psi](\omega)d\omega$$

ein paar einfache Beobachtungen:

Theorem 6.4 *Die Translate einer Funktion $\varphi \in L_2(IR^d)$ 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 6.5 *Haben φ und ψ kompakten Träger, so ist das Klammerprodukt ein trigonometrisches Polynom.*

Theorem 6.6 *Sind f und φ beide in $L_2(IR)$ 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:

$$\begin{aligned} 1/2\pi &= [\varphi_c, \varphi_c](\omega) \\ &= \sum_{j\in\mathbb{Z}} \hat{\varphi}_c(\omega+2\pi j)\overline{\hat{\varphi}_c(\omega+2\pi j)} \\ &= \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\sigma_c(\omega+2\pi j)\overline{\hat{\varphi}(\omega+2\pi j)\sigma_c(\omega+2\pi j)} \\ &= \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\sigma_c(\omega)\overline{\hat{\varphi}(\omega+2\pi j)\sigma_c(\omega)} \\ &= |\sigma_c(\omega)|^2 \sum_{j\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi j)\overline{\hat{\varphi}(\omega+2\pi j)} \\ &= |\sigma_c(\omega)|^2 [\varphi,\varphi](\omega). \end{aligned} \tag{6.3}$$

Theorem 6.7 *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:

$$0 < A^2 \le [\varphi,\varphi](\omega) \le B^2. \tag{6.4}$$

Diese Situation wird manchmal auch "stabil" genannt. Aus (6.2) 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 6.8 *Es sei $\varphi \in L_2(I\!R)$ 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*

$$\begin{aligned} \{f \in L_2(I\!R) \;&:\; \hat{f} = \tau\cdot\hat{\varphi},\ \tau \in L_{2,2\pi}\} =: S_1 \\ \{f \in L_2(I\!R) \;&:\; f = \varphi_c,\ c \in \ell_2\} =: S_2. \end{aligned}$$

Beweis: Beide Räume liegen in S_φ, wenn man die Definitionen zunächst auf endliche Folgen c und trigonometrische Polynome τ einschränkt. Mit (6.2) kann man dann aber auch im Falle von S_2 wie folgt abschätzen:

$$A^2\|c\|_{\ell_2}^2 \;\le\; \|f\|_{L_2(I\!R)}^2 \;=\; \|\varphi_c\|_{L_2(I\!R)}^2 \;\le\; B^2\|c\|_{\ell_2}^2.$$

Damit kann man zum Abschluß übergehen. Die Situation von S_1 ist analog wegen $\tau = \sigma_c$ für $f = \varphi_c$ und $\|c\|_{\ell_2} = \|\tau\|_{L_{2,2\pi}}$. □

6.3 Projektion

Wir wollen jetzt die L_2–Projektion von $L_2(I\!R)$ auf S_φ ausrechnen, wie bei der Shannon-Situation. Der Projektor, nennen wir ihn P_φ, muss existieren, und im Falle eines orthogonalen Generators is 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 6.8 an, dass er über Koeffizienten $c_f \in \ell_2$ mit $P_\varphi f = c_f * \varphi$ parametrisierbar ist.

Es folgt

$$\begin{aligned}
0 &= (f - I\!P_\varphi f, \varphi(\cdot - k))_{L_2(I\!R)} \\
(f, \varphi(\cdot - k))_{L_2(I\!R)} &= (c_f * \varphi, \varphi(\cdot - k))_{L_2(I\!R)} \\
\int_{-\pi}^{\pi} [f,\varphi](\omega) e^{-ik\omega} d\omega &= \int_{-\pi}^{\pi} [c_f * \varphi, \varphi](\omega) e^{-ik\omega} d\omega \\
&= \int_{-\pi}^{\pi} \sum_{j \in \mathbb{Z}\!\!\mathbb{Z}} \sigma_c(\omega) \hat{\varphi}(\omega + 2\pi j) \overline{\hat{\varphi}(\omega + 2\pi j)} e^{-ik\omega} d\omega \\
&= \int_{-\pi}^{\pi} \sigma_c(\omega) [\varphi,\varphi](\omega) e^{-ik\omega} d\omega
\end{aligned}$$

d.h.

$$[f,\varphi](\omega) = \sigma_c(\omega)[\varphi,\varphi](\omega)$$

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:

$$P_\varphi f = \sum_{k \in \mathbb{Z}\!\!\mathbb{Z}} c_k \varphi(\cdot - k),\ c_k = \int_{-\pi}^{\pi} \frac{[f,\varphi](\omega)}{[\varphi,\varphi](\omega)} e^{-ik\omega} d\omega$$

oder im Fourierraum

$$(P_\varphi f)\hat{}(\omega) = \frac{[f,\varphi](\omega)}{[\varphi,\varphi](\omega)} \hat{\varphi}(\omega).$$

Man braucht diese Gleichung später bei der wavelet–Konstruktion.

6.4 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

$$P_{\varphi,h}(f)(x) := P_\varphi(f(\cdot h))(x/h)$$

nehmen. Diese Art der Skalierung wird in der Literatur auch "**stationär**" genannt. Definiert man den Projektor so, ergibt sich die Orthogonalität

$$\begin{aligned}
(f - P_{\varphi,h}(f), \tfrac{1}{h}\varphi((\cdot - kh)/h))_{L_2(I\!R)} &= \frac{1}{h} \int_{I\!R} (f(x) - P_{\varphi,h}(f)(x)) \overline{\varphi(x/h - k)} dx \\
&= \frac{1}{h} \int_{I\!R} (f(x) - P_\varphi(f(\cdot h))(x/h)) \overline{\varphi(x/h - k)} dx \\
&= \int_{I\!R} (f(hy) - P_\varphi(f(\cdot h))(y)) \overline{\varphi(y - k)} dy \\
&= 0.
\end{aligned}$$

Genauso rechnen wir den Fehler aus, und zwar

$$\begin{aligned}\|f - P_{\varphi,h}(f)\|^2_{L_2(I\!R)} &= \int_{I\!R} |f(x) - P_\varphi(f(\cdot h))(x/h)|^2 dx \\ &= h\int_{I\!R} |f(hy) - P_\varphi(f(\cdot h))(y)|^2 dy \\ &= h\|f_h - P_\varphi(f_h)\|^2_{L_2(I\!R)}\end{aligned}$$

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 6.2 *Das Projektionsverfahren hat bezüglich eines Unterraums W von $L_2(I\!R)$ die Approximationsordnung m, wenn für alle $f \in W$ eine Abschätzung der Form*

$$\|f - P_{\varphi,h}(f)\|_{L_2(I\!R)} \le C_f h^m$$

mit einer von h unabhängigen Konstanten C_f gilt.

Definition 6.3 *Für beliebige positive κ kann man den* **Sobolewraum**

$$W_2^\kappa(I\!R) := \{f \in L_2(I\!R) \ : \ \int_{I\!R} |\hat{f}(\omega)|^2(1+|\omega|^2)^k d\omega < \infty\}$$

mit dem Skalarprodukt

$$(f,g)_{W_2^\kappa(I\!R)} := \int_{I\!R} \hat{f}(\omega)\overline{\hat{g}(\omega)}(1+|\omega|^2)^k d\omega$$

definieren.

Der obige Raum besteht aus allen Funktionen, die durch Fouriertransformation definierte verallgemeinerte Ableitungen bis zur Ordnung κ haben, die noch in $L_2(I\!R)$ liegen. Wir haben solche Räume schon bei den Fourierreihen gesehen, dort aber im periodischen Fall. 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 $W_2^m(I\!R)$ die Ordnung m.

Theorem 6.9 *Gilt*

$$\|f - P_\varphi f\|_2 \le C|f|_m \tag{6.5}$$

für alle $f \in W_2^m(I\!R)$ mit der Seminorm

$$\|f\|_m^2 := \int_{I\!R} |\hat{f}(\omega)|^2 |\omega|^{2m} d\omega,$$

so hat der Projektor $P_{\varphi,h}$ die Approximationsordnung m im Raum $W_2^m(I\!R)$.

Der **Beweis** folgt aus einem einfachen Skalierungsargument:

$$\begin{aligned}
\|f - P_{\varphi,h} f\|_2^2 &= h\|f_h - P_\varphi f_h\|_2^2 \\
&\le Ch|f_h|_m^2 \\
&= Ch\|\hat{f}_h(\omega)|\omega|^m\|_2^2 \\
&= Ch\left\|\frac{1}{h}\hat{f}(\frac{\omega}{h})|\omega|^m\right\|_2^2 \\
&= Ch\frac{1}{h^2}h^{2m}\int_{IR)}|\hat{f}(\frac{\omega}{h})^2|\frac{\omega}{h}|^{2m}d\omega \\
&\le Ch^{2m}\left\||\hat{f}(\omega)|^2|\omega|^m\right\|_2^2 \\
&= Ch^{2m}|f|_m^2 \\
&\le Ch^{2m}\|f\|_{W_2^m(IR)}^2.
\end{aligned}$$

□

Wären Polynome in $L_2(IR)$, so könnte man aus (6.5) 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

$$\begin{aligned}
P_s &:= P_{sinc} \\
\hat{P}_s &= \hat{f} \cdot \hat{\chi}_{[-\pi,\pi]} \\
\|f - P_s f\|_2^2 &= \|\hat{f} - \hat{P}_s f\|_2^2 \\
&= \int_{|\omega|\ge\pi}|\hat{f}(\omega)|^2 d\omega \\
&\le \int_{|\omega|\ge\pi}|\hat{f}(\omega)|^2\frac{|\omega|^{2m}}{\pi^{2m}}d\omega \\
&\le \frac{1}{\pi^{2m}}\int_{IR}|\hat{f}(\omega)|^2|\omega|^{2m}d\omega.
\end{aligned}$$

□

Wir benutzen das, um auf den Fehler anderer Projektoren zu schließen.

Theorem 6.10 *Gilt*

$$\|f - P_\varphi f\|_2 \le C_{\varphi,s}|f|_m$$

für alle bandbreitenbeschränkten $f \in P_s(L_2(IR))$, so hat $P_{\varphi,h}$ die Approximationsordnung m in $W_2^m(IR)$.

Beweis: Wir schätzen folgendermaßen ab:

$$\begin{aligned}
\|f - P_\varphi f\|_2 &\le \|f - P_s f\|_2 + \|P_s f - P_\varphi P_s f\|_2 + \|P_\varphi P_s f - P_\varphi f\|_2 \\
&\le C_s|f|_m + C_{\varphi,s}|P_s f|_m + \|P_\varphi\|\|P_s f - f\|_2 \\
&\le C_s|f|_m + C_{\varphi,s}|f|_m + C_s|f|_m
\end{aligned}$$

weil die Projektoren die Norm 1 in L_2 haben und $|P_s f|_m \le |f|_m$ gilt. □

6.5 Fehlerabschätzung

Unter den Voraussetzungen des Satzes 6.8 betrachten wir den Fehler der Projektion. Wegen der üblichen Orthogonalität hat man

$$\begin{aligned}
\|f - P_\varphi f\|^2_{L_2(I\!R)} &= \|f\|^2_{L_2(I\!R)} - \|P_\varphi f\|^2_{L_2(I\!R)} \\
&= \|\hat{f}\|^2_{L_2(I\!R)} - \|(P_\varphi f)^\wedge\|^2_{L_2(I\!R)} \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{I\!R} \frac{|[f,\varphi]|^2(\omega)}{[\varphi,\varphi]^2(\omega)} |\hat{\varphi}(\omega)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \sum_{k\in I\!R} \int_{-\pi}^{\pi} \frac{|[f,\varphi]|^2(\omega)}{[\varphi,\varphi]^2(\omega)} |\hat{\varphi}(\omega+2k\pi)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|[f,\varphi]|^2(\omega)}{[\varphi,\varphi](\omega)} d\omega.
\end{aligned}$$

