

# **Iterative Fourier–Rekonstruktion**

## **Diplomarbeit**

im Rahmen des Diplomstudiengangs Informatik

vorgelegt von

**Stefan Kunis**

Thema ausgegeben von

**Prof. Dr. Jürgen Prestin**

Betreuer

**Dr. Daniel Potts**

Universität zu Lübeck  
Technisch-Naturwissenschaftliche Fakultät  
Institut für Mathematik

August 2003

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Lübeck, den 29. August 2003

## **Dank**

An dieser Stelle danke ich all jenen, die durch ihre fachliche und persönliche Unterstützung zum Gelingen dieser Arbeit beigetragen haben.

Besonders herzlich danke ich meinen Eltern, sie haben mir dieses Studium durch ihre unschätzbare Unterstützung ermöglicht.

Mein Dank gilt Prof. Dr. J. Prestin für die Betreuung meiner Diplomarbeit. Darüberhinaus danke ich Dr. D. Potts für die intensive Förderung während der Erstellung dieser Arbeit. Von seinem Wissen und seiner Betreuung habe ich während meines gesamten Studiums in besonderer Weise profitiert.



## Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Notation . . . . .	5
2.2	Polynomräume und Abtastung . . . . .	10
2.3	Interpolationsprobleme . . . . .	16
2.4	Diskretes Approximationsproblem . . . . .	22
2.5	Regularisierungsverfahren . . . . .	29
2.6	Zusammenfassung . . . . .	32
<b>3</b>	<b>Synthese</b>	<b>33</b>
3.1	Diskrete Fourier–Transformation für beliebige Knoten . . . . .	33
3.2	Diskrete Polynomtransformation . . . . .	36
3.3	Diskrete Fourier–Transformation auf der Sphäre für beliebige Knoten . . . . .	40
3.4	Zusammenfassung . . . . .	42
<b>4</b>	<b>Analyse</b>	<b>43</b>
4.1	Grundlagen . . . . .	43
4.2	Einfache Verfahren . . . . .	45
4.3	CG–Verfahren . . . . .	48
4.4	Weitere Verfahren . . . . .	52
4.5	Zusammenfassung . . . . .	54
<b>5</b>	<b>Entwurf und Implementierung NFFT1.1</b>	<b>55</b>
5.1	NFFT1.0 und Entwurf . . . . .	55
5.2	Implementierung . . . . .	56
5.3	Technische Dokumentation . . . . .	57
<b>6</b>	<b>Numerische Tests</b>	<b>59</b>
<b>7</b>	<b>Ausblick</b>	<b>63</b>
<b>A</b>	<b>Index der wichtigsten definierten Symbole</b>	<b>65</b>



# 1 Einleitung

Die Auswertung trigonometrischer oder sphärischer Polynome an endlich vielen Knoten ist eine grundlegende Problemstellung der numerischen Mathematik. Die bezüglich einer geeigneten Basis gegebenen Fourier–Koeffizienten werden mit einer „einfachen“ Rechenvorschrift auf die Abtastwerte des durch die Fourier–Koeffizienten definierten Polynoms abgebildet. Wir sprechen in diesem Sinne von einem *direkten Problem*, welches wir als Matrix–Vektor–Multiplikationen interpretieren.

Für trigonometrische Polynome und äquidistante Knoten heißt das direkte Problem (äquidistante) *diskrete Fourier–Transformation* (DFT). Der dafür entwickelte schnelle Algorithmus, siehe [5], wurde als schnelle Fourier–Transformation (engl. fast Fourier transform, FFT) berühmt. Die Beschränkung auf äquidistante Knoten stellt für viele Anwendungen einen Nachteil dar. Die Verallgemeinerung auf beliebige Knoten und darüberhinaus die Erweiterung für sphärische Polynome ist für uns interessant.

Die Entwicklung schneller Algorithmen für die diskreten Fourier–Transformationen für beliebige Knoten, siehe z.B. [11, 45, 40, 39, 25, 36] und die darin enthaltenen Referenzen, ist in den letzten Jahren Gegenstand intensiver Forschung und von essentiellen Interesse. Die Verfügbarkeit von Programmbibliotheken ermöglicht dem Anwender eine einfache Benutzung in seinem speziellen Problemfeld.

Den Schwerpunkt dieser Arbeit bildet die Berechnung von Fourier–Koeffizienten aus gegebenen Abtastwerten zu nicht äquidistanten Knoten. Wir sprechen in diesem Sinne von einem *inversen Problem*, einen Algorithmus zur Lösung in einem geeigneten Sinn bezeichnen wir in dieser Arbeit kurz als *inverse Fourier–Transformation*.

Die wesentlichen Schwierigkeiten liegen bei vielen Anwendungen in der geeigneten Wahl eines Polynomraumes und der Formulierung geeigneter Interpolations– beziehungsweise diskreter Approximationsbedingungen.

Von diesen Bedingungen ausgehend, führen wir die gewichteten und gedämpften Normalgleichungen erster und zweiter Art ein. Die auftretenden Gleichungssysteme sind voll besetzt, besitzen jedoch spezielle Struktur. Mit dem Ausnutzen dieser speziellen Struktur für schnelle Matrix–Vektor–Multiplikationen ist es uns möglich, effiziente iterative Lösungsverfahren für die Gleichungssysteme zu entwickeln. Die hier vorgeschlagenen inversen Fourier–Transformationen sind für eine große Klasse von Problemen geeignet und verallgemeinern sehr erfolgreich eingesetzte Algorithmen, siehe insbesondere [46, 12, 47].

Die Implementierung der Verfahren beschränkt sich auf den Fall der trigonometrischen Polynome. Wir erweitern die frei verfügbare Programmbibliothek NFFT1.0 um die inversen Fourier–Transformationen, ein von Anwendern gewünschtes Leistungsmerkmal.

Die Arbeit ist wie folgt gegliedert: Das Kapitel 2 stellt die im darauf Folgenden benötigten mathematischen Grundlagen vor. Nachdem wir uns in Abschnitt 2.1 mit den verwendeten Schreibweisen für Vektoren, Matrizen und Funktionen vertraut gemacht haben, widmen wir uns in Abschnitt 2.2 der Abtastung von Polynomen. Dabei wiederholen wir insbesondere die regelmäßige Abtastung und dafür bekannte direkte Inversionsformeln für spezielle uni– und multivariate Fälle.

Die Behandlung des allgemeineren, über– beziehungsweise unterbestimmten Gleichungssystems ist schwieriger. Für exakte Abtastwerte eines Polynoms mit bekanntem beschränkten Grad leiten wir in Abschnitt 2.3 die gedämpfte Normalgleichung zweiter Art her. Die gegebenen Daten werden dabei interpoliert. In dem daran anschließenden Abschnitt 2.4 fordern

wir nur eine Approximation an die gegebenen Daten, wobei wir unter Berücksichtigung der speziellen Abtastmenge die gewichtete Normalengleichung erster Art aufstellen.

Abschnitt 2.5 beschäftigt sich kurz mit Regularisierungsverfahren für die formulierten Interpolations- und Approximationsprobleme. Insbesondere motivieren wir die Verwendung von iterativen und mehrstufigen Verfahren und schließen das Kapitel 2 in Abschnitt 2.6 mit einer Darstellung der formulierten Probleme.

In Kapitel 3 wiederholen wir die schnellen Algorithmen für die diskreten Fourier-Transformationen für beliebige Knoten. Besonderen Wert legen wir dabei auf die Darstellung der Fourier-Transformationen als näherungsweise Faktorisierung der Vandermonde-ähnlichen-Matrix in ein Produkt schwach besetzter Matrizen. Wir stellen in Abschnitt 3.1 die Fourier-Transformation für beliebige Knoten, in Abschnitt 3.2 die Polynomtransformation und in Abschnitt 3.3 die Fourier-Transformation auf der Sphäre für beliebige Knoten dar. Das Kapitel 3 wird durch einen Überblick über die verfügbaren schnellen Algorithmen abgeschlossen.

Wir leiten schließlich in Kapitel 4 schnelle iterative Algorithmen zur näherungsweisen Berechnung der inversen Fourier-Transformation für beliebige Knoten her. Nach der Einführung grundlegender Begriffe in Abschnitt 4.1 stellen wir in Abschnitt 4.2 einfache iterative Verfahren vor. Diese Verfahren nutzen die in Kapitel 3 entwickelten schnellen Algorithmen zur Matrix-Vektor-Multiplikation. Die Verallgemeinerungen einfacher Verfahren führen in Abschnitt 4.3 auf die sehr effizienten CG-Typ-Verfahren. Im anschließenden Abschnitt 4.4 ordnen wir die in [11, 9] und [46, 12, 47] vorgeschlagenen Verfahren ein. Wir geben in Abschnitt 4.5 eine kurze Zusammenfassung der vorgeschlagenen Algorithmen.

Das Kapitel 5 widmet sich der Implementierung der entwickelten Algorithmen. Wir schränken uns auf trigonometrische Polynome ein und erweitern das Programmpaket NFFT. Wir gliedern die Weiterentwicklung zur Version NFFT1.1 in den Entwurf, siehe Abschnitt 5.1; die Implementierung, siehe Abschnitt 5.2, und die Dokumentation, siehe Abschnitt 5.3.

Die vorliegende Arbeit wird mit einigen einfachen Beispielen in Kapitel 6 und einem Ausblick auf offene Problemstellungen in Kapitel 7 abgerundet.

## 2 Grundlagen

In diesem Kapitel stellen wir die mathematischen Grundlagen für das weitere Vorgehen dar. Nach der Einführung üblicher Bezeichnungen für Vektoren, Matrizen und Funktionen in Abschnitt 2.1 wenden wir uns in Abschnitt 2.2 Polynomräumen zu. Wir untersuchen insbesondere den Vektorraum der Fourier–Koeffizienten, den Vektorraum der Polynome und Operatoren zwischen diesen Räumen. Daran anschließend formulieren wir das lineare Gleichungssystem, welches sich durch die Abtastung der betrachteten Polynome ergibt. Gesucht sind geeignete Fourier–Koeffizienten zu einem Polynom, welches die gegebenen Abtastwerte in einem geeigneten Sinn approximiert. Wir wiederholen insbesondere die regelmäßige Abtastung und bekannte direkte Inversionsformeln für spezielle uni– und multivariate Fälle.

In Abschnitt 2.3 gehen wir von exakt gegebenen Abtastwerten aus und betrachten Interpolationsprobleme. Wir leiten die gedämpfte Normalengleichung zweiter Art her. Die Dämpfung nutzt dabei die nicht durch die Interpolationsbedingungen festgelegten Freiheitsgrade, um eine glatte Lösung zu erzielen.

Weiterhin betrachten wir in Abschnitt 2.4 Approximationsprobleme, d.h. die gegebenen Abtastwerte werden nur angenähert. Wir formulieren das Approximationsproblem unter Berücksichtigung der speziellen Abtastmenge in naheliegender Weise, leiten die gewichtete Normalengleichung erster Art her und weisen unter einfachen Voraussetzungen die Existenz einer Lösung nach.

Abschnitt 2.5 beschäftigt sich kurz mit Regularisierungsverfahren für die formulierten Interpolations– und Approximationsprobleme. Wir motivieren insbesondere die Verwendung von iterativen und mehrstufigen Verfahren. Dieses Kapitel wird in Abschnitt 2.6 mit einer Gesamtdarstellung der formulierten Probleme abgeschlossen.

### 2.1 Notation

#### Symbole und Schreibweisen

Die zu definierenden Gegenstände sind in der gesamten Arbeit *kursiv* gesetzt. Für die Menge der auftretenden Skalare verwenden wir im Allgemeinen das Symbol  $\mathbb{K}$ . Typischerweise ist  $\mathbb{K}$  die Menge der reellen Zahlen  $\mathbb{R}$  beziehungsweise der komplexen Zahlen  $\mathbb{C}$ . Mit  $\mathbb{R}^+$  bezeichnen wir die nichtnegativen reellen Zahlen. Weiterhin bezeichnet  $\mathbb{Z}$  die ganzen,  $\mathbb{N}_0$  die nichtnegativen ganzen und  $\mathbb{N}$  die natürlichen Zahlen. Für eine komplexe Zahl  $a \in \mathbb{C}$  heißt  $\bar{a}$  konjugiert komplexe Zahl. Die Landau–Notation  $\mathcal{O}$  verwenden wir in der erweiterten Bedeutung, siehe z.B. [36, S. 11].

#### Vektoren und Matrizen

Vektoren und Matrizen sind in kleinen beziehungsweise großen fetten Buchstaben gesetzt. Ein Vektor  $\mathbf{a} \in \mathbb{K}^n$  der Dimension  $n \in \mathbb{N}$  hat die Elemente  $a_l \in \mathbb{K}$ ,  $l = 0, \dots, n-1$ . In den Vektorräumen  $\mathbb{K}^n$  benutzen wir das euklidische Skalarprodukt und die davon induzierte Norm

$$\langle \mathbf{a}, \mathbf{b} \rangle_2 := \sum_{l=0}^{n-1} a_l \bar{b}_l, \quad \|\mathbf{a}\|_2 := \langle \mathbf{a}, \mathbf{a} \rangle_2^{\frac{1}{2}}.$$

Eine Matrix  $\mathbf{A} \in \mathbb{K}^{m \times n}$  der Dimension  $m \times n$ ,  $m, n \in \mathbb{N}$ , hat die Elemente  $a_{k,l} \in \mathbb{K}$ ,  $k = 0, \dots, m-1$ ;  $l = 0, \dots, n-1$ , d.h.,  $\mathbf{A} := (a_{k,l})_{k=0, \dots, m-1; l=0, \dots, n-1}$ . Wir verwenden die Schreib-

weisen  $\mathbf{A}^T$  für die transponierte,  $\overline{\mathbf{A}}$  für die konjugierte und  $\mathbf{A}^H = \overline{\mathbf{A}}^T$  für die adjungierte Matrix. Die Einheitsmatrix ist  $\mathbf{E} := (\delta_{k,l})_{k,l=0,\dots,m-1}$ , wobei wir das Kronecker-Symbol

$$\delta_{k,l} := \begin{cases} 1 & \text{für } k = l \\ 0 & \text{sonst} \end{cases}$$

verwenden. Weiterhin definieren wir für gerade  $m \in \mathbb{N}$  die *Fourier-Matrix* mit

$$\mathbf{F} := \left( e^{-2\pi i \frac{kl}{m}} \right)_{k,l=-\frac{m}{2}, \dots, \frac{m}{2}-1}.$$

Die Einheitsmatrix  $\mathbf{E}$  und die Fourier-Matrix  $\mathbf{F}$  sind quadratisch und tragen keinen Hinweis auf ihre Größe, diese ist immer aus dem Zusammenhang klar.

Für die folgenden Definitionen und Aussagen verweisen wir auf [2, Kapitel 1]. Sei die Matrix  $\mathbf{S} \in \mathbb{K}^{m \times n}$  beliebig, so definieren wir den Bildraum (engl. range)

$$\mathcal{R}(\mathbf{S}) := \left\{ \mathbf{f} = \mathbf{S}\hat{\mathbf{f}} : \hat{\mathbf{f}} \in \mathbb{K}^n \right\}$$

und den Nullraum

$$\mathcal{N}(\mathbf{S}) := \left\{ \hat{\mathbf{f}} \in \mathbb{K}^n : \mathbf{S}\hat{\mathbf{f}} = \mathbf{0} \right\}.$$

Die Dimension des Bildraumes bezeichnen wir mit  $\text{rang}(\mathbf{S})$ . Für das orthogonale Komplement des Nullraumes

$$\mathcal{N}(\mathbf{S})^\perp := \left\{ \hat{\mathbf{g}} \in \mathbb{K}^n : \langle \hat{\mathbf{g}}, \hat{\mathbf{f}} \rangle_2 = 0 \text{ für alle } \hat{\mathbf{f}} \in \mathcal{N}(\mathbf{S}) \right\}$$

gilt die Identität  $\mathcal{N}(\mathbf{S})^\perp = \mathcal{R}(\mathbf{S}^H)$ .

Weiterhin existieren zu beliebigen Matrizen  $\mathbf{S} \in \mathbb{K}^{m \times n}$  unitäre Matrizen  $\mathbf{U} \in \mathbb{C}^{m \times m}$ ,  $\mathbf{V} \in \mathbb{C}^{n \times n}$  und eine Diagonalmatrix  $\Sigma = \text{diag}(\sigma_0, \dots, \sigma_{\text{rang}(\mathbf{S})-1}) \in \mathbb{R}^{\text{rang}(\mathbf{S}) \times \text{rang}(\mathbf{S})}$  mit  $\sigma_l > 0$ ,  $l = 0, \dots, \text{rang}(\mathbf{S}) - 1$ , so dass

$$\mathbf{S} = \mathbf{U} \begin{pmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{V}^H \in \mathbb{K}^{m \times n}. \quad (2.1)$$

Die Zerlegung (2.1) heißt Singulärwertzerlegung, mit ihrer Hilfe ist die Moore–Penrose–Pseudo-inverse

$$\mathbf{S}^\dagger := \mathbf{V} \begin{pmatrix} \Sigma^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \mathbf{U}^H \in \mathbb{K}^{n \times m}$$

definiert. Ihre wichtigsten Eigenschaften fassen wir in folgendem Lemma 2.1 zusammen.

**Lemma 2.1.** Sei die Matrix  $\mathbf{S} \in \mathbb{K}^{m \times n}$  und der Vektor  $\mathbf{f} \in \mathbb{K}^m$  beliebig, so gilt für den Vektor  $\mathbf{f}^\dagger = \mathbf{S}^\dagger \mathbf{f}$  die Abschätzung

$$\|\mathbf{S}\mathbf{f}^\dagger - \mathbf{f}\|_2 \leq \|\mathbf{S}\hat{\mathbf{f}} - \mathbf{f}\|_2 \quad \text{für alle } \hat{\mathbf{f}} \in \mathbb{K}^n.$$

Weiterhin hat der Vektor  $\mathbf{f}^\dagger$  kleinste Norm, d.h.,

$$\|\mathbf{f}^\dagger\|_2 \leq \|\hat{\mathbf{f}}\|_2 \quad \text{für alle } \hat{\mathbf{f}} \text{ mit } \|\mathbf{S}\hat{\mathbf{f}} - \mathbf{f}\|_2 = \|\mathbf{S}\mathbf{f}^\dagger - \mathbf{f}\|_2.$$

*Beweis.* Siehe z.B. [2, Kapitel 1]. ■

Der Vektor  $\mathbf{f}^\dagger$  ist also Lösung des *Approximationsproblems*

$$\|\mathbf{S}\hat{\mathbf{f}} - \mathbf{f}\|_2 \xrightarrow{\hat{\mathbf{f}}} \min \quad (2.2)$$

und unter den Lösungen des Approximationsproblems ist der Vektor  $\mathbf{f}^\dagger$  die Lösung des *Minimierungsproblems*

$$\|\hat{\mathbf{f}}\|_2 \xrightarrow{\hat{\mathbf{f}}} \min. \quad (2.3)$$

Die so eindeutig bestimmte Lösung  $\mathbf{f}^\dagger$  nennen wir *Minimallösung*. Wird bei dem Approximationsproblem (2.2) das Minimum  $\|\mathbf{S}\hat{\mathbf{f}} - \mathbf{f}\|_2 = 0$  angenommen, so hat die Gleichung  $\mathbf{S}\hat{\mathbf{f}} = \mathbf{f}$  mindestens eine Lösung und wir nennen die Gleichung *konsistent*.

Weiterhin betrachten wir für reguläre Matrizen  $\mathbf{T} \in \mathbb{K}^{n \times n}$  die Neumann-Reihe, mit deren Hilfe die Inverse  $\mathbf{T}^{-1}$  approximiert werden kann.

**Lemma 2.2.** Sei die Matrix  $\mathbf{T} \in \mathbb{K}^{n \times n}$  regulär mit  $\|\mathbf{E} - \mathbf{T}\|_2 < 1$ , so gilt

$$\mathbf{T}^{-1} = \sum_{l=0}^{\infty} (\mathbf{E} - \mathbf{T})^l, \quad (2.4)$$

für die  $l_{\max}$ -te Partialsumme gilt die a-priori-Fehlerabschätzung

$$\|\mathbf{T}^{-1} - \sum_{l=0}^{l_{\max}} (\mathbf{E} - \mathbf{T})^l\|_2 \leq \|\mathbf{E} - \mathbf{T}\|_2^{l_{\max}+1} \frac{1 + \|\mathbf{E} - \mathbf{T}\|_2}{1 - \|\mathbf{E} - \mathbf{T}\|_2}.$$

*Beweis.* Siehe z.B. [28, S. 193ff]. ■

## Funktionen

Für einen *Urbildraum*  $\mathbb{D}$ , hier ein Intervall, ein Torus oder eine Sphäre, und einen *skalaren Bildraum*  $\mathbb{K}$  betrachten wir die quadratintegrierbaren Funktionen  $g : \mathbb{D} \rightarrow \mathbb{K}$ . Diese Funktionen bilden einen *Hilbertraum*  $L = L^2(\mathbb{D}, \mathbb{K})$  mit dem Skalarprodukt  $\langle \cdot, \cdot \rangle_L$  und der induzierten Norm  $\|\cdot\|_L := \langle \cdot, \cdot \rangle_L^{\frac{1}{2}}$ .

Für einen *Polynomgrad*  $N \in \mathbb{N}$  interessieren wir uns für den *Polynomraum*  $L_N \subset L$ , die *Dimension des Polynomraumes* bezeichnen wir mit  $N_L \in \mathbb{N}$ . Wir indizieren eine Basis eines Polynomraumes  $L_N$  im Allgemeinen mit  $k = 0, \dots, N_L - 1$ . Ist das Funktionensystem strukturiert, so verwenden wir auch Multiindizes. Diese werden der Übersichtlichkeit wegen verwendet, müssten aber strenggenommen mit einer Indexfunktion in die Menge der natürlichen Zahlen abgebildet werden.

## Torus

Wir betrachten den *d-Torus*  $\mathbb{T}^d := \mathbb{R}^d / \mathbb{Z}^d$  mit der darauf definierten Metrik

$$\text{dist}_{\mathbb{T}^d}(\mathbf{x}, \mathbf{y}) := \min_{\mathbf{j} \in \mathbb{Z}^d} \|(\mathbf{x} + \mathbf{j}) - \mathbf{y}\|_2.$$

Die quadratintegrierbaren Funktionen  $f, g : \mathbb{T}^d \rightarrow \mathbb{K}$  bilden mit dem Skalarprodukt

$$\langle f, g \rangle_L := \int_{\mathbb{T}^d} f(\mathbf{x}) \overline{g(\mathbf{x})} \, d\mathbf{x}$$

den Hilbertraum  $L = L^2(\mathbb{T}^d, \mathbb{K})$ .

Mit Hilfe der Indexmenge  $I_N^d := \left\{ \mathbf{k} = (k_1, \dots, k_d)^T \in \mathbb{Z}^d : -\frac{N}{2} \leq k_c < \frac{N}{2}, c = 1, \dots, d \right\}$  definieren wir den Unterraum der *trigonometrischen Polynome*

$$L_N := \text{span} \left( e^{-2\pi i \mathbf{k}^T \cdot} : \mathbf{k} \in I_N^d \right).$$

Die Dimension dieses Polynomraumes ist  $N^d$ . Die Funktionen  $e^{-2\pi i \mathbf{k}^T \cdot}$ ,  $\mathbf{k} \in I_N^d$  bilden die Standardorthonormalbasis des Hilbertraumes  $L$ .

## 2–Sphäre

Wir betrachten die 2–Sphäre  $\mathbb{S}^2 := \{ \mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\|_2 = 1 \}$  mit der Metrik

$$\text{dist}_{\mathbb{S}^2}(\mathbf{x}, \mathbf{y}) := \arccos(\mathbf{x}^T \mathbf{y}).$$

Die quadratintegrierbaren Funktionen  $f, g : \mathbb{S}^2 \rightarrow \mathbb{K}$  bilden mit dem Skalarprodukt

$$\begin{aligned} \langle f, g \rangle_L &:= \int_{\mathbb{S}^2} f(\mathbf{x}) \overline{g(\mathbf{x})} \, d\omega_{\mathbb{S}^2}(\mathbf{x}) \\ &= \int_0^{2\pi} \int_0^\pi f(\theta, \phi) \overline{g(\theta, \phi)} \sin \theta \, d\theta \, d\phi \end{aligned}$$

den Hilbertraum  $L = L^2(\mathbb{S}^2, \mathbb{K})$ .

Mit Hilfe der Indexmenge  $J_N^2 := \left\{ \mathbf{k} = (k, n)^T \in \mathbb{N}_0 \times \mathbb{Z} : |n| \leq k < N \right\}$  definieren wir den Unterraum der *sphärischen Polynome* (Kugelflächenfunktionen, engl. Spherical Harmonics)

$$L_N := \text{span} (Y_{\mathbf{k}} : \mathbf{k} \in J_N^2).$$

Die Dimension dieses Polynomraumes ist  $N^2$ . Weiterhin ist der Polynomraum  $L_N$  genau der Raum der (algebraischen) Polynome in drei Variablen mit Gesamtgrad kleiner als  $N$  eingeschränkt auf die 2–Sphäre.

