Wavelets: Anwendungen in Bildanalyse und Numerik

Stephan Dahlke
Institut für Geometrie und
Praktische Mathematik
RWTH Aachen
Templergraben 55
52056 Aachen

1. Anwendungen in der Bildanalyse
2. Anwendungen in der Numerik

1. Anwendungen von Wavelets in der Signal/Bildanalyse

Zunächst Signalanalyse, $f \in L_2(\mathbb{R})$
Daubechies–Wavelets $\phi = \phi^N, \ \psi = \psi^N$
$T_{2^j} \longleftrightarrow$ orthogonalem Projektor P_j auf V_j,

$$\begin{aligned} P_j(f)(x) &= \sum_{k \in \mathbb{Z}} \langle f(\cdot), 2^{j/2}\phi(2^j \cdot -k)\rangle 2^{j/2}\phi(2^j x - k) \\ &=: \sum_{k \in \mathbb{Z}} \langle f, \phi_{j,k}\rangle \phi_{j,k} \end{aligned}$$

'measuring device' $\rightsquigarrow \{\lambda_k^j\}_{k \in \mathbb{Z}}, \qquad \lambda_k^j = \langle f, \phi_{j,k}\rangle$

5.1 Zerlegung des Signals

Kausalitätsprinzip $\Longrightarrow P_{j-1}$ durch P_j bestimmt

$$\begin{aligned} \lambda_l^{j-1} &= \langle f, \phi_{j-1,l}\rangle \\ &= \langle f, 2^{(j-1)/2}\phi(2^{j-1} \cdot -l)\rangle \\ &= \langle f, 2^{(j-1)/2} \sum_{k \in \mathbb{Z}} a_k \phi(2^j \cdot -2l - k)\rangle \\ &= 2^{-1/2} \sum_{k \in \mathbb{Z}} \overline{a}_{k-2l} \langle f, 2^{j/2}\phi(2^j \cdot -k)\rangle \end{aligned}$$

$$= \quad 2^{-1/2} \sum_{k\in\mathbb{Z}} \overline{a}_{k-2l} \lambda_k^j$$

Also gilt:

$$\begin{aligned} P_{j-1}(f) &= \sum_{l\in\mathbb{Z}} \lambda_l^{j-1} \phi_{j-1,l} \\ &= \sum_{l\in\mathbb{Z}} 2^{-1/2} (\sum_{k\in\mathbb{Z}} \overline{a}_{k-2l} \lambda_k^j) \phi_{j-1,l}. \end{aligned}$$

$\mathbf{a}$ induziert diskreten Filter H

$$\lambda^{j-1} = H\lambda^j, \qquad \lambda_l^{j-1} = \sum_{k\in\mathbb{Z}} 2^{-1/2} \overline{a}_{k-2l} \lambda_k^j. \qquad (1)$$

$H \hat{=}$ Tiefpaßfilter

W_j beschreiben die von V_j nach V_{j+1} hinzukommende Detailinformation!

$$\begin{aligned} Q_j(f) &:= \sum_{l\in\mathbb{Z}} \langle f, \psi_{j,l} \rangle \psi_{j,l} = \sum_{l\in\mathbb{Z}} c_l^j \psi_{j,l} \\ &= \sum_{l\in\mathbb{Z}} \langle f, \sum_{k\in\mathbb{Z}} 2^{j/2} b_k \phi(2^{j+1} \cdot -2l - k) \rangle \psi_{j,l} \\ &= \sum_{l\in\mathbb{Z}} 2^{-1/2} (\sum_{k\in\mathbb{Z}} \overline{b}_{k-2l} \langle f, \phi_{j+1,k} \rangle) \psi_{j,l} \\ &= \sum_{l\in\mathbb{Z}} 2^{-1/2} (\sum_{k\in\mathbb{Z}} \overline{b}_{k-2l} \lambda_k^{j+1}) \psi_{j,l} \end{aligned}$$

b induziert ebenfalls diskreten Filter D
$D \hat{=}$ Hochpaßfilter

$$c^j = D\lambda^{j+1}, \qquad c_l^j = 2^{-1/2} \sum_{k \in \mathbb{Z}} \overline{b}_{k-2l} \lambda_l^{j+1} \tag{2}$$

Iteration $\Longrightarrow$ schnelle Wavelet–Transformation, Pyramiden–Algorithmus:

$$\begin{tikzcd}
\lambda^j \arrow[r, "H"] \arrow[dr, "D"'] & \lambda^{j-1} \arrow[r, "H"] \arrow[dr, "D"'] & \lambda^{j-2} \arrow[r, "H"] \arrow[dr, "D"'] & \lambda^{j-3} \arrow[r, dotted, no head] \arrow[dr, dotted, no head] & {} \\
& c^{j-1} & c^{j-2} & c^{j-3} & {}
\end{tikzcd}$$