Jetzt machen wir wie beim Shannon sampling die Annahme

$$\hat{f}(\omega) = 0 \text{ für alle } |\omega| > \pi. \tag{6.6}$$

Dann folgt für alle $\omega \in [-\pi, \pi]$ die Gleichung

$$\begin{aligned}
[f,\varphi](\omega) &= \sum_{k\in I\!R} \hat{f}(\omega+2\pi k)\overline{\hat{\varphi}(\omega+2\pi k)} \\
&= \hat{f}(\omega)\overline{\hat{\varphi}(\omega)}.
\end{aligned}$$

Das liefert

$$\begin{aligned}
\|f - P_\varphi f\|^2_{L_2(I\!R)} &= \int_{-\pi}^{\pi} |\hat{f}(\omega)|^2 \underbrace{\left(1 - \frac{|\hat{\varphi}(\omega)|^2}{[\varphi,\varphi](\omega)}\right)}_{=:L_\varphi(\omega)} d\omega \\
&= \int_{-\pi}^{\pi} |\hat{f}(\omega)|^2 L_\varphi(\omega) d\omega.
\end{aligned} \tag{6.7}$$

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 6.9 mit (6.7) vergleichen, liegt nahe, dass wir die verschärfte Voraussetzung

$$0 \le L_\varphi(\omega) \le C_L|\omega|^{2m},\ |\omega| \le \pi \tag{6.8}$$

machen sollten. Dann wird aus (6.7) genau die Voraussetzung von Theorem 6.10 und wir bekommen unser Hauptergebnis

Theorem 6.11 *Gilt* (6.8) *mit einem punktweise wohldefinierten L_φ, so hat die duch φ definierte skalierte Projektion $P_{\varphi,h}$ im Sobolevraum $W_2^m(I\!R)$ die Approximationsordnung m.*

6.6 Strang–Fix–Bedingungen

Dies sind Bedingungen an φ, um (6.8) zu erreichen. Wir setzen wie bisher Stabilität von φ und zusätzlich Wohldefiniertheit von L_φ voraus, und dann ist es für (6.8) 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 6.12 *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 $W_2^m(IR)$ die Approximationsordnung m. □

6.7 B–Spline–Generatoren

Wir definieren

$$\varphi_1(x) := \chi_{[0,1]}(x) = \begin{cases} 1 & 0 \leq x \leq 1 \\ 0 & |x| > 1 \end{cases}$$

als die Haarsche Skalierungsfunktion, aber wir falten sie rekursiv zu

$$\varphi_n(x) := (\varphi_{n-1} * \varphi_1)(x) := \int_0^1 \varphi_{n-1}(x-t)dt,\ x \in IR,\ n > 1.$$

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 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 müssen nur $\hat{\varphi}^1$ ausrechnen:

$$\begin{aligned}
\hat{\varphi}^1(\omega) &= \frac{1}{\sqrt{2\pi}} \int_0^1 e^{-i\omega x} dx \\
&= \frac{1}{\sqrt{2\pi}} \frac{1}{-i\omega} (e^{-i\omega} - 1) \\
&= \frac{1}{\sqrt{2\pi}} \frac{2i}{i\omega} \frac{e^{i\omega/2} - e^{-i\omega/2}}{2i} e^{-i\omega/2} \\
&= \frac{1}{\sqrt{2\pi}} \frac{\sin(\omega/2)}{\frac{\omega}{2}} e^{-i\omega/2} \\
&= \frac{1}{\sqrt{2\pi}} \mathrm{sinc}\left(\frac{\omega}{2\pi}\right) e^{-i\omega/2}.
\end{aligned}$$

Also gilt

$$\hat{\varphi}_n(\omega) = (2\pi)^{-n/2} \mathrm{sinc}^n\left(\frac{\omega}{2\pi}\right) e^{-in\omega/2}.$$

Wir sollten nachprüfen, ob die Translate stabil sind. Dazu müssen wir das Klammerprodukt

$$\begin{aligned}
[\varphi_n,\varphi_n](\omega) &= \sum_{k\in\mathbb{Z}} |\hat{\varphi}_n(\omega+2\pi k)|^2 \\
&= (2\pi)^{-n} \sum_{k\in\mathbb{Z}} \operatorname{sinc}^{2n}\left(\frac{\omega+2\pi k}{2\pi}\right) \\
&= (2\pi)^{-n} \sum_{k\in\mathbb{Z}} \frac{\sin^{2n}\left(\frac{\omega}{2}+k\pi\right)}{\left(\frac{\omega}{2}+k\pi\right)^{2n}} \\
&= (2\pi)^{-n} \sin^{2n}\left(\frac{\omega}{2}\right) \sum_{k\in\mathbb{Z}} \frac{1}{\left(\frac{\omega}{2}+k\pi\right)^{2n}} \\
&= (2\pi)^{-n} \frac{\sin^{2n}\left(\frac{\omega}{2}\right)}{\left(\frac{\omega}{2}\right)^{2n}} \sum_{k\in\mathbb{Z}} \frac{\left(\frac{\omega}{2}\right)^{2n}}{\left(\frac{\omega}{2}+k\pi\right)^{2n}}
\end{aligned}$$

untersuchen. Weil $|\operatorname{sinc}(x)| \le 1$ global gilt, folgt

$$[\varphi_n,\varphi_n](\omega) \le (2\pi)^{-n},$$

und daraus folgt sofort, daß die Translate von φ_n nur im Falle $n = 1$ orthonormal sind. Weil man ω auf $|\omega| \le \pi$ einschränken kann, folgt auch

$$[\varphi_n,\varphi_n](\omega) \ge (2\pi)^{-n} \min_{|\omega|\le\pi} \frac{\sin^{2n}\left(\frac{\omega}{2}\right)}{\left(\frac{\omega}{2}\right)^{2n}} > 0$$

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 6.13 *Die Approximation in durch B–Splines φ_n erzeugten translationsinvarianten Räumen ist stabil und hat Approximationsordnung m in den Sobolevräumen $W_2^m(IR)$ für alle $m \le n$.* □

$$\begin{aligned}
\|f-Pf\|^2_{L_2(IR)} &= \|\hat{f}-(Pf)^\wedge\|^2_{L_2(IR)} \\
&= \|\hat{f}\|^2_{L_2(IR)} - \|(Pf)^\wedge\|^2_{L_2(IR)} \\
&= \int_{IR} |\hat{f}(\omega)|^2 d\omega - \int_{IR} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi]^2(\omega)} |\hat{\varphi}(\omega)|^2 d\omega \\
&= \int_{IR} |\hat{f}(\omega)|^2 d\omega - \sum_{m\in\mathbb{Z}} \int_{-\pi}^{\pi} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi]^2(\omega)} |\hat{\varphi}(\omega+2\pi m)|^2 d\omega \\
&= \int_{IR} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi](\omega)} d\omega \\
&= \int_{IR} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|\sum_{k\in\mathbb{Z}} \hat{f}(\omega+2\pi k)\overline{\hat{\varphi}(\omega+2\pi k)}|^2}{[\varphi,\varphi](\omega)} d\omega
\end{aligned}$$

$$
\begin{aligned}
\|f-Pf\|^2_{L_2(I\!R)} &= \|\hat{f}-(Pf)^\wedge\|^2_{L_2(I\!R)} \\
&= \int_{I\!R} |\hat{f}(\omega)-(Pf)^\wedge(\omega)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)-\frac{[f,\varphi](\omega)}{[\varphi,\varphi](\omega)}\hat{\varphi}(\omega)|^2 d\omega \\
&= \sum_{k\in\mathbb{Z}} \int_{-\pi}^{\pi} |\hat{f}(\omega+2\pi k)-\frac{[f,\varphi](\omega)}{[\varphi,\varphi](\omega)}\hat{\varphi}(\omega+2\pi k)|^2 d\omega \\
&= \sum_{k\in\mathbb{Z}} \int_{-\pi}^{\pi} |\hat{f}(\omega+2\pi k)-\frac{\hat{\varphi}(\omega+2\pi k)}{[\varphi,\varphi](\omega)}\sum_{m\in\mathbb{Z}}\hat{f}(\omega+2\pi m)\overline{\hat{\varphi}(\omega+2\pi m)}|^2 d\omega
\end{aligned}
$$

$$
\begin{aligned}
\|f-Pf\|^2_{L_2(I\!R)} &= \|\hat{f}-(Pf)^\wedge\|^2_{L_2(I\!R)} \\
&= \|\hat{f}\|^2_{L_2(I\!R)} - \|(Pf)^\wedge\|^2_{L_2(I\!R)} \\
&= \int_{I\!R} \left(|\hat{f}(\omega)|^2 - |(Pf)^\wedge(\omega)|^2\right) d\omega \\
&= \int_{I\!R} \left(|\hat{f}(\omega)|^2 - |\frac{[f,\varphi](\omega)}{[\varphi,\varphi](\omega)}\hat{\varphi}(\omega)|^2\right) d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{I\!R} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi]^2(\omega)}|\hat{\varphi}(\omega)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \sum_{m\in\mathbb{Z}}\int_{-\pi}^{\pi} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi]^2(\omega)}|\hat{\varphi}(\omega+2\pi m)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi](\omega)} d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|\sum_{k\in\mathbb{Z}}\hat{f}(\omega+2\pi k)\overline{\hat{\varphi}(\omega+2\pi k)}|^2}{[\varphi,\varphi](\omega)} d\omega
\end{aligned}
$$

$$
\begin{aligned}
\|f-Pf\|^2_{L_2(I\!R)} &= \|\hat{f}-(Pf)^\wedge\|^2_{L_2(I\!R)} \\
&= \|\hat{f}\|^2_{L_2(I\!R)} - \|(Pf)^\wedge\|^2_{L_2(I\!R)} \\
&= \int_{I\!R} \left(|\hat{f}(\omega)|^2 - |(Pf)^\wedge(\omega)|^2\right) d\omega \\
&= \int_{I\!R} \left(|\hat{f}(\omega)|^2 - |\frac{[f,\varphi](\omega)}{[\varphi,\varphi](\omega)}\hat{\varphi}(\omega)|^2\right) d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{I\!R} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi]^2(\omega)}|\hat{\varphi}(\omega)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \sum_{m\in\mathbb{Z}}\int_{-\pi}^{\pi} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi]^2(\omega)}|\hat{\varphi}(\omega+2\pi m)|^2 d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|[f,\varphi](\omega)|^2}{[\varphi,\varphi](\omega)} d\omega \\
&= \int_{I\!R} |\hat{f}(\omega)|^2 d\omega - \int_{-\pi}^{\pi} \frac{|\sum_{k\in\mathbb{Z}}\hat{f}(\omega+2\pi k)\overline{\hat{\varphi}(\omega+2\pi k)}|^2}{[\varphi,\varphi](\omega)} d\omega
\end{aligned}
$$

Theorem 6.14 *Es seien φ und ψ beide in $L_2(I\!R)$, ebenso sei das Klammerprodukt $[\varphi,\psi]$ in $L_2[-\pi,\pi]$. Ferner soll ψ senkrecht stehen auf allen φ_c mit $c \in \ell_2$ und $\varphi_c \in L_2$. Dann folgt $\psi = 0$.*

Wir setzen an und rechnen folgendes aus:

$$\begin{aligned}
(\varphi_c, \psi)_{L_2} &= \int_{-\pi}^{\pi} [\varphi_c, \psi](\omega) d\omega \\
&= \int_{-\pi}^{\pi} \sum_{j \in \mathbb{Z}} \hat{\varphi}_c(\omega + 2\pi j) \overline{\hat{\psi}(\omega + 2\pi j)} d\omega \\
&= \int_{-\pi}^{\pi} \sigma_c(\omega) \sum_{j \in \mathbb{Z}} \hat{\varphi}(\omega + 2\pi j) \overline{\hat{\psi}(\omega + 2\pi j)} d\omega \\
&= \int_{-\pi}^{\pi} \sigma_c(\omega) [\varphi, \psi](\omega) d\omega \\
&= \int_{-\pi}^{\pi} \sum_{k \in \mathbb{Z}} c_k e^{-ik\omega} [\varphi, \psi](\omega) d\omega \\
&= \sum_{k \in \mathbb{Z}} c_k \int_{-\pi}^{\pi} e^{-ik\omega} [\varphi, \psi](\omega) d\omega
\end{aligned}$$

und deshalb folgt aus den Voraussetzungen des Satzes daß $[\varphi, \psi]$ verschwindet, weil alle Fourierkoeffizienten verschwinden.