Im Folgenden geben wir die Standardorthonormalbasis  $(Y_{\mathbf{k}})_{\mathbf{k} \in J_\infty^2}$  des Hilbertraumes  $L$  an, siehe z.B. [4]. Seien zunächst die Legendre–Polynome

$$P_k(t) := \frac{1}{2^k k!} \frac{d^k}{dt^k} (t^2 - 1)^k, \quad k \in \mathbb{N}_0$$

und die zugeordneten Legendre–Funktionen

$$P_k^{|n|}(t) := \left( \frac{(k - |n|)!}{(k + |n|)!} \right)^{\frac{1}{2}} (1 - t^2)^{\frac{|n|}{2}} \frac{d^{|n|}}{dt^{|n|}} P_k(t), \quad (k, n)^T \in J_\infty^2$$

definiert, so bilden die Funktionen

$$Y_{\mathbf{k}}(\mathbf{x}) := Y_k^n(\theta, \phi) = \left( \frac{2k + 1}{4\pi} \right)^{\frac{1}{2}} P_k^{|n|}(\cos(\theta)) e^{in\phi}, \quad \mathbf{k} \in J_\infty^2$$

die Standardorthonormalbasis des Hilbertraumes  $L$ .

**Reelles Intervall**

Weiterhin betrachten wir die quadratintegrierbaren Funktionen  $f : [-1, 1] \rightarrow \mathbb{K}$  mit dem Skalarprodukt

$$\langle f, g \rangle_L := \int_{-1}^1 f(x) \overline{g(x)} \omega(x) \, dx,$$

wobei  $\omega : [-1, 1] \rightarrow \mathbb{R}$  eine nichtnegative Gewichtsfunktion ist. Wir erhalten den Hilbertraum  $L = L_\omega^2([-1, 1], \mathbb{K})$ . Im Besonderen interessieren wir uns für den  $N$ -dimensionalen Unterraum der (*algebraischen*) *Polynome*

$$L_N := \text{span} \left( \cdot^k : k = 0, \dots, N-1 \right).$$

## 2.2 Polynomräume und Abtastung

Sei  $L_N \subset L$  einer der oben beschriebenen Polynomräume mit einem Vektor der Basiselemente

$$\Phi := (\phi_k)_{k=0, \dots, N_L-1}^T = (\phi_0, \phi_1, \dots, \phi_{N_L-1}),$$

so hat jedes Polynom  $f \in L_N$  mit den *Fourier-Koeffizienten*  $\hat{f}_k \in \mathbb{K}$  die eindeutige Darstellung

$$f = \sum_{k=0}^{N_L-1} \hat{f}_k \phi_k. \quad (2.5)$$

Den Vektor  $\hat{\mathbf{f}} = \left( \hat{f}_k \right)_{k=0, \dots, N_L-1}$  nennen wir *Vektor der Fourier-Koeffizienten*.

### Synthese

Den durch Gleichung (2.5) gegebenen linearen Operator

$$\mathcal{S} : \mathbb{K}^{N_L} \rightarrow L_N, \hat{\mathbf{f}} \mapsto f = \mathcal{S}\hat{\mathbf{f}} \quad (2.6)$$

nennen wir *Synthese*. Die Synthese bildet den Vektor der Fourier-Koeffizienten  $\hat{\mathbf{f}}$  bezüglich des Vektors der Basiselemente  $\Phi$  auf das Polynom  $f \in L_N$  ab. Weiterhin betrachten wir für Funktionen  $g \in L$  den adjungierten Operator  $\mathcal{S}^H : L \rightarrow \mathbb{K}^{N_L}$  mit

$$\mathcal{S}^H g := (\langle g, \phi_k \rangle_L)_{k=0, \dots, N_L-1}.$$

**Lemma 2.3.** Für beliebige Vektoren  $\hat{\mathbf{f}} \in \mathbb{K}^{N_L}$  und Funktionen  $g \in L$  gilt

$$\langle \hat{\mathbf{f}}, \mathcal{S}^H g \rangle_2 = \langle \mathcal{S}\hat{\mathbf{f}}, g \rangle_L.$$

*Beweis.* Mit Hilfe der Symmetrie und der Linearität der Skalarprodukte sehen wir

$$\langle \hat{\mathbf{f}}, \mathcal{S}^H g \rangle_2 = \sum_{k=0}^{N_L-1} \hat{f}_k \overline{\langle g, \phi_k \rangle_L} = \sum_{k=0}^{N_L-1} \hat{f}_k \langle \phi_k, g \rangle_L = \left\langle \sum_{k=0}^{N_L-1} \hat{f}_k \phi_k, g \right\rangle_L = \langle \mathcal{S}\hat{\mathbf{f}}, g \rangle_L. \quad \blacksquare$$

### Normalengleichung, Gram-Matrix

Aus Gleichung (2.5) ergeben sich durch die Skalarmultiplikationen mit den Basiselementen  $\phi_{k_2}$ ,  $k_2 = 0, \dots, N_L - 1$ , die Normalengleichungen

$$\sum_{k_1=0}^{N_L-1} \hat{f}_{k_1} \langle \phi_{k_1}, \phi_{k_2} \rangle_L = \langle f, \phi_{k_2} \rangle_L, \quad k_2 = 0, \dots, N_L - 1. \quad (2.7)$$

Wenden wir den adjungierten Operator  $\mathcal{S}^H$  auf das Polynom  $f$  an, so erhalten wir mit Definition (2.6) die Gleichung

$$\mathcal{S}^H \mathcal{S}\hat{\mathbf{f}} = \mathcal{S}^H f.$$

Die Gram-Matrix  $G := S^H S$  hat die explizite Darstellung

$$\mathbf{G} = \begin{pmatrix} \langle \phi_0, \phi_0 \rangle_L & \cdots & \langle \phi_{N_L-1}, \phi_0 \rangle_L \\ \vdots & & \vdots \\ \langle \phi_0, \phi_{N_L-1} \rangle_L & \cdots & \langle \phi_{N_L-1}, \phi_{N_L-1} \rangle_L \end{pmatrix} \in \mathbb{K}^{N_L \times N_L}.$$

Mit ihrer Hilfe erhalten wir analog zu den Gleichungen (2.7) die Normalengleichung in Matrixschreibweise

$$\mathbf{G} \hat{\mathbf{f}} = \mathbf{S}^H f. \quad (2.8)$$

Die Gram-Matrix  $\mathbf{G}$  ist stets hermitesch und positiv definit, siehe zum Beispiel [20, S. 282f]. Hat die Matrix  $\mathbf{G}$  Diagonalgestalt, so heißt die Basis  $\Phi$  orthogonal, gilt sogar  $\mathbf{G} = \mathbf{E}$ , so heißt  $\Phi$  orthonormal. Allgemein erhalten wir für beliebige Polynome  $f \in L_N$  eine Parseval-Gleichung der Form

$$\|f\|_L^2 = \langle f, f \rangle_L = \hat{\mathbf{f}}^H \mathbf{G} \hat{\mathbf{f}}. \quad (2.9)$$

### Analyse

Unter der *Analyse* eines Polynoms  $f \in L_N$  verstehen wir den linearen Operator, der dem gegebenen Polynom  $f \in L_N$  seinen Vektor der Fourier-Koeffizienten  $\hat{\mathbf{f}}$  bezüglich  $\Phi$  zuordnet. Wir können die Analyse

$$\mathcal{A} : L_N \rightarrow \mathbb{K}^{N_L}, f \mapsto \hat{\mathbf{f}} = \mathcal{A}f$$

mit der Normalengleichung (2.8) durch

$$\mathcal{A}f = \mathbf{G}^{-1} \mathbf{S}^H f \quad (2.10)$$

für alle  $f \in L_N$  ausdrücken. Für Orthonormalbasen vereinfacht sich dies weiter zur Identität  $\mathcal{A} = \mathbf{S}^H$ .

### Orthogonalprojektion

Sei die Funktion  $g \in L$  beliebig, so hat die „kontinuierliche“ Approximationsaufgabe

$$\|f - g\|_L \xrightarrow{f} \min$$

eine eindeutige Lösung  $f \in L_N$ , welche durch

$$\langle f - g, h \rangle_L = 0 \quad \text{für alle } h \in L_N$$

charakterisiert ist, diese heißt Orthogonalprojektion der Funktion  $g \in L$  in den Polynomraum  $L_N$  bezüglich des Skalarproduktes  $\langle \cdot, \cdot \rangle_L$ .

**Lemma 2.4.** Die Abbildung

$$\mathcal{P} : L \rightarrow L_N, g \mapsto \mathcal{S} \mathcal{A} g = \mathbf{S} \mathbf{G}^{-1} \mathbf{S}^H g$$

ist der *orthogonale Projektor* vom Funktionenraum  $L$  in den Polynomraum  $L_N$  bezüglich des Skalarproduktes  $\langle \cdot, \cdot \rangle_L$ . Dabei sei Gleichung (2.10) analog auf die Funktion  $g \in L$  angewandt.

*Beweis.* Für beliebige Polynome  $f \in L_N$  gilt  $\mathcal{P}f = f$  und somit  $\mathcal{P}^2 = \mathcal{P}$ , d.h.,  $\mathcal{P}$  ist Projektor. Weiterhin gilt mit Lemma 2.3 und den Eigenschaften der Gram-Matrix  $\mathbf{G}$  für beliebige Funktionen  $g, h \in L$ , dass  $\langle g, \mathcal{P}h \rangle_L = \langle \mathcal{P}g, h \rangle_L$ , d.h.,  $\mathcal{P}$  ist selbstadjungiert. In [23, S.481] wird gezeigt, dass jeder selbstadjungierte Projektor bereits der orthogonale Projektor ist. ■

### Abtastung

In einem diskreten Modell kennen wir eine Funktion  $g \in L$  im Allgemeinen nur an  $M \in \mathbb{N}$  vielen Stellen, den *Abtaststellen*, auch *Knoten*,  $\mathbf{x}_j \in \mathbb{D}$ ,  $j = 0, \dots, M-1$ . Die Gesamtheit der Abtaststellen bezeichnen wir als *Abtastmenge* (engl. sampling set)  $\mathbf{X} = (\mathbf{x}_j)_{j=0, \dots, M-1}$ . Die Notation als Vektor ist insbesondere notwendig, da die  $\mathbf{x}_j$  nicht notwendigerweise paarweise verschieden sind. Die Werte einer Funktion  $g$  an den Abtaststellen sind die *Abtast-* beziehungsweise *Funktionswerte*  $g_j = g(\mathbf{x}_j)$ . Die *ideale Abtastung*

$$\mathcal{D} : L \rightarrow \mathbb{K}^M, g \mapsto \mathbf{g} = \mathcal{D}g$$

ordnet einer genügend glatten Funktion  $g \in L$  den *Vektor der Abtastwerte*  $\mathbf{g} \in \mathbb{K}^M$  zu, dieser hat die *abgetastete Energie*  $\|\mathbf{g}\|_2^2$ . Selbst bei glatten Funktionen kann dabei ein Informationsverlust auftreten, siehe auch Lemma 2.5. Im Folgenden werden wir daher die Abtastung der Polynome  $f \in L_N$  betrachten, beziehungsweise annehmen, dass der Vektor der Abtastwerte  $\mathbf{f}$  zu einem Polynom  $f \in L_N$  gehört.

Mit der *Vandermonde-ähnlichen-Matrix* (engl. Vandermonde-like-matrix)

$$\begin{aligned} \mathbf{S} &:= \mathcal{D}\mathcal{S} \\ &= \begin{pmatrix} \phi_0(\mathbf{x}_0) & \dots & \phi_{N_L-1}(\mathbf{x}_0) \\ \vdots & & \vdots \\ \phi_0(\mathbf{x}_{M-1}) & \dots & \phi_{N_L-1}(\mathbf{x}_{M-1}) \end{pmatrix} \in \mathbb{K}^{M \times N_L} \end{aligned}$$

erhalten wir analog zur Synthese, siehe Gleichung (2.6), das lineare Gleichungssystem

$$\mathbf{S}\hat{\mathbf{f}} = \mathbf{f}. \quad (2.11)$$

Das Lösen dieses Gleichungssystems in einem geeigneten Sinn ist das zentrale Problem dieser Arbeit. Hat die Vandermonde-ähnliche-Matrix  $\mathbf{S}$  vollen Spaltenrang  $\text{rang}(\mathbf{S}) = N_L$ , so ist notwendigerweise die Anzahl der Abtaststellen mindestens so groß wie die Dimension des Polynomraumes, d.h.,  $M \geq N_L$ . Die Umkehrung gilt im Allgemeinen nicht. Voller Spaltenrang der Vandermonde-ähnliche-Matrix  $\mathbf{S}$  ist äquivalent mit der Bedingung  $\mathbf{f} = \mathbf{0} \Rightarrow f = 0$  für beliebige Polynome  $f \in L_N$  und deren Abtastwerte  $\mathbf{f} = \mathcal{D}f$ .

### Spezielle Abtastgeometrien

Wir betrachten im Folgenden den Fall, dass die Anzahl und Lage der Knoten  $\mathbf{x}_j$  frei bestimmbar ist. Die unter gewissen Voraussetzungen konstruierbaren Newton-Cotes-Quadraturen sollen hier nicht untersucht werden, da die Gewichte  $\omega_j \in \mathbb{R}$ ,  $j = 0, \dots, M-1$ , d.h. die bestimmten

Integrale der Fundamentalinterpolanten zu den Knoten  $\mathbf{x}_j$ , schwierig (numerisch) zu berechnen wären.

Für einen gegebenen Polynomgrad  $N$  werden die Anzahl der Knoten  $M$ , die Knoten  $\mathbf{x}_j$  und die Gewichte  $\omega_j \in \mathbb{R}^+$ ,  $j = 0, \dots, M-1$ , derart konstruiert, dass für Polynome  $h$  bis zu einem gewissen Grad die Identität

$$\int_{\mathbb{D}} h(\mathbf{x}) \, d\omega_{\mathbb{D}}(\mathbf{x}) = \sum_{j=0}^{M-1} \omega_j h(\mathbf{x}_j)$$

gilt. Derartige Quadraturformeln sind beispielsweise die Gauß–Quadratur und die Clenshaw–Curtis–Quadratur, siehe auch [20, Kapitel VII,IX], [40].

Berechnen wir mit einer genügend genauen Quadraturformel das Skalarprodukt  $\langle f, g \rangle_L$  für  $f, g \in L_N$  allein aus den Vektoren der Abtastwerte  $\mathbf{f} = \mathcal{D}f$ ,  $\mathbf{g} = \mathcal{D}g$ , so erhalten wir

$$\langle f, g \rangle_L = \langle \mathbf{f}, \mathbf{g} \rangle_{\mathbf{W}},$$

wobei

$$\langle \mathbf{f}, \mathbf{g} \rangle_{\mathbf{W}} := \mathbf{g}^H \mathbf{W} \mathbf{f} = \sum_{j=0}^{M-1} \omega_j f_j \bar{g}_j.$$

Die *Gewichtsmatrix*  $\mathbf{W}$  sei dabei  $\mathbf{W} := \text{diag}(\boldsymbol{\omega})$ ,  $\boldsymbol{\omega} = (\omega_j)_{j=0, \dots, M-1}$ ,  $\omega_j \in \mathbb{R}^+$ . Anschaulich erhält diese Art der Abtastung die Energie für beliebige Polynome  $f \in L_N$ , d.h.,

$$\|f\|_L^2 = \|\mathbf{f}\|_{\mathbf{W}}^2.$$

Berechnen wir  $\langle f, \phi_k \rangle_L = \langle \mathbf{f}, \boldsymbol{\phi}_k \rangle_{\mathbf{W}}$  und  $\langle \phi_{k_1}, \phi_{k_2} \rangle_L = \langle \boldsymbol{\phi}_{k_1}, \boldsymbol{\phi}_{k_2} \rangle_{\mathbf{W}}$ , so gilt

$$\mathcal{S}^H f = \mathcal{S}^H \mathbf{W} \mathbf{f} \text{ und } \mathbf{G} = \mathcal{S}^H \mathcal{S} = \mathcal{S}^H \mathbf{W} \mathcal{S}.$$

Als diskretes Analogon zu Gleichung (2.8) erhalten wir

$$\mathcal{S}^H \mathbf{W} \mathcal{S} \hat{\mathbf{f}} = \mathcal{S}^H \mathbf{W} \mathbf{f},$$

für Orthonormalbasen vereinfacht sich dies analog zur Analyse, siehe Gleichung (2.10), zu

$$\hat{\mathbf{f}} = \mathcal{S}^H \mathbf{W} \mathbf{f}. \quad (2.12)$$

Die Spezialfälle der Gleichung (2.12) sind die Diskrete Fourier–Transformation und die Diskrete Fourier–Transformation auf der Sphäre, siehe z.B. [20, Kapitel VII,IX], [40]. Die Berechnung der Abtastwerte aus den Fourier–Koeffizienten eines Polynoms mithilfe der Gleichung (2.11) wird häufig als (inverse) Diskrete Fourier–Transformation (auf der Sphäre) bezeichnet.

### Aliaseffekte

Wir betrachten im Folgenden die Auswirkungen, falls der tatsächliche Polynomgrad unterschätzt wird.

**Lemma 2.5.** Sei  $\Phi_L := (\phi_k)_{k \in \mathbb{N}_0}^T$  eine Folge orthonormaler Basiselemente von  $L$  und die Fourier-Reihe

$$\sum_{k \in \mathbb{N}_0} \hat{g}_k \phi_k(\mathbf{x})$$

konvergiere gleichmäßig und absolut gegen die Funktion  $g \in L$ . Weiterhin sei das Polynom  $f = \mathcal{P}g \in L_N$  die in Lemma 2.4 gegebene Orthogonalprojektion und wir berechnen analog zu Gleichung (2.12)

$$\hat{\mathbf{h}} := \mathbf{S}^H \mathbf{W} \mathbf{g},$$

dann gilt

$$\hat{h}_k = \hat{f}_k + \sum_{l \geq N_L} \hat{g}_l \langle \phi_l, \phi_k \rangle_{\mathbf{W}}, \quad k = 0, \dots, N_L - 1. \quad (2.13)$$

*Beweis.* Für die Elemente des Vektors  $\hat{\mathbf{h}}$  gilt

$$\hat{h}_k = \langle \mathbf{g}, \phi_k \rangle_{\mathbf{W}} = \left\langle \sum_{l=0}^{N_L-1} \hat{g}_l \phi_l, \phi_k \right\rangle_{\mathbf{W}} + \sum_{l \geq N_L} \hat{g}_l \langle \phi_l, \phi_k \rangle_{\mathbf{W}} = \hat{f}_k + \sum_{l \geq N_L} \hat{g}_l \langle \phi_l, \phi_k \rangle_{\mathbf{W}}.$$

■

Die Anteile unter dem Summenzeichen in (2.13) nennen wir Aliaskomponenten. Im folgenden Beispiel 2.6 wird für den Spezialfall des 1-Torus  $\mathbb{T}^1$  sogar gezeigt, dass als Aliaskomponenten nur „Vielfache der Grundschwingung“ auftreten. Eine ähnliche Aussage für die sphärischen Polynome beziehungsweise die orthogonalen Polynome auf dem Intervall  $[-1, 1]$  ist uns nicht bekannt.

**Beispiel 2.6.** (1-Torus  $\mathbb{T}^1$ )

Mit  $\Phi_L = (e^{-2\pi i k \cdot})_{k \in \mathbb{Z}}^T$  sei eine Folge orthogonaler Basiselemente von  $L$  gegeben. Seien weiterhin die Knoten  $x_j = \frac{j}{N}$ ,  $j \in I_N^1$ , und die Gewichtsmatrix  $\mathbf{W} = \frac{1}{N} \mathbf{E}$  gewählt, so gilt

$$\hat{h}_k = \hat{f}_k + \sum_{l \geq N} \hat{g}_l \sum_{j \in I_N^1} e^{-2\pi i l \frac{j}{N}} \frac{1}{N} e^{2\pi i k \frac{j}{N}} = \hat{f}_k + \sum_{r \in \mathbb{Z} \setminus \{0\}} \hat{g}_{k+rN}, \quad k \in I_N^1.$$

□

## Separation der Variablen

Haben Abtastmengen und Basisfunktionen Tensorproduktstruktur, so kann das  $d$ -variante Problem in geschachtelte univariate Probleme zerlegt werden. Die Vandermonde-ähnliche-Matrix  $\mathbf{S}$  kann mithilfe des Kroneckerproduktes, siehe z.B. [20, S. 702ff], in der Form

$$\mathbf{S} = \mathbf{S}_1 \otimes \mathbf{S}_2 \otimes \dots \otimes \mathbf{S}_d$$

zerlegt werden. Die Matrizen  $\mathbf{S}_c$ ,  $c = 1, \dots, d$ , sind die zu einem univariaten Problem gehörenden Vandermonde-ähnlichen-Matrizen. Algorithmen, welche derartige Faktorisierung nutzen, werden auch als Zeilen-Spalten-Algorithmus bezeichnet. Das folgende Beispiel 2.7 ist eine direkte Anwendung der dargestellten Idee, das Beispiel 2.8 zeigt eine Verallgemeinerung.

**Beispiel 2.7.** ( $d$ -Torus  $\mathbb{T}^d$ )

Wählen wir die äquidistanten Knoten  $\mathbf{x}_j = \frac{1}{N}\mathbf{j}$ ,  $\mathbf{j} \in I_N^d$ , so besitzt die Vandermonde-ähnliche-Matrix  $\mathbf{S}$  die Faktorisierung

$$\mathbf{S} = \left( e^{-2\pi i \mathbf{k}^T \mathbf{j} / N} \right)_{\mathbf{j}, \mathbf{k} \in I_N^d} = \bigotimes_{c=1}^d \mathbf{F}.$$

□

**Beispiel 2.8.** (2-Sphäre  $\mathbb{S}^2$ )

Lassen sich die Knoten  $(\theta_j, \phi_j)$ ,  $j = 0, \dots, M-1$ , mit

$$\left( \theta_{j_1}, \phi_{(j_1, j_2)^T} \right)_{j_1=0, \dots, M_1-1; j_2=0, \dots, M_2, j_1-1} \quad \text{indizieren, wobei } M = \sum_{j_1=0}^{M_1-1} M_{2, j_1},$$

so gilt für den Vektor  $\mathbf{f} = \mathbf{S}\hat{\mathbf{f}}$  in Multiindexschreibweise

$$\begin{aligned} f_{(j_1, j_2)^T} &= f\left(\theta_{j_1}, \phi_{(j_1, j_2)^T}\right) \\ &= \sum_{\mathbf{k} \in J_N^2} \hat{f}_{\mathbf{k}} Y_{\mathbf{k}}\left(\theta_{j_1}, \phi_{(j_1, j_2)^T}\right) \\ &= \sum_{n=-(N-1)}^{N-1} \left( \sum_{k=|n|}^{N-1} \hat{f}_k \left(\frac{2n+1}{4\pi}\right)^{\frac{1}{2}} P_k^{|n|}(\cos(\theta_{j_1})) \right) e^{in\phi_{(j_1, j_2)^T}} \end{aligned} \quad (2.14)$$

$$= \sum_{n=-(N-1)}^{N-1} F_{j_1}^n e^{in\phi_{(j_1, j_2)^T}}. \quad (2.15)$$

Eine algorithmische Umsetzung für die Matrix-Vektor-Multiplikation  $\mathbf{S}\hat{\mathbf{f}}$  hat zunächst die „quartische“ arithmetische Komplexität von  $\mathcal{O}(MN^2)$ . Durch Ausnutzen der speziellen Struktur der Knoten berechnen wir zunächst die  $F_{j_1}^n$  für  $j_1 = 0, \dots, M_1$ ,  $n = -(N-1), \dots, N-1$  in (2.14) mit einer arithmetischen Komplexität von je  $\mathcal{O}(N)$ . Anschließend werten wir für jeden Knoten  $(\theta_j, \phi_j)$ ,  $j = 0, \dots, M-1$ , die Summe in (2.15) mit einer Komplexität von je  $\mathcal{O}(N)$  aus. Wir erhalten eine „kubische“ Gesamtkomplexität von  $\mathcal{O}(M_1 N^2 + MN)$ .

□

### 2.3 Interpolationsprobleme

Im folgenden Abschnitt beschäftigen wir uns mit Problemen, bei denen die gegebenen Abtastwerte  $\mathbf{f} \in \mathbb{K}^M$  interpoliert werden. Für Interpolationsprobleme ist die Gleichung (2.11) konsistent, wir beschäftigen uns deshalb mit dem Minimierungsproblem (2.3).

Wir unterscheiden zunächst die Fälle  $M = N_L$ ,  $M \leq N_L$  und  $M \geq N_L$ , wobei Lösungen durch zusätzliche Forderungen an die Glattheit des Interpolationspolynoms eindeutig charakterisiert werden.

#### Vandermonde–Matrix

Sei die Anzahl der Knoten gleich der Dimension des Polynomraumes, d.h.,  $M = N_L$ , so heißt die quadratische Matrix  $\mathbf{S} \in \mathbb{K}^{N_L \times N_L}$  *Vandermonde–Matrix*. Für den univariaten Fall  $d = 1$  erhalten wir für Abtastmengen  $\mathbf{X}$  mit paarweise verschiedenen Knoten  $x_j \in \mathbb{D}$  durch den Fundamentalsatz der Algebra, siehe z.B. [52, S. 576ff],  $\det(\mathbf{S}) \neq 0$ . Das Interpolationsproblem ist somit für  $[-1, 1]$  und die zueinander isometrischen Mengen, siehe z.B. [23, S. 41],  $\mathbb{T}^1$  und  $\mathbb{S}^1$  eindeutig. Das lineare Gleichungssystem (2.11) besitzt in diesem Fall genau eine Lösung.