Filter $(\mathcal{F}\lambda)_l := \sum_{k \in \mathbb{Z}} h_{l-k} \lambda_k = \{h * \lambda\}_l$

$$\begin{aligned}
(\mathcal{H}\lambda)_l &= 2^{-1/2} \sum_{k \in \mathbb{Z}} \overline{a}_{k-l} \lambda_k = \{2^{-1/2} \tilde{\mathbf{a}} * \lambda\}_l, \tilde{a}_l = \overline{a}_{-l} \\
(\mathcal{D}\lambda)_l &= 2^{-1/2} \sum_{k \in \mathbb{Z}} \overline{b}_{k-l} \lambda_k = \{2^{-1/2} \tilde{\mathbf{b}} * \lambda\}_l, \tilde{b}_l = \overline{b}_{-l}
\end{aligned}$$

Interpretation von (1), (2)

- berechne $2^{-1/2}\tilde{\mathbf{a}} * \lambda^j, \; 2^{-1/2}\tilde{\mathbf{b}} * \lambda^j$

- vernachlässige ungerade Indizes downsampling

5.2 Rekonstruktion

$$V_j = V_{j-1} \oplus W_{j-1} \Longrightarrow$$

$$\begin{aligned}
\phi_{j,l} &= \sum_{k\in\mathbb{Z}} \langle \phi_{j,l}, \phi_{j-1,k}\rangle \phi_{j-1,k} + \sum_{n\in\mathbb{Z}} \langle \phi_{j,l}, \psi_{j-1,n}\rangle \psi_{j-1,n} \\
&= 2^{-1/2} \sum_{k\in\mathbb{Z}} \overline{a}_{l-2k}\phi_{j-1,k} + 2^{-1/2} \sum_{n\in\mathbb{Z}} \overline{b}_{l-2n}\psi_{j-1,n} \Longrightarrow
\end{aligned}$$

$$\begin{aligned}
P_j(f) &= \sum_{l\in\mathbb{Z}} \langle f, \phi_{j,l}\rangle \phi_{j,l} = \sum_{l\in\mathbb{Z}} \lambda_l^j \phi_{j,l} \\
&= \frac{1}{\sqrt{2}} \sum_{l\in\mathbb{Z}} (\langle f, \sum_{k\in\mathbb{Z}} \overline{a}_{l-2k}\phi_{j-1,k} + \sum_{n\in\mathbb{Z}} \overline{b}_{l-2n}\psi_{j-1,n}\rangle)\phi_{j,l} \\
&= \frac{1}{\sqrt{2}} \sum_{l\in\mathbb{Z}} (\sum_{k\in\mathbb{Z}} a_{l-2k}\langle f, \phi_{j-1,k}\rangle + \sum_{n\in\mathbb{Z}} b_{l-2n}\langle f, \psi_{j-1,n}\rangle)\phi_{j,l} \\
&= \frac{1}{\sqrt{2}} \sum_{l\in\mathbb{Z}} (\sum_{k\in\mathbb{Z}} a_{l-2k}\lambda_k^{j-1} + \sum_{n\in\mathbb{Z}} b_{l-2n}c_n^{j-1})\phi_{j,l}
\end{aligned}$$

$$\lambda_l^j = 2^{-1/2} \sum_{k\in\mathbb{Z}} a_{l-2k}\lambda_k^{j-1} + 2^{-1/2} \sum_{n\in\mathbb{Z}} b_{l-2n}c_n^{j-1} \qquad (3)$$

$$
\begin{aligned}
(\overline{\mathcal{H}}\lambda)_l &= 2^{-1/2}\sum_{k\in\mathbb{Z}} a_{l-k}\lambda_k = \{2^{-1/2}\mathbf{a}*\lambda\}_l,\\
(\overline{\mathcal{D}}\lambda)_l &= 2^{-1/2}\sum_{k\in\mathbb{Z}} b_{l-k}\lambda_k = \{2^{-1/2}\mathbf{b}*\lambda\}_l
\end{aligned}
$$

Interpretation von (3)

- füge an ungeraden Stellen Nullen ein upsampling
- berechne $2^{-1/2}\mathbf{a}*\lambda^{j-1},\ 2^{-1/2}\mathbf{b}*c^{j-1}$
- addiere

$$
\lambda^j \xrightarrow{\text{Zerlegung}} \binom{\lambda^{j-1}}{c^{j-1}} \xrightarrow[\text{Entrauschen}]{\text{Kompression}} \binom{\tilde\lambda^{j-1}}{\tilde c^{j-1}} \xrightarrow{\text{Rekonstruktion}} \tilde\lambda^j
$$