Wir nehmen nun Funktionen aus Sobolewräumen

$$W_2^k(IR) := \{f \in L_2(IR) \ : \ \int_{IR} |\hat{f}(\omega)|^2 (1 + |\omega|^2)^k d\omega < \infty\}$$

mit der Norm

$$\|f\|^2_{W_2^k(IR)} := |\hat{f}(\omega)|^2 (1 + |\omega|^2)^k d\omega$$

her und stellen fest, daß diese Funktionen im Fouriersinne verallgemeinerte Ableitungen bis zur Ordnung k in $L_2(IR)$ haben.

7 Wavelets

(Folie zur Vorlesung)

Kapitel 7

Wavelets

(Folie zur Vorlesung)

Inhalt dieses Kapitels (Vorschau)

- Grundlagen
- Haarsches wavelet
- Algorithmen
- Wavelet–Theorie
- Konvergenzsätze

7.1 Grundlagen

(Folie zur Vorlesung)

Grundlagen

- Dieses Kapitel wird vertretungsweise von Christian Rieger vorgetragen.
- *Danke!!!*
- *Literatur: 7.1 bis 7.3 des Skripts von Tomas Sauer (siehe website der Vorlesung)*
- *Multiresolutionsanalyse*
- *Skalierungsfunktion, Verfeinerungsgleichung*
- *Orthonormale Translate von Skalierungsfunktionen*
- *Orthogonale wavelets*

7.2 Haar wavelet

(Folie zur Vorlesung)

Haarsches Wavelet

- Spezialfall des Haarschen Wavelets
- Wavelet–Transformationen dazu
- Siehe Übungsblatt 11
- Literatur: Sondertext auf der website

7.3 Algorithmen

(Folie zur Vorlesung)

Algorithmen

- Verfeinerungsgleichung
- Wavelet–Darstellung
- Modifikationen im orthonormalen Fall
- Schnelle wavelet–Transformation
- Demonstrationen bei Bildverarbeitung
- Algorithmen auf Masken

7.4 Wavelet–Theorie

(Folie zur Vorlesung)

Spezielle wavelets

- Siehe Sondertext auf der website
- Existenz von wavelets
- Orthogonale wavelets
- Spezialfälle: Spline wavelets
- Daubechies wavelets

(Folie zur Vorlesung)

Fehlerabschätzungen

- Projektionsoperatoren auf wavelets
- Strang-Fix-Bedingungen

Der folgende, sehr elementare Stoff ist aus einer Vorversion für das Buch Numerische Mathematik (Schaback/Wendland, bei Springer) entnommen und in eine Spezialversion für die Vorlesung über Approximationsverfahren, WS 2006/07 konvertiert worden. Eine verschärfte Version der wavelet–Theorie folgt weiter unten.

Wichtige Änderungen hier gegenüber der Buchversion:
p_k als Koeff. der Verfeinerungsgleichung
Korrektur der Koeff. des wavelets $(-1)^k p_{1-k}$.

7.5 Haarsche Skalierungsfunktion

Wir wollen den Gedanken der effizienten Speicherung von zwei Zahlen benutzen, um das Haar-Wavelet 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:

$$s = \frac{a+b}{2}, \qquad d = b - a.$$

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 weglassen und erreicht so eine Speicherplatzersparnis auf Kosten eines zu analysierenden Fehlers.

Die Rekonstruktion der Orginalwerte ist gegeben durch

$$a = s - \frac{d}{2}, \qquad b = s + \frac{d}{2}.$$

Nehmen wir nun an, dass wir nicht nur zwei Zahlen sondern ein Signal $f^{(n)}$ bestehend aus 2^n Werten gegeben haben, d.h. $f^{(n)} = \{f_k^{(n)} : 0 \leq k < 2^n\}$. 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 $2^{-n}k$ vorstellen, d.h. $f_k^{(n)} = f(k2^{-n})$, $0 \leq k < 2^n$. Wenn wir nun die Durchschnitts- und Differenzbildung auf jedes der Paare $a = f_{2k}^{(n)}$ und $b = f_{2k+1}^{(n)}$, $0 \leq k < 2^{n-1}$, anwenden erhalten wir zwei neue Vektoren $f^{(n-1)}$ und $r^{(n-1)}$ vermöge

$$f_k^{(n-1)} = \frac{f_{2k}^{(n)} + f_{2k+1}^{(n)}}{2}, \qquad r_k^{(n-1)} = f_{2k+1}^{(n)} - f_{2k}^{(n)}.$$

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.

Wendet man den eben beschrieben 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. Man kann das Ganze so interpretieren, dass vom Übergang $f^{(j)} \to f^{(j-1)}$ geglättet wird, und die verlorengegangenen Details in $r^{(j-1)}$ gesammelt werden. Abbildung 30 zeigt die Zerlegung schematisch. Die Bezeichnungen H und G in der Abbildung stehen für

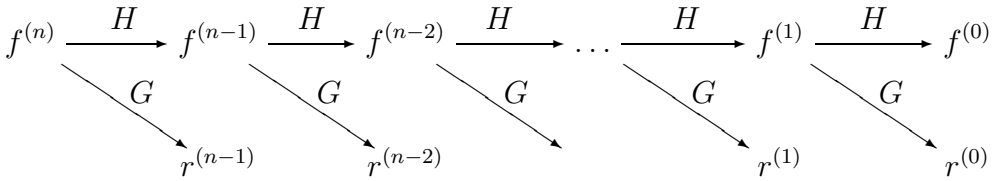

Abbildung 30: Schematische Darstellung der Wavelet Zerlegung.

den jeweiligen Übergang. Natürlich lässt sich auch die Rekonstruktion auf diese Weise rekursiv realisieren, wie in Abbildung 31 dargestellt, wobei wir bewusst dieselben Buchstaben für die Übergangsbezeichnung benutzt haben, dies später noch genauer erklärt. Der Aufwand, um die

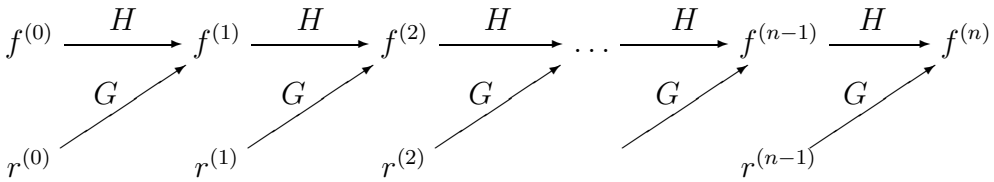

Abbildung 31: Schematische Darstellung der Wavelet Rekonstruktion.

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.

An dieser Stelle fügen wir die Übungsaufgabe aus Blatt 11 ein:

1. Wir wollen heute "wavelets für Dummies" machen. Erster Schritt ist eine Transformation `w=haar(v)` auf einem Vektor mit Komponenten $v_1, \ldots, v_{2n}$, die diesen $2n$-Vektor in $w_1, \ldots, w_n, w_{n+1}, \ldots ..., w_{2n}$ mit $w_i = (v_{2i-1} + v_{2i})/\sqrt{2}$ und $w_{n+i} = (v_{2i-1} - v_{2i})/\sqrt{2}$ für $1 \leq i \leq n$ transformiert. Die inverse Transformation ist $v_{2i-1} = (w_i + w_{n+i})/\sqrt{2}$ und $v_{2i} = (w_i - w_{n+i})/\sqrt{2}$ für $1 \leq i \leq n$. Die Grundidee dieser Transformation ist, dass bei einem "fast konstanten" Inputvektor v der Outputvektor w zur Hälfte aus "Fast-Nullen" besteht. Man bekommt einen Kompressionsalgorithmus für vektoriell geschriebene digitale Signale $v \in IR^{2n}$, indem man erst transformiert, dann die "Fast-Nullen" wegwirft, das Signal speichert oder überträgt und schließlich zurücktransformiert. Man mache sich auch klar, daß die Transformation die Quadratsumme und damit auch die euklidische Länge des Vektors konstant läßt. Sie beschreibt geometrisch einen (orthogonalen) Basiswechsel (warum?).

2. Man programmiere die Transformationen als m-files `haar.m` und `ihaar.m` mit der Startzeile
`function w=haar(v)` bzw. `v=ihaar(w)`.

3. Es sei jetzt $n = 2^p$ eine reine Zweierpotenz. Die **Haar-wavelet-Transformation** `y=hwt(v)` transformiert nacheinander `v(1:m,1)=haar(v(1:m,1))` für $m = n, n/2, n/4, \ldots, 2$ und gibt das Resultat als `y` zurück. Man realisiere das als Funktion `hwt.m` wie oben.

4. Die `inverse Haar-wavelet-Transformation y=ihwt(v)` macht dasselbe mit `ihaar.m`, aber **rückwärts**, d.h. für $m = 2, 4, 8, \ldots, n = 2^p$. Man realisiere das als Funktion `ihwt.m` wie oben.

5. Man teste die Transformation und ihre Inverse auf ein paar gut plotbaren synthetischen "Signalen", z.B. einem Sinus, einem Hut und einem Sprung. Dazu ist schon ein m-file `testhwt.m` und ein Ergebnis auf der website vorgegeben.

6. Und jetzt kommt das verspätete Weihnachtsgeschenk:
Man hole sich das m-file `wavsound4.m` und die wave-Dateien von der website, baue die eigenen m-Files ein und höre sich den Effekt der wavelet-Komprimierung an. Dazu muss man aber leider in den NAM-WAP-Raum gehen, weil die MI-Rechner meines Wissens keinen Sound zulassen. Aus reinem Jux kann man das dann auch mal mit der Cosinustransformation und der FFT versuchen. Wir haben aber nicht den Ehrgeiz, auf Anhieb MP3-Qualität zu erreichen.

Für unser weiteres Vorgehen nehmen wir an, dass die Samples tatsächlich von einer Funktion f kommen, die auf den Intervallen $[2^{-n}k, 2^{-n}(k+1))$ konstant ist. Daher definieren wir

Definition 1 *Die Skalierungsfunktion nach Haar ist definiert durch*

$$\phi(x) = \begin{cases} 1, & \textit{falls } 0 \leq x < 1, \\ 0, & \textit{sonst.} \end{cases}$$

Desweiteren setzen wir

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

und

$$V_j = \overline{span\{\phi_{j,k} : k \in \mathbb{Z}\}}, \tag{7.0}$$

wobei der Abschluss der Abschluss in $L_2(IR)$ sein soll.

Der Raum V_j besteht also aus allen Funktionen aus $L_2(IR)$, die auf den Intervallen $[2^{-j}k, 2^{-j}(k+1))$ konstant sind. Der Faktor $2^{j/2}$ in der Definition von $\phi_{j,k}$ ist so gewählt, dass

$$\|\phi_{j,k}\|^2_{L_2(IR)} = \int_{-\infty}^{\infty} |\phi_{j,k}(x)|^2 dx = 1$$

gilt. Desweiteren hat $\phi_{j,k}$ offensichtlich den Träger

$$\text{supp}(\phi_{j,k}) = [2^{-j}k, 2^{-j}(k+1)].$$

Die Haarsche Skalierungsfunktion wird uns im Folgenden immer als Muster-Beispiel dienen. Ebenso wird der Index j immer die *Skalierung* und der Index k die *Verschiebung* oder auch *Translation* bezeichnen.

7.6 Multi-Skalen-Analyse und Wavelets

Die Räume V_j aus (7.0) haben einige nützliche Eigenschaften, die wir nun zusammenstellen wollen.