Überraschenderweise liefert der Satz von Mairhuber–Curtis, siehe zum Beispiel [3], [44, S. 54], dass es sogenannte Haar–Räume, also Räume in denen das Interpolationsproblem eindeutig lösbar ist, nur über Mengen  $\mathbb{D}$  geben kann, welche homeomorph, siehe z.B. [23, S. 46], zu Teilmengen des 1–Torus  $\mathbb{T}^1$  sind. Für den multivariaten Fall  $d > 1$  heißt dies insbesondere, dass Abtastmengen mit paarweise verschiedenen Knoten existieren für die identisch verschwindende Abtastwerte  $\mathbf{f} = \mathbf{0}$  eines Polynoms  $f \in L_N$  nicht  $f = 0$  implizieren.

#### Spezielles Optimierungsproblem

Sei die Anzahl der Knoten nicht größer als die Dimension des Polynomraumes, d.h.,  $M \leq N_L$ , die Gleichung (2.11) ist also unterbestimmt, so betrachten wir das *Minimierungsproblem*

$$\|\hat{\mathbf{f}}\|_2 \xrightarrow{f} \min \quad \text{mit Nebenbedingung } \mathbf{S}\hat{\mathbf{f}} = \mathbf{f}, \quad (2.16)$$

d.h. wir fordern insbesondere, dass Gleichung (2.11) konsistent ist. Das Problem (2.16) formulieren wir äquivalent um zu

$$\hat{\mathbf{f}} \perp \mathcal{N}(\mathbf{S}) \quad \text{und } \mathbf{S}\hat{\mathbf{f}} = \mathbf{f}.$$

Somit ist der Vektor  $\hat{\mathbf{f}} \in \mathcal{R}(\mathbf{S}^H)$ , siehe Abschnitt 2.1, also darstellbar als  $\hat{\mathbf{f}} = \mathbf{S}^H \tilde{\mathbf{f}}$ ,  $\tilde{\mathbf{f}} \in \mathbb{K}^M$ . Aufgrund der Nebenbedingung muss der Vektor  $\tilde{\mathbf{f}}$  eine Lösung der *Normalengleichung zweiter Art*

$$\mathbf{S}\mathbf{S}^H \tilde{\mathbf{f}} = \mathbf{f}$$

sein. Die Matrix  $\mathbf{S}\mathbf{S}^H$  ist offensichtlich hermitesch und stets positiv semidefinit. Eine spezielle Lösung der Normalengleichung zweiter Art wird durch die anschließende Multiplikation mit der Matrix  $\mathbf{S}^H$  auf die eindeutige Lösung  $\hat{\mathbf{f}}$  des Minimierungsproblems (2.16) abgebildet. Hat die Vandermonde–ähnliche-Matrix  $\mathbf{S}$  vollen Zeilenrang  $\text{rang}(\mathbf{S}) = M$ , so ist die Matrix  $\mathbf{S}\mathbf{S}^H$  positiv definit, die Lösung der Normalengleichung zweiter Art  $\tilde{\mathbf{f}}$  ist dann eindeutig bestimmt und es gilt  $\mathbf{S}^\dagger = \mathbf{S}^H (\mathbf{S}\mathbf{S}^H)^{-1}$ , siehe z.B. [2, Kapitel 1], [6, Kapitel 8].

Sind wir an „glatten“ Lösungen des Interpolationsproblems interessiert, so können wir dies als Abklingbedingung an die Fourier–Koeffizienten formulieren. Wir erhalten das *gedämpfte*

*Minimierungsproblem*

$$\|\hat{\mathbf{f}}\|_{\hat{\mathbf{W}}} \xrightarrow{\hat{\mathbf{f}}} \min \quad \text{mit der Nebenbedingung } \mathbf{S}\hat{\mathbf{f}} = \mathbf{f}. \quad (2.17)$$

Die Matrix  $\hat{\mathbf{W}} := \text{diag}(\hat{\boldsymbol{\omega}})$  nennen wir *Dämpfungsmatrix*, der Vektor  $\hat{\boldsymbol{\omega}} = (\hat{\omega}_k)_{k=0, \dots, N_L-1}$  enthält die Dämpfungsfaktoren  $\hat{\omega}_k \in \mathbb{R}^+$ . Spezielle Dämpfungsfaktoren werden nach den verwandten Normen beziehungsweise Kernen benannt, siehe die Beispiele 2.9, 2.10, 2.11 und 2.12. Die Bezeichnung „gedämpft“ bezieht sich auf ein gewünschtes Abklingen der Fourier-Koeffizienten und ist nicht zu verwechseln mit dem Filtern von Singulärwerten bei Regularisierungsverfahren, siehe Abschnitt 2.5 und z.B. [27, Kapitel 4].

Das gedämpfte Minimierungsproblem (2.17) läßt sich mittels der Substitution  $\tilde{\mathbf{f}}^{\omega} := \hat{\mathbf{W}}^{\frac{1}{2}} \hat{\mathbf{f}}$  in ein Minimierungsproblem (2.16) überführen. Bei konsistenter Nebenbedingung erhalten wir die *gedämpfte Normalgleichung zweiter Art*

$$\mathbf{S}\hat{\mathbf{W}}^{-1}\mathbf{S}^H\tilde{\mathbf{f}} = \mathbf{f}, \quad \hat{\mathbf{f}} = \hat{\mathbf{W}}^{-1}\mathbf{S}^H\tilde{\mathbf{f}}. \quad (2.18)$$

Wir betrachten die Beispiele 2.9 und 2.10 um die ungedämpften Lösungen des speziellen Optimierungsproblems mit den gedämpften zu vergleichen.

**Beispiel 2.9.** (1-Torus  $\mathbb{T}^1$ )

Wählen wir die speziellen „Sobolev“-Dämpfungsfaktoren  $\hat{\omega}_k = 1 + (2\pi k)^2$ ,  $k \in I_N^1$ , so erhalten wir für das spezielle Optimierungsproblem mit  $M = 5$  zufällig gewählten Abtaststellen und einem Polynomgrad  $N = 50$  bei Abtastwerten  $f$  zu zufällig gewählten Fourier-Koeffizienten das folgende Ergebnis in Abbildung 1. Mit Hilfe der Parseval-Gleichung (2.9) für das Polynom  $f$  und seine Ableitung  $f'$  erhalten wir, dass das ungedämpfte Interpolationspolynom die Norm  $\|f\|_L$  minimiert, wohingegen das gedämpfte Interpolationspolynom die Sobolev-Norm  $\|f\|_{W_1^2(\mathbb{T}^1)} := \|f\|_L + \|f'\|_L$ , siehe z.B. [20, S. 393ff] minimiert.

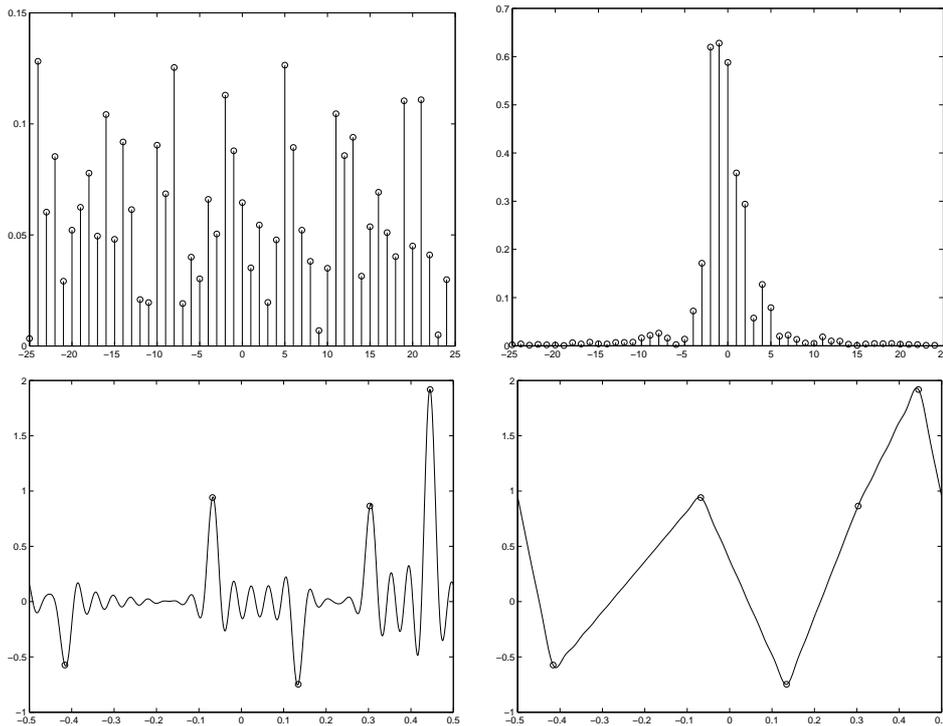


Abbildung 1: Vergleich der Lösungen des speziellen Optimierungsproblems, oben: Absolutbetrag der Fourierkoeffizienten  $\hat{f}_k$ ,  $k \in I_N^1$ , unten: Realteil der Interpolationspolynome, die Interpolationsbedingungen sind durch einen Kreis gekennzeichnet; links: ungedämpft, d.h.  $\|f\|_L \rightarrow \min$ , rechts: gedämpft, d.h.  $\|f\|_{W_1^2(\mathbb{T}^1)} \rightarrow \min$ .

□

**Beispiel 2.10.** (2-Sphäre  $\mathbb{S}^2$ )

Wählen wir die speziellen „Gauß-Weierstraß“-Dämpfungsfaktoren, siehe z.B. [14, S. 112],  $\hat{\omega}_k^n = e^{\frac{k(k+1)}{100}}$ ,  $(k, n)^T \in J_N^2$ , so erhalten wir für das spezielle Optimierungsproblem mit  $M = 10$  zufällig gewählten Abtaststellen und einem Polynomgrad  $N = 20$ , d.h. einer Dimension des Polynomraumes  $N_L = 400$ , bei Abtastwerten  $f_j = \delta_{j,0}$  folgende Ergebnisse in Abbildung 2.

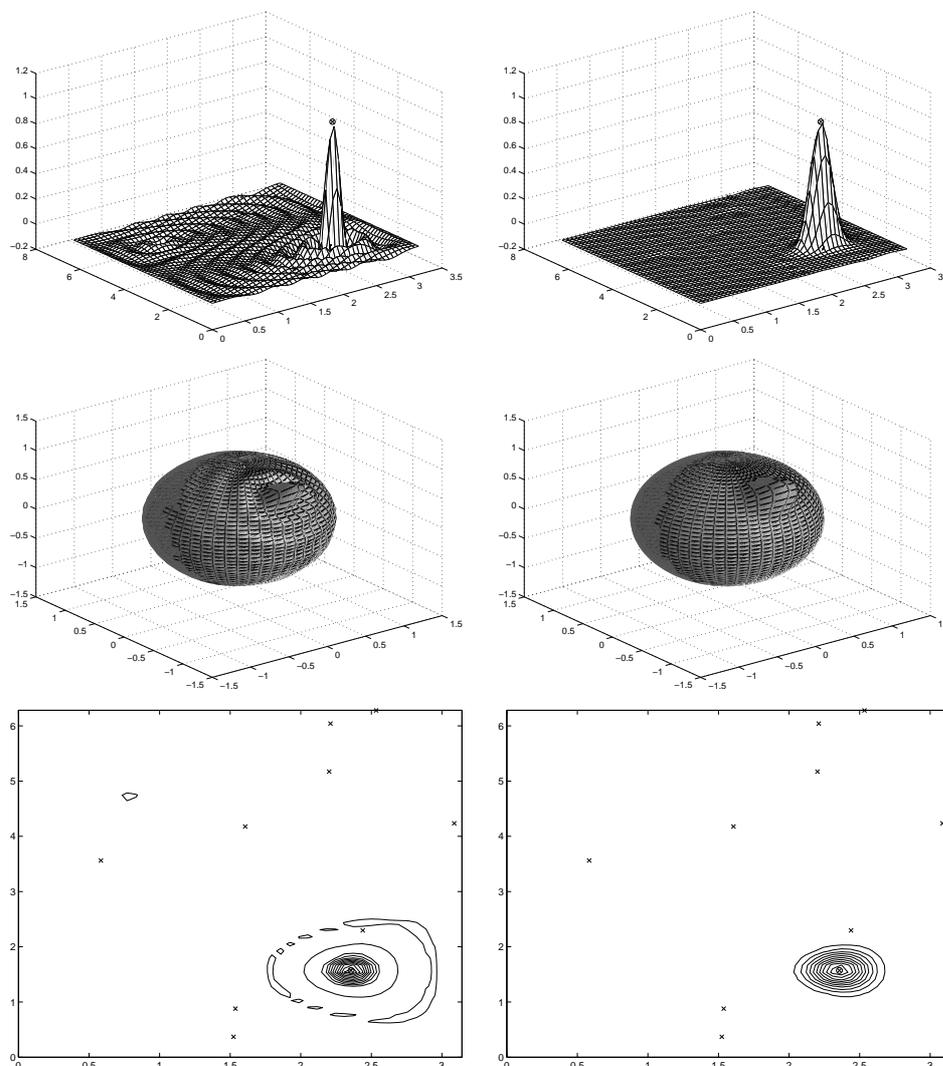


Abbildung 2: Vergleich der Lösungen des speziellen Optimierungsproblems, Realteil der Interpolationspolynome, oben: Funktionswerte über der  $(\theta, \phi)$ -Ebene, mitte: Funktionswerte relativ über der Sphäre, unten: Konturplot der Funktionswerte über der  $(\theta, \phi)$ -Ebene, die Abtaststellen sind durch ein Kreuz gekennzeichnet; links: ungedämpft, d.h.  $\hat{\omega}_k^n = 1$ , rechts: gedämpft, d.h.  $\hat{\omega}_k^n = e^{\frac{k(k+1)}{100}}$ ,  $(k, n)^T \in J_N^2$ .

□

**Polynomiale Kerne**

Sei die Anzahl der Knoten mindestens so groß wie die Dimension des Polynomraumes, d.h.,  $M \geq N_L$ , so betrachten wir zur Herleitung der gedämpften Normalgleichung zweiter Art die

polynomialen Kerne

$$K_j := \sum_{k=0}^{N_L-1} \phi_k \hat{\omega}_k^{-1} \overline{\phi_k(x_j)}. \quad (2.19)$$

Mit den Dämpfungsfaktoren  $\hat{\omega}_k$ ,  $k = 0, \dots, N_L - 1$ , steuern wir die Eigenschaften des Kernes, siehe Beispiel 2.11 und Beispiel 2.12.

**Beispiel 2.11.** (1-Torus  $\mathbb{T}^1$ )

Seien die inversen Dämpfungsfaktoren als die „Dirichlet-Gewichte“  $\hat{\omega}_k^{-1} = 1$ , die „Fejer-Gewichte“  $\hat{\omega}_k^{-1} = \frac{N}{2} + 1 - |k|$  beziehungsweise als „Gauß-Gewichte“  $\hat{\omega}_k^{-1} = e^{-k^2/N}$  gewählt, siehe z.B. [1], so erhalten wir für den Polynomgrad  $N = 20$  mit  $x_0 = -0.2$  nach entsprechender Normalisierung  $K_0(x_0) = 1$  die folgenden polynomialen Kerne in Abbildung 3.

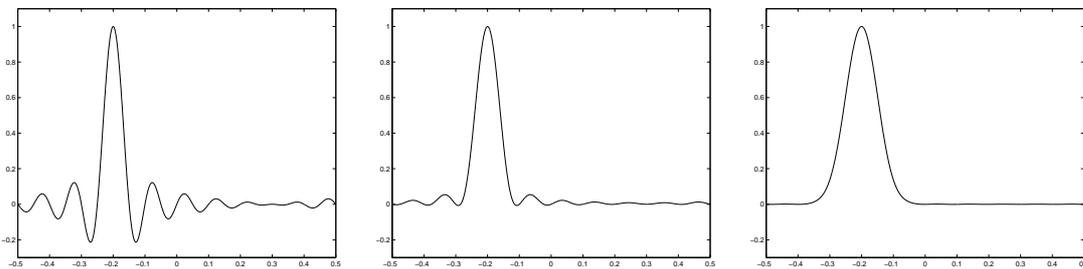


Abbildung 3: Realteil der polynomialen Kerne, links: „Dirichlet“, mitte: „Fejer“, rechts: „Gauß“.

□

**Beispiel 2.12.** (2-Sphäre  $S^2$ )

Seien für  $(k, n)^T \in J_N^2$  die inversen Dämpfungsfaktoren als die „Dirichlet-Gewichte“  $\hat{\omega}_{(k,n)^T}^{-1} = 1$ , die speziellen „Gauß-Weierstraß-Gewichte“  $\hat{\omega}_{(k,n)^T}^{-1} = e^{-\frac{k(k+1)}{100}}$  beziehungsweise als die speziellen „Poisson-Gewichte“  $\hat{\omega}_{(k,n)^T}^{-1} = 0.7^k$  gewählt, siehe z.B. [14, S. 81ff], so erhalten wir für den Polynomgrad  $N = 20$  nach entsprechender Normalisierung  $K_0(x_0) = 1$  folgende polynomiale Kerne in Abbildung 4.

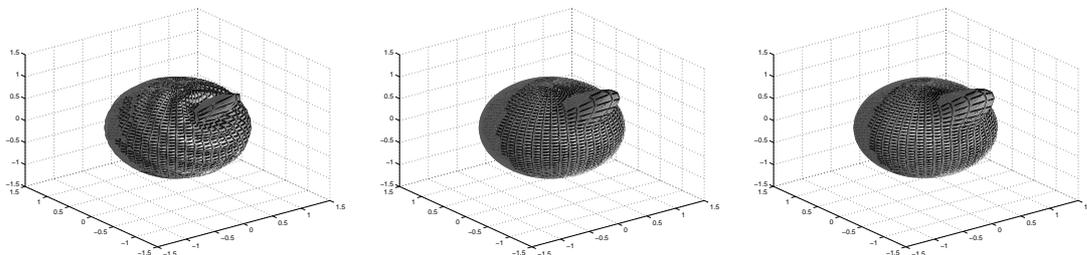


Abbildung 4: Realteil der polynomialen Kerne, links: „Dirichlet“, mitte: „Gauß-Weierstraß“, rechts: „Poisson“.

□

Die *Linearkombination der Kerne*

$$\mathcal{K} : \mathbb{K}^M \rightarrow L_N, \tilde{\mathbf{f}} \mapsto f = \mathcal{K}\tilde{\mathbf{f}} := \sum_{j=0}^{M-1} \tilde{f}_j K_j$$

bildet einen Vektor von „Kernkoeffizienten“  $\tilde{\mathbf{f}} \in \mathbb{K}^M$  auf eine Funktion  $f \in L_N$  ab.

Spannen die Kerne  $K_j$ ,  $j = 0, \dots, M-1$ , den Polynomraum  $L_N$  auf, wofür die Bedingung  $M \geq N_L$  notwendig ist, so ist die Gleichung

$$\mathcal{K}\tilde{\mathbf{f}} = f$$

für alle Polynome  $f \in L_N$  konsistent. Sie kann mit der Definition (2.19) als

$$S\hat{W}^{-1}S^H\tilde{\mathbf{f}} = f$$

ausgedrückt werden. Durch die Abtastung, d.h.,  $S = \mathcal{D}S$  und  $f = \mathcal{D}f$  erhalten wir weiterhin die gedämpfte Normalgleichung zweiter Art

$$S\hat{W}^{-1}S^H\tilde{\mathbf{f}} = f.$$

Die Approximation (nichtpolynomialer) Kerne und damit die schnelle Summation radialer Basisfunktionen wird in [38, 36] dargestellt. Für nichtsinguläre Kerne hat die approximative Auswertung der Kerne zur Abtastmenge  $\mathbf{X}_2$  an den Knoten der Abtastmenge  $\mathbf{X}_1$  die Form

$$f \approx S_1\hat{W}^{-1}S_2^H\tilde{\mathbf{f}}.$$

### Zusammenfassung

Sei die Anzahl der Abtaststellen  $M$  beliebig und die Gleichung (2.11) konsistent, so können wir die gedämpfte Normalgleichung zweiter Art

$$S\hat{W}^{-1}S^H\tilde{\mathbf{f}} = f \tag{2.20}$$

aufstellen. Jede spezielle Lösung  $\tilde{\mathbf{f}}$  dieser Gleichung wird mit  $\hat{\mathbf{f}} = \hat{W}^{-1}S^H\tilde{\mathbf{f}}$  auf die eindeutig bestimmte Lösung des gedämpften Minimierungsproblems

$$\|\hat{\mathbf{f}}\|_{\hat{W}} \xrightarrow{\hat{\mathbf{f}}} \min \tag{2.21}$$

mit der Nebenbedingung (2.11) abgebildet. Die Wahl der Dämpfungsmatrix  $\hat{W}$  bestimmt dabei die Glattheitseigenschaften des Interpolationspolynoms, insbesondere können die im Fourier-Bereich einfach formulierbaren Sobolev-Normen verwendet werden.

## 2.4 Diskretes Approximationsproblem

Für die speziell konstruierten Abtastmengen in Abschnitt 2.2 erhielten wir zwischen einem Polynom  $f$  und seinen Abtastwerten  $\mathbf{f}$  einen Energievergleich der Form  $\|\mathbf{f}\|_{\mathbf{W}}^2 = \|f\|_2^2$ . Im Folgenden werden wir für eine große Klasse von Abtastmengen zumindest noch Ungleichungen zum Energievergleich aufstellen.

### Normalengleichung erster Art und Stabilität von Abtastmengen

Sei die Anzahl der Knoten  $M$  beliebig in Bezug auf die Dimension des Polynomraumes, so betrachten wir das *Approximationsproblem*

$$\|\mathbf{S}\hat{\mathbf{f}} - \mathbf{f}\|_2 \xrightarrow{\hat{\mathbf{f}}} \min. \quad (2.22)$$

Mit [2, Kapitel 1] ist dieses Approximationsproblem (2.22) äquivalent zur Normalengleichung erster Art

$$\mathbf{S}^H \mathbf{S} \hat{\mathbf{f}} = \mathbf{S}^H \mathbf{f}. \quad (2.23)$$

Die Matrix  $\mathbf{S}^H \mathbf{S}$  ist offensichtlich hermitesch und positiv semidefinit. Hat die Vandermonde-ähnliche-Matrix  $\mathbf{S}$  vollen Spaltenrang  $\text{rang}(\mathbf{S}) = N_L$ , d.h., der Polynomgrad  $N$  ist niedrig genug gewählt, so ist die Matrix  $\mathbf{S}^H \mathbf{S}$  positiv definit und es gilt  $\mathbf{S}^\dagger = (\mathbf{S}^H \mathbf{S})^{-1} \mathbf{S}^H$ .

Weiterhin erhalten wir in Lemma 2.13 folgenden engen Zusammenhang zwischen der Abtastmenge  $\mathbf{X}$  und der Vandermonde-ähnliche-Matrix  $\mathbf{S}$ .

**Lemma 2.13.** Sei  $\Phi$  ein Vektor orthonormaler Basiselemente des komplexen Polynomraumes  $L_N$  und  $\mathbf{S}$  die zur Abtastmenge  $\mathbf{X}$  gehörende Vandermonde-ähnliche-Matrix, so sind die bestmöglichen Konstanten in der Ungleichung, siehe z.B. [18, S. 75], [29, 17, 10],

$$\xi \|f\|_L^2 \leq \|\mathbf{f}\|_2^2 \leq \Xi \|f\|_L^2, \quad f \in L_N, \quad (2.24)$$

gegeben durch

$$\xi = \lambda_{\min}(\mathbf{S}^H \mathbf{S}), \quad \Xi = \lambda_{\max}(\mathbf{S}^H \mathbf{S}).$$

Es gilt insbesondere für beliebige Konstanten  $\xi, \Xi, \xi > 0$ , welche die Ungleichung (2.24) erfüllen, dass  $\text{cond}_2(\mathbf{S}^H \mathbf{S}) \leq \frac{\Xi}{\xi}$ .

*Beweis.* Wir verwenden die Parseval-Gleichung (2.9) für  $\|f\|_L^2$  und die Gleichung (2.11) für  $\|\mathbf{f}\|_2^2$  in der Ungleichung (2.24) und erhalten

$$\xi \leq \frac{\hat{\mathbf{f}}^H \mathbf{S}^H \mathbf{S} \hat{\mathbf{f}}}{\hat{\mathbf{f}}^H \hat{\mathbf{f}}} \leq \Xi.$$

Weiterhin gilt mit [20, S. 207f], dass

$$\left\{ \frac{\hat{\mathbf{f}}^H \mathbf{S}^H \mathbf{S} \hat{\mathbf{f}}}{\hat{\mathbf{f}}^H \hat{\mathbf{f}}} : \hat{\mathbf{f}} \in \mathbb{C}^{N_L} \right\} = [\lambda_{\min}(\mathbf{S}^H \mathbf{S}), \lambda_{\max}(\mathbf{S}^H \mathbf{S})].$$

Wir sehen leicht, dass  $\text{cond}_2(\mathbf{S}^H \mathbf{S}) = \frac{\lambda_{\max}(\mathbf{S}^H \mathbf{S})}{\lambda_{\min}(\mathbf{S}^H \mathbf{S})} \leq \frac{\Xi}{\xi}$ . ■

Sind positive Konstanten für die Ungleichung (2.24) bekannt, so erhalten wir mit dem folgenden Lemma 2.14 eine Möglichkeit die Normalengleichung erster Art (2.23) zu lösen.