5.3 Datenkompression

lineare Verfahren:
Approximiere f mittels $\{V_j\}_{j\in\mathbb{Z}}$, d.h. setze:

$$d_k^n = \begin{cases} c_k^n & n \le j \\ 0 & \text{sonst} \end{cases}$$

Jackson–Ungleichung:

$$\inf_{g\in V_j} \|f-g\|_{L_2(\mathbb{R}^d)} \le C_6 2^{-jr} |f|_{H^r} \qquad \Longrightarrow$$

lineare Verfahren gut, falls f hinreichend regulär!

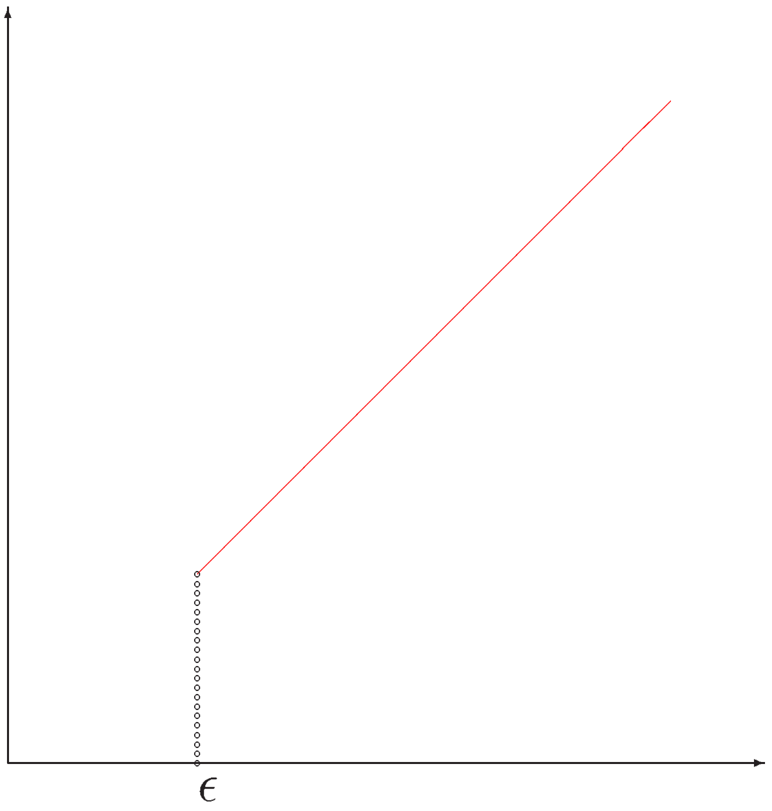
nichtlinerare Verfahren:
Idee: f lokal “glatt” $\Longrightarrow$

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x-x_0) + \frac{1}{2} f''(x_0)(x-x_0)^2 + \text{Rest} \\ c_k^j &= \langle f, \psi_{j,k}\rangle \\ &= \underbrace{\langle f(x_0), \psi_{j,k}\rangle}_{=0} + \underbrace{\langle f'(x_0)(x-x_0), \psi_{j,k}\rangle}_{=0} \\ &\qquad + \frac{1}{2} \underbrace{\langle f''(x_0)(x-x_0)^2, \psi_{j,k}\rangle}_{=0} + \langle \text{Rest}, \psi_{j,k}\rangle \end{aligned}$$

hard thresholding

$$d_k^j = \begin{cases} c_k^j & |c_k^j| > \epsilon \\ 0 & \text{sonst} \end{cases} \Longrightarrow$$

$$f^\epsilon = \sum_{j,k} d_k^j \psi_{j,k} = \sum_{|c_k^j| \geq \epsilon} c_k^j \psi_{j,k}$$



5.4 Entrauschen

$$\tilde{f} = \sum_{j,k} \tilde{c}_k^j \psi_{j,k}$$

$$\tilde{c}_k^j = c_k^j + \eta_k^j, \qquad \eta_k^j \text{ unabhängig } N(0,\sigma)\text{–verteilt}$$