Theorem 7.1 *Die V_j sind abgeschlossene Unterräume von $L_2(IR)$ 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. $\overline{\bigcup_{j\in\mathbb{Z}} V_j} = L_2(IR)$,
4. $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$,
5. $\{\phi(\cdot - k) : k \in \mathbb{Z}\}$ *ist eine orthonormale Basis von* V_0.

Beweis: Die Eigenschaften (1) und (2) sind offensichtlich erfüllt, (3) folgt aus der Tatsache, dass sich jede $L_2(IR)$-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 $L_2(IR)$ 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 $L_2(IR)$, da Redundanzen auftreten.

Die in Satz 7.1 hergeleiteten Eigenschaften sind in der Wavelet-Theorie enorm wichtig und geben Anlass zu folgender Definition.

Definition 2 *Sei $\{V_j\}_{j\in\mathbb{Z}}$ eine Familie von abgeschlossenen Unterräumen, zu der es eine Funktion $\phi \in L_2(IR)$ gibt, sodass die Eigenschaften (1)-(5) aus Satz 7.1 gelten. Dann heißt $\{V_j\}$ eine* Multi-Skalen-Analyse (Multiresolution Analysis) *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 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

$$\phi(x) = \sum_{k\in\mathbb{Z}} p_k \phi(2x-k), \tag{7.0}$$

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

$$\phi(x) = \phi(2x) + \phi(2x-1), \tag{7.0}$$

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*.

Wir wollen uns dies zunächst für die Haarsche Skalierungsfunktion exemplarisch überlegen. Da ϕ hier die charakteristische Funktion von $[0,1)$ ist, liegt es wegen $(\phi,\psi)_{L_2(\mathbb{R})} = 0$ Nahe, ψ folgendermaßen anzusetzen:

Definition 3 *Das Haar Wavelet ist die Funktion*

$$\psi(x) = \phi(2x) - \phi(2x-1) = \begin{cases} 1, & \textit{falls } 0 \le x < 1/2, \\ -1, & \textit{falls } 1/2 \le x < 1, \\ 0 & \textit{sonst.} \end{cases}$$

Theorem 18 *Sei ψ das Haar-Wavelet. Dann ist die Familie $\{\psi_{j,k} : k \in \mathbb{Z}\}$ 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:

$$V_{j+1} = W_j \oplus V_j = W_j \oplus W_{j-1} \oplus V_{j-1} = \ldots = \bigoplus_{\ell\le j} W_i.$$

Der Grenzwert liefert dann die Behauptung. Schließlich bilden die $\{\psi_{j,k} : j,k \in \mathbb{Z}\}$ tatsächlich eine orthonormale Basis für $L_2(IR)$. 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, den wir hier nicht beweisen wollen. Wir werden aber im Ramen der schnellen Wavelet-Transformation zumindest zeigen, dass die Shifts von ϕ und ψ den vollen Raum V_0 ergeben. Einen vollständigen, elementaren Beweis findet man in der Literatur. Man beachte, dass die im Satz angegebene Konstruktion bei der Haarschen Skalierungsfunktion bis auf das Vorzeichen zu obigem Haar-Wavelet führt.

Theorem 19 *Sei (V_j) eine MRA mit orthogonaler Skalierungsfunktion $\phi \in V_0$. Seien $\{c_k\} \in \ell_2$ die Koeffizienten der Verfeinerungsgleichung (7.0). Setzt man*

$$\psi = \sum_{k\in\mathbb{Z}} (-1)^k p_{1-k}\phi(2x-k), \tag{7.0}$$

so 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 $L_2(IR)$.

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 Textes hinaus geht. Wir verweisen daher auf die Literatur. 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.

7.7 Die schnelle Wavelet-Transformation

Wie sehen nun die Wavelet Zerlegung und die Rekonstruktion aus? Eine entscheidende Rolle spielen dabei die Verfeinerungsgleichung und die Wavelet-Definition, die wir jetzt mit $h_k = c_k/\sqrt{2}$ und $g_k = (-1)^k h_{1-k}$ folgendermaßen schreiben wollen:

$$\phi_{j,k} = \sum_\ell h_\ell \phi_{j+1,2k+\ell} \qquad \text{und} \qquad \psi_{j,k} = \sum_\ell g_\ell \phi_{j+1,2k+\ell}.$$

Der erste Schritt ist die Projektion der gegebenen Funktion $f \in L_2(I\!R)$ in einen der Räume V_n für hinreichend großes n. Diese Projektion lässt sich schreiben als

$$P_n f = \sum_{k\in\mathbb{Z}} c_k^{(n)}(f)\phi_{n,k}.$$

Der Rest erfolgt mit den hierbei berechneten Koeffizienten. Daher wollen wir von nun an annehmen, dass bereits $f \in V_{j+1}$ gilt. Bei der schnellen Wavelet Transformation wollen wir aus der Darstellung

$$f = \sum_k c_k^{(j+1)} \phi_{j+1,k} \tag{7.0}$$

auf dem feineren $(j+1)$-ten Level die Darstellung

$$f = \sum_k c_k^{(j)} \phi_{j,k} + \sum_k d_k^{(j)} \psi_{j,k} \tag{7.0}$$

berechnen. Dies ist möglich wegen $V_{j+1} = V_j \oplus W_j$. Es handelt sich dabei um eine Transformation der Koeffizientenfolgen. Aus der Orthonormalität erhält man

$$\begin{aligned} c_k^{(j)} &= (f,\phi_{j,k})_{L_2(I\!R)} = \sum_\ell h_\ell (f,\phi_{j+1,2k+\ell})_{L_2(I\!R)} = \sum_\ell h_\ell c_{2k+\ell}^{(j+1)} \\ &= \sum_\ell h_{\ell-2k} c_\ell^{(j+1)} \end{aligned}$$

und genauso

$$d_k^{(j)} = (f,\psi_{j,k})_{L_2(I\!R)} = \sum_\ell g_\ell (f,\phi_{j+1,2k+\ell})_{L_2(I\!R)} = \sum_\ell g_{\ell-2k} c_\ell^{(j+1)}.$$

Die Veranschaulichung wird nun gerade wieder durch Abbildung 30 gewährleistet. Die bei der Zerlegung auftretenden Summen sind diskrete Faltungen mit den *Filtern* $H = \{h_\ell\}$ und $G = \{g_\ell\}$, was die Bezeichnungen in Abbildung 30 noch einmal erklärt. Bei der Wavelet-Transformation 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. Speichern muss man dabei nur die Koeffizienten auf auf dem gröbsten Level und sämtliche Details.

Kommen wir nun zur Rekonstruktion. Hier soll aus der Darstellung (7.0) die Darstellung (7.0) wieder gewonnen werden. Dies ist natürlich wieder eine Operation auf den Koeffizienten. Zunächst einmal notieren wir

$$(\phi_{j,k}, \phi_{j+1,k})_{L_2(I\!R)} = \sum_n h_n (\phi_{j+1,2\ell+n}, \phi_{j+1}, k)_{L_2(I\!R)} = h_{k-2\ell}$$

und

$$(\psi_{j,\ell}, \phi_{j+1,k})_{L_2(I\!R)} = \sum_n g_n (\phi_{j+1,2\ell+n}, \phi_{j+1}, k)_{L_2(I\!R)} = g_{k-2\ell}.$$

Damit erhalten wir

$$\begin{aligned} c_k^{(j+1)} &= (f, \phi_{j+1,k})_{L_2(I\!R)} \\ &= \sum_\ell c_\ell^{(j)} (\phi_{j,\ell}, \phi_{j+1,k})_{L_2(I\!R)} + \sum_\ell d_\ell^{(j)} (\psi_{j,\ell}, \phi_{j+1,k})_{L_2(I\!R)} \\ &= \sum_\ell \left[c_\ell^{(j)} h_{k-2\ell} + d_\ell^{(j)} g_{k-2\ell} \right], \end{aligned}$$

sodass die Wavelet-Rekonstruktion sich wieder mit den Filtern G und H wie in Abbbildung 31 veranschaulichen lässt. Kompression lässt sich nun erreichen, indem man "kleine" Koeffizienten $d_k^{(j)}$ nicht mehr speichert.

Für die Haarsche Skalierungsfunktion und das Haar-Wavelet sind die Filter G und H besonders einfach. Wir erhalten für die Wavelet-Transformation

$$\begin{aligned} c_k^{(j-1)}(f) &= \frac{1}{\sqrt{2}} \left(c_{2k}^{(j)}(f) + c_{2k+1}^{(j)}(f) \right) \\ d_k^{(j-1)}(f) &= \frac{1}{\sqrt{2}} \left(c_{2k}^{(j)}(f) - c_{2k+1}^{(j)}(f) \right). \end{aligned}$$

Dies entspricht bis auf die Normierung genau der Mittelung und Restbildung, die wir am Anfang des Kapitels als Motivation hatten. Entsprechend ist die Wavelet-Rekonstruktion gegeben durch

$$\begin{aligned} c_{2k}^{(j)}(f) &= \frac{1}{\sqrt{2}} \left(c_k^{(j-1)}(f) + d_k^{(j-1)}(f) \right) \\ c_{2k+1}^{(j)}(f) &= \frac{1}{\sqrt{2}} \left(c_k^{(j-1)}(f) - d_k^{(j-1)}(f) \right) \end{aligned}$$

Beim Haar-Wavelet lassen sich auch die Koeffizienten auf dem höchsten Level leicht (wenigstens näherungsweise) berechnen. Da die $\phi_{n,k}$, $k \in \mathbb{Z}$, eine orthonormal Basis bilden, gilt

$$c_k^{(n)}(f) = (f, \phi_{n,k}) = \int_{-\infty}^{\infty} f(x) \phi_{n,k}(x) dx = 2^{n/2} \int_{2^{-n}k}^{2^{-n}(k+1)} f(x) dx,$$

und der letzte Ausdruck kann z.B. durch eine Quadraturformel genähert werden.

7.8 Verfeinerbare Funktionen

Hier beginnt ein "verschärfter" Text zu wavelets. Dieser Text setzt den über translationsinvariante Räume voraus, denn z.B. kommt hier das Klammerprodukt vor. Ebenso sollte der obige elementare Text über das Haar-wavelet schon gelesen sein.

Es gelte die Verfeinerungsgleichung

$$\varphi(x) = \sum_{k \in \mathbb{Z}} p_k \varphi(2x - k)$$

für eine Funktion $\varphi \in L_2(IR)$ unter geeigneten Voraussetzungen an φ bzw. die Koeffizienten $p_k \in IR$ 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.

Lemma 7.1 *Die Fouriertransformierte einer verfeinerbaren Funktion φ mit*

$$\sum_{k \in \mathbb{Z}} |p_k| < \infty$$

erfüllt

$$\hat{\varphi}(\omega) = \hat{\varphi}\left(\frac{\omega}{2}\right) P\left(e^{-i\omega/2}\right) \tag{7.1}$$

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

$$\begin{aligned} (\hat{\varphi}(2 \cdot -k))(\omega) &= \frac{1}{\sqrt{2\pi}} \int_{IR} \varphi(2x - k) e^{-ix\omega} dx \\ &= \frac{1}{2} \frac{1}{\sqrt{2\pi}} \int_{IR} \varphi(y) e^{-iy\omega/2} e^{-ik\omega/2} dx \\ &= \frac{1}{2} \hat{\varphi}\left(\frac{\omega}{2}\right) e^{-ik\omega/2} \end{aligned} \tag{7.2}$$

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.

Iteriert man die Beziehung (7.1) 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 später. Aber wir folgern aus (7.1) noch, dass aus der Gleichung im Nullpunkt folgt, dass $P(1) = 1$ und damit

$$\sum_{k \in \mathbb{Z}} p_k = 2 \tag{7.3}$$

gelten sollte, wenn man sich an die Konstruktion wagt.