**Lemma 2.14.** Sei  $\Phi$  ein Vektor orthonormaler Basiselemente des Polynomraumes  $L_N$  und die Ungleichung (2.24) sei für beliebige Polynome  $f \in L_N$  mit der Konstanten  $\xi > 0$  erfüllt, so gilt mit dem Parameter  $\alpha$ ,  $0 < \alpha < \frac{1}{\Xi}$ , und  $\mathbf{T} := \alpha \mathbf{S}^H \mathbf{S}$ , dass

$$\mathbf{T} \text{ regulär ist und } \|\mathbf{E} - \mathbf{T}\|_2 < 1.$$

Somit sind die Voraussetzungen von Lemma 2.2 erfüllt und die Inverse  $\mathbf{T}^{-1}$  kann explizit nach (2.4) angegeben werden.

*Beweis.* Mit  $\xi > 0$  ist  $\mathbf{T}$  regulär und weiterhin gilt mit den Spektren  $\sigma$  und dem Spektralradius  $\rho$ , siehe z.B. [28, S.19ff],

$$\sigma(\mathbf{S}^H \mathbf{S}) \subseteq [\xi, \Xi] \Rightarrow \sigma(\mathbf{T}) \subseteq (0, 1) \Rightarrow \|\mathbf{E} - \mathbf{T}\|_2 = \rho(\mathbf{E} - \mathbf{T}) < 1.$$

■

Im Folgenden weisen wir kurz auf die auftretenden Schwierigkeiten im Zusammenhang mit der Ungleichung (2.24) und den minimalen und maximalen Eigenwerten der Matrix  $\mathbf{S}^H \mathbf{S}$  hin. Da für  $M < N_L$  die Matrix  $\mathbf{S}$  höchstens Rang  $M$  haben kann, erhalten wir  $\lambda_{\min}(\mathbf{S}^H \mathbf{S}) = 0$ . Im univariaten Fall  $d = 1$  erhalten wir für mindestens  $N_L$ , d.h. insbesondere  $M \geq N_L$ , paarweise verschiedene Abtaststellen  $\lambda_{\min}(\mathbf{S}^H \mathbf{S}) > 0$ . Für den multivariaten Fall  $d > 1$  impliziert der Satz von Mairhuber–Curtis, dass selbst  $M \geq N_L$  paarweise verschiedene Abtaststellen nicht ausreichen um  $\lambda_{\min}(\mathbf{S}^H \mathbf{S}) > 0$  zu gewährleisten. Wir nennen die Abtastmenge  $\mathbf{X}$  *stabil*, falls  $\lambda_{\min}(\mathbf{S}^H \mathbf{S}) > 0$ . Diese Eigenschaft einer Abtastmenge ist qualitativer Natur.

Für die beste obere Schranke in der Ungleichungen (2.24), d.h. den maximalen Eigenwert der Matrix  $\mathbf{S}^H \mathbf{S}$ , betrachten wir folgendes Beispiel 2.15.

**Beispiel 2.15.** (1–Torus  $\mathbb{T}^1$ )

Wir betrachten den Dirichlet–Kern  $D_N$  für  $L_N(\mathbb{T}^1, \mathbb{C})$  und seine Reproduktionseigenschaft für beliebige Polynome  $f \in L_N$ , d.h.,

$$D_N := \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} e^{-2\pi i k \cdot}, \quad f(x) = \int_{\mathbb{T}^1} f(t) D_N(t-x) dt.$$

Damit gilt insbesondere

$$\|f\|_{\infty} \leq \int_{\mathbb{T}^1} |f(t)| |D_N(t-x)| dt \leq \|f\|_L \|D_N\|_L = \sqrt{N} \|f\|_L,$$

wobei die Ungleichungen für  $f = D_N$  zur Gleichung  $\|D_N\|_{\infty} = \sqrt{N} \|D_N\|_L = N$  übergehen, siehe z.B. [50, S. 229].

Für beliebig gewählte Abtastmengen erhalten wir im Folgenden eine optimale Abschätzung für den größten Eigenwert der Matrix  $\mathbf{S}^H \mathbf{S}$ . Es gilt

$$\|\mathbf{f}\|_2^2 \leq MN \|f\|_L^2,$$

d.h., der größte Eigenwert der Matrix  $\mathbf{S}^H \mathbf{S}$  wächst höchstens wie  $MN$ . Gleichheit wird angenommen, falls für alle  $j = 0, \dots, M-1$  gilt  $|f_j| = \|f\|_{\infty}$ . □

### Gewichtetes Approximationsproblem

Im Weiteren werden wir die Struktur der gegebenen Abtastmenge  $\mathbf{X}$  bei der Konstruktion der Normalgleichung erster Art einbeziehen und können unter gewissen Voraussetzungen an die Abtastmenge explizite Schranken für die Eigenwerte der entstandenen Matrix angeben, welche nur vom größten „Loch“ in der Abtastmenge abhängen. Dies ermöglicht dann eine explizite Parameterwahl bei einfachen Lösungsverfahren, siehe Abschnitt 4.2, und Aussagen zum Konvergenzverhalten bei allen Verfahren in Kapitel 4.

Für eine Gewichtsmatrix  $\mathbf{W} = \text{diag}(\boldsymbol{\omega})$ ,  $\boldsymbol{\omega} = (\omega_j)_{j=0, \dots, M-1}$ ,  $\omega_j \in \mathbb{R}^+$ , betrachten wir das *gewichtete Approximationsproblem*

$$\|\mathbf{S}\hat{\mathbf{f}} - \mathbf{f}\|_{\mathbf{W}} \xrightarrow{\hat{\mathbf{f}}} \min. \quad (2.25)$$

Mit der Substitution  $\mathbf{S}^\omega := \mathbf{W}^{\frac{1}{2}}\mathbf{S}$  und  $\mathbf{f}^\omega := \mathbf{W}^{\frac{1}{2}}\mathbf{f}$  erhalten wir ein Approximationsproblem (2.22) und damit die *gewichtete Normalgleichung erster Art*

$$\mathbf{S}^H \mathbf{W} \mathbf{S} \hat{\mathbf{f}} = \mathbf{S}^H \mathbf{W} \mathbf{f}. \quad (2.26)$$

Die Matrix  $\mathbf{S}^H \mathbf{W} \mathbf{S}$  ist hermitesch und positiv semidefinit. Hat die Vandermonde-ähnliche Matrix  $\mathbf{S}$  vollen Spaltenrang  $\text{rang}(\mathbf{S}) = N_L$ , d.h., der Polynomgrad  $N$  ist niedrig genug gewählt, so ist die Matrix  $\mathbf{S}^H \mathbf{W} \mathbf{S}$  positiv definit. Im Weiteren klären wir, wie die Gewichtsmatrix  $\mathbf{W}$  gewählt werden kann.

### Abtastmenge und erlaubte Partition

Die folgenden Ideen sind den Arbeiten [30, 12, 16, 17] entnommen. Für die Abtastmenge  $\mathbf{X}$  definieren wir die *Abtastmengen-Norm* (engl. mesh-norm, siehe z.B. [30])

$$\delta_{\mathbf{X}} = \max_{\mathbf{x} \in \mathbb{D}} \min_{j=0, \dots, M-1} \text{dist}_{\mathbb{D}}(\mathbf{x}, \mathbf{x}_j). \quad (2.27)$$

Die Abtastmengen-Norm  $\delta_{\mathbf{X}} \in \mathbb{R}^+$  gibt also den „Radius des größten Loches“ der Abtastmenge  $\mathbf{X}$  im Urbildraum  $\mathbb{D}$  an. Werden „Kreisscheiben“ auf die Abtastpunkte  $\mathbf{x}_j$  gelegt, so benötigen diese einen Radius größer als die Abtastmengen-Norm um unter Vereinigung den Urbildraum  $\mathbb{D}$  zu überdecken. In Anlehnung an [17] nennen wir die Abtastmenge  $\delta$ -*dicht*, falls  $\delta_{\mathbf{X}} < \delta \in \mathbb{R}^+$ . In [17] wird die äquivalente Charakterisierung

$$\bigcup_{j=0}^{M-1} B_{j,\delta} = \mathbb{D} \quad \text{für } B_{j,\delta} := \{\mathbf{x} \in \mathbb{D} : \text{dist}_{\mathbb{D}}(\mathbf{x}_j, \mathbf{x}) < \delta\}$$

gegeben.

Weiterhin betrachten wir eine Partition der Urbildmenge  $\mathbb{D}$ , d.h. eine Menge

$$\mathbf{R} := \{R_{j'} \subset \mathbb{D} : j' = 0, \dots, M' - 1\}$$

mit  $R_{j'_1} \cap R_{j'_2} = \emptyset$ ,  $j'_1 \neq j'_2$ , und  $\bigcup_{j'=0}^{M'-1} R_{j'} = \mathbb{D}$ . Eine Partition heißt *erlaubte Partition in Bezug auf die Abtastmenge  $\mathbf{X}$*  (engl.  $\mathbf{X}$ -compatible decomposition, siehe [29]), falls jede Region  $R_{j'}$ ,  $j' = 0, \dots, M' - 1$ , mindestens einen Knoten der Abtastmenge  $\mathbf{X}$  enthält. Gehört zu jeder Region der Partition  $\mathbf{R}$  genau ein Knoten der Abtastmenge, so sprechen wir von einer

1-Partition. In dieser Arbeit sei angenommen, dass kein Knoten  $\mathbf{x}_j$ ,  $j = 0, \dots, M - 1$ , auf dem Rand einer Regionen  $R_{j'}$  liegt.

Wir definieren die *Norm der Partition*

$$\|\mathbf{R}\| := \max_{j'=0, \dots, M'-1} \text{diam}(R_{j'}), \quad \text{diam}(R) := \sup_{\mathbf{x}, \mathbf{y} \in R} \text{dist}_{\mathbb{D}}(\mathbf{x}, \mathbf{y}).$$

Wählen wir zu jeder Region  $R_{j'}$ ,  $j' = 0, \dots, M' - 1$  einen Knoten  $\mathbf{x}_j \in R_{j'}$  als Repräsentant, so erhalten wir die *reduzierte Abtastmenge*  $\mathbf{X}'$ . Offensichtlich ist  $\mathbf{R}$  eine erlaubte Partition in Bezug auf die reduzierte Abtastmenge  $\mathbf{X}'$  und es gilt  $\delta_{\mathbf{X}'} \leq \delta_{\mathbf{X}} < \|\mathbf{R}\|$ . Da zu jeder Region der Partition  $\mathbf{R}$  genau ein Knoten der reduzierten Abtastmenge  $\mathbf{X}'$  gehört, ist  $\mathbf{R}$  eine 1-Partition in bezug auf  $\mathbf{X}'$ .

### Voronoi-Region, Voronoi-Gewicht

Sei die Abtastmenge  $\mathbf{X}$  beliebig, so heißt

$$R_{V,j} := \left\{ \mathbf{x} \in \mathbb{D} : \arg \min_{l=0, \dots, M-1} \text{dist}_{\mathbb{D}}(\mathbf{x}, \mathbf{x}_l) = j \right\}$$

*Voronoi-Region des Knotens  $\mathbf{x}_j$* . Die Partition  $\mathbf{R}_V := \{R_{V,j} \subset \mathbb{D} : j = 0, \dots, M - 1\}$  nennen wir im Folgenden *Voronoi-Partition*. Das Gewicht

$$\omega_j := \int_{R_{V,j}} d\omega_{\mathbb{D}}(\mathbf{x})$$

heißt *Voronoi-Gewicht des Knotens  $\mathbf{x}_j$* .

### Beispiel 2.16. (1-Torus $\mathbb{T}^1$ )

Wir wählen zufällig  $M = 5$  Abtaststellen und berechnen die Voronoi-Partition. Durch die Wegnahme der mit einem Kreis gekennzeichneten Abtaststelle erhalten wir eine Partition, nicht erlaubt ist, durch die Wegnahme einer Regionengrenze, d.h. die Vereinigung zweier Regionen, bleibt die Partition jedoch erlaubt, siehe Abbildung 5.

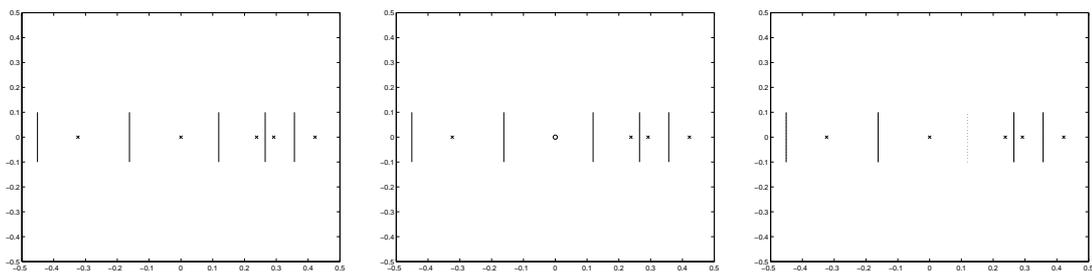


Abbildung 5: Partitionen des 1-Torus  $\mathbb{T}^1$ , links: Voronoi-Partition, mitte: nicht erlaubte Partition, rechts: erlaubte Partition. □

### Beispiel 2.17. (2-Torus $\mathbb{T}^2$ )

Analog zu Beispiel 2.16 erhalten wir für  $M = 10$  auf dem 2-Torus die Abbildung 6. □

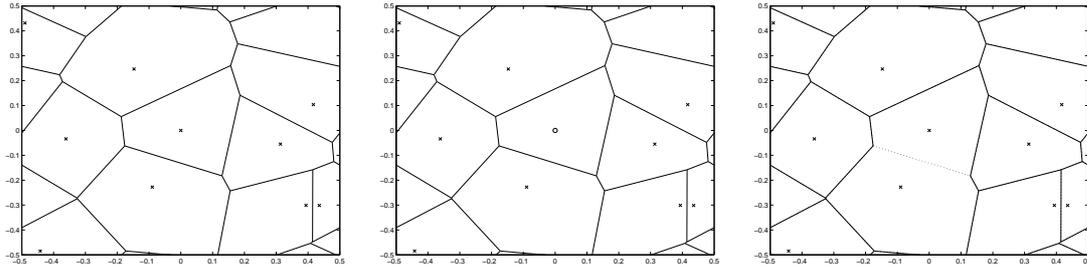


Abbildung 6: Partitionen des 2-Torus  $\mathbb{T}^2$ , links: Voronoi-Partition, mitte: nicht erlaubte Partition, rechts: erlaubte Partition.

### Spezielle Ungleichungen

Im Folgenden zitieren wir tief liegendere Sätze zur Abtastung trigonometrischer und sphärischer Polynome. Wir folgern mithilfe von Lemma 2.13 explizite Schranken für die Konditionszahl der Matrix  $\mathbf{S}^H \mathbf{W} \mathbf{S}$ . Weiterhin quantifizieren wir mit den Schranken der Marcinkiewicz-Zygmund-Ungleichungen, siehe z.B. [31], die Güte der Abtastmenge  $\mathbf{X}$ .

**Satz 2.18.** (1-Torus  $\mathbb{T}^1$ , Feichtinger, Gröchenig, Strohmer)

Sei die Abtastmenge  $\mathbf{X}$   $\delta$ -dicht im 1-Torus  $\mathbb{T}^1$ , so gilt für beliebigen Polynomgrad  $N$  mit

$$N < \frac{1}{\delta}$$

und den Voronoi-Gewichten  $\omega_j$ ,  $j = 0, \dots, M - 1$ , für alle Polynome  $f \in L_N$

$$(1 - \delta N)^2 \|f\|_L^2 \leq \|f\|_{\mathbf{W}}^2 \leq (1 + \delta N)^2 \|f\|_L^2 \quad (2.28)$$

und somit

$$\text{cond}_2(\mathbf{S}^H \mathbf{W} \mathbf{S}) \leq \left( \frac{1 + \delta N}{1 - \delta N} \right)^2.$$

*Beweis.* Siehe [16, Satz 5], [12, Satz 1]. ■

**Satz 2.19.** ( $d$ -Torus  $\mathbb{T}^d$ , Gröchenig, Strohmer, Bass)

Sei die Abtastmenge  $\mathbf{X}$   $\delta$ -dicht im  $d$ -Torus  $\mathbb{T}^d$ , so gilt für beliebigen Polynomgrad  $N$  mit

$$N < \frac{\log 2}{\pi \delta d}$$

und den Voronoi-Gewichten  $\omega_j$ ,  $j = 0, \dots, M - 1$ , für alle Polynome  $f \in L_N$

$$(2 - e^{\pi N \delta d})^2 \|f\|_L^2 \leq \|f\|_{\mathbf{W}}^2 \leq 4 \|f\|_L^2 \quad (2.29)$$

und somit

$$\text{cond}_2(\mathbf{S}^H \mathbf{W} \mathbf{S}) \leq \left( \frac{2}{2 - e^{\pi N \delta d}} \right)^2.$$

*Beweis.* Siehe [17, Satz 5], [1, Satz 4.1]. ■

Wir bemerken, dass Satz 2.18 und Satz 2.19 ihre Gültigkeit behalten, falls  $f \in L_N$  zusätzliche nicht verschwindende Fourierkoeffizienten  $\hat{f}_k$ ,  $k_c = \frac{N}{2}$ ,  $c = 1, \dots, d$  erhält. Weiterhin wurde Satz 2.19 in [1] verschärft, die  $\delta$ -Dichtheit reicht bezüglich der „schwächeren“ Distanz  $\min_{j \in \mathbb{Z}^d} \|(\mathbf{x} + \mathbf{j}) - \mathbf{y}\|_\infty$  aus.

**Satz 2.20.** (2-Sphäre  $\mathbb{S}^2$ , Mhaskar, Narcowich, Ward)

Nach [30, S. 17] existieren eine gleichmäßig nach unten beschränkte Konstante  $\varrho \leq \frac{1}{4}$  und ein „Mindestgrad“  $\tilde{N} \in \mathbb{N}$ . Sei weiterhin  $\mathbf{R}$  eine erlaubte Partition in Bezug auf eine reduzierte Abtastmenge  $\mathbf{X}'$ , so gilt für  $\nu \in (0, 1)$  und beliebigen Polynomgrad  $N$  mit

$$\tilde{N} \leq N < \frac{\nu \varrho}{\|\mathbf{R}\|},$$

dass alle Polynome  $f \in L_N$  die Ungleichung

$$(1 - \nu)^2 \|f\|_L^2 \leq \|\mathbf{f}\|_{\mathbf{W}'}^2 \leq (1 + \nu)^2 \|f\|_L^2 \quad (2.30)$$

erfüllen, wobei die Gewichte mit

$$\omega'_j = \begin{cases} \int_{R_j} d\omega_{\mathbb{S}^2} & \text{für } \mathbf{x}_j \in \mathbf{X}', \\ 0 & \text{sonst} \end{cases}$$

gewählt sind. Somit ist

$$\text{cond}_2(\mathbf{S}^H \mathbf{W}' \mathbf{S}) \leq \left( \frac{1 + \nu}{1 - \nu} \right)^2.$$

*Beweis.* Siehe [30, Satz 3.1]. ■

**Lemma 2.21.** (2-Sphäre  $\mathbb{S}^2$ )

Sei  $\mathbf{R}_V$  die Voronoi-Partition in Bezug auf die Abtastmenge  $\mathbf{X}$ . Weiter sei  $\mathbf{X}'$  eine reduzierte Abtastmenge mit einer 1-Partition  $\mathbf{R}$ , welche aus Vereinigungen von Voronoi-Regionen  $R_{V,j}$ ,  $j = 0, \dots, M - 1$  besteht.

Sind die Voraussetzungen des Satzes 2.20 für die Partition  $\mathbf{R}$  gegeben, so gilt insbesondere die Ungleichung

$$(1 - \nu)^2 \|f\|_L^2 \leq \|\mathbf{f}\|_{\mathbf{W}}^2 \leq (1 + \nu)^2 \|f\|_L^2 \quad (2.31)$$

mit den üblichen Voronoi-Gewichten  $\omega_j$ ,  $j = 0, \dots, M - 1$ .

*Beweis.* Die Vereinigung der Voronoi-Regionen wird „rückgängig“ gemacht. Für den Vektor der Abtastwerte

$$\mathbf{f}_{\min} \in \mathbb{K}^{M'} \quad \text{zu den „Minimumrepräsentanten“ } \mathbf{x}_{\min,j'} = \arg \min_{\mathbf{x} \in R_{j'}} |f(\mathbf{x})|$$

und den Vektor der Abtastwerte

$$\mathbf{f}_{\max} \in \mathbb{K}^{M'} \quad \text{zu den „Maximumrepräsentanten“ } \mathbf{x}_{\max,j'} = \arg \max_{\mathbf{x} \in R_{j'}} |f(\mathbf{x})|$$

der Regionen  $R_{j'}$ ,  $j' = 0, \dots, M' - 1$  hat der Satz 2.20 jeweils Gültigkeit. Weiter gilt

$$\|\mathbf{f}_{\min}\|_{\mathbf{W}'}^2 \leq \|\mathbf{f}\|_{\mathbf{W}}^2 \leq \|\mathbf{f}_{\max}\|_{\mathbf{W}'}^2.$$

■

### Zusammenfassung

Sei die Anzahl der Abtaststellen  $M$  beliebig, so können wir die gewichtete Normalgleichung erster Art

$$\mathbf{S}^H \mathbf{W} \mathbf{S} \hat{\mathbf{f}} = \mathbf{S}^H \mathbf{W} \mathbf{f} \quad (2.32)$$

formulieren. Die Lösungen  $\hat{\mathbf{f}}$  dieser Normalgleichung sind genau die Lösungen des gewichteten Approximationsproblems

$$\|\mathbf{S} \hat{\mathbf{f}} - \mathbf{f}\|_{\mathbf{W}} \xrightarrow{\hat{\mathbf{f}}} \min. \quad (2.33)$$

Die Wahl der Gewichtsmatrix  $\mathbf{W}$  gleicht dabei Ungleichmässigkeiten der Abtastmenge aus, beziehungsweise steuert wie gut in einem bestimmten Gebiet approximiert wird. Für eine gegebene Abtastmenge konnten wir darüberhinaus explizite Schranken für die Kondition der gewichteten Normalgleichung erster Art (2.32) geben, falls der Polynomgrad  $N$  niedrig genug gewählt wird.

## 2.5 Regularisierungsverfahren

Durch das Abtasten der Gleichung (2.6), d.h.

$$\mathcal{S}\hat{\mathbf{f}} = \mathbf{f}, \quad \text{erhalten wir die Gleichung (2.11), d.h. } \mathcal{S}\hat{\mathbf{f}} = \mathbf{f}.$$

Dieses lineare Gleichungssystem können wir unter gewissen Voraussetzungen und in einem geeigneten Sinn (theoretisch) lösen. Die numerischen Eigenschaften der Gleichung (2.11) haben wir bis jetzt vernachlässigt, was bei gestörten Abtastwerten  $\mathbf{f}$  jedoch zu starken Fehlern in der Lösung  $\hat{\mathbf{f}}$  führen kann.

Im Folgenden klären wir zunächst die verwendeten Begriffe, geben Standardverfahren zur Regularisierung an und leiten schließlich für unsere Problemstellung geeignete Verfahren ab.

### Diskret schlecht gestellte Probleme

In Anlehnung an die Definition für gut gestellte Probleme in [27, S. 8ff] wollen wir ein Problem  $(\Phi, \mathbf{X}, \mathbb{K}^{N_L}, \mathbb{K}^M)$  *diskret gut gestellt* nennen, falls die Gleichung  $\mathcal{S}\hat{\mathbf{f}} = \mathbf{f}$

1. für jedes  $\mathbf{f} \in \mathbb{K}^M$  eine Lösung hat,
2. diese Lösung eindeutig bestimmt ist,
3. die betrachtete Normalengleichung gut konditioniert ist.

Ist eine Bedingung verletzt, so nennen wir das Problem *diskret schlecht gestellt*.

Die erste Bedingung ist verletzt, falls  $\mathbf{f} \notin \mathcal{R}(\mathcal{S})$ , d.h., falls ein zu niedriger Polynomgrad angenommen wurde. Die Gleichung (2.11) ist für solch einen Vektor von Abtastwerten  $\mathbf{f}$  inkonsistent. Eine Lösung kann dann nur approximativ sein, siehe Abschnitt 2.4. Im Abschnitt 2.3 haben wir uns auf die konsistenten Probleme  $(\Phi, \mathbf{X}, \mathbb{K}^{N_L}, \mathcal{DL}_N)$  eingeschränkt, d.h. insbesondere, wir gehen von ungestörten Abtastwerten aus. Damit war die erste Bedingung erfüllt und wir haben uns der zweiten gewidmet. Die Forderung nach guter Kondition kann durch die in [27, Kapitel 4] vorgestellten *Regularisierungsverfahren* erreicht werden.