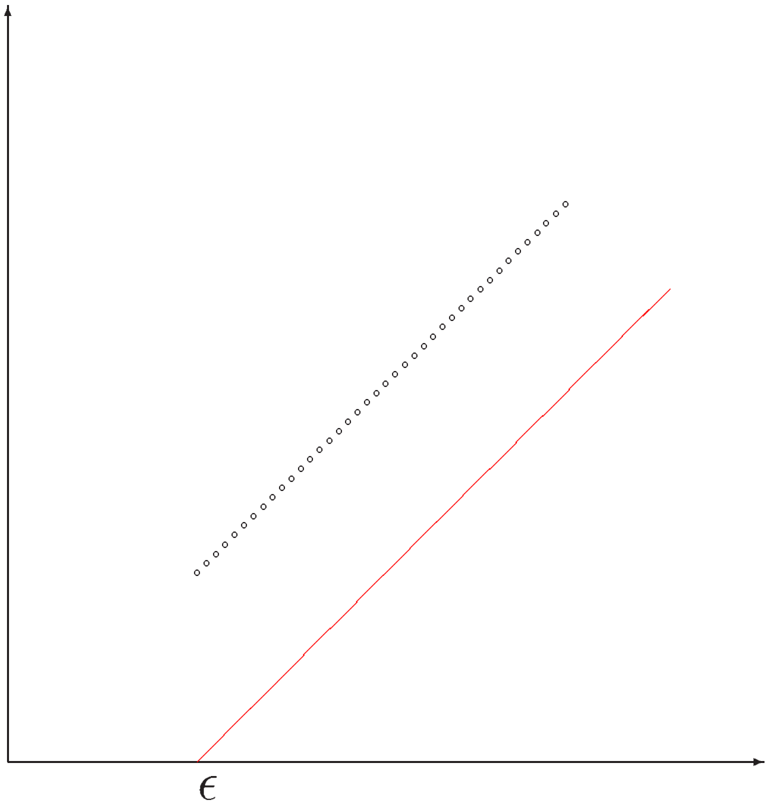
soft thresholding, shrinkage

$$s_\epsilon(t) = \begin{cases} t-\epsilon, & t > \epsilon \\ 0, & |t| \le \epsilon \\ t+\epsilon, & t < -\epsilon \end{cases} \Longrightarrow$$

$$\tilde{f}_\epsilon = \sum_{j,k} s_\epsilon(c_k^j + \eta_k^j)\psi_{j,k}$$

ϵ kann so bestimmt werden, daß

$$E(\|f - \tilde{f}_\epsilon\|) = \min!$$


ϵ

5.5 Der mehrdimensionale Fall, Bildanalyse

Tensorprodukt–Ansatz $(d = 2)$

$$\{\phi(x)\psi(y), \psi(x)\phi(y), \psi(x)\psi(y)\}$$

ist System von Mother Wavelets,

$$V_0 = \overline{\text{span}\{\phi(x-k)\phi(y-l) \mid k, l \in \mathbb{Z}\}}$$

$$\begin{aligned}
\langle f, \phi_{j-1,k}\phi_{j-1,l}\rangle &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{a}_{n-2k}\overline{a}_{t-2l}\langle f, \phi_{j,n}\phi_{j,t}\rangle \\
\langle f, \phi_{j,k}\psi_{j,l}\rangle &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{a}_{n-2k}\overline{b}_{t-2l}\langle f, \phi_{j+1,n}\phi_{j+1,t}\rangle \\
\langle f, \psi_{j,k}\phi_{j,l}\rangle &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{b}_{n-2k}\overline{a}_{t-2l}\langle f, \phi_{j+1,n}\phi_{j+1,t}\rangle \\
\langle f, \psi_{j,k}\psi_{j,l}\rangle &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{b}_{n-2k}\overline{b}_{t-2l}\langle f, \phi_{j+1,n}\phi_{j+1,t}\rangle
\end{aligned}$$

$$
\begin{aligned}
(H\lambda)_{k,l} &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{a}_{n-2k}\overline{a}_{t-2l}\lambda_{n,t} \\
(D^1\lambda)_{k,l} &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{a}_{n-2k}\overline{b}_{t-2l}\lambda_{n,t} \\
(D^2\lambda)_{k,l} &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{b}_{n-2k}\overline{a}_{t-2l}\lambda_{n,t} \\
(D^3\lambda)_{k,l} &= \frac{1}{2}\sum_{n,t\in\mathbb{Z}} \overline{b}_{n-2k}\overline{b}_{t-2l}\lambda_{n,t}
\end{aligned}
$$