Wir rechnen jetzt mal das Klammerprodukt einer verfeinerbaren Funktion aus:

$$\begin{aligned}
[\varphi,\varphi](\omega) &= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega+2\pi k)|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi k)P(e^{-i(\omega/2+\pi k)})|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi 2k)P(e^{-i(\omega/2+\pi 2k)})|^2 \\
&\quad + \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+\pi(2k+1))P(e^{-i(\omega/2+\pi(2k+1))})|^2 \\
&= \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2\pi k)P(e^{-i\omega/2})|^2 \\
&\quad + \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2k\pi+\pi)P(e^{-i(\omega/2+\pi)})|^2 \\
&= |P(e^{-i\omega/2})|^2 \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2\pi k)|^2 \\
&\quad + |P(e^{-i(\omega/2+\pi)})|^2 \sum_{k\in\mathbb{Z}} |\hat{\varphi}(\omega/2+2k\pi+\pi)|^2 \\
&= |P(e^{-i\omega/2})|^2[\varphi,\varphi](\omega/2) + |P(e^{-i(\omega/2+\pi)})|^2[\varphi,\varphi](\omega/2+\pi) \\
&= |P(z)|^2[\varphi,\varphi](\omega/2) + |P(-z)|^2[\varphi,\varphi](\omega/2+\pi),
\end{aligned}$$

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 7.2 *Im Falle orthogonaler Translate von φ gilt*

$$1 = |P(z)|^2 + |P(-z)|^2 \tag{7.4}$$

auf dem Einheitskreis. Diese Beziehung ist äquivalent zu

$$2\delta_{j0} = \sum_{k\in\mathbb{Z}} p_k p_{k-2j},\ j \in \mathbb{Z}.$$

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. □

7.8.1 Strang–Fix–Bedingungungen

Jetzt wollen wir untersuchen, wann **verfeinerbare** Skalierungsfunktionen die Strang–Fix–Bedingungen erfüillen. Wir setzen die Verfeinerungsgleichung in der Form

$$\begin{aligned}
\hat\varphi(\omega) &= \hat\varphi(\omega/2) P(z) \\
&=: \hat\varphi(\omega/2) H(\omega/2) \\
z &= e^{-i\omega/2} \\
H(\omega/2) &:= P(e^{-i\omega/2})
\end{aligned}$$

voraus und nehmen an, daß $\hat\varphi$ glatt ist und schnell nach $\pm\infty$ abfällt.

Theorem 7.2 *Gilt*

$$P^{(j)}(-1) = 0,\ 1 \le j < n, \tag{7.5}$$

so erfüllt φ die Strang–Fix–Bedingungen der Ordnung n.

Beweis: Zuerst einmal setzen wir $\omega = 2\pi$ in $H(\omega/2) = P(e^{-i\omega/2})$ und bekommen $P(-1) = H(\pi)$. Ferner kann man relativ leicht induktiv beweisen, daß die Voraussetzung (7.5) zu

$$H^{(j)}(\pi) = 0,\ 1 \le j < m$$

äquivalent ist, denn die Transformation $t \mapsto e^{-it}$ ist lokal sehr gutartig: alle Ableitungen verschwinden nicht.

Nun differenzieren wir die Verfeinerungsgleichung j–mal und bekommen

$$\hat{\varphi}^{(j)}(\omega) = 2^{-j} \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(\omega/2) H^{(j-m)}(\omega/2).$$

Das werten wir in $\omega_k = 2\pi k$ aus:

$$\hat{\varphi}^{(j)}(2\pi k) = 2^{-j} \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(\pi k) H^{(j-m)}(k\pi).$$

Ist k in obiger Gleichung gerade, so folgt für alle $k \in \mathbb{Z}$

$$\hat{\varphi}^{(j)}(4\pi k) = 2^{-j} \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(2\pi k) H^{(j-m)}(2k\pi),$$

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 $k \in \mathbb{Z}$

$$\hat{\varphi}^{(j)}(4\pi k + 2\pi) = 2^{-j} \sum_{m=0}^{j} \binom{m}{j} \hat{\varphi}^{(m)}(2\pi k + \pi) H^{(j-m)}(2k\pi + \pi).$$

Weil $H(t) = e^{-it}$ die Periode 2π hat, folgt

$$\hat{\varphi}^{(j)}(4\pi k + 2\pi) = 0,\ k \in \mathbb{Z}, 0 \le j < n.$$

An einer beliebigen Stelle der Form $2\pi m$ mit $m \neq 0$ zerlegen wir $m =: 2^p(2q+1)$ mit $p \ge 0$ und $q \in \mathbb{Z}$. Dann wissen wir, daß

$$\hat{\varphi}^{(j)}((2q+1)2\pi) = 0,\ 0 \le j < n$$

gilt, und nach p-maliger Anwendung des vorigen Vererbungsarguments ergibt das auch

$$\hat{\varphi}^{(j)}(2^p(2q+1)2\pi) = 0,\ 0 \le j < n. \square$$

Somit gilt

Theorem 7.3 *Ist $\hat{\varphi}$ mindestens n–mal stetig differenzierbar und verfeinerbar mit* (7.1) *und* (7.5)*, 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 $W_2^n(IR)$.* $\square$

7.8.2 Allgemeine Wavelets

Wenn man von einer verfeinerbaren Funktion φ ausgeht, bekommt man erst einmal den shift–invarianten Raum

$$V_0 := S_\varphi := \operatorname{clos}_{L_2(IR)} \operatorname{span} \{\varphi(\cdot - k)\ :\ k \in \mathbb{Z}\}.$$

Die Verfeinerbarkeit sichert die Inklusion $V_0 \subset V_1$ mit

$$V_1 := S_{\varphi(2\cdot)} := \operatorname{clos}_{L_2(IR)} \operatorname{span} \{\varphi(2\cdot - k)\ :\ k \in \mathbb{Z}\}.$$

Wir wollen nun die Orthogonalzerlegung

$$V_1 = V_0 + W_0$$

durchführen und W_0 durch ein **wavelet** ψ erzeugen als

$$W_0 := S_\psi := \text{clos}_{L_2(IR)} \text{span}\{\psi(\cdot - k) \ : \ k \in \mathbb{Z}\}.$$

Wir definieren

$$\eta(x) := \varphi(2x), \ x \in IR$$

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. Weil es in $W_0 \subset V_1$ liegt, müßte es dann auch eine Gleichung der Form

$$\psi(x) := \sum_{k \in \mathbb{Z}} q_k \varphi(2x - k) \tag{7.6}$$

erfüllen. Rechnen wir due 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 (7.1) 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

$$\begin{aligned} [\eta, \varphi](\omega)\hat{\varphi}(\omega) &= \hat{\varphi}(\omega/2)P(z) \sum_{k \in \mathbb{Z}} \hat{\eta}(\omega + 2\pi k)\overline{\hat{\varphi}(\omega + 2\pi k)} \\ &= \frac{1}{2}\hat{\varphi}(\omega/2)P(z) \sum_{k \in \mathbb{Z}} \hat{\varphi}(\omega/2 + \pi k)\overline{\hat{\varphi}(\omega/2 + \pi k)P(e^{-i(\omega + 2\pi k)/2})} \\ &= \frac{1}{2}\hat{\varphi}(\omega/2)P(z) \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega/2 + \pi k)|^2 \overline{P(z(-1)^k)} \\ &= \frac{1}{2}\hat{\varphi}(\omega/2)P(z)\left([\varphi, \varphi](\omega/2)\overline{P(z)} + [\varphi, \varphi](\omega/2 + \pi)\overline{P(-z)}\right). \end{aligned}$$

Insgesamt ist das

$$\begin{aligned} 2\hat{\psi}_1(\omega) &= \hat{\varphi}(\omega/2)[\varphi, \varphi](\omega/2 + \pi)\left(|P(-z)|^2 - P(z)\overline{P(-z)}\right) \\ &= \hat{\varphi}(\omega/2)[\varphi, \varphi](\omega/2 + \pi)\overline{P(-z)}z \underbrace{z^{-1}\left(P(-z) - P(z)\right)}_{=:A(\omega)} \end{aligned}$$

mit dem 2π–periodischen Teil

$$\begin{aligned} A(\omega) &= z^{-1}\left(P(-z)-P(z)\right) \\ &= e^{i\omega/2}\left(P(-e^{-i\omega/2})-P(e^{-i\omega/2})\right) \\ A(\omega+2\pi) &= e^{i(\omega+2\pi)/2}\left(P(-e^{-i(\omega+2\pi)/2})-P(e^{-i(\omega+2\pi)/2})\right) \\ &= (-z^{-1})\left(P(z)-P(-z)\right) \\ &= A(\omega). \end{aligned}$$

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 unter schwachen Zusatzvoraussetzungen 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

$$Q(z) := \frac{1}{2}\sum_{k\in\mathbb{Z}} q_k z^k.$$

Das liefert die Existenz einer Verfeinerungsleichung (7.6) und wir haben ψ_2 als Kandidaten für ein wavelet. Diese Konstruktionstechnik führt in den allermeisten konkreten Fällen 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 später.

Weil wir einige Vereinfachungen vorgenommen haben, sind wir nicht sicher, ob das Ergebnis alle gewünschten Eigenschaften hat.

Theorem 7.4 *Die Translate von ψ_2 sind orthogonal zu denen von φ, und sie spannen zusammen mit diesen den Raum V_1 auf.*

Beweis: Zunächst mal müssen wir $[\varphi,\psi_2]=0$ nachweisen, um zu zeigen, dass alle Translate von ψ_2 zu denen von φ orthogonal sind. Wir rechnen das aus:

$$\begin{aligned} [\varphi,\psi_2](\omega) &= \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega+2\pi k)\overline{\hat{\psi}_2(\omega+2\pi k)} \\ &= \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi k)P(e^{-i(\omega+2\pi k)/2})\cdot \\ &\quad \cdot\overline{\hat{\varphi}(\omega/2+k\pi)[\varphi,\varphi](\omega/2+\pi k+\pi)e^{-i(\omega+2\pi k)/2}\overline{P(-e^{-i(\omega+2\pi k)/2})}} \\ &= \sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi 2k)P(e^{-i(\omega+2\pi 2k)/2})\cdot \\ &\quad \cdot\overline{\hat{\varphi}(\omega/2+2k\pi)[\varphi,\varphi](\omega/2+2\pi k+\pi)e^{-i(\omega+2\pi 2k)/2}\overline{P(-e^{-i(\omega+2\pi 2k)/2})}} \\ &\quad +\sum_{k\in\mathbb{Z}} \hat{\varphi}(\omega/2+\pi 2k+\pi)P(e^{-i(\omega+2\pi(2k+1))/2})\cdot \\ &\quad \cdot\overline{\hat{\varphi}(\omega/2+2k\pi+\pi)[\varphi,\varphi](\omega/2+\pi(2k+1)+\pi)e^{-i(\omega+2\pi(2k+1))/2}\overline{P(-e^{-i(\omega+2\pi(2k+1))/2})}} \\ &= [\varphi,\varphi](\omega/2)[\varphi,\varphi](\omega/2+\pi)P(z)\overline{z}P(-z) \\ &\quad +[\varphi,\varphi](\omega/2+\pi)[\varphi,\varphi](\omega/2)P(-z)\overline{(-z)}P(z) \\ &= [\varphi,\varphi](\omega/2+\pi)[\varphi,\varphi](\omega/2)\overline{z}\left(P(z)P(-z)-P(-z)P(z)\right) \\ &= 0. \end{aligned}$$

Jetzt nehmen wir eine beliebige Funktion $f \in V_1$ her und müssen zeigen, dass sie im Abschluss des Spans der Translate von φ und ψ_2 liegt. Jede solche Funktion f hat die Eigenschaft

$$f(x) = \sum_{k \in \mathbb{Z}} c_k \varphi(2x - k)$$

und deshalb wegen (7.2) auch

$$\hat{f}(\omega) = \frac{1}{2} \hat{\varphi}(\omega/2) \sigma_c(e^{-i\omega/2}).$$

Es genügt zu zeigen, dass aus $[f, \varphi] = [f, \psi_2] = 0$ auch $f = 0$ folgt. Wie bei den bisherigen Rechnungen bekommt man

$$\begin{aligned} [f, \varphi](\omega) &= [\varphi, \varphi](\omega/2) \sigma_c(z) \overline{P(z)} + [\varphi, \varphi](\omega/2 + \pi) \sigma_c(-z) \overline{P(-z)} \\ [f, \psi_2](\omega) &= [\varphi, \varphi](\omega/2)[\varphi, \varphi](\omega/2 + \pi) \overline{z} \left(\sigma_c(z) P(-z) - \sigma_c(-z) P(z) \right). \end{aligned}$$

Wenn man $[f, \varphi] = [f, \psi_2] = 0$ und strikte Positivität von $[\varphi, \varphi]$ voraussetzt, bekommt man ein homogenes lineares Gleichungssystem

$$\begin{pmatrix} [\varphi, \varphi](\omega/2) \overline{P(z)} & [\varphi, \varphi](\omega/2 + \pi) \overline{P(-z)} \\ P(-z) & -P(z) \end{pmatrix} \begin{pmatrix} \sigma_c(z) \\ \sigma_c(-z) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

Die Determinante ist bis auf das Vorzeichen gleich

$$[\varphi, \varphi](\omega/2) \overline{P(z)} P(z) + [\varphi, \varphi](\omega/2 + \pi) \overline{P(-z)} P(-z) = [\varphi, \varphi](\omega) > 0$$

so dass wir auf $\sigma_c(z) = \sigma_c(-z) = 0$ und dann auf $f = 0$ schliessen können.