### Ursprüngliches Problem und abgeschnittene Singulärwertzerlegung

Die ersten beiden Forderungen an ein diskret gut gestelltes Problem werden durch die Pseudoinverse-Lösung  $\mathbf{f}^\dagger$  erfüllt, diese hat mit der Singulärwertzerlegung (2.1) der Matrix  $\mathcal{S}$ , den Spalten  $\mathbf{u}_j$ ,  $j = 0, \dots, M-1$ , der Matrix  $\mathbf{U}$ , den Spalten  $\mathbf{v}_k$ ,  $k = 0, \dots, N_L-1$ , der Matrix  $\mathbf{V}$  und  $l_{\max} = \text{rang}(\mathcal{S})$  die Darstellung

$$\mathbf{f}^\dagger = \sum_{l=0}^{l_{\max}} (\sigma_l^{-1} \mathbf{u}_l^H \mathbf{f}) \mathbf{v}_l.$$

Für einen Regularisierungsparameter  $l_{\max} < \text{rang}(\mathcal{S})$  werden Anteile zu kleinen Singulärwerten ausgeblendet. Wir wollen dieses Verfahren hier nicht weiter verfolgen, da es im Allgemeinen eine hohe arithmetische Komplexität mit sich bringt.

### Tikhonov–Philips–Regularisierung

Ein Verfahren zur Regularisierung des Approximationsproblems, siehe Abschnitt 2.4, ist die sogenannte Tikhonov–Philips–Regularisierung. Dabei wird mittels eines „Straftermes“  $\Omega : \mathbb{K}^{N_L} \rightarrow \mathbb{R}_0^+$  Zusatzinformation über die Lösung  $\hat{f}$  eingebracht. Das Tikhonov–Philips–Funktional bildet ein Polynom  $f$ , gegeben durch seine Fourier–Koeffizienten in die positiv reellen Zahlen ab, es hat mit dem Regularisierungsparameter  $\gamma \in \mathbb{R}_0^+$  die Form

$$J_\gamma \hat{f} := \|\mathcal{S}\hat{f} - f\|_W^2 + \gamma^2 \Omega \hat{f}.$$

In [49] schlagen die Autoren bei der Rekonstruktion von Magnetresonanzbildern einen speziellen Strafterm  $\Omega$  vor. Für die Verwendung in iterativen Verfahren ist es insbesondere wichtig den Strafterm  $\Omega$  schnell berechnen zu können. Die praktischen Probleme sind also die geschickte Wahl eines Straftermes und die an die (unbekannt große) Störung der Daten  $f$  angepaßte Schätzung des Regularisierungsparameters.

### Iterationsverfahren

Die Verwendung von Iterationsverfahren zur Regularisierung diskret schlecht gestellter Probleme wird beispielsweise in [27, S. 103ff], die Anwendung von CG–Algorithmen insbesondere in [19] behandelt. Der Regularisierungsparameter ist in diesem Fall die Anzahl der Iterationen, für spezielle Verfahren kann die Wirkung auf die Singulärwerte von  $\mathcal{S}$  explizit angegeben werden.

In [32, S. 89] wird die Konvergenz am Beginn der Iteration und die anschließende Divergenz als Semikonvergenz bezeichnet. Unser Ziel ist ein Kompromiss zwischen Approximation an die gegebenen Abtastwerte bei gleichzeitiger Verhinderung der Verstärkung von Störungen aus den Abtastwerten.

### Mehrstufige Algorithmen

Der tatsächliche Polynomgrad  $N$  ist oft nicht a–priori bekannt. Legen wir uns für einen zu niedrigen Polynomgrad fest, so werden die Daten im Allgemeinen schlecht approximiert. Benutzen wir einen zu hohen Polynomgrad, so können unerwünschte „hochfrequente“ Anteile in der Lösung auftreten, siehe z.B. [47, Abbildung 1].

Die grundlegende Idee besteht im Folgenden darin, in einem *mehrstufigen Algorithmus* (engl. multilevel algorithm) wachsende Polynomgrade zu verwenden und mit einem „geeigneten“ Grad abzurechnen. In [47] wird dazu ein Iterationsverfahren um eine zusätzliche äußere Iteration ergänzt, welche den Polynomgrad erhöht.

Wir formulieren den mehrstufigen Algorithmus, indem wir die Dämpfungsfaktoren  $\hat{\omega}_k$  in Abhängigkeit des Iterationsindex verändern. Die Beschränkung auf einen kleineren Polynomgrad während eines mehrstufigen Algorithmus läßt sich in ihrer Wirkung auf den Vektor  $\hat{f}$  als Dämpfung interpretieren. Das folgende Beispiel 2.22 veranschaulicht den Zusammenhang zwischen einem niedrig gewählten Polynomgrad während eines mehrstufigen Algorithmus und den Dämpfungsfaktoren aus Abschnitt 2.3.

#### Beispiel 2.22. (1–Torus $\mathbb{T}^1$ )

Wir schränken den Polynomgrad  $N = 20$  während eines mehrstufigen Algorithmus auf  $N = 10$  ein, weiterhin verwenden wir als inverse Dämpfungsfaktoren die „Gauß–Gewichte“  $\hat{\omega}_k^{-1} = e^{-k^2/N}$ . Abbildung 7 zeigt die Wirkung im Einzelnen und in Kombination.

□

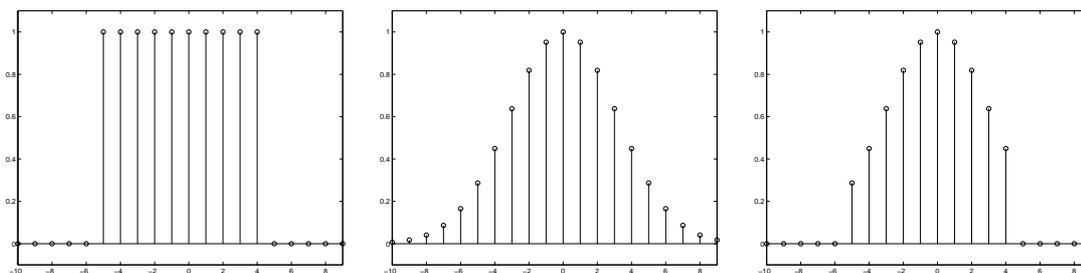


Abbildung 7: Zusammenhang zwischen den Dämpfungsfaktoren und mehrstufigen Algorithmen, links: Verwenden eines niedrigen Polynomgrades, mitte: „Gauß“-Dämpfungsfaktoren, rechts: Kombination der Dämpfungsfaktoren.

### Zusammenfassung

In diesem Abschnitt haben wir uns hauptsächlich mit Fragen der numerischen Lösbarkeit der Gleichung (2.11) auseinander gesetzt. Wir halten Iterationsverfahren zur Lösung der Normalgleichungen für die Methode der Wahl, da sie ohne umfangreiches Vorwissen über die Art und die Größe der Störungen in den Abtastwerten angewendet werden können.

Weiterhin sind wir in der Lage, die mehrstufigen Algorithmen, siehe z.B. [47], durch eine geeignete Veränderung der Dämpfungsfaktoren zwischen einzelnen Iterationsschritten zu realisieren.

## 2.6 Zusammenfassung

Wir untersuchen zu gegebenen Abtastwerten  $f_j = f(\mathbf{x}_j) \in \mathbb{K}$ ,  $j = 0, \dots, M-1$  eines Polynoms

$$f \in L_N, \quad N \in \mathbb{N},$$

womit wir bereits eine Annahme an den Polynomgrad  $N$  formulieren, das gewichtete Approximationsproblem

$$\|S\hat{\mathbf{f}} - \mathbf{f}\|_{\mathbf{W}} \xrightarrow{\hat{\mathbf{f}}} \min$$

und charakterisieren die Lösung eindeutig durch das gedämpfte Minimierungsproblem

$$\|\hat{\mathbf{f}}\|_{\hat{\mathbf{W}}} \xrightarrow{\hat{\mathbf{f}}} \min.$$

Die Matrix  $\mathbf{S} = (\phi_k(\mathbf{x}_j))_{j=0, \dots, M-1; k=0, \dots, N_L-1}$  enthält die Abtastwerte der Basisfunktionen. Die Gewichtsmatrix  $\mathbf{W}$  bezieht Informationen über die Abtastmenge, die Dämpfungsmatrix  $\hat{\mathbf{W}}$  Annahmen an die Glattheit der Lösung ein.

Wir erhalten mittels der Substitution  $\hat{\mathbf{f}}^\omega = \hat{\mathbf{W}}^{\frac{1}{2}} \hat{\mathbf{f}}$ ,  $\mathbf{S}^\omega = \mathbf{W}^{\frac{1}{2}} \mathbf{S} \hat{\mathbf{W}}^{-\frac{1}{2}}$  und  $\mathbf{f}^\omega = \mathbf{W}^{\frac{1}{2}} \mathbf{f}$  die Standardprobleme (2.2) und (2.3).

Bei konsistenter Nebenbedingung, d.h.  $\mathbf{f} \in \mathcal{DL}_N$ , können wir immer die gedämpfte und gewichtete Normalgleichung zweiter Art

$$\mathbf{W}^{\frac{1}{2}} \mathbf{S} \hat{\mathbf{W}}^{-1} \mathbf{S}^H \mathbf{W}^{\frac{1}{2}} \tilde{\mathbf{f}} = \mathbf{W}^{\frac{1}{2}} \mathbf{f}, \quad \hat{\mathbf{f}} = \hat{\mathbf{W}}^{-1} \mathbf{S}^H \mathbf{W}^{\frac{1}{2}} \tilde{\mathbf{f}}$$

aufstellen. Die Lösungen  $\hat{\mathbf{f}}$  dieser Gleichung sind genau die Lösungen der gedämpften Normalgleichung zweiter Art (2.20).

Für einen beliebigen Vektor von Abtastwerten  $\mathbf{f} \in \mathbb{K}^M$  können wir die gedämpfte und gewichtete Normalgleichung erster Art

$$\hat{\mathbf{W}}^{-\frac{1}{2}} \mathbf{S}^H \mathbf{W} \mathbf{S} \hat{\mathbf{W}}^{-\frac{1}{2}} \hat{\mathbf{W}}^{\frac{1}{2}} \hat{\mathbf{f}} = \hat{\mathbf{W}}^{-\frac{1}{2}} \mathbf{S}^H \mathbf{W} \mathbf{f}$$

aufstellen. Die Lösungen  $\hat{\mathbf{f}}$  dieser Gleichung sind genau die Lösungen der gewichteten Normalgleichung erster Art (2.26), falls wir echt positive Dämpfungsfaktoren  $\hat{\omega}_k$ ,  $k = 0, \dots, N_L-1$  verwenden.

In Abschnitt 2.5 haben wir uns für die Verwendung von Iterationsverfahren zur Lösung der Normalgleichungen entschieden. Die dazu notwendigen schnellen Matrix–Vektor–Multiplikationen stellen wir im folgenden Kapitel 3 vor. In Kapitel 4 kommen wir auf das inverse Problem und die damit verbundenen Normalgleichungen zurück.

### 3 Synthese

In diesem Kapitel wiederholen wir die schnellen Algorithmen für die Matrix–Vektor–Multiplikationen  $S\hat{f}$  beziehungsweise  $S^H f$ . Wir legen dabei Wert auf die Interpretation eines schnellen Algorithmus als näherungsweise Faktorisierung der Matrix  $S$  in schwach besetzte Matrizen.

Wir stellen in Abschnitt 3.1 die Fourier–Transformation für beliebige Knoten, in Abschnitt 3.2 die Polynomtransformation und in Abschnitt 3.3 die Fourier–Transformation auf der Sphäre für beliebige Knoten dar. Wir schließen das Kapitel mit einem Überblick über die verfügbaren schnellen Algorithmen ab.

#### 3.1 Diskrete Fourier–Transformation für beliebige Knoten

Die folgenden Resultate sind den Arbeiten [45, 41, 36] entnommen. Wir interessieren uns für den Unterraum der trigonometrischen Polynome, die Vandermonde–ähnliche–Matrix  $S$  hat in Multiindexschreibweise die explizite Darstellung

$$S = \left( e^{-2\pi i \mathbf{k}^T \mathbf{x}_j} \right)_{j=0, \dots, M-1; \mathbf{k} \in I_N^d} \in \mathbb{C}^{M \times N^d}.$$

Die Abbildung die jedem Vektor  $\hat{f} \in \mathbb{C}^{N^d}$  für beliebige Knoten  $\mathbf{x}_j \in \mathbb{T}^d$  den Vektor  $f \in \mathbb{C}^M$  zuordnet, nennen wir *diskrete Fourier–Transformation für beliebige Knoten* (NDFT). Wir unterscheiden zunächst die NDFT

$$f_j = f(\mathbf{x}_j) := \sum_{\mathbf{k} \in I_N^d} \hat{f}_{\mathbf{k}} e^{-2\pi i \mathbf{k}^T \mathbf{x}_j}, \quad j = 0, \dots, M-1, \quad (3.1)$$

kurz  $f = S\hat{f}$ , und die adjungierte NDFT

$$\hat{h}_{\mathbf{k}} := \sum_{j=0}^{M-1} f_j e^{2\pi i \mathbf{k}^T \mathbf{x}_j}, \quad \mathbf{k} \in I_N^d, \quad (3.2)$$

kurz  $\hat{h} = S^H f$ .

Für einen wichtigen Spezialfall mit  $M = N^d$  und den äquidistanten Knoten  $\mathbf{x}_j = \frac{1}{N} \mathbf{j}$ ,  $\mathbf{j} \in I_N^d$ , ist die Vandermonde–Matrix  $S$  die  $d$ –variate Fourier–Matrix. Die Multiplikation mit der Matrix  $S$  heißt *diskrete Fourier–Transformation* (DFT), es gilt  $S^H S = N^d \mathbf{E}$ . Eine schnelle Realisierung der DFT ist durch die *schnelle Fourier–Transformation* (FFT) gegeben.

Die direkte Berechnung von (3.1) oder (3.2) eine arithmetische Komplexität von  $\mathcal{O}(N^d M)$ . Die in der Signalverarbeitung und der Astrophysik bekannten Verfahren der „Gitterinterpolation“ (engl. gridding) stellen eine erste Realisierung der schnellen, approximativen Berechnung der NDFT dar. Eine Reihe von Arbeiten zur schnellen Realisierung der NDFT beschäftigt sich seit etwa einem Jahrzehnt verstärkt mit der Abhängigkeit von Berechnungsgüte (Genauigkeit) und Berechnungszeit (arithmetische Komplexität), siehe [36] und die darin enthaltenen Referenzen. Mit [26] wurde ein frei zugängliches Programmpaket entwickelt. Im Folgenden werden wir unsere Notation an [45, 41, 26, 36] anlehnen.

Die *schnelle Fourier–Transformation für beliebige Knoten* (NFFT) ist ein approximativer Algorithmus mit einer arithmetischen Komplexität von  $\mathcal{O}(N^d \log N + m^d M)$ , wobei der Abschneideparameter  $m$  nur von der gewünschten Genauigkeit abhängt.

## Grundlagen

Die Hauptidee des Algorithmus besteht in der Nutzung einer *Fensterfunktion*  $\varphi$ , welche im Zeit-beziehungsweise Ortsraum  $\mathbb{R}^d$  und im Frequenzraum  $\mathbb{R}^d$  gut lokalisiert ist. Unschärferelationen geben dabei natürliche Grenzen vor.

Sei  $\tilde{\varphi} \in L(\mathbb{T}^d, \mathbb{R})$  die 1-Periodisierung von  $\varphi$ , welche also gut im Zeit-beziehungsweise Ortsraum  $\mathbb{T}^d$  und im Frequenzraum  $\mathbb{Z}^d$  lokalisiert sein soll. Wir approximieren das (unbekannte) trigonometrische Polynom  $f \in L_N$ , welches der Gleichung (3.1) zugrunde liegt, durch eine Linearkombination von Translaten

$$s_1(\mathbf{x}) := \sum_{\mathbf{l} \in I_n^d} g_{\mathbf{l}} \tilde{\varphi}\left(\mathbf{x} - \frac{1}{n}\mathbf{l}\right), \quad (3.3)$$

wobei  $n := \alpha N$ . Der Faktor der Aufwärtsabtastung (engl. oversampling factor)  $\alpha > 1$  wird so gewählt, dass die diskrete Fourier-Transformation der Länge  $n$  effizient berechnet werden kann. Wir entwickeln die Funktion  $s_1$  in ihre Fourier-Reihe

$$s_1(\mathbf{x}) = \sum_{\mathbf{k} \in I_n^d} \hat{g}_{\mathbf{k}} c_{\mathbf{k}}(\tilde{\varphi}) e^{-2\pi i \mathbf{k}^T \mathbf{x}} + \sum_{\mathbf{r} \in \mathbb{Z}^d \setminus \{0\}} \sum_{\mathbf{k} \in I_n^d} \hat{g}_{\mathbf{k}} c_{\mathbf{k} + n\mathbf{r}}(\tilde{\varphi}) e^{-2\pi i (\mathbf{k} + n\mathbf{r})^T \mathbf{x}} \quad (3.4)$$

mit den Fourier-Koeffizienten

$$c_{\mathbf{k}}(\tilde{\varphi}) := \int_{\mathbb{T}^d} \tilde{\varphi}(\mathbf{x}) e^{2\pi i \mathbf{k}^T \mathbf{x}} d\mathbf{x} = \int_{\mathbb{R}^d} \varphi(\mathbf{x}) e^{2\pi i \mathbf{k}^T \mathbf{x}} d\mathbf{x}, \quad \mathbf{k} \in \mathbb{Z}^d$$

und den diskreten Fourier-Koeffizienten

$$\hat{g}_{\mathbf{k}} := \sum_{\mathbf{l} \in I_n^d} g_{\mathbf{l}} e^{2\pi i \mathbf{k}^T \mathbf{l} / n}. \quad (3.5)$$

Vernachlässigen wir die Aliasanteile in Gleichung (3.4), so legt ein Koeffizientenvergleich im Frequenzraum  $I_n^d$  die Definition

$$\hat{g}_{\mathbf{k}} := \begin{cases} \hat{f}_{\mathbf{k}} / c_{\mathbf{k}}(\tilde{\varphi}) & \mathbf{k} \in I_N^d, \\ 0 & \mathbf{k} \in I_n^d \setminus I_N^d \end{cases}$$

nahe. Die Koeffizienten  $g_{\mathbf{l}}$  in Gleichung (3.3) werden dann aus den Koeffizienten  $\hat{g}_{\mathbf{k}}$  mittels einer  $d$ -variaten schnellen Fourier-Transformation der Länge  $n$  berechnet, d.h.,

$$g_{\mathbf{l}} = \frac{1}{n^d} \sum_{\mathbf{k} \in I_n^d} \hat{g}_{\mathbf{k}} e^{-2\pi i \mathbf{k}^T \mathbf{l} / n}.$$

Da die Funktion  $\tilde{\varphi}$  weiterhin gut im  $d$ -Torus  $\mathbb{T}^d$  lokalisiert ist, wird sie durch eine Funktion  $\tilde{\psi}$  mit  $\text{supp } \tilde{\psi} \subseteq \frac{m}{n}\mathbb{T}^d$  ( $m \ll n$ ) approximiert und wir erhalten schließlich

$$f_j = f(\mathbf{x}_j) \approx s_1(\mathbf{x}_j) \approx \sum_{\mathbf{l} \in I_{n,m}(\mathbf{x}_j)} g_{\mathbf{l}} \tilde{\psi}\left(\mathbf{x}_j - \frac{1}{n}\mathbf{l}\right),$$

wobei die Indexmenge  $I_{n,m}(\mathbf{x}_j) := \{\mathbf{l} \in I_n^d : n\mathbf{x}_j - m\mathbf{1} \leq \mathbf{l} \leq n\mathbf{x}_j + m\mathbf{1}\}$  und  $\mathbf{1}$  ein Vektor der Länge  $d$  mit den Elementen eins ist. Die Indexmenge  $I_{n,m}(\mathbf{x}_j)$  enthält höchstens  $(2m+2)^d$  Elemente, wodurch sich insgesamt folgender Algorithmus mit einer arithmetischen Komplexität von  $\mathcal{O}((\alpha N)^d \log(\alpha N) + m^d M)$  ergibt.

```

1: Eingabe:  $N, M, m \in \mathbb{N}, \alpha > 1$ 
2: Vorberechnung:  $c_{\mathbf{k}}(\tilde{\varphi})$  ( $\mathbf{k} \in I_N^d$ )
3:  $\tilde{\psi}\left(\mathbf{x}_j - \frac{\mathbf{l}}{n}\right)$  ( $j \in I_M^1, \mathbf{l} \in I_{n,m}(\mathbf{x}_j)$ )
4: for  $\mathbf{k} \in I_N^d$  do
5:    $\hat{g}_{\mathbf{k}} = \frac{\hat{f}_{\mathbf{k}}}{n^d c_{\mathbf{k}}(\tilde{\varphi})}$ 
6: end for
7:  $\mathbf{g} = \text{FFT}_d(\hat{\mathbf{g}})$ 
8: for  $j = 0, \dots, M - 1$  do
9:    $s_j = \sum_{\mathbf{l} \in I_{n,m}(\mathbf{x}_j)} g_{\mathbf{l}} \tilde{\psi}\left(\mathbf{x}_j - \frac{\mathbf{l}}{n}\right)$ 
10: end for
11: Ausgabe:  $\mathbf{s}$ , wobei  $\mathbf{s} \approx \mathbf{f}$ 

```

**Algorithmus 1:** NFFT**Matrixdarstellung**

In Matrix–Vektor–Notation ist also

$$\mathbf{S} \approx \mathbf{BFR},$$

wobei  $\mathbf{B} \in \mathbb{R}^{M \times n^d}$  die dünnbesetzte Matrix

$$\mathbf{B} := \left( \tilde{\psi}\left(\mathbf{x}_j - \frac{\mathbf{l}}{n}\right) \right)_{j=0, \dots, M; \mathbf{l} \in I_n^d},$$

$\mathbf{F} \in \mathbb{C}^{n^d \times n^d}$  die übliche  $d$ -variate Fourier–Matrix und  $\mathbf{R} \in \mathbb{R}^{n^d \times N^d}$  die Diagonalmatrix

$$\mathbf{R} := \left( \mathbf{O} \mid \text{diag}\left(\frac{1}{n^d c_{\mathbf{k}}(\tilde{\varphi})}\right)_{\mathbf{k} \in I_N^d} \mid \mathbf{O} \right)^T$$

mit den Nullmatrizen  $\mathbf{O} \in \mathbb{R}^{N^d \times (n-N)^d/2}$  bezeichne. Aus der Matrixdarstellung ist insbesondere die adjungierte NFFT abzuleiten, es gilt  $\mathbf{S}^H \approx \mathbf{R}^T \mathbf{F}^H \mathbf{B}^T$ . Details zur Implementierung der NFFT finden wir in [26]. Fehlerabschätzungen, welche einen exponentiellen Abfall des Approximationsfehlers in Abhängigkeit des Abschneideparameters  $m$  nachweisen, werden insbesondere in [45, 41, 36] gegeben.

### 3.2 Diskrete Polynomtransformation

Die folgenden Resultate sind den Arbeiten [8, 33, 39, 35] entnommen. Wir interessieren uns für den Unterraum der (algebraischen) Polynome, die Vandermonde-ähnliche-Matrix  $S$  hat mit den Polynomen  $p_k$ ,  $k = 0, \dots, N - 1$  die explizite Darstellung

$$S = (p_k(x_j))_{j=0, \dots, M-1; k=0, \dots, N-1} \in \mathbb{R}^{M \times N}.$$

Wir leiten im Folgenden analog zu [39] einen schnellen Algorithmus zur Auswertung von Polynomen an speziellen Knoten her. Die Polynome sind dabei durch ihre Fourier-Koeffizienten bezüglich einer Orthonormalbasis gegeben. Nach einer Darstellung der Grundlagen zu orthogonalen Polynomen, veranschaulichen wir die Faktorisierung der Matrix  $S$ , welche einer schnellen Polynomtransformation zugrundeliegt. Eine Verallgemeinerung für beliebige Knoten  $x_j \in [-1, 1]$ ,  $j = 0, \dots, M - 1$ , wird in [35] gegeben.

#### Orthogonale Polynome

Im Weiteren interessieren wir uns für orthogonale Polynome  $p_k$ ,  $k = 0, \dots, N - 1$ , diese genügen bekannterweise einer Dreitermrekursion, siehe z.B. [20, S. 288ff],

$$p_{k+1}(x) = (\alpha_k x + \beta_k) p_k(x) + \gamma_k p_{k-1}(x), \quad k \in \mathbb{N} \quad (3.6)$$

mit  $p_0(x) := c$ ,  $c \in \mathbb{R}$  konstant,  $p_{-1}(x) := 0$  und  $\alpha_k, \beta_k, \gamma_k \in \mathbb{R}$ ,  $k \in \mathbb{N}$ .