Zerlegung und Rekonstruktion analog zum Fall $d = 1$

2. Anwendungen von Wavelets in der Numerik

2.1 Einleitung

Numerische Behandlung von Operatorgleichungen

$$Au = F$$
$$A: \; H^1 \longrightarrow H^2,$$

$H^1, \; H^2$ Sobolev–Räume

Beispiel 1:
Poisson–Gleichung

$$\begin{aligned} -\triangle u &= F \quad \text{in} \quad \Omega \subset \mathbb{R}^d \\ u &= 0 \quad \text{auf} \quad \partial\Omega \\ \triangle u &:= \sum_{i=1}^{d} \frac{\partial^2 u}{\partial x_i^2} \end{aligned} \tag{4}$$

u: Verschiebung einer Membran
F: äußere Kraft

Beispiel 2:
Doppelschicht–Potential

$$(\frac{1}{2}I + \mathcal{K})v = g \quad \text{auf } \partial\Omega,\ \Omega \subset I\!R^d$$

$$\mathcal{K}(f)(x) = \frac{1}{\omega_d}\int_{\partial\Omega}\frac{\langle x-y, \vec{n}(y)\rangle}{|x-y|^d}f(y)d\sigma(y)$$

Beispiel 3:

$$\begin{aligned} -\frac{d^2u}{dx^2} &= F \quad \text{in} \quad [0,1] \\ u(0) &= u(1) = 0 \end{aligned} \tag{5}$$

Existenz und Eindeutigkeit (häufig) geklärt
Aber: wie sieht die Lösung aus?

Galerkin–Verfahren

Multiplikation von (5) mit Testfunktion $\eta \in C_0^\infty$, supp $\eta \subset [0,1]$, partielle Integration $\Longrightarrow$

$$\begin{aligned} -\int_0^1 u''(x)\eta(x)dx &= \int_0^1 F(x)\eta(x)dx \\ \Longleftrightarrow \int_0^1 u'(x)\eta'(x)dx &= \int_0^1 F(x)\eta(x)dx \\ a(u,\eta) &= \mathcal{F}(\eta) \end{aligned} \tag{6}$$

Idee: Projiziere (6) auf endlich–dimensionalen, linearen Teilraum $V_M = \text{span}\{\eta_1, \ldots, \eta_M\}$

Aufgabe: Finde $u^M \in V_M$ mit

$$
\begin{aligned}
a(u^M, v) &= \mathcal{F}(v) \qquad \forall\, v \in V_M \\
\Longleftrightarrow a(u^M, \eta_i) &= \mathcal{F}(\eta_i), \qquad i = 1, \ldots, M
\end{aligned}
\tag{7}
$$

$$u^M = \sum_{j=1}^{M} d_j \eta_j \quad \rightsquigarrow$$

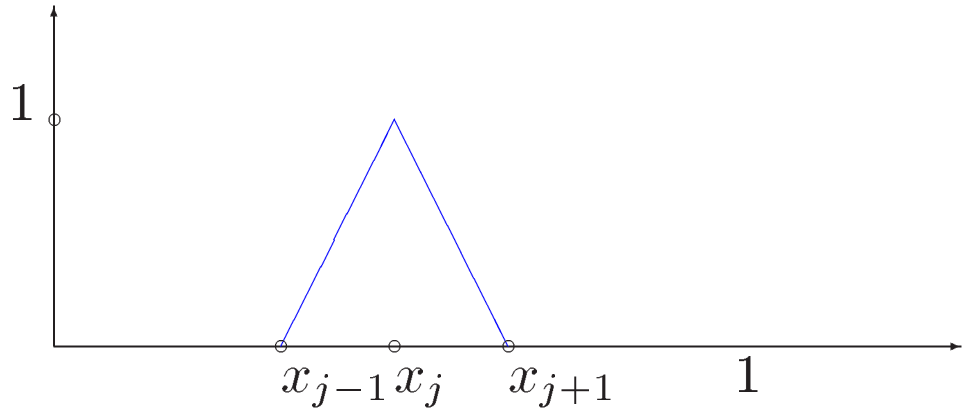
$$
\begin{aligned}
a(u^M, \eta_i) &= a(\sum_{j=1}^{M} d_j \eta_j, \eta_i) \\
&= \sum_{j=1}^{M} d_j a(\eta_j, \eta_i) = \mathcal{F}(\eta_i), \quad i = 1, \ldots, M \\
\Longleftrightarrow \mathbf{A}_M \mathbf{d} &= \mathbf{b} \\
\mathbf{A}_M &= (a(\eta_j, \eta_i))_{i,j=1,\ldots,M}, \qquad \mathbf{b} = (\mathcal{F}(\eta_i))_{i=1}^{M}
\end{aligned}
$$

$\mathbf{A}_M$ symmetrisch, positiv definit

Frage: Wahl der Ansatzräume V_M?