Deshalb leistet ψ_2 das Verlangte, hat aber nicht notwendig orthogonale Translate, ebensowenig wie φ. □

Immerhin gilt

Theorem 7.5 *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

$$\begin{aligned} [\psi_2, \psi_2](\omega) &= \sum_{k \in \mathbb{Z}} |\hat{\psi}_2(\omega + 2\pi k)|^2 \\ &= \sum_{k \in \mathbb{Z}} |\hat{\varphi}(\omega/2 + \pi k)|^2 [\varphi, \varphi]^2(\omega/2 + \pi k + \pi) |P(-e^{-i(\omega + 2\pi k)/2})|^2 \\ &= [\varphi, \varphi]^2(\omega/2 + \pi)[\varphi, \varphi](\omega/2) |P(-z)|^2 \\ &\quad + [\varphi, \varphi]^2(\omega/2)[\varphi, \varphi](\omega/2 + \pi) |P(z)|^2 \\ &= [\varphi, \varphi](\omega/2 + \pi)[\varphi, \varphi](\omega/2) \cdot \\ &\quad \cdot \left([\varphi, \varphi](\omega/2 + \pi) |P(-z)|^2 + [\varphi, \varphi](\omega/2) |P(z)|^2 \right) \\ &= [\varphi, \varphi](\omega/2 + \pi)[\varphi, \varphi](\omega/2)[\varphi, \varphi](\omega). \square \end{aligned}$$

7.8.3 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. Wer genug Nerven hat, kann folgendes benutzen:

$$\begin{aligned}
\cot x &= \lim_{k\to\infty}\sum_{j=-k}^{k}\frac{1}{x+j\pi} \\
-\frac{1}{(2n-1)!}\frac{d^{2n-1}}{dx^{2n-1}}\cot x &= \sum_{j\in\mathbb{Z}}\frac{1}{(x+2j\pi)^{2n}}
\end{aligned}$$

und daraus für festes n das Klammerprodukt als positives trigonometrisches Polynom explizit ausrechnen.

Man kann dann das Klammerprodukt in das obige Kalkül einsetzen und dazu ein wavelet ausrechnen. Leider bekommt es eine infinite Maske, die aber immerhin exponentiell abfällt. Details lassen wir aber hier weg. Stattdessen lassen wir die verfeinerbaren Skalierungsfunktionen der B–Splines bei den biorthogonalen wavelets wieder auferstehen.

7.8.4 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 7.6 *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

$$0=\sum_{k\in\mathbb{Z}} p_k q_{k-2j},\ j\in\mathbb{Z}. \tag{7.7}$$

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 7.7 *Wenn φ orthonormale Translate hat, und wenn man ψ über* (7.6) *definiert, so folgt aus den simultanen Gleichungen*

$$\begin{aligned} 1 &= |P(z)|^2 + |P(-z)|^2 \\ 1 &= |Q(z)|^2 + |Q(-z)|^2 \\ 0 &= P(z)\overline{Q(z)} + P(-z)\overline{Q(-z)} \end{aligned} \tag{7.8}$$

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 Ausagen 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 (7.7) 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

$$\begin{aligned}
\varphi &= \sum_{k\in\mathbb{Z}} p_k f_k \\
\psi &= \sum_{k\in\mathbb{Z}} q_k f_k \\
(f_k, f_\ell)_2 &= \int_{I\!R} \varphi(2x - k)\varphi(2x - \ell)dx \\
&= \frac{1}{2}\int_{I\!R} \varphi(y - k)\varphi(y - \ell)d4 \\
&= \frac{1}{2}\delta_{k\ell} \\
f_k(x - m) &= \varphi(2(x - m) - k) \\
&= \varphi(2x - 2m + k) \\
&= f_{2m+k}(x)
\end{aligned}$$

.

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. □

7.8.5 Die wavelets von Ingrid Daubechies

Die Gleichungen (7.8) 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 von Ingrid Daubechies.

Aus (7.1) folgte $P(1) = 1$ und damit auch (7.3). Wenn wir Orthogonalität haben wollen, muss (7.4) 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

$$P(z) = (1+z)^n R_1(z)$$

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

$$\begin{aligned} |r(\omega)|^2 &= |R_1(z^2)|^2 \\ &= T(\cos\omega) \\ &= T(1 - 2\sin^2(\omega)/2)) \\ &=: R(\sin^2(\omega)/2)) \end{aligned}$$

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

$$\begin{aligned} \frac{1 \pm z^2}{1 \pm z^2} &= 1 \pm e^{-i\omega} \\ |(1 \pm z^2)|^2 &= 1 \pm e^{+i\omega} \\ &= 2 \pm (e^{+i\omega} + e^{-i\omega}) \\ &= 2 \pm 2\cos\omega \\ 1 + \cos\alpha &= 2\cos^2(\alpha/2) \\ 1 - \cos\alpha &= 2\sin^2(\alpha/2). \end{aligned} \tag{7.9}$$

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

$$4^{-n} = (1-t)^n R(t) + t^n R(1-t) \tag{7.10}$$

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

$$R(t) = \sum_{j=0}^{n-1} \rho_j \binom{n-1}{j} t^j (1-t)^j$$

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 (7.10) ist erfüllt, aber wir brauchen ein trigonometrisches Polynom r mit

$$|r(\omega)|^2 = R(\sin^2(\omega/2)). \tag{7.11}$$

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 (7.10) durch die Konstante $R=\frac{1}{4}$ lösen und (7.11) 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 (7.10) 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 (7.11) 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.

7.8.6 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 so kann man die obige Gleichung 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

$$L^-_{m+1} := L^-_m + 2^m k^-, \;\; L^+_{m+1} := L^+_m + 2^m k^+.$$

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

$$C^{(m)}_i = c^{(m)}_{L^-_m + i - 1}, \; 1 \le i \le L^+_m - L^-_m + 1,$$

$$c^{(m)}_r = C^{(m)}_{r - L^-_m + 1} =, \; L^-_m \le r \le L^+_m.$$

Die Indexumrechnung der linken Seite ist dieselbe, aber mit $m+1$ anstelle von m. Es folgt

$$\begin{aligned} c^{(m+1)}_\ell &= \sum_k p_k c^{(m)}_{\ell - 2^m k} \\ C^{(m+1)}_{\ell - L^-_{m+1} + 1} &= \sum_{k=k^-}^{k^+} P_{k-k^-+1} C^{(m)}_{\ell - 2^m k - L^-_m + 1} \\ C^{(m+1)}_j &= \sum_{s=1}^{k^+ - k^- + 1} P_s C^{(m)}_{j + L^-_{m+1} - 1 - 2^m(s + k^- - 1) - L^-_m + 1} \\ &= \sum_{s=1}^{k^+ - k^- + 1} P_s C^{(m)}_{j - 2^m(s-1)} \end{aligned}$$

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

$$C^{(m+1)}_j = \sum_{s=1}^{k^+ - k^- + 1} P_s C^{(m)}_{j - 2^m(s-1)}, \; 1 \le j \le L^+_{m+1} - L^-_{m+1} + 1$$

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 $c^{(m+1)}_\ell$ 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.

7.8.7 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 hatten wir schon vorgerechnet. 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

$$x_{neu}^{\pm} = \frac{1}{2}(x_{alt}^{\pm} + n^{\pm})$$

angepaßt.