#### Clenshaw-Algorithmus

Wir betrachten zunächst ein festes  $x_0 \in [-1, 1]$  und wollen

$$f(x_0) = \sum_{k=0}^{N-1} \hat{f}_k p_k(x_0) \quad (3.7)$$

berechnen. Die grundlegende Idee des folgenden Algorithmus 2 ist es, die Summe in jedem Schritt mittels der Dreitermrekursion um einen Summanden zu verkürzen. Die auf den ortho-

```

1: Eingabe:  $N \in \mathbb{N}$ ,  $N \geq 2$ ,  $\hat{f} \in \mathbb{R}^N$ ,  $x_0 \in \mathbb{R}$ 
2: for  $k = N - 1, \dots, 2$  do
3:    $\hat{f}_{k-1} = \hat{f}_{k-1} + \hat{f}_k (\alpha_k x_0 + \beta_k)$ 
4:    $\hat{f}_{k-2} = \hat{f}_{k-2} + \hat{f}_k \gamma_k$ 
5: end for
6:  $p(x_0) = \hat{f}_0 p_0(x_0) + \hat{f}_1 p_1(x_0)$ 
7: Ausgabe:  $f(x_0)$ 

```

#### Algorithmus 2: Clenshaw-Algorithmus

gonalen Polynomen  $p_k$ ,  $k = 0, \dots, N - 1$ , vorliegende Rekursion wird auf die Koeffizienten  $\hat{f} = (\hat{f}_k)_{k=0, \dots, N-1}$  angewandt, siehe z.B. [20, S. 287]. Graphisch ist ein einzelner solcher Schritt in Abbildung 8 veranschaulicht.



Abbildung 8: Schritt im Clenshaw-Algorithmus.

### Zugeordnete Polynome

Wenden wir die gegebene Dreitermrekursion iteriert an, so erhalten wir eine verallgemeinerte Rekursion. An die Stelle von  $(\alpha_k x_0 + \beta_k)$  und  $\gamma_k$  müssen dann Terme höheren Grades in  $x_0$  treten. Dies wird in folgendem Beispiel 3.1 veranschaulicht.

#### Beispiel 3.1. (Diskrete Polynomtransformation)

Für die Indizes  $k = 8, \dots, 11$  gilt

$$\begin{aligned} p_{10} &= (\alpha_9 x_0 + \beta_9) p_9 + \gamma_9 p_8, \\ p_{11} &= (\alpha_{10} x_0 + \beta_{10}) p_{10} + \gamma_{10} p_9 \\ &= (\alpha_{10} x_0 + \beta_{10}) ((\alpha_9 x_0 + \beta_9) p_9 + \gamma_9 p_8) + \gamma_{10} p_9. \end{aligned}$$

Wenden wir diese Rekursion wiederum auf die Koeffizienten an, so erhalten wir mit

$$\begin{pmatrix} \hat{f}_8^{(1)} \\ \hat{f}_9^{(1)} \end{pmatrix} = \begin{pmatrix} \hat{f}_8 \\ \hat{f}_9 \end{pmatrix} + \begin{pmatrix} \gamma_9 & (\alpha_{10} x_0 + \beta_{10}) \gamma_9 \\ \alpha_9 x_0 + \beta_9 & (\alpha_{10} x_0 + \beta_{10}) (\alpha_9 x_0 + \beta_9) + \gamma_{10} \end{pmatrix} \begin{pmatrix} \hat{f}_{10} \\ \hat{f}_{11} \end{pmatrix}$$

die Gleichung

$$\sum_{k=8}^{11} \hat{f}_k p_k(x_0) = \sum_{k=8}^9 \hat{f}_k^{(1)} p_k(x_0).$$

Die Koeffizienten  $\hat{f}_k^{(1)}$  sind wiederum abhängig von  $x_0$ . Abbildung 9 zeigt die Aufdatierung von  $\hat{f}_8$  und  $\hat{f}_9$ .

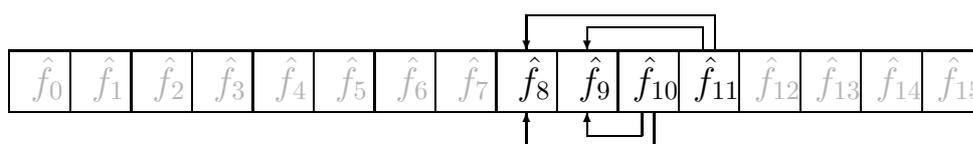


Abbildung 9: Zweifache Anwendung der Dreitermrekursion.

□

Der Zusammenhang zwischen der Dreitermrekursion und der verallgemeinerten Rekursion wird in folgendem Satz 3.2 hergestellt.

**Satz 3.2.** Sei die Familie der Polynome

$$p_{c+1}(x, k) := (\alpha_{c+k} x + \beta_{c+k}) p_c(x, k) + \gamma_{c+k} p_{c-1}(x, k) \quad (k, c \in \mathbb{N}_0) \quad (3.8)$$

mit  $p_{-1}(x, k) := 0$  und  $p_0(x, k) := 1$  den orthogonalen Polynomen  $(p_k)_{k \in \mathbb{N}_0}$  zugeordnet, so gilt, siehe z.B. [39], dass

$$p_{k+c}(x) = p_c(x, k) p_k(x) + \gamma_k p_{c-1}(x, k+1) p_{k-1}(x).$$

*Beweis.* Sei  $k \in \mathbb{N}_0$  beliebig, so gilt für  $c = 0$

$$\begin{aligned} p_k(x) &= 1p_k(x) + \gamma_k 0p_{k-1} \\ &\stackrel{\text{mit (3.8)}}{=} p_0(x, k)p_k(x) + \gamma_k p_{-1}(x, k+1)p_{k-1}(x), \end{aligned}$$

weiterhin für  $c = 1$

$$\begin{aligned} p_{k+1}(x) &\stackrel{\text{mit (3.6)}}{=} (\alpha_k x + \beta_k)p_k(x) + \gamma_k 1p_{k-1}(x) \\ &\stackrel{\text{mit (3.8)}}{=} p_1(x, k)p_k(x) + \gamma_k p_0(x, k+1)p_{k-1}(x) \end{aligned}$$

und im Induktionsschritt schließlich

$$\begin{aligned} p_{k+c+1}(x) &\stackrel{\text{mit (3.6)}}{=} (\alpha_{k+c}x + \beta_{k+c})p_{k+c}(x) + \gamma_{k+c}p_{k+c-1}(x) \\ &\stackrel{\text{mit (IV)}}{=} (\alpha_{k+c}x + \beta_{k+c})(p_c(x, k)p_k(x) + \gamma_k p_{c-1}(x, k+1)p_{k-1}(x)) \\ &\quad + \gamma_{k+c}(p_{c-1}(x, k)p_k(x) + \gamma_k p_{c-2}(x, k+1)p_{k-1}(x)) \\ &= [(\alpha_{k+c}x + \beta_{k+c})p_c(x, k) + \gamma_{k+c}p_{c-1}(x, k)]p_k(x) \\ &\quad + \gamma_k [(\alpha_{k+c}x + \beta_{k+c})p_{c-1}(x, k+1) + \gamma_{k+c}p_{c-2}(x, k+1)]p_{k-1}(x) \\ &\stackrel{\text{mit (3.8)}}{=} p_{c+1}(x, k)p_k(x) + \gamma_k p_c(x, k+1)p_{k-1}(x). \end{aligned}$$

■

Eine direkte Anwendung der verallgemeinerten Rekursion für einzelne Knoten  $x_j$  führt zu keinem effizienten Verfahren, deshalb betrachten wir im Folgenden eine geschicktere Organisation.

### Schnelle Polynomtransformation - Matrixdarstellung

In [39], [33, S. 28ff] wird die schnelle Polynomtransformation ausführlich dargestellt. Wir zeigen im folgenden Beispiel 3.3 einen Ausschnitt des Algorithmus. Dazu seien die Polynome  $\hat{f}_k^{(0)}(x) := \hat{f}_k$ ,  $k = 0, \dots, N-1$ , definiert.

**Beispiel 3.3.** (Faktorisierung von  $S$ )

Wir verwenden die Kurzschreibweisen  $p_{c,j,k} := p_c(x_j, k)$  und  $\hat{f}_{k,j}^{(\tau)} := \hat{f}_k^{(\tau)}(x_j)$ . Seien die Polynome  $\hat{f}_k^{(0)}$ ,  $k = 8, \dots, 11$ , an den Knoten  $x_j$ ,  $j = 0, \dots, 4$ , gegeben, so berechnen wir die Abtastwerte der Polynome  $\hat{f}_k^{(1)}$ ,  $k = 8, 9$ , mit

$$\begin{pmatrix} \hat{f}_{8,0}^{(1)} \\ \hat{f}_{9,0}^{(1)} \\ \hat{f}_{8,1}^{(1)} \\ \hat{f}_{9,1}^{(1)} \\ \hat{f}_{8,2}^{(1)} \\ \hat{f}_{9,2}^{(1)} \\ \hat{f}_{8,3}^{(1)} \\ \hat{f}_{9,3}^{(1)} \\ \hat{f}_{8,4}^{(1)} \\ \hat{f}_{9,4}^{(1)} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \gamma_9 p_{0,0,10} & \gamma_9 p_{1,0,10} \\ 0 & 1 & p_{1,0,9} & p_{2,0,9} \\ & & \dots & \\ & & & 1 & 0 & \gamma_9 p_{0,4,10} & \gamma_9 p_{1,4,10} \\ & & & 0 & 1 & p_{1,4,9} & p_{2,4,9} \end{pmatrix} \begin{pmatrix} \hat{f}_{8,0}^{(0)} \\ \hat{f}_{9,0}^{(0)} \\ \hat{f}_{10,0}^{(0)} \\ \hat{f}_{11,0}^{(0)} \\ \vdots \\ \hat{f}_{8,4}^{(0)} \\ \hat{f}_{9,4}^{(0)} \\ \hat{f}_{10,4}^{(0)} \\ \hat{f}_{11,4}^{(0)} \end{pmatrix}.$$

Das wesentliche, verbleibende Problem um in der dargestellten Weise fortzufahren, besteht in der Berechnung von je  $M^{(2)} = 9$  Abtastwerten der Polynome  $\hat{f}_k^{(1)}$ ,  $k = 8, 9$ .  $\square$

Die Berechnung von weiteren Abtastwerten eines Polynoms aus gegebenen Abtastwerten ist eine wiederkehrende Aufgabe, die in [7] mittels der diskreten Fourier-Transformation, in [22, 39, 25] mittels der diskreten Kosinustransformation und in [48] mittels der schnellen Multiplimethode gelöst wird.

### 3.3 Diskrete Fourier–Transformation auf der Sphäre für beliebige Knoten

Die folgenden Resultate sind den Arbeiten [7, 40, 36, 25] entnommen. Wir interessieren uns für den Unterraum der sphärischen Polynome, die Vandermonde–ähnliche–Matrix  $\mathbf{S}$  hat in Multiindexschreibweise die explizite Darstellung

$$\mathbf{S} = (Y_{\mathbf{k}}(x_j))_{j=0,\dots,M-1; \mathbf{k} \in J_N^2} \in \mathbb{C}^{M \times N^2}.$$

Wir leiten im Folgenden analog zu [25, 24] einen schnellen Algorithmus zur Auswertung von sphärischen Polynomen an beliebigen Knoten her. Die sphärischen Polynome sind dabei durch ihre Fourier–Koeffizienten bezüglich der Standardorthonormalbasis gegeben. Wir zerlegen das Problem in die Teilprobleme der diskreten Polynomtransformation und der diskreten Fourier–Transformation an beliebigen Knoten. Dabei veranschaulichen wir insbesondere die Faktorisierung der Matrix  $\mathbf{S}$ , welche die Grundlage einer schnellen Fourier–Transformation auf der Sphäre für beliebige Knoten ist.

#### Grundlagen

Wir fassen die Ergebnisse aus [24] kurz zusammen, wir berechnen die Elemente des Vektors  $\mathbf{f} = \mathbf{S}\hat{\mathbf{f}}$  wie folgt. Die Abtastwerte der sphärischen Polynome  $f \in L_N(\mathbb{S}^2, \mathbb{C})$  lassen sich darstellen als

$$\begin{aligned} f_j &= \sum_{\mathbf{k} \in J_N^2} \hat{f}_{\mathbf{k}} Y_{\mathbf{k}}(\mathbf{x}_j) \\ &= \sum_{k=0}^N \sum_{n=-k}^k \hat{f}_k^n Y_k^n(\theta_j, \phi_j) \\ &= \sum_{n=-N}^N h_n(\cos \theta_j) e^{in\phi_j}. \end{aligned}$$

Die trigonometrischen Polynome  $h_n(\cos \cdot)$  sind dabei gegeben durch

$$h_n(\cos \theta) := \begin{cases} g_n(\cos \theta) & \text{für gerades } n, \\ (\sin \theta) g_n(\cos \theta) & \text{sonst,} \end{cases} \quad (3.9)$$

wobei die (algebraischen) Polynome  $g_n$  mit

$$g_n(x) := \begin{cases} \sum_{k=|n|}^N \hat{f}_k^n P_k^{|n|}(x) & \text{für gerades } n, \\ \frac{1}{\sqrt{1-x^2}} \sum_{k=|n|}^N \hat{f}_k^n P_k^{|n|}(x) & \text{sonst,} \end{cases}$$

definiert sind. Mit Hilfe der diskreten Polynomtransformation, siehe Abschnitt 3.2, überführen wir für beliebige  $n = -N, \dots, N$  die Polynome  $g_n$  aus der Darstellung bezüglich der zugeordneten Legendre–Funktionen  $P_k^{|n|}$  mit den Fourier–Koeffizienten  $\hat{f}_k^n$ ,  $k = |n|, \dots, N$  in

eine Darstellung bezüglich der Chebyshev–Polynome erster Art  $T_k$ . Da weiterhin  $T_k(\cos \theta) = \cos(k\theta)$  gilt, erhalten wir schließlich

$$f_j = \sum_{(k,n)^T \in I_{2N+2}^2} c_k^n e^{i(k\theta_j + n\phi_j)} = \sum_{\mathbf{k} \in I_{2N+2}^2} c_k^n e^{-2\pi i \mathbf{k}^T \mathbf{x}_j},$$

wobei  $\mathbf{x}_j = -\frac{1}{2\pi} (\theta_j, \phi_j)^T$ ,  $j = 0, \dots, M-1$ . Die verbleibende Aufgabe ist eine bivariate diskrete Fouriertransformation für beliebige Knoten.

### Matrixdarstellung

Berechnen wir das (algebraische) Polynom  $g_n$  an den Chebyshev–Knoten  $(\cos(\frac{j\pi}{N}))_{j=0,\dots,N}$ , so bezeichnen wir die entsprechende Vandermonde–Matrix mit  $\mathbf{S}^{\text{DPT}_n}$ , die Darstellung des Polynoms  $g_n$  bezüglich der Basis der Chebyshev–Polynome erster Art erhalten wir mittels einer geeigneten diskreten Kosinustransformation, hier mit der Matrix  $\mathbf{C}$  bezeichnet. Für den in [24] dargestellten Übergang zu Fourier–Koeffizienten verwenden wir die Matrix  $\mathbf{D}_1$ . Wir erhalten für die skalierten Knoten  $\mathbf{x}_j = -\frac{1}{2\pi} (\theta_j, \phi_j)^T$ ,  $j = 0, \dots, M-1$ , mithilfe der NFFT–Matrix  $\mathbf{S}^{\text{NFFT}_2} \in \mathbb{C}^{M \times (2N+2)^2}$  schließlich die Faktorisierung

$$\mathbf{S} = \mathbf{S}^{\text{NFFT}_2} \mathbf{D}_1 (\mathbf{E} \otimes \mathbf{C}) \begin{pmatrix} \mathbf{S}^{\text{DPT}_{-N}} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{S}^{\text{DPT}_N} \end{pmatrix}.$$

### 3.4 Zusammenfassung

Wir haben in diesem Kapitel schnelle Algorithmen für die direkten Probleme der diskreten Fourier–Transformation dargestellt. Aufgrund der Vielzahl der Algorithmen schlagen wir wie in [36] folgende Namenskonventionen vor. Die Abkürzungen setzen sich aus den englischen Begriffen cosine – C, discrete – D, fast/Fourier – F, nonequispaced – N, polynomial – P, sine – S und transform – T zusammen. Verfügbare Implementierungen sind mit  $\checkmark$  für die Multiplikation mit der Matrix  $S$  und mit  $^H$  für die Multiplikation mit  $S^H$  gekennzeichnet.

$d$ –Torus	2–Sphäre	Intervall (polynomiell)	Intervall (trigonometrisch)
DFT $\checkmark, ^H$	DSFT $\checkmark, ^H$	DPT $\checkmark, ^H$	DCT, DST $\checkmark, ^H$
FFT $\checkmark, ^H$	FSFT $\checkmark, (^H)$	FPT $\checkmark, (^H)$	FCT, FST $\checkmark, ^H$
NDFT $\checkmark, ^H$	NDSFT $\checkmark, ^H$	NDPT $\checkmark, ^H$	NDCT, NDST $\checkmark, ^H$
NFFT $\checkmark, ^H$	NFSFT $\checkmark$	NFPT $\checkmark$	NFCT, NFST

Tabelle 1: Namenskonventionen für die direkten Algorithmen zur diskreten Fourier–Transformation

## 4 Analyse

In diesem Kapitel beschäftigen wir uns mit der iterativen Lösung von Normalgleichungen. Direkte Verfahren zur Lösung der Normalgleichungen verbieten sich häufig aufgrund der Rechenzeit- und Speicherplatzanforderungen, die Gauß-Elimination als bekanntester Vertreter direkter Verfahren hat beispielsweise eine arithmetischen Komplexität von  $\mathcal{O}(n^3)$  und einen Speicherplatzbedarf von  $\mathcal{O}(n^2)$  bei  $n$  Unbekannten.

Demgegenüber kommen Iterationsverfahren meist ohne die Kenntnis der Einträge der Matrix  $S$  aus, haben also einen niedrigeren Speicherplatzbedarf und sind mit der Nutzung der schnellen Matrix-Vektor-Multiplikationen, siehe Kapitel 3, sehr schnell. Da die Algorithmen zur Matrix-Vektor-Multiplikation approximativer Natur sind, kann nur ein mit dem Gleichungssystem (2.11) verwandtes System (engl. nearby system) gelöst werden. Von dieser Einschränkung wird zunächst abgesehen, in Kapitel 6 kommen wir kurz darauf zurück. Ein wesentlicher Vorteil von Iterationsverfahren liegt in der Erzeugung zunächst besser werdender Approximationen aus einer (schlechten) Approximation.

Nach der Einführung grundlegender Begriffe im folgenden Abschnitt 4.1 stellen wir in Abschnitt 4.2 einfache iterative Verfahren vor. Daran anschließend stellen wir in Abschnitt 4.3 die sehr effizienten CG-Typ-Verfahren vor. Abschnitt 4.4 ordnet die in [11, 9] und [46, 12, 47] vorgeschlagenen Verfahren zur inversen Fourier-Transformation ein. Wir geben schließlich in Abschnitt 4.5 eine kurze Zusammenfassung der vorgeschlagenen Algorithmen.

### 4.1 Grundlagen

Durch die Formulierung von Approximations- und Minimierungsproblemen in Kapitel 2 haben wir mithilfe der Standard-Substitutionen  $\hat{\mathbf{f}}^\omega = \hat{\mathbf{W}}^{\frac{1}{2}} \hat{\mathbf{f}}$ ,  $\mathbf{S}^\omega = \mathbf{W}^{\frac{1}{2}} \mathbf{S} \hat{\mathbf{W}}^{-\frac{1}{2}}$ ,  $\mathbf{f}^\omega = \mathbf{W}^{\frac{1}{2}} \mathbf{f}$  die Normalgleichungen, siehe Abschnitt 2.6, erster Art

$$\mathbf{S}^{\omega H} \mathbf{S}^\omega \hat{\mathbf{f}}^\omega = \mathbf{S}^{\omega H} \mathbf{f}^\omega$$

und zweiter Art

$$\mathbf{S}^\omega \mathbf{S}^{\omega H} \tilde{\mathbf{f}}^\omega = \mathbf{f}^\omega, \quad \hat{\mathbf{f}}^\omega = \mathbf{S}^{\omega H} \tilde{\mathbf{f}}^\omega$$

erhalten. Sie beziehen durch die Gewichtung bereits die verfügbaren Informationen über die Abtaststellen und durch die Dämpfung weitere Annahmen über die Glattheit der Lösung ein.

Ausgehend von den gedämpften und gewichteten Normalgleichungen formulieren wir verschiedene Verfahren, wobei wir immer die *Originalvektoren*  $\hat{\mathbf{f}}_l$  iterieren. Wir definieren zur  $l$ -ten Iterierten  $\hat{\mathbf{f}}_l$  der folgenden Verfahren den *Originalresiduenvektor*

$$\mathbf{r}_l = \mathbf{f} - \mathbf{S} \hat{\mathbf{f}}_l,$$

den *Residuenvektor der gewichteten Normalgleichung erster Art*

$$\hat{\mathbf{z}}_l = \mathbf{S}^H \mathbf{W} \mathbf{r}_l$$

und zur  $l$ -ten Iterierten  $\tilde{\mathbf{f}}_l$  der Normalgleichung zweiter Art den *Residuenvektor der gedämpften Normalgleichung zweiter Art*

$$\tilde{\mathbf{z}}_l = \mathbf{f} - \mathbf{S} \hat{\mathbf{W}}^{-1} \mathbf{S}^H \tilde{\mathbf{f}}_l.$$

Mit dem Zusammenhang  $\hat{\mathbf{f}}_l = \hat{\mathbf{W}}^{-1} \mathbf{S}^H \tilde{\mathbf{f}}_l$  gilt natürlich  $\tilde{\mathbf{z}}_l = \mathbf{r}_l$ . Da jedoch bei iterativen Verfahren für die Normalengleichung erster Art statt des Residuenvektors der gewichteten Normalengleichung erster Art  $\hat{\mathbf{z}}_l$  der Originalresiduenvektor  $\mathbf{r}_l$  iteriert wird, assoziieren wir den Vektor  $\mathbf{r}_l$  mit der Normalengleichung erster Art und den Vektor  $\tilde{\mathbf{z}}_l$  mit der Normalengleichung zweiter Art.

## 4.2 Einfache Verfahren

Wir betrachten in diesem Abschnitt die gedämpfte und gewichtete Normalgleichung erster Art, für sie wurden einige Verfahren vorgeschlagen, die wir hier einordnen.

### Einfache Approximation

Mit Gleichung (2.10) lassen sich die Fourier-Koeffizienten direkt durch die Anwendung des adjungierten Operators  $S^H$  berechnen, werden die auftretenden Integrale einfach diskretisiert, so erhalten wir

$$\hat{f} = S^H f \approx S^H W f.$$

Die gewichtete Rechteckregel fällt in diese Kategorie. Im Kontext der nachfolgenden Approximationen für  $\hat{f}$  ist

$$\hat{f}_0 = S^H W f$$

ein möglicher Startwert. Wird eine Dämpfung der Fourier-Koeffizienten gewünscht, so kann die Approximation  $\hat{f}_0$  einfach mit der Dämpfungsmatrix  $\hat{W}^{-1}$  multipliziert werden.

Anwendung findet diese Approximation in der Computertomographie (CT), siehe z.B. [37], [36, Bemerkung 3.4], und der Magnetresonanztomographie (MRT), siehe z.B. [13].

### Lineare Iteration

Wir betrachten zunächst nur die gewichtete Normalgleichung erster Art. Approximieren wir  $(S^H W S)^{-1}$  mittels der  $l$ -ten Neumann-Summe, siehe Lemma 2.2, so folgt für die Iterierte  $\hat{f}_l$  induktiv der Zusammenhang

$$\hat{f}_{l+1} = \hat{f}_l + \alpha \hat{z}_l. \quad (4.1)$$

Diese Iteration nennen wir *Landweber-Iteration*. Sie ist *linear*, da  $(\hat{f} + \hat{g})_l = \hat{f}_l + \hat{g}_l$  und *stationär*, da der Parameter  $\alpha$  nicht vom Iterationsindex  $l$  abhängig ist. Die in der Literatur, siehe z.B. [21, S. 138], [42, S. 234], [12], verwendeten Bezeichnungen sind Landweber-Iteration, (stationäres) Richardson-, Cimino-, Picard-, Fridman-Verfahren und frame-algorithm (engl.). Wir fassen die wichtigsten Eigenschaften des Verfahrens in folgendem Satz 4.1 zusammen.

#### Satz 4.1. (Landweber-Iteration)

Sei die Abtastmenge  $X$  stabil und die Gewichte  $\omega_j > 0$ ,  $j = 0, \dots, M-1$ , beliebig, so ist die Matrix  $S^H W S$  positiv definit. Mit den optimalen Konstanten  $\xi = \lambda_{\min}(S^H W S)$  und  $\Xi = \lambda_{\max}(S^H W S)$  gilt:

1. Die Iteration (4.1) konvergiert genau dann, wenn  $0 < \alpha < \frac{2}{\Xi}$ .
2. Der Spektralradius, siehe z.B. [28, Kapitel 2], der Iterationsmatrix  $E - \alpha S^H W S$  wird minimal für

$$\alpha = \alpha_{\text{opt}} = \frac{2}{\xi + \Xi}.$$

3. Für  $0 < \alpha < \frac{2}{\Xi}$  ist die Iteration (4.1) ein lineares Regularisierungsverfahren.

*Beweis.* Siehe z.B. [28, S. 82ff] und [27, S. 108ff]. ■

Als Anwendung des Satzes 2.18 und des Satzes 4.1 erhalten wir beispielsweise folgendes Lemma 4.2.