Eine Möglichkeit: finite Elemente
$0 = x_0 < x_1 < \ldots < x_M < x_{M+1} = 1$,
$x_i = ih,\ h = 1/(M+1),\ \eta_i$ stückweise linear mit

$$\eta_j(x_i) = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j, \quad j = 1, \ldots, N \end{cases}$$



$h \rightsquigarrow h/2 \rightsquigarrow h/4 \ldots$
Man kann zeigen:

$$\mathbf{A}_M = \frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & \ldots & \ldots & 0 \\ -1 & 2 & -1 & 0 & \ldots & 0 \\ 0 & -1 & 2 & & & \\ & 0 & & & & \\ & & & & & \\ & & & & 2 & -1 \\ 0 & \ldots & \ldots & 0 & -1 & 2 \end{pmatrix}$$

Im $\mathbb{R}^2$: benutze Folge von Triangulierungen

Vorteil: $\mathbf{A}_M$ sparse $\longrightarrow$ iterative Gleichungslöser

$$
\begin{aligned}
x^{j+1} &= Tx^j + c, \qquad \text{z.B.} \\
x^{j+1} &= x^j - D^{-1}(Ax^j - b), \qquad \mathbf{D} = \text{diag}(A)
\end{aligned}
$$

Nachteil: Kondition von $\mathbf{A}_M = \kappa(\mathbf{A}_M) = \frac{\lambda_{\max}}{\lambda_{\min}} \sim \mathcal{O}(h^{-2})$

Kondition bestimmt

- Konvergenzgeschwindigkeit iterativer Verfahren
- Fehleranfälligkeit

2.2 Vorkonditionierung

Idee: Benutze Wavelets als Basis für das Galerkin–Verfahren $\longmapsto$ Wavelet–Galerkin–Verfahren !
Wähle die Räume $\{V_j\}_{j\geq 0}$ der M.R.A. als Ansatzräume!

$$V_j = \overline{\text{span}\{\varphi(\cdot - l), \psi_{i,j,k},\ k,l \in \mathbb{Z}^d, j \geq 0, i \in I\}}$$

Vorkonditionierung möglich!

$$\mathbf{A}_M\mathbf{d} = \mathbf{b} \rightsquigarrow \mathbf{D}^{-1}\mathbf{A}_M\mathbf{D}^{-1}\mathbf{D}\mathbf{d} = \mathbf{D}^{-1}\mathbf{b}$$

Wähle etwa für die Poisson–Gleichung

$$\begin{aligned} \mathbf{D}_{(j,i,k),(j',i',k')} &= 2^j \delta_{j,j'}\delta_{i,i'}\delta_{k,k'} \Longrightarrow \\ \kappa(\mathbf{D}^{-1}\mathbf{A}_M\mathbf{D}^{-1}) &= \mathcal{O}(1)\ ! \end{aligned}$$

Diagonale Vorkonditionierung ist für eine riesige Klasse von Problemen möglich!

2.3 Matrixkompression

Integralgleichungen

$$\begin{aligned} Au &= F \quad \text{in} \quad \Omega \subset \mathbb{R}^d \\ (Au)(x) &= \int_\Omega K(x,y)u(y)dy \\ |\partial_x^\alpha \partial_y^\beta K(x,y)| &\leq c_{\alpha,\beta}\mathrm{dist}(x,y)^{(-d+2t+|\alpha|+|\beta|)} \qquad (8) \end{aligned}$$

Beispiel: Hilbert–Transformation

$$K(x,y) = \frac{1}{x-y}, \qquad (Au)(x) = \int_{\mathbb{R}} \frac{u(y)}{x-y}dy$$

Galerkin–Verfahren mit finiten Elementen $\Longrightarrow \mathbf{A}_M$ nicht sparse!

Wir benutzen wieder Wavelets! Viele Einträge a priori klein!

Beachte: Wavelets haben verschwindende Momente,

$$\int_{\mathbb{R}} x^\alpha \psi(x)dx = 0, \qquad 0 \leq \alpha \leq N-1$$

Typischer Eintrag:

$$\langle A(\psi_{i,j,k}), \psi_{i',j',k'}\rangle = \int_\Omega \int_\Omega K(x,y)\psi_{i',j',k'}(x)\psi_{i,j,k}(y)dxdy$$
$$= \int_\Omega \int_\Omega (K(x,y) - p(x,y))\psi_{i',j',k'}(x)\psi_{i,j,k}(y)dxdy$$

$p(x,y)$ Polynom

Analog zur Datenkompression: Wähle $p(x,y)$ als Taylor–Polynom $\Longrightarrow$

$$K(x,y) - p(x,y) \sim \sum_{|\alpha|,|\beta|=N} (x - x_{i',j',k'})^\alpha (y - y_{i,j,k})^\beta \partial_x^\alpha \partial_y^\beta K(x_{i',j',k'}, y_{i,j,k})$$