7.8.8 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');
```

7.8.9 Ein paar Bilder

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 32 zeigt den kubischen Fall, aber beim wavelet haben wir etwas gemogelt, weil wir die Formel fest einprogrammiert haben. Aber im letzten Abschnitt zeigt sich, dass wir dadurch zwar nicht das wavelet zu φ, sondern das wavelet zu einer "dualen" Skalierungsfunktion ausgerechnet haben. Immerhin.

Ein orthogonales wavelet vom Daubechies–Typ ist in Abbildung 33 zu sehen.

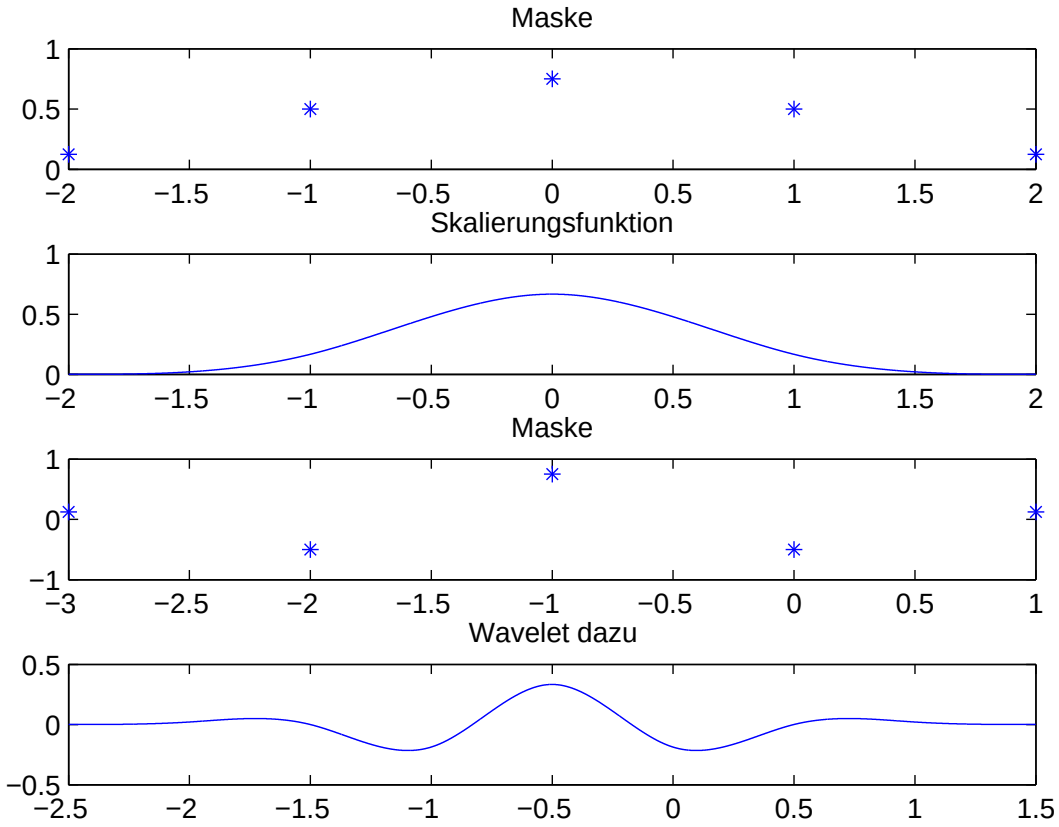

Abbildung 32: Kubisches B–Spline wavelet

Wählt man irgendwelche wilden Masken, so bekommt man oft fraktale Gebilde, siehe Abbildung 34.

7.8.10 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 $f \in W_2^m(IR)$ aus, so kann man die Projektionen im Level j als

$$f_j := P_{\varphi,2^{-j}}(f)$$

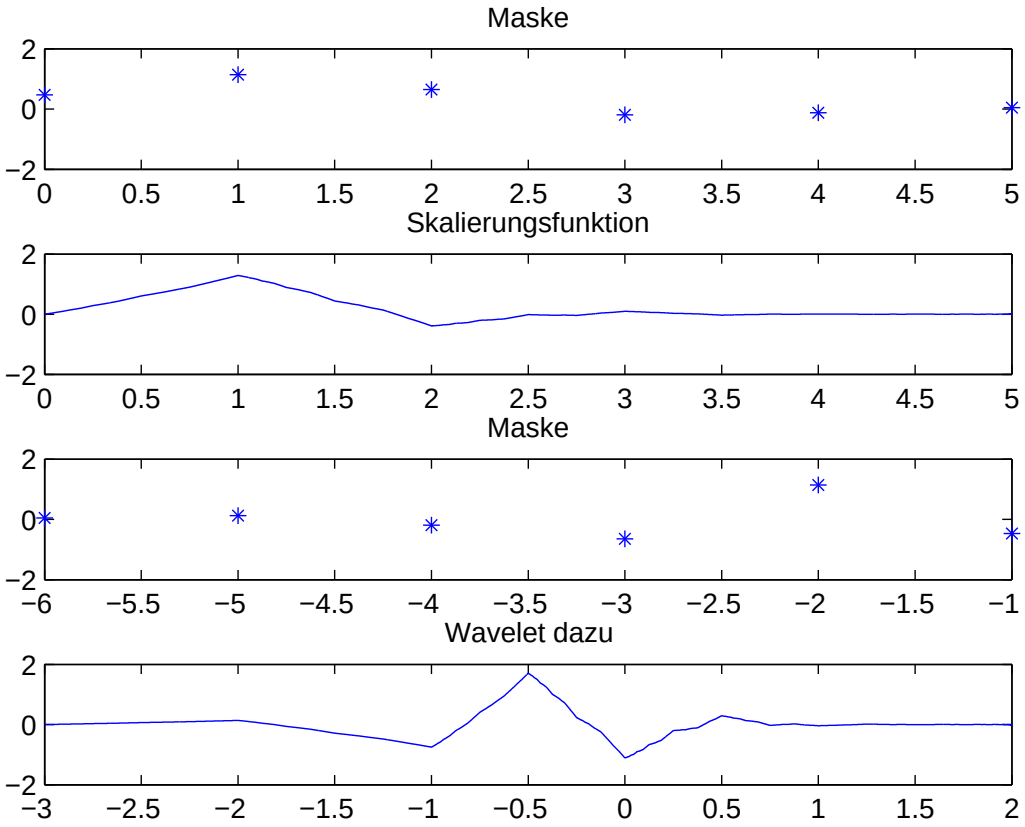

Abbildung 33: Daubechies wavelet

ansetzen und bekommt unter unseren Voraussetzungen aus der Fehlerabschätzung in translationsinvarianten Räumen die Aussage

$$\|f - f_j\|_{L_2(I\!R)} = \|f - P_{\varphi,2^{-j}}(f)\|_{L_2(I\!R)} \leq C 2^{-jm} \|f\|_{W_2^m(I\!R)}.$$

Das ist nicht nur eine Fehlerabschätzung, sondern auch eine Konvergenzaussage für $j \to \infty$.

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

$$\begin{aligned}
(g_j, v_j)_{L_2(I\!R^d)} &= (f_{j+1} - f_j, v_j)_{L_2(I\!R^d)} \\
&= (f_{j+1} - f + f - f_j, v_j)_{L_2(I\!R^d)} \\
&= (f_{j+1} - f, v_j)_{L_2(I\!R^d)} + (f - f_j, v_j)_{L_2(I\!R^d)} \\
&= 0
\end{aligned}$$

für alle $v_j \in V_j$ gilt. Also ist $g_j \in W_j$ der wavelet–Anteil, und wir können die wavelet–Zerlegung

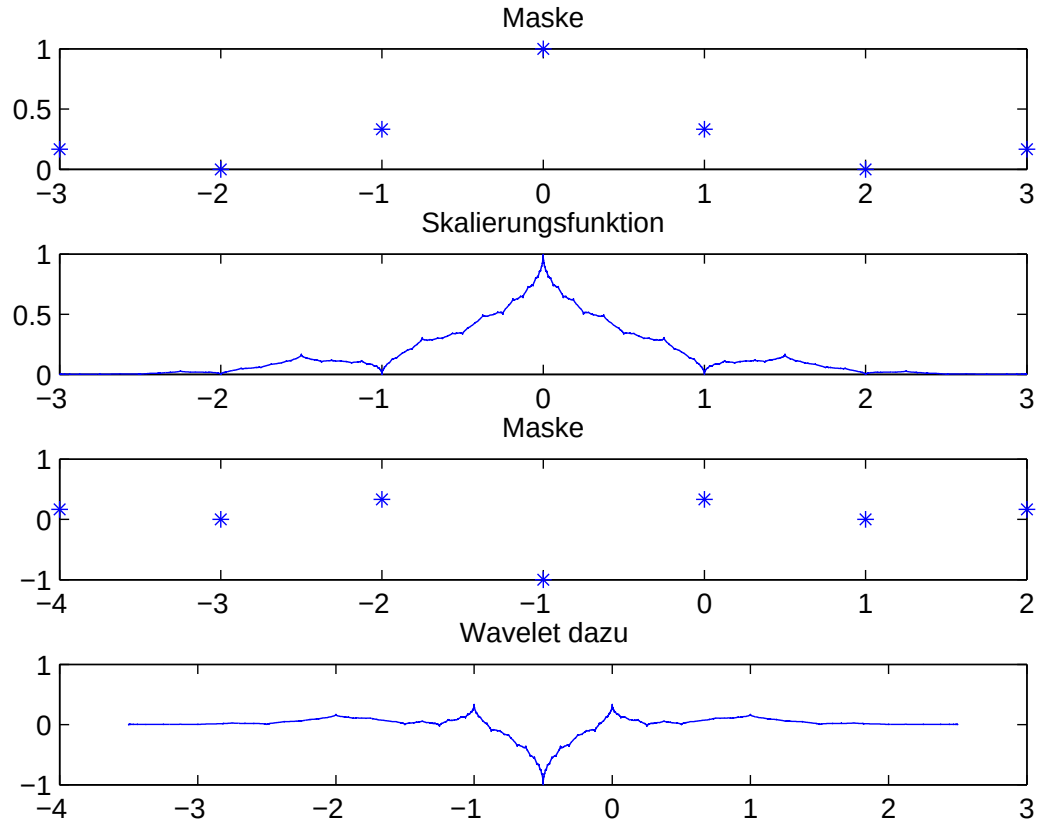

Abbildung 34: Irgendein fraktales wavelet

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 $\|g_j\|_{L_2(I\!R)}$ und es folgt

$$\begin{aligned} \|g_j\|_{L_2(I\!R)} &= \|f_{j+1} - f_j\|_{L_2(I\!R)} \\ &\leq \|f - f_{j+1}\|_{L_2(I\!R)} + \|f - f_j\|_{L_2(I\!R)} \\ &\leq 2C2^{-jm}\|f\|_{W_2^m(I\!R)}. \end{aligned}$$

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.

7.8.11 Biorthogonale Spline–Wavelets

Wir nehmen eine verfeinerbare Funktion φ mit nicht notwendig orthogonalen Translaten her, z.B. den B–Spline φ_n der Ordnung n auf $[0, n]$. Unser Ziel ist es, eine "duale" verfeinerbare

Funktion $\tilde{\varphi}$ zu finden, so daß die **Biorthogonalität**

$$(\varphi(\cdot - j), \tilde{\varphi}(\cdot - k))_{L_2(I\!R)} = \delta_{jk},\ j,k \in Z\!\!Z$$

gilt. Mit der Parsevalschen Gleichung und dem üblichen Transformieren ist das äquivalent zu

$$[\tilde{\varphi}, \varphi](\omega) = \ [\varphi, \tilde{\varphi}](\omega) = \frac{1}{2\pi}.$$

Natürlich setzen wir auch die Verfeinerungsgleichung als

$$\tilde{\varphi}^\wedge(\omega) = \tilde{\varphi}^\wedge(\omega/2)\tilde{P}(z)$$

an und bekommen aus den obigen Gleichungen

$$\begin{aligned}
[\tilde{\varphi}, \varphi](\omega) &= \sum_{k \in Z\!\!Z} \tilde{\varphi}^\wedge(\omega + 2\pi k)\overline{\varphi^\wedge(\omega + 2\pi k)} \\
&= [\tilde{\varphi}, \varphi](\omega/2)\tilde{P}(z)\overline{P(z)} + [\tilde{\varphi}, \varphi](\omega/2 + \pi)\tilde{P}(-z)\overline{P(-z)} \\
1 &= \tilde{P}(z)\overline{P(z)} + \tilde{P}(-z)\overline{P(-z)}.
\end{aligned}$$

Wir wollen obendrein dazu wavelets bauen, die wir dann ψ und $\tilde{\psi}$ nennen, und wir wollen, dass deren span zu dem der "dualen" Skal;ierungsfunktionen orthogonal ist, also

$$\begin{aligned}
(\varphi(\cdot - j), \tilde{\psi}(\cdot - k))_{L_2(I\!R)} &= 0,\ j,k \in Z\!\!Z, \\
(\tilde{\varphi}(\cdot - j), \psi(\cdot - k))_{L_2(I\!R)} &= 0,\ j,k \in Z\!\!Z
\end{aligned}$$

oder

$$\begin{aligned}
[\varphi, \tilde{\psi}] &= 0, \\
[\tilde{\varphi}, \psi] &= 0,
\end{aligned}$$

und sie sollen wie die Skalierungsfunktionen dual zueinander sein, d.h.

$$\begin{aligned}
(\psi(\cdot - j), \tilde{\psi}(\cdot - k))_{L_2(I\!R)} &= \delta_{jk},\ j,k \in Z\!\!Z, \\
[\psi, \tilde{\psi}](\omega) &= \tfrac{1}{2\pi}.
\end{aligned}$$

Mit entsprechenden Verfeinerungsgleichungen im Fourierrraum, also

$$\psi^\wedge(\omega) = \varphi^\wedge(\omega/2)Q(z),\ \tilde{\psi}^\wedge(\omega) = \tilde{\varphi}^\wedge(\omega/2)\tilde{Q}(z)$$

bekommen wir die Gleichungen

$$\begin{aligned}
1 &= \tilde{P}(z)\overline{P(z)} &+ \tilde{P}(-z)\overline{P(-z)} \\
1 &= \tilde{Q}(z)\overline{Q(z)} &+ \tilde{Q}(-z)\overline{Q(-z)} \\
0 &= P(z)\overline{\tilde{Q}(z)} &+ P(-z)\overline{\tilde{Q}(-z)} \\
0 &= \tilde{P}(z)\overline{Q(z)} &+ \tilde{P}(-z)\overline{Q(-z)}.
\end{aligned}$$

Wir machen uns das Erstellen der wavelets sehr einfach, wenn wir wie im orthogonalen Fall, aber mit "dualer" Modifikation den Ansatz

$$\begin{aligned}
Q(z) &:= -z\overline{\tilde{P}(-z)} \\
\tilde{Q}(z) &:= -z\overline{P(-z)}
\end{aligned}$$

machen. Dann sind die obigen Gleichungen bis auf die erste erfüllt, wie man leicht nachrechnet. Die zweite reduziert sich auf die erste, und die anderen verschwinden. Das zeigt, daß wir beim Berechnen unserer wavelets aus Masken im B-Spline–Fall keinen allzu grossen Unsinn gemacht haben, denn wir haben das duale wavelet ausgerechnet ohne zu wissen, was das ist.

Es bleibt also nur noch

$$1 \quad = \quad \tilde{P}(z)\overline{P(z)} \quad + \quad \tilde{P}(-z)\overline{P(-z)}$$

und wir wollen natürlich wegen der Strang–Fix–Bedingungen kräftige Nullstellen bei -1 haben. Man könnte das auch allgemeiner durchrechnen, aber wir machen das hier nur im B–Spline–Fall und setzen einfach $\tilde{\varphi}$ als B–Spline der Ordnung n an, indem wir

$$\tilde{P}(z) = \frac{1}{2^n}(1+z)^n$$

setzen und die obige Gleichung nach P auflösen. Das geht schrittweise mit gewissen Substitutionen wie bei der Herleitung der Daubechies–wavelets.

Wir rechnen erst einmal gewisse Polynome um in

$$\begin{aligned}
(1 \pm z^2)^n &= (1 \pm e^{-i\omega})^n \\
&= (e^{i\omega/2} \pm e^{-i\omega/2})^n e^{-in\omega/2} \\
2^{-n}(1+z^2)^n &= \cos^n(\omega/2)e^{-in\omega/2} \\
2^{-n}(1-z^2)^n &= i^n \sin^n(\omega/2)e^{-in\omega/2}, \\
z^{-2}2^{-2}(1+z^2)^2 &= \cos^2(\omega/2) \\
-z^{-2}2^{-2}(1-z^2)^2 &= \sin^2(\omega/2).
\end{aligned} \tag{7.12}$$

Das ergibt

$$\begin{aligned}
1 &= \tilde{P}(z^2)\overline{P(z^2)} + \tilde{P}(-z^2)\overline{P(-z^2)} \\
&= 2^{-n}(1+z^2)^n\overline{P(z^2)} + 2^{-n}(1-z^2)^n\overline{P(-z^2)} \\
&= \cos^n(\omega/2)e^{-in\omega/2}\overline{P(z^2)} + i^n \sin^n(\omega/2)e^{-in\omega/2}\overline{P(-z^2)}.
\end{aligned} \tag{7.13}$$

Wir wollen wie bei der Herleitung der Daubechies–wavelets wieder

$$\begin{aligned}
y &:= \sin^2(\omega/2) \\
1-y &= \cos^2(\omega/2)
\end{aligned}$$

setzen. Leider haben wir aber keine geraden Potenzen des Sinus und Cosinus in unserer Gleichung. Deshalb mogeln wir die fehlenden Potenzen in die Gleichung herein, indem wir fordern