**Lemma 4.2.** (1-Torus  $\mathbb{T}^1$ )

Unter den Voraussetzungen von Satz 2.18 und  $0 < \alpha < \frac{2}{(1+\delta N)^2}$  konvergiert die Iteration (4.1) gegen die Lösung der gewichteten Normalengleichung erster Art (2.26). Die Eigenwerte der Iterationsmatrix  $\mathbf{E} - \alpha \mathbf{S}^H \mathbf{W} \mathbf{S}$  liegen im Intervall  $[1 - \alpha(1 + \delta N)^2, 1 - \alpha(1 - \delta N)^2]$ .

*Beweis.* Siehe [16, S. 143]. ■

Unter Einbeziehung der Dämpfung erhalten wir den Algorithmus 3.

```

1: Eingabe:  $\mathbf{f} \in \mathbb{C}^M, \hat{\mathbf{f}}_0 \in \mathbb{C}^{N_L}$ 
2:  $\mathbf{r}_0 = \mathbf{f} - \mathbf{S}\hat{\mathbf{f}}_0$ 
3:  $\hat{\mathbf{z}}_0 = \mathbf{S}^H \mathbf{W} \mathbf{r}_0$ 
4: for  $l = 0, \dots$  do
5:    $\hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha^\omega \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l$ 
6:    $\mathbf{r}_{l+1} = \mathbf{f} - \mathbf{S}\hat{\mathbf{f}}_{l+1}$ 
7:    $\hat{\mathbf{z}}_{l+1} = \mathbf{S}^H \mathbf{W} \mathbf{r}_{l+1}$ 
8: end for
9: Ausgabe:  $\hat{\mathbf{f}}_l$ 

```

**Algorithmus 3:** LANDWEBER

Das Hauptproblem des Verfahrens liegt in seiner langsamen Konvergenz. Diese wird hervorgerufen durch eine nur schlecht mögliche optimale Wahl des Parameters  $\alpha$  und die nicht optimale Wahl der „Suchrichtung“ mit  $\hat{\mathbf{z}}_l$ , siehe z.B. [12, 16, 46]. Beide Probleme werden im Weiteren behoben.

### Gradientenverfahren

Behalten wir die „Suchrichtung“  $\hat{\mathbf{z}}_l$  zunächst noch bei, passen die Schrittweite  $\alpha$  jedoch in jedem Schritt an, so erhalten wir die Iteration

$$\hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha_l \hat{\mathbf{z}}_l, \quad (4.2)$$

wobei die Schrittweite

$$\alpha_l^\omega = \frac{\hat{\mathbf{z}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l}{\mathbf{v}_l^H \mathbf{W} \mathbf{v}_l}$$

mit  $\mathbf{v}_l = \mathbf{S} \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l$  in jedem Schritt des Verfahrens optimal ist, siehe z.B. [28, S. 112]. Wir erhalten das *Gradientenverfahren*, siehe Algorithmus 4.

1: Eingabe:  $\mathbf{f} \in \mathbb{C}^M$ ,  $\hat{\mathbf{f}}_0 \in \mathbb{C}^{N_L}$   
2:  $\mathbf{r}_0 = \mathbf{f} - \mathbf{S}\hat{\mathbf{f}}_0$   
3:  $\hat{\mathbf{z}}_0 = \mathbf{S}^H \mathbf{W} \mathbf{r}_0$   
4: **for**  $l = 0, \dots$  **do**  
5:    $\mathbf{v}_l = \mathbf{S} \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l$   
6:    $\alpha_l^\omega = \frac{\hat{\mathbf{z}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l}{\mathbf{v}_l^H \mathbf{W} \mathbf{v}_l}$   
7:    $\hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha_l^\omega \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l$   
8:    $\mathbf{r}_{l+1} = \mathbf{r}_{l+1} - \alpha_l^\omega \mathbf{v}_l$   
9:    $\hat{\mathbf{z}}_{l+1} = \mathbf{S}^H \mathbf{W} \mathbf{r}_{l+1}$   
10: **end for**  
11: Ausgabe:  $\hat{\mathbf{f}}_l$

**Algorithmus 4:** STEEPEST\_DESCENT

### 4.3 CG–Verfahren

Das *Verfahren der konjugierten Gradienten (CG–Verfahren)* wurde von Hestens und Stiefel 1952 vorgeschlagen, zunächst als direktes Verfahren zur Lösung hermitescher positiv definierter Gleichungssysteme angesehen und erlangte später große Bedeutung als iteratives Lösungsverfahren, siehe z.B. [28, 43]. Zur Anwendung auf Ausgleichsprobleme siehe [19, 2], für die Verwendung von CG–Typ–Verfahren bei der Approximation mit trigonometrischen Polynomen siehe z.B. [34].

Gegen das Formulieren der Normalgleichung erster Art und die Anwendung des CG–Verfahrens wird häufig eingewendet, dass für die Konditionszahl  $\text{cond}_2(\mathbf{S}^H \mathbf{S}) = \text{cond}_2(\mathbf{S})^2$  gilt und damit Standardaussagen für die Konvergenz des CG–Verfahren „schlechter“ werden, siehe z.B. [28, S. 126]. Für das ursprüngliche unter- bzw. überbestimmte Problem (2.11) sind uns jedoch keine iterativen Algorithmen bekannt, welche eine Konvergenzaussage in Abhängigkeit von einer Kondition  $\text{cond}(\mathbf{S})$  machen.

#### Krylov–Räume, CGNR und CGNE

Im Weiteren formulieren wir die Minimierungseigenschaften der nachfolgenden CG–Verfahren. Wir folgen der Notation in [2, S. 288ff], wobei wir die Substitutionen bezüglich der Gewichtung und der Dämpfung wieder zugunsten der Originalvektoren rückgängig machen. Mithilfe der Standardsubstitutionen ergeben sich weiterhin  $\mathbf{r}_l^\omega = \mathbf{W}^{\frac{1}{2}} \mathbf{r}_l$  und  $\hat{\mathbf{z}}_l^\omega = \hat{\mathbf{W}}^{\frac{1}{2}} \hat{\mathbf{z}}_l$ .

Wir betrachten die gedämpften und gewichteten *Krylov–Räume*

$$\begin{aligned} \mathcal{K}_l^\omega(\mathbf{S}, \hat{\mathbf{r}}_0) &:= \hat{\mathbf{W}}^{-\frac{1}{2}} \mathcal{K}_l(\mathbf{S}^\omega H \mathbf{S}^\omega, \hat{\mathbf{z}}_0^\omega) \\ &= \hat{\mathbf{W}}^{-\frac{1}{2}} \text{span} \left( \hat{\mathbf{z}}_0^\omega, \mathbf{S}^\omega H \mathbf{S}^\omega \hat{\mathbf{z}}_0^\omega, \dots, (\mathbf{S}^\omega H \mathbf{S}^\omega)^{l-1} \hat{\mathbf{z}}_0^\omega \right) \end{aligned}$$

und suchen in den affinen Teilräumen  $\hat{\mathbf{f}}_0 + \mathcal{K}_l^\omega(\mathbf{S}, \hat{\mathbf{r}}_0) \subset \mathbb{K}^{N_L}$  Lösungen, welche die Fehlerfunktionale

$$E_0^\omega(\hat{\mathbf{f}}_l) := \|\hat{\mathbf{f}}^\dagger - \hat{\mathbf{f}}_l\|_{\hat{\mathbf{W}}}, \quad E_1^\omega(\hat{\mathbf{f}}_l) := \|\mathbf{r}_l\|_{\mathbf{W}} - \|\mathbf{r}^\dagger\|_{\mathbf{W}}$$

minimieren. Mit den in [2] angegebenen Fehlerfunktionalen

$$E_0(\hat{\mathbf{f}}_l^\omega) := \|\hat{\mathbf{f}}^{\omega\dagger} - \hat{\mathbf{f}}_l^\omega\|_2, \quad E_1(\hat{\mathbf{f}}_l^\omega) := \|\hat{\mathbf{f}}^{\omega\dagger} - \hat{\mathbf{f}}_l^\omega\|_{\mathbf{S}^\omega H \mathbf{S}^\omega} = \|\mathbf{r}^{\omega\dagger} - \mathbf{r}_l^\omega\|_2$$

gilt der Zusammenhang

$$E_0^\omega(\hat{\mathbf{f}}_l) = E_0(\hat{\mathbf{f}}_l^\omega), \quad E_1^\omega(\hat{\mathbf{f}}_l) = E_1(\hat{\mathbf{f}}_l^\omega).$$

Die Iterierten  $\hat{\mathbf{f}}_l = \hat{\mathbf{W}}^{-1} \mathbf{S}^H \tilde{\mathbf{f}}_l$  des folgenden Verfahrens *CGNE* (engl. **C**onjugated **G**radient (applied to the) **N**ormal (equation), **E**rror (minimisation); Benennung z.B. durch Saad in [42, Kapitel 8]) minimieren das Fehlerfunktional  $E_0^\omega$  über den affinen Teilräumen  $\hat{\mathbf{f}}_0 + \mathcal{K}_l^\omega(\mathbf{S}, \hat{\mathbf{r}}_0)$ , allerdings nur, falls die Gleichung (2.11) konsistent ist. Das CGNE–Verfahren ist äquivalent mit der Anwendung des CG–Verfahrens auf die gedämpfte und gewichtete Normalgleichung zweiter Art.

Unglücklicherweise ist die Namensgebung für dieses Verfahren nicht einheitlich, so verwendet Hanke in [19] CGME (engl. Conjugated Gradient Minimal Error) und Björck in [2] CGLS

(engl. Conjugated Gradient for Least Squares ( $\mu = 0$ )) und weist auf die weitere Bezeichnung Craigs-Method hin.

Das zweite für uns interessante Verfahren heißt *CGNR* (engl. Conjugated Gradient (applied to the) Normal (equation), Residual (minimisation); Benennung z.B. durch Saad in [42, Kapitel 8]). Die Iterierten  $\hat{\mathbf{f}}_l$  minimieren das Fehlerfunktional  $E_1^\omega$  über den gleichen affinen Teilräumen  $\hat{\mathbf{f}}_0 + \mathcal{K}_l^\omega(\mathbf{S}, \hat{\mathbf{r}}_0)$ . Das CGNR-Verfahren ist äquivalent mit der Anwendung des CG-Verfahrens auf die gedämpfte und gewichtete Normalgleichung erster Art. Weitergehende Eigenschaften des CGNR-Verfahren finden wir in [19, S. 16f], [20, Bemerkung 15.2] und [2]. So ist das CGNR-Verfahren äquivalent zu einem MR-Verfahren (engl. Minimal Residual method) für die gedämpfte und gewichtete Normalgleichung zweiter Art, der durch das CGNE-Verfahren minimierte Fehler  $\|\hat{\mathbf{f}}_l\|_{\hat{\mathbf{W}}}$  fällt beim CGNR-Verfahren monoton und die Iterierten streben gegen die Lösung der Normalgleichung erster Art, welche zum Startwert  $\hat{\mathbf{f}}_0$  minimalen (gedämpften) Abstand hat.

Auch hier gibt es wieder verschiedene Bezeichnungen für das Verfahren, so nennt es Björck CGLS (engl. Conjugated Gradient for Least Squares ( $\mu = 1$ )) und Hanke in [19] CGNE (engl. Conjugated Gradient Normal Equation). Darüberhinaus haben Paige und Saunders mit LSQR eine mathematisch äquivalente Implementierung vorgeschlagen.

Wir formulieren die Algorithmen 5 und 6, wobei durch den engen Zusammenhang der Verfahren immer die Iterierte des jeweils anderen Verfahrens mitberechnet werden kann, siehe dazu [19, S. 22ff]. Es sind beide Algorithmen angegeben und implementiert, siehe auch Kapitel 5. Wir halten dies für sinnvoll, da in endlicher Arithmetik die mitberechneten Iterierten nicht mit ihren Äquivalenten übereinstimmen müssen und weiterhin nur die iterierten Residuen eines Verfahrens zur Verfügung stehen.

### Zusammenfassung

Die beiden vorgestellten CG-Typ-Verfahren erzeugen Iterierte aus dem gleichen affinen Krylov-Raum  $\hat{\mathbf{f}}_0 + \mathcal{K}_l^\omega(\mathbf{S}, \hat{\mathbf{r}}_0)$ . Die Wahl des Krylov-Raum steuern wir mit der Gewichtsmatrix  $\mathbf{W}$  und der Dämpfungsmatrix  $\hat{\mathbf{W}}$ . Das CGNE-Verfahren minimiert in jedem Schritt den gedämpften Fehler

$$\|\hat{\mathbf{f}}^\dagger - \hat{\mathbf{f}}_l\|_{\hat{\mathbf{W}}},$$

das CGNR-Verfahren das gewichtete Residuum

$$\|\mathbf{r}_l\|_{\mathbf{W}}.$$

Das CGNR-Verfahren verallgemeinert insbesondere den ACT-Algorithmus, siehe Abschnitt 4.4.

```

1: Eingabe:  $\mathbf{f} \in \mathbb{C}^M$ ,  $\hat{\mathbf{f}}_0 \in \mathbb{C}^{N_L}$ 
2: if ITERATE_2nd then
3:    $\hat{\mathbf{f}}_0^{\text{cgne}} = \hat{\mathbf{f}}_0$ 
4: end if
5:  $\mathbf{r}_0 = \mathbf{f} - \mathbf{S}\hat{\mathbf{f}}_0$ 
6:  $\hat{\mathbf{z}}_0 = \mathbf{S}^H \mathbf{W} \mathbf{r}_0$ 
7:  $\hat{\mathbf{p}}_0 = \hat{\mathbf{z}}_0$ 
8: for  $l = 0, \dots$  do
9:    $\mathbf{v}_l = \mathbf{S}\hat{\mathbf{W}}^{-1} \hat{\mathbf{p}}_l$ 
10:   $\alpha_l^\omega = \frac{\hat{\mathbf{z}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l}{\mathbf{v}_l^H \mathbf{W} \mathbf{v}_l}$ 
11:   $\hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha_l^\omega \hat{\mathbf{W}}^{-1} \hat{\mathbf{p}}_l$ 
12:  if ITERATE_2nd then
13:     $\alpha_l^{\text{cgne},\omega} = \frac{\mathbf{r}_l^H \mathbf{W} \mathbf{r}_l}{\hat{\mathbf{z}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l}$ 
14:     $\hat{\mathbf{f}}_{l+1}^{\text{cgne}} = \hat{\mathbf{f}}_l^{\text{cgne}} + \alpha_l^{\text{cgne},\omega} \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l$ 
15:  end if
16:   $\mathbf{r}_{l+1} = \mathbf{r}_l - \alpha_l^\omega \mathbf{v}_l$ 
17:   $\hat{\mathbf{z}}_{l+1} = \mathbf{S}^H \mathbf{W} \mathbf{r}_{l+1}$ 
18:   $\beta_l^\omega = \frac{\hat{\mathbf{z}}_{l+1}^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_{l+1}}{\hat{\mathbf{z}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l}$ 
19:   $\hat{\mathbf{p}}_{l+1} = \beta_l^\omega \hat{\mathbf{p}}_l + \hat{\mathbf{z}}_{l+1}$ 
20: end for
21: Ausgabe:  $\hat{\mathbf{f}}_l$ ,  $\hat{\mathbf{f}}_l^{\text{cgne}}$ 

```

Algorithmus 5: CGNR\_E

```

1: Eingabe:  $\mathbf{f} \in \mathbb{C}^M, \hat{\mathbf{f}}_0 \in \mathbb{C}^{N_L}$ 
2: if ITERATE_2nd then
3:    $\gamma_0^\omega = 1$ 
4:    $\hat{\mathbf{f}}_0^{\text{cgnr}} = \hat{\mathbf{f}}_0$ 
5: end if
6:  $\tilde{\mathbf{z}}_0 = \mathbf{f} - \mathbf{S}\hat{\mathbf{f}}_0$ 
7:  $\hat{\mathbf{p}}_0 = \mathbf{S}^H \mathbf{W} \tilde{\mathbf{z}}_0$ 
8: for  $l = 0, \dots$  do
9:    $\alpha_l^\omega = \frac{\tilde{\mathbf{z}}_l^H \mathbf{W} \tilde{\mathbf{z}}_l}{\hat{\mathbf{p}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{p}}_l}$ 
10:   $\hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha_l^\omega \hat{\mathbf{W}}^{-1} \hat{\mathbf{p}}_l$ 
11:   $\tilde{\mathbf{z}}_{l+1} = \tilde{\mathbf{z}}_l - \alpha_l^\omega \mathbf{S} \hat{\mathbf{W}}^{-1} \hat{\mathbf{p}}_l$ 
12:   $\beta_l^\omega = \frac{\tilde{\mathbf{z}}_{l+1}^H \mathbf{W} \tilde{\mathbf{z}}_{l+1}}{\tilde{\mathbf{z}}_l^H \mathbf{W} \tilde{\mathbf{z}}_l}$ 
13:  if ITERATE_2nd then
14:     $\gamma_{l+1}^\omega = \beta_l^\omega \gamma_l^\omega + 1$ 
15:     $\hat{\mathbf{f}}_{l+1}^{\text{cgnr}} = \frac{1}{\gamma_{l+1}^\omega} \left( \beta_l^\omega \gamma_l^\omega \hat{\mathbf{f}}_l^{\text{cgnr}} + \hat{\mathbf{f}}_l \right)$ 
16:  end if
17:   $\hat{\mathbf{p}}_{l+1} = \beta_l^\omega \hat{\mathbf{p}}_l + \mathbf{S}^H \mathbf{W} \tilde{\mathbf{z}}_{l+1}$ 
18: end for
19: Ausgabe:  $\hat{\mathbf{f}}_l, \hat{\mathbf{f}}_l^{\text{cgnr}}$ 

```

Algorithmus 6: CGNE\_R

#### 4.4 Weitere Verfahren

Bisher haben die spezielle Struktur der Vandermonde-ähnlichen-Matrix ausgenutzt. Nur für den Nachweis der Verwendbarkeit von iterativen Verfahren haben wir auch die Eigenschaften der Matrizen in den Normalgleichungen genutzt.

Im Weiteren wollen wir neben einer in [9, 11] vorgeschlagenen direkten inversen Fourier-Transformation für  $\mathbb{T}^1$  insbesondere die speziellen Strukturen der Matrizen in den Normalgleichungen klären.

##### Faktorisierung der Matrix $S^{-1}$ für den 1-Torus $\mathbb{T}^1$

Für den univariaten Fall und die Vandermonde-Matrix  $S$  wird in [9, 11] eine Approximation an die Inverse  $S^{-1}$  vorgeschlagen. Ersetzen wir bei der approximativen Matrix-Vektor-Multiplikation die schnelle Multipolmethode (engl. fast multipol method, FMM) durch den entsprechenden, auf der NFFT beruhenden Algorithmus zur schnellen Summation, siehe [38], so erhalten wir mit geeigneten Diagonalmatrizen  $D_1$ ,  $D_2$ ,  $D_3$ , einer Bandmatrix  $N$  und der gewöhnlichen Fourier-Matrix  $F$  die Approximation

$$S^{-1} \approx F^H D_1 (F D_2 S^H + N) D_3.$$

Numerische Ergebnisse in [9] zeigen, dass diese Algorithmen numerisch instabil sind, falls einige Knoten besonders eng zusammen liegen oder die Abtastmenge größere „Löcher“ aufweist. Abgesehen vom Tensorproduktfall sind uns keine Verallgemeinerungen für die multivariaten Fälle  $d > 1$  bekannt.

##### (P)ACT-Algorithmus für den $d$ -Torus $\mathbb{T}^d$

Wir betrachten im Folgenden die gewichtete Normalgleichung erster Art. Unser weiteres Vorgehen führt zu dem in [12, 16, 17, 46] entwickelten Verfahren (P)ACT (engl. (Preconditioned) Adaptive weights Conjugate gradient Toeplitz method).

Die Matrix  $T^\omega = S^H W S$  ist wie bereits erwähnt hermitesch und positiv semidefinit. Für den 1-Torus  $\mathbb{T}^1$  ist die Matrix  $T^\omega$  weiterhin eine Toeplitz-Matrix mit den Einträgen

$$T_{k_1, k_2}^\omega = \sum_{j=0}^{M-1} \omega_j e^{-2\pi i (k_2 - k_1) x_j}.$$

Für den  $d$ -Torus  $\mathbb{T}^d$ ,  $d > 1$ , ist sie eine Mehrstufen-Toeplitz-Matrix. Wir formulieren die gewichtete Normalgleichung erster Art als das lineare Gleichungssystem

$$T^\omega \hat{f} = S^H W f. \quad (4.3)$$

Eine Matrix-Vektor-Multiplikation  $T^\omega \hat{p}$  kann mittels der Einbettung in eine zirkulante Matrix effizient berechnet werden kann. Die Dimension  $N_L$  der Matrix müssen wir dabei nicht auf eine Zweierpotenz einschränken, eine Matrix-Vektor-Multiplikation hat eine arithmetische Komplexität von  $\mathcal{O}(N_L \log N_L)$ . Der folgende Algorithmus 7 zeigt den „Standard-CG-Algorithmus“, d.h. ohne Vorkonditionierung, angewandt auf das Gleichungssystem (4.3).

Dieses Verfahren formuliert die gewichtete Normalgleichung erster Art explizit. Wir sehen für die mit Maschinengenauigkeit aufgestellten Gleichungssysteme die folgenden numerischen Nachteile: In [2, S. 293] wird sich gegen eine Rekursion basierend auf den Residuenvektoren der gewichteten Normalgleichung erster Art  $\hat{z}_l$  ausgesprochen, weiterhin können

1: Eingabe: $\mathbf{f} \in \mathbb{C}^M$ , $\hat{\mathbf{f}}_0 \in \mathbb{K}^{N_L}$
2: $\hat{\mathbf{z}}_0 = \mathbf{S}^H \mathbf{W} \mathbf{f} - \mathbf{T}^\omega \hat{\mathbf{f}}_0$
3: $\hat{\mathbf{p}}_0 = \hat{\mathbf{z}}_0$
4: <b>for</b> $l = 0, \dots$ <b>do</b>
5: $\hat{\mathbf{v}}_l = \mathbf{T}^\omega \hat{\mathbf{p}}_l$
6: $\alpha_l = \frac{\hat{\mathbf{z}}_l^H \hat{\mathbf{z}}_l}{\hat{\mathbf{p}}_l^H \hat{\mathbf{v}}_l}$
7: $\hat{\mathbf{f}}_{l+1} = \hat{\mathbf{f}}_l + \alpha_l \hat{\mathbf{p}}_l$
8: $\hat{\mathbf{z}}_{l+1} = \hat{\mathbf{z}}_l - \alpha_l \hat{\mathbf{v}}_l$
9: $\beta_l = \frac{\hat{\mathbf{z}}_{l+1}^H \hat{\mathbf{z}}_{l+1}}{\hat{\mathbf{z}}_l^H \hat{\mathbf{z}}_l}$
10: $\hat{\mathbf{p}}_{l+1} = \hat{\mathbf{z}}_{l+1} + \beta_l \hat{\mathbf{p}}_l$
11: <b>end for</b>
12: Ausgabe: $\hat{\mathbf{f}}_l$

**Algorithmus 7: ACT**

die Einträge der Matrix  $\mathbf{T}^\omega$  nur mit Rechnergenauigkeit dargestellt werden, weshalb in [15, S. 106f] vom expliziten Aufstellen der Normalengleichung erster Art abgeraten wird.

Weitere Nachteile sehen wir in der Einschränkung auf die ungedämpfte und gewichtete Normalengleichung erster Art. Darüberhinaus wird zur effizienten Berechnung der rechten Seite  $\mathbf{S}^H \mathbf{f}$  und der Einträge der Matrix  $\mathbf{T}^\omega$  ohnehin die NFFT benötigt.

### Zur Struktur der Normalengleichung zweiter Art für die 2-Sphäre $\mathbb{S}^2$

Seien die inversen Dämpfungsfaktoren mit

$$\hat{\omega}_{(k,n)}^{-1} = \frac{a_k}{2k+1}, \quad a_k > 0$$

gewählt, so hat die gedämpfte Normalengleichung zweiter Art mit den Legendre-Polynomen  $P_k$  die spezielle Struktur

$$\mathbf{S} \hat{\mathbf{W}}^{-1} \mathbf{S}^H = \left( \sum_{k=0}^{N_L-1} a_k P_k(\mathbf{x}_{j_1}^T \mathbf{x}_{j_2}) \right)_{j_1, j_2=0, \dots, M-1},$$

siehe z.B. [4].

## 4.5 Zusammenfassung

Wir haben in diesem Kapitel effiziente iterative Verfahren zur Lösung der in Kapitel 2 formulierten Probleme entwickelt. Diese nutzen die in Kapitel 3 vorgestellten Algorithmen zur schnellen Matrix–Vektor–Multiplikation. Neben einfachen Verfahren konnten wir erstmals gedämpfte und gewichtete Versionen der Verfahren CGNR und CGNE entwickeln.

Bekanntere Verfahren haben wir folgendermaßen verallgemeinert: Die Dämpfung ermöglicht die Nutzung von Vorwissen über die Lösung bei der Auswahl der Krylov–Räume, die Dämpfung umfasst darüberhinaus mehrstufige Algorithmen. Durch das Ausnutzen der Struktur der Vandermonde–ähnlichen–Matrix konnten wir Verfahren für die Normalgleichungen erster und zweiter Art entwickeln, die insbesondere auch für sphärische Polynome verwendbar sind. Wir glauben mit den entwickelten Algorithmen einer inversen Fourier–Transformation für beliebige Knoten gerecht zu werden.