Kombiniere dies mit (8) $\Longrightarrow$

$$2^{-(j'+j)t}|\langle A(\psi_{i,j,k}), \psi_{i',j',k'}\rangle| \leq C_{10}\frac{2^{-|j-j'|\sigma}}{(1 + 2^{\min(j,j')}\mathrm{dist}(\mathrm{supp}\psi_{i',j',k'}, \mathrm{supp}\psi_{i,j,k})^{d+2\tilde{m}+2t}}$$

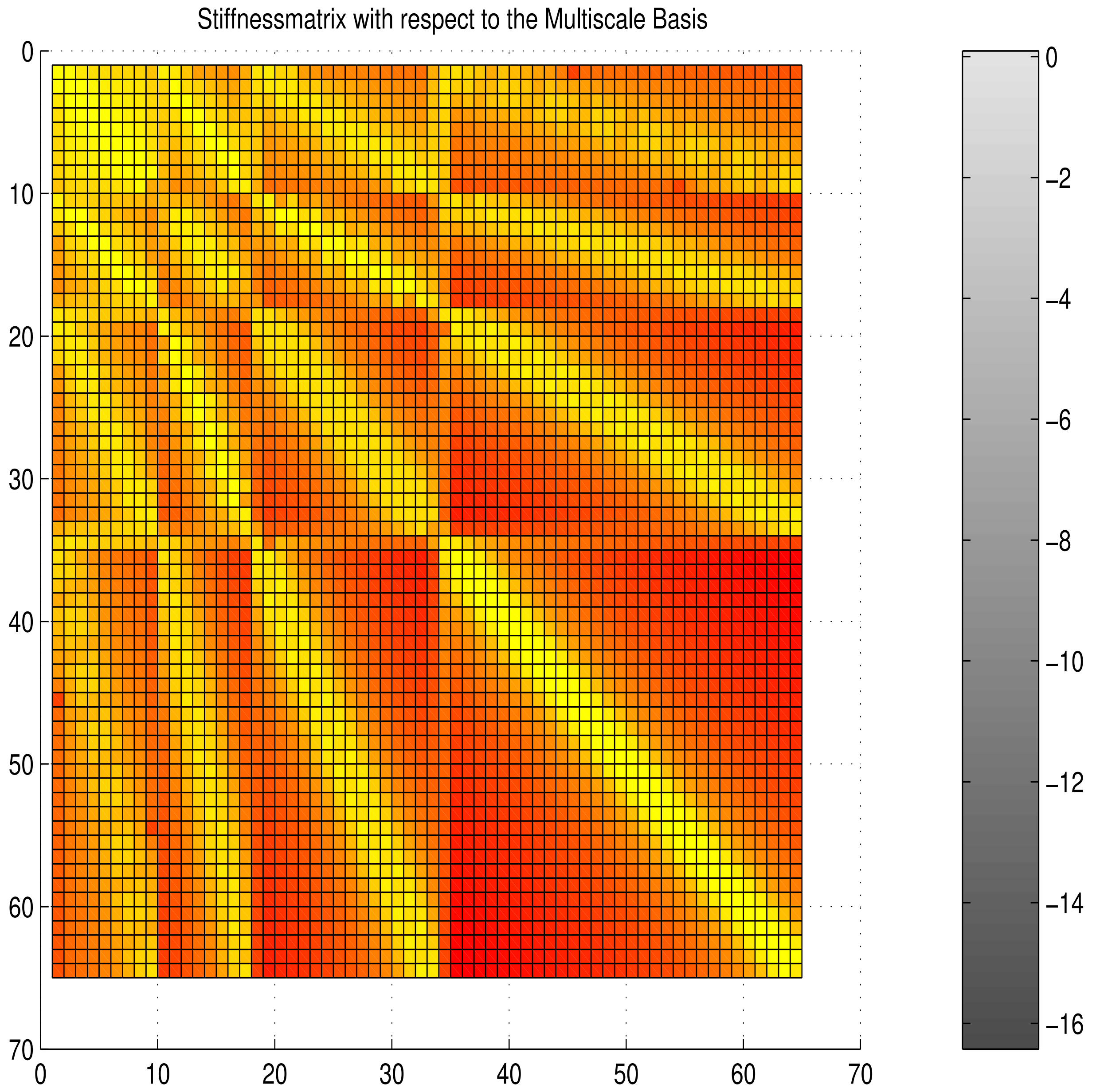
Einträge klein, falls

- Abstand der Skalen groß
- Abstand der Träger groß

Wir schmeißen diese einfach weg!