$$e^{-in\omega/2}\overline{P(z^2)} = \cos^{n+2m}(\omega/2)p_{n+m}(\sin^2(\omega/2))$$

mit einem noch zu bestimmenden reellen algebraischen Polynom p_{n+m}, das wir in weiser Voraussicht mit $n+m$ indizieren, obwohl das hier alles andere als klar ist. Diese Forderung sieht wild aus, macht aber durchaus Sinn, wenn wir sie umschreiben in

$$\begin{aligned}
\overline{P(z^2)} &= \overline{P(e^{-i\omega})} \\
&= e^{in\omega/2}\cos^n(\omega/2)\cos^{2m}(\omega/2)p_{n+m}(\sin^2(\omega/2)) \\
&= 2^{-n}e^{in\omega/2}(e^{i\omega/2}+e^{-i\omega/2})^n \\
&= 2^{-n}(e^{i\omega}+1)^n\cos^{2m}(\omega/2)p_{n+m}(\sin^2(\omega/2)) \\
&= \overline{\tilde{P}(z^2)}\cos^{2m}(\omega/2)p_{n+m}(\sin^2(\omega/2))
\end{aligned} \tag{7.14}$$

weil jetzt beide Seiten wegen (7.9) die Periode 2π haben.

Unser Trick sorgt aber nur für den ersten Term in (7.13), wir hätten aber gerne

$$\begin{aligned} 1 &= \cos^{2n+2m}(\omega/2)p_{n+m}(\sin^2(\omega/2)) + \sin^{2n+2m}(\omega/2)p_{n+m}(\cos^2(\omega/2)) \\ &= (1-y)^{n+m}p_{n+m}(y) + y^{n+m}p_{n+m}(1-y). \end{aligned}$$

Nun ja, das kann man prüfen über

$$\begin{aligned} & \sin^{2n+2m}(\omega/2)p_{n+m}(\cos^2(\omega/2)) \\ = \; & \cos^{2n+2m}((\omega+\pi)/2)p_{n+m}(\sin^2((\omega+\pi)/2)) \\ = \; & \cos^{n}((\omega+\pi)/2)e^{-in(\omega+\pi)/2}\overline{P(e^{i(\omega+\pi)})} \\ = \; & i^n \sin^n(\omega/2)e^{-in\omega/2}\overline{P(-z^2)} \end{aligned}$$

Jetzt sind wir also bei

$$1 = (1-y)^{n+m}p_{n+m}(y) + y^{n+m}p_{n+m}(1-y)$$

und können diese Gleichung zu lösen versuchen. Hier wird auch die Indizierung klar. Natürlich macht man wieder einen Ansatz mit Bernsteinpolynomen

$$p_{n+m}(y) := \sum_{j=0}^{n+m-1} c_j^{(n+m)} y^j (1-y)^{n+m-1-j}$$

und bekommt einen Koeffizientenvergleich zwischen

$$\begin{aligned} 1 &= \sum_{j=0}^{n+m-1} c_j^{(n+m)} \left(y^j(1-y)^{2n+2m-1-j} + (1-y)^j y^{2n+2m-1-j} \right) \\ &= \sum_{j=0}^{n+m-1} c_j^{(n+m)} y^j (1-y)^{2n+2m-1-j} \\ &\quad + \sum_{k=n+m}^{2n+2m-1} c_{2n+2m-1-k}^{(n+m)} (1-y)^{2n+2m-1-k} y^k \\ 1 &= (y+1-y)^{2n+2m-1} \\ &= \sum_{j=0}^{2n+2m-1} \binom{2n+2m-1}{j} y^j (1-y)^{2n+2m-1-j}. \end{aligned}$$

Das ergibt

$$\begin{aligned} c_j^{(n+m)} &= \binom{2n+2m-1}{j}, \; 0 \le j \le n+m-1 \\ c_{2n+2m-1-k}^{(n+m)} &= \binom{2n+2m-1}{k}, \; n+m \le k \le 2n+2m-1 \end{aligned}$$

und diese Gleichungen sind wie durch Wunder nicht widersprüchlich, wie man wieder durch Substitution $k = 2n+2m-1-j$ sieht.

Jetzt geht es rückwärts. Wir haben (7.14), und daraus können wir sofort $P(z^2)$ ausrechnen, weil wir mit (7.12) die fehlenden Terme als Funktion von z^2 schreiben können. Dabei wird P eine rationale Funktion, aber das macht nichts. Es folgt

$$P(z^2) = 2^{-n}(1+z^2)^n \left(z^{-2}2^{-2}(1+z^2)^2 \right)^m p_{n+m}\left(-z^{-2}2^{-2}(1-z^2)^2 \right)$$

und daraus kann man mit einigen Nerven die Maskenkoeffizienten für die entsprechende verfeinerbare Funktion ausrechnen. Man muss mit der Wahl von m und n etwas aufpassen, weil man sicherstellen muß, daß φ noch in $L_2(IR)$ liegt.

Diese biorthogonalen wavelets kann man sich in der wavelet–Toolbox von MATLAB ansehen.

... inkomplett, es fehlen die Rechenformeln im biorthogonalen Fall, und es sollte noch ein Beispiel durchgerechnet werden....

Literatur

1. **Alles aus dem A-Standort der NAM-Bibliothek**

2. Klassische Werke:

   - Cheney 1998 [4]
   - DeVore-Lorentz 1993 [11]

3. Moderneres:

   - Christensen 2005 [5]
   - Stepanets 2005 [17]
   - Tikhomirov 2006 [18]
   - Steffens 2006 [16]

4. Spezielles:

   - Splines: de Boor 2001 [10]
   - Finite Elemente: Braess 2002 [1]
   - Finite Elemente: Brenner und Scott 2002 [2]
   - Lernverfahren: Shawe-Taylor und Cristianini 2004: [15]
   - Lernverfahren: Schölkopf und Smola 2002: [14]
   - Lernverfahren: Cristianini und Shawe-Taylor 2000: [9]
   - Radiale Basisfunktionen: Wendland 2005 [19]
   - Radiale Basisfunktionen: Buhmann 2003 [3]
   - Multivariate Datenmodellierung: Iske 2004 [13]
   - wavelets: Cohen 2003 [8]
   - wavelets: Cohen 2000 [7]
   - wavelets: Chui 1992 [6]
   - wavelets: noch Blatter, Y. Meyer, Louis, Wickerhauser
   - Weitere Numerik, z.B. SVD : Golub-van Loan [12]

Literatur

[1] D. Braess. *Finite Elements. Theory, Fast Solvers and Applications in Solid Mechanics.* Cambridge University Press, 2001.

[2] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods, Second edition.* Springer, 2002.

[3] M. D. Buhmann. *Radial basis functions: theory and implementations*, volume 12 of *Cambridge Monographs on Applied and Computational Mathematics.* Cambridge University Press, Cambridge, 2003.

[4] E. W. Cheney. *Introduction to approximation theory.* AMS Chelsea Publishing, Providence, RI, 1998. Reprint of the second (1982) edition.

[5] Ole Christensen and Khadija L. Christensen. *Approximation theory.* Applied and Numerical Harmonic Analysis. Birkhäuser Boston Inc., Boston, MA, 2005. From Taylor polynomials to wavelets, Corrected second printing of the 2004 original.

[6] Charles K. Chui. *An introduction to wavelets*, volume 1 of *Wavelet Analysis and its Applications.* Academic Press Inc., Boston, MA, 1992.

[7] Albert Cohen. Wavelet methods in numerical analysis. In *Handbook of numerical analysis, Vol. VII*, Handb. Numer. Anal., VII, pages 417–711. North-Holland, Amsterdam, 2000.

[8] Albert Cohen. *Numerical analysis of wavelet methods*, volume 32 of *Studies in Mathematics and its Applications.* North-Holland Publishing Co., Amsterdam, 2003.

[9] N. Cristianini and J. Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods.* Cambridge University Press, Cambridge, 2000.

[10] Carl de Boor. *A practical guide to splines*, volume 27 of *Applied Mathematical Sciences.* Springer-Verlag, New York, revised edition, 2001.

[11] Ronald A. DeVore and George G. Lorentz. *Constructive approximation*, volume 303 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, 1993.

[12] G. Golub and C. van Loan. *Matrix computations.* The Johns Hopkins University Press, 1996. Third edition.

[13] A. Iske. *Multiresolution methods in scattered data modelling*, volume 37 of *Lecture Notes in Computational Science and Engineering.* Springer-Verlag, Berlin, 2004.

[14] B. Schölkopf and A.J. Smola. *Learning with Kernels.* MIT Press, Cambridge, 2002.

[15] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

[16] Karl-Georg Steffens. *The history of approximation theory.* Birkhäuser Boston Inc., Boston, MA, 2006. From Euler to Bernstein.

[17] A. I. Stepanets. *Methods of approximation theory.* VSP, Leiden, 2005.

[18] V. M. Tikhomirov. Approximation theory in the twentieth century. In *Mathematical events of the twentieth century*, pages 409–436. Springer, Berlin, 2006.

[19] Holger Wendland. *Scattered data approximation*, volume 17 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2005.