Rigorose Fehlerabschätzung möglich

Beispiel: Hilbert–Transformation



2.4 Adaptivität

Problem: $\mathbf{A}_N$ kann sehr groß werden!

Möglicher Ausweg:
Adaptive Verfahren: Benutze nicht die vollen Räume V_j, sondern geeignete Teilräume $S_\Lambda \subset V_j$

Aufdatierungsstrategie:

- S_Λ angepaßt an u
- selbststeuernd

lösen	–	abschätzen	–	verfeinern
$\mathbf{A}_\Lambda \mathbf{c}_\Lambda = \mathbf{F}_\Lambda$		$\|u - u_\Lambda\| = ?$ a–posteriori Fehlerschätzer		verfeinere Partition, addiere Funktionen falls notwendig

Man kann zeigen:

Verwendung von Wavelets führt auf adaptive Strategie, die

- tatsächlich konvergiert
- optimale Konvergenzordnung realisiert

Prinzipielle Idee:

- Fehlerschätzer

$$\begin{aligned}
\|u - u_\Lambda\|_{H^1} &\leq C_{11} \|A(u - u_\Lambda)\|_{H^2} \\
&\leq C_{11} \|F - A(u_\Lambda)\|_{H^2} \\
&\leq C_{11} \|r_\Lambda\|_{H^2} \\
&\leq C_{12} \left(\sum_{(i,j,k) \notin \Lambda} \underbrace{2^{-2tj} |\langle r_\Lambda, \psi_{i,j,k} \rangle|^2}_{\delta^2_{i,j,k}} \right)^{1/2}
\end{aligned}$$

$$\delta_{i,j,k} = 2^{-tj} |F_{i,j,k} - \sum_{(i',j',k') \in \Lambda} \langle A\psi_{i',j',k'}, \psi_{i,j,k} \rangle d_{i',j',k'}|,$$

$$F_{i,j,k} = \langle F, \psi_{i,j,k} \rangle$$

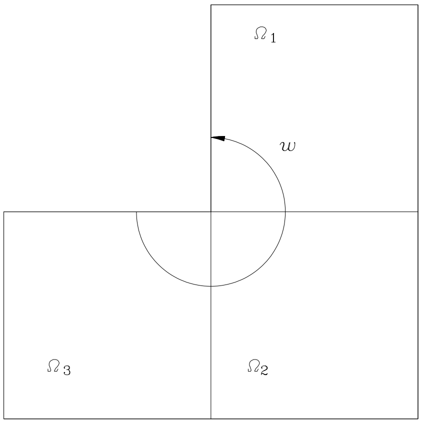
- adaptive Strategie
  Wähle $\tilde{\Lambda}$ so, daß

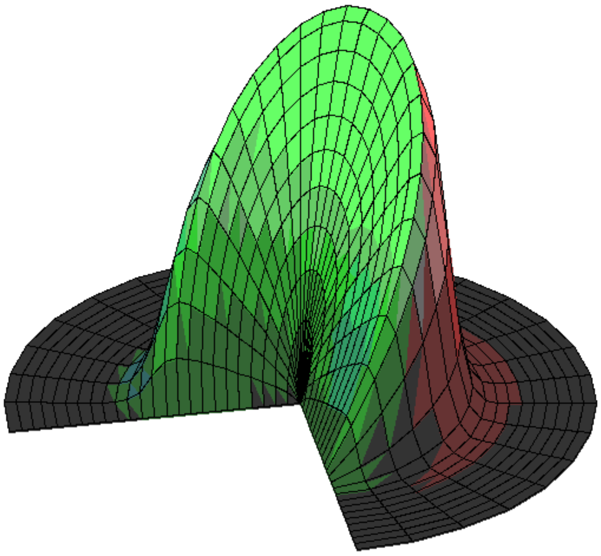
$$\sum_{(i,j,k)\in\tilde{\Lambda}\setminus\Lambda} 2^{-2tj}|\langle r_\Lambda, \psi_{i,j,k}\rangle|^2 \quad \geq \quad \gamma \sum_{(i,j,k)\notin\Lambda} 2^{-2tj}|\langle r_\Lambda, \psi_{i,j,k}\rangle|^2$$

$$\Longrightarrow \|u - u_{\tilde{\Lambda}}\| \quad \leq \quad \kappa\|u - u_\Lambda\|, \qquad \kappa \in (0,1)$$

Beispiel:

$$\begin{aligned} -\triangle u &= F \quad \text{in} \quad \Omega, \\ u|_{\partial\Omega} &= 0 \end{aligned}$$





die rechte Seite