## 5 Entwurf und Implementierung NFFT1.1

In den vorangegangenen Kapiteln haben wir die theoretischen Grundlagen für eine inverse Fourier-Transformation für beliebige Knoten in einem allgemeinen Rahmen dargestellt. In diesem Kapitel widmen wir uns der Erweiterung des Programmpaketes NFFT. Neben den vielen Anwendungen der direkten NFFT, siehe z.B. [36], haben Anwender auch den Wunsch nach einer inversen NFFT geäußert. Der Entwurf, siehe Abschnitt 5.1; die Implementierung, siehe Abschnitt 5.2, und die Dokumentation, siehe Abschnitt 5.3, zur Weiterentwicklung der Programmbibliothek werden in diesem Kapitel durchgeführt.

### 5.1 NFFT1.0 und Entwurf

Wir gehen von der frei verfügbaren Bibliothek NFFT1.0, siehe [26], aus. Im Vordergrund der Entwicklung stehen wie bereits bei Version 1.0 eine einfache Handhabbarkeit und hohe Anpassbarkeit an die spezielle Anwendung. Die Anpassbarkeit zielt auch hier darauf ab, den Benutzer zwischen Speicherplatz und Rechenzeit ausbalancieren zu lassen. Der generelle Ablauf wird in folgender Abbildung 10 dargestellt.

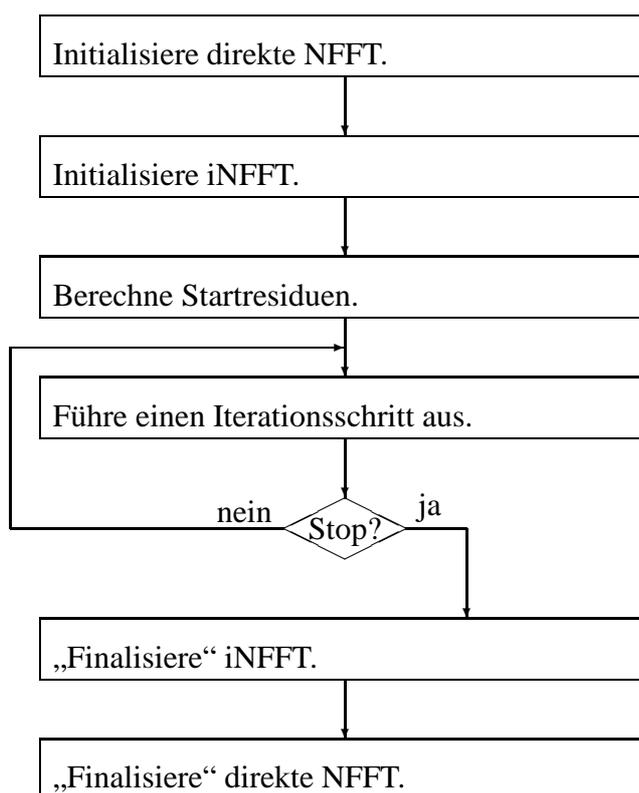


Abbildung 10: Struktogramm.

Wir stellen dem Nutzer alle verfügbaren Daten nach jedem Iterationsschritt komfortabel zur Verfügung, dies soll insbesondere helfen anwendungsspezifische Abbruchkriterien und Regularisierungsverfahren einfach zu implementieren.

## 5.2 Implementierung

Die Programmbibliothek NFFT1.0 bleibt unverändert erhalten, die einzigen von uns vorgenommenen Veränderungen daran sind die Umbenennung der Funktionen `nfft_dD_transpose` in `nfft_dD_adjoint` und die Ergänzung der Datenstruktur `nfft_dD_plan` um die Dimension  $N_L$  der Polynomräume.

Analog zu Version 1.0 enthält der Plan zur inversen Transformation sämtliche Daten, wodurch eine schmale Schnittstelle der Benutzerfunktionen erreicht wird. Dies dient momentan insbesondere der Sicherheit bei der Benutzung. Da die aktuelle Version auf den unveränderten Routinen der vorigen Version aufsetzt, wurde für  $d = 1, 2, 3$  jeweils eigener Quellcode erzeugt. Dies wird bei vereinheitlichter direkter Transformation sofort überflüssig.

Sämtliche Vektor-Routinen, d.h. die Berechnung von Skalarprodukten und die Aufdatierung von Vektoren, sind in die Hilfsbibliothek `utils.c` ausgelagert worden. Diese können einfach durch effizientere Vektor-Routinen ausgetauscht werden.

Die in Kapitel 4 vorgeschlagenen Algorithmen `CGNR_E`, `CGNE_R`, `STEEPEST_DESCENT` und `LANDWEBER` sind in der NFFT1.1 implementiert.

### 5.3 Technische Dokumentation

Wir stellen im Folgenden den erweiterten Teil der Anwendungsschnittstelle des Programmpaketes vor.

#### Parameter

Die Parameter der zu benutzenden direkten Transformation legen wir im Plan der direkten NFFT `direct_plan` vom Typ `nfft_dD_plan*` fest.

Durch Setzen von Schaltern (engl. `flags`) in `infft_flags` vom Typ `int` legen wir alle weiteren Parameter fest. Das sind der Iterationstyp, die gewünschte Gewichtung und Dämpfung. Beim Landweber-Verfahren können wir darüberhinaus die zusätzliche Berechnung von Residuen festlegen, bei den CG-Verfahren jeweils noch die zusätzliche Berechnung der „verwandten“ Iterierten. Die verfügbaren Schalter sind im Einzelnen: `LANDWEBER`, `STEEPEST_DESCENT`, `CGNR_E`, `CGNE_R`, `ITERATE_2nd`, `NORMS_FOR_LANDWEBER`, `PRECOMPUTE_WEIGHT` und `PRECOMPUTE_DAMP`.

#### Datenstrukturen der iNFFT

Neben den Parametern der iNFFT speichert die Datenstruktur `infft_dD_plan` noch alle Zustandsgrößen der Iterationen. Diese stehen uns zwischen den Iterationsschritten zur Verfügung. Die folgende Tabelle 2 listet sämtliche Vektoren, die Tabelle 3 sämtliche skalare Größen der Iteration auf.

Typ	Name	Beschreibung	Größe	Parameterwahl
<code>fftw_real*</code>	<code>w</code>	$\omega$	$M$	<code>PRECOMPUTE_WEIGHT</code>
<code>fftw_real*</code>	<code>w_hat</code>	$\hat{\omega}$	$N_L$	<code>PRECOMPUTE_DAMP</code>
<code>fftw_complex*</code>	<code>given_f</code>	$f$	$M$	
<code>fftw_complex*</code>	<code>f_hat_iter</code>	$\hat{f}_l$	$N_L$	
<code>fftw_complex*</code>	<code>f_hat_iter_2nd</code>	$\hat{f}_l^{\text{cgn}(e,r)}$	$N_L$	<code>ITERATE_2nd</code>
<code>fftw_complex*</code>	<code>r_iter</code>	$r_l$	$M$	außer <code>CGNE_R</code>
<code>fftw_complex*</code>	<code>z_hat_iter</code>	$\hat{z}_l$	$N_L$	außer <code>CGNE_R</code>
<code>fftw_complex*</code>	<code>z_tilde_iter</code>	$\tilde{z}_l$	$M$	nur <code>CGNE_R</code>
<code>fftw_complex*</code>	<code>p_hat_iter</code>	$\hat{p}_l$	$N_L$	nur <code>CGNxxx</code> ,
<code>fftw_complex*</code>	<code>v_iter</code>	$v_l$	$M$	außer <code>LANDWEBER</code>

Tabelle 2: Vektoren der Pläne `infft_dD_plan`.

#### Funktionen der iNFFT

Die Funktionen der iNFFT arbeiten vollständig auf einer Variablen des Typs `infft_dD_plan`. Der Rückgabotyp sämtlicher Funktionen ist `void`, das erste Argument immer die Variable des Typs `infft_dD_plan`. Die Funktionsnamen und weitere Argumente finden wir in folgender Tabelle 4. Die Standardeinstellungen bei der einfachen Initialisierung ist `CGNR_E`, d.h., es wird nur die CGNR-Iterierte für das ungedämpfte und ungewichtete Problem erzeugt.

Name	Beschreibung
alpha_iter	$\alpha_l$
alpha_iter_2nd	$\alpha_l^{\text{cgn}(e,r)}$
beta_iter	$\alpha_l$
gamma_iter	$\gamma_{l+1}$
gamma_iter_old	$\gamma_l$
w_norm_r_iter	$\mathbf{r}_l^H \mathbf{W} \mathbf{z}_l$
w_hat_norm_z_hat_iter	$\hat{\mathbf{z}}_{l+1}^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_{l+1}$
w_hat_norm_z_hat_iter_old	$\hat{\mathbf{z}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{z}}_l$
w_norm_z_tilde_iter	$\tilde{\mathbf{z}}_{l+1}^H \mathbf{W} \tilde{\mathbf{z}}_{l+1}$
w_norm_z_tilde_iter_old	$\tilde{\mathbf{z}}_l^H \mathbf{W} \tilde{\mathbf{z}}_l$
w_hat_norm_p_hat_iter	$\hat{\mathbf{p}}_l^H \hat{\mathbf{W}}^{-1} \hat{\mathbf{p}}_l$
w_norm_v_iter	$\mathbf{v}_l^H \mathbf{W} \mathbf{v}_l$

Tabelle 3: Skalare und Normen der Pläne `infft_dD_plan`.

Name	(weitere Argumente)
<code>infft_dD_init</code>	<code>nfft_dD_plan *direct_plan</code>
<code>infft_dD_init_specific</code>	<code>nfft_dD_plan *direct_plan, int infft_flags</code>
<code>infft_dD_before_loop</code>	
<code>infft_dD_loop_one_step</code>	
<code>infft_dD_finalize</code>	

Tabelle 4: Benutzerfunktionen.

## 6 Numerische Tests

In diesem Kapitel testen wir die implementierten Algorithmen an einfachen Beispielen. Alle Angaben beziehen sich auf das durch die Approximation bei der Matrix–Vektor–Multiplikation erzeugte verwandte System (engl. nearby system). Beispiel 6.1 vergleicht für den Fall gestörter Abtaststellen die verschiedenen Verfahren gewichtet und ungewichtet.

### Beispiel 6.1. (Jitter, 1–Torus $\mathbb{T}^1$ )

Für einen Polynomgrad  $N = 16$  wählen wir nacheinander  $M = 8, 16, 32$  Knoten. Die Knoten sind jeweils mithilfe der in  $[0, \frac{1}{32}]$  unabhängig gleichverteilten Zufallsvariablen  $\nu_j$  gegeben durch  $x_j = \frac{j}{M} - 0.5 + \nu_j$ ,  $j = 0, \dots, M - 1$ . Weiterhin wählen wir unabhängig gleichverteilte Abtastwerte  $f_j \in [0, 1] \times [0, 1]$ ,  $j = 0, \dots, M - 1$ . Wir wählen die konstanten Dämpfungsfaktoren  $\hat{\omega}_k = 1$ ,  $k = 0, \dots, 16$ ; d.h., es findet keine Dämpfung statt, und nacheinander die konstanten Gewichte  $\omega_j = 1$ ,  $j = 0, \dots, M - 1$  beziehungsweise die Voronoi–Gewichte  $\omega_j$ ,  $j = 0, \dots, M - 1$ . Für die Landweber–Iteration wählen wir den Relaxationsparameter  $\alpha = \frac{2}{(1+\delta_X N)^2}$  im gewichteten Fall und  $\alpha = \frac{2}{MN}$  im ungewichteten Fall.

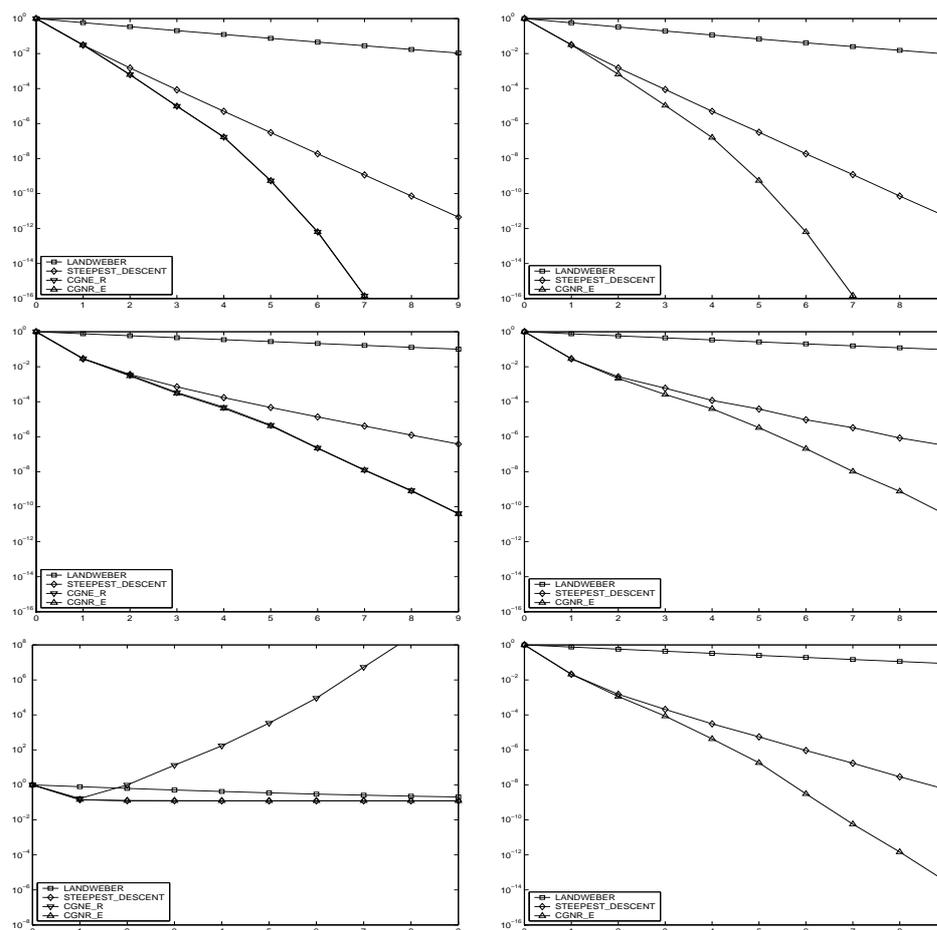


Abbildung 11: Residuenverläufe für die ungewichteten Probleme, relativ zum Startresiduum, logarithmische Darstellung; links: Originalresiduen beziehungsweise Residuen der Normalengleichung zweiter Art, rechts: Residuen der Normalengleichung erster Art; oben:  $M = 8$ , mitte:  $M = 16$ , unten:  $M = 32$ .

Zu den Verfahren LANDWEBER, STEEPEST\_DESCENT, CGNR\_E und CGNE\_R sind in Abbildung 11 die Originalresiduen beziehungsweise für das CGNE–Verfahren die Residuen der Normalgleichung zweiter Art dargestellt. Für die ersten drei Verfahren ist außerdem der Verlauf der Residuen der Normalgleichung erster Art angegeben. Alle Residuen sind jeweils relativ zu ihrem Startresiduum berechnet.

Abbildung 12 zeigt analog die Residuenverläufe für den gewichteten Fall, die Originalresiduen sind ungewichtet, d.h.  $\|r_l\|_2$ , und gewichtet, d.h.  $\|r_l\|_W$ , kaum zu unterscheiden.

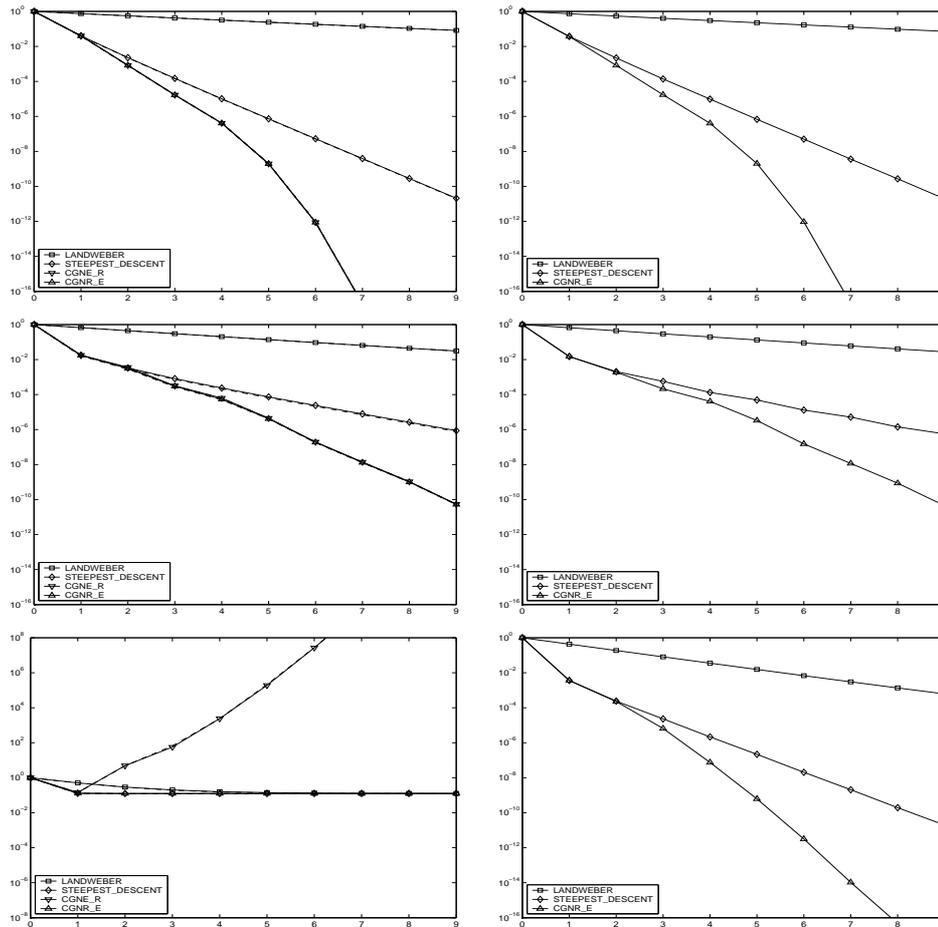


Abbildung 12: Residuenverlauf für gewichtetes Problem, analog zu Abbildung 11.

□

Das Beispiel 6.1 bestätigt die theoretischen Ergebnisse, die Landweber–Iteration zeigt eine lineare Konvergenz. Die CG–Typ–Verfahren sind den anderen Verfahren überlegen, das CGNE–Verfahren ist allerdings nur bei konsistenten Abtastwerten einzusetzen. Weiterhin scheint die Gewichtung hauptsächlich im überbestimmten Fall  $M > N_L$  sinnvoll, dort führt sie zu einer schnelleren Konvergenz.

In dem zweiten Beispiel 6.2 untersuchen wir die Auswirkung der Dämpfungsfaktoren an dem einfachen Beispiel der konstanten Funktion  $f = 1$ , d.h., für die exakten Fourier–Koeffizienten gilt  $\hat{f}_k = \delta_{k,0}$ .

### Beispiel 6.2. (1-Torus $\mathbb{T}^1$ )

Wir betrachten die konstante Funktion  $f = 1$ , diese wird an  $M = 20$  in  $[-0.5, 0.5]$  unabhängig gleichverteilten Knoten abgetastet. Die Grundannahme der getesteten Algorithmen ist ein Polynomgrad  $N = 10$ , die Iterierten wurden alle mit dem CGNR-Verfahren erzeugt.

Unser Vergleich zielt auf die Güte der Rekonstruktion in Abhängigkeit verschiedener Dämpfungsfaktoren ab. Wir wählen die inversen Dämpfungsfaktoren  $\hat{\omega}_k^{-1} = \delta_{0,k}$  (exakter Polynomgrad bekannt),  $\hat{\omega}_k^{-1} = \delta_{-1,k} + \delta_{0,k} + \delta_{1,k}$  (fast exakter Polynomgrad),  $\hat{\omega}_k^{-1} = e^{-k^2}$  (Gauß-Dämpfungsfaktoren),  $\hat{\omega}_k^{-1} = \frac{1}{1+k^2}$  („Sobolev“-Dämpfungsfaktoren) und  $\hat{\omega}_k^{-1} = 1$  (keine Dämpfung). Wir erhalten die folgenden Ergebnisse in Abbildung 13.

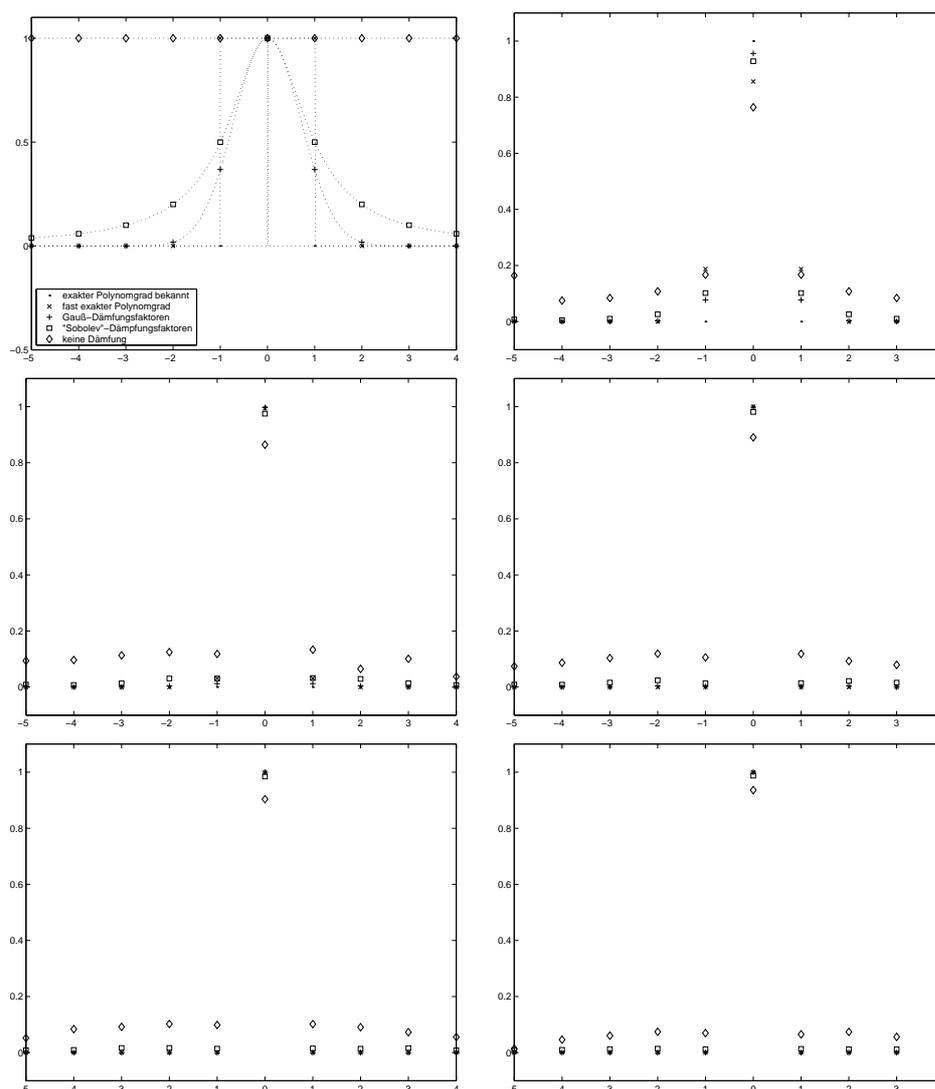


Abbildung 13: Rekonstruktion der Fourier-Koeffizienten zur konstanten Funktion  $f = 1$ ; links oben: benutzte Dämpfungsfaktoren; zeilenweise: Absolutbetrag der ersten bis fünften Iterierten  $\hat{f}_l$ ,  $l = 1, \dots, 5$ .

□

Die iterativen Algorithmen finden (optimale) Lösungen in gewissen Teilräumen. Haben wir a-priori Wissen über die gesuchte Lösung, so können wir dies in Form der Dämpfung einbringen. Dies führt im Beispiel 6.2 zu einer schnelleren Konvergenz.

Das letzte Beispiel 6.3 zeigt schließlich eine Anwendung bei der Orts- und Ortsfrequenzraum die Bedeutung tauschen.

**Beispiel 6.3.** (2-Torus  $\mathbb{T}^2$ )

Wir gehen von dem sogenannten Shepp-Logan-Phantom der Größe  $256 \times 256$  (MATLAB-Funktion `phantom(256)`) aus. Die auf dem Gitter  $[0, 255]^2$  gegebenen Bildwerte interpretieren wir als Fourier-Koeffizienten. Das entsprechende trigonometrische Polynom werten wir auf dem Polargitter  $\mathbf{x}_{r,t} = \left( \frac{r}{2R} \cos\left(\frac{2\pi t}{T}\right), \frac{r}{2R} \sin\left(\frac{2\pi t}{T}\right) \right)$ ,  $r = 0, \dots, R-1$ ,  $t = 0, \dots, T-1$ ,  $R = 256$ ,  $T = 512$ , aus.

Zur Rekonstruktion verwenden wir das CGNR-Verfahren mit den Gewichten des abgetasteten Hann-Fensters  $\omega_{r,t} = \frac{r}{2R} \left( 1 + \cos\left(\frac{\pi r}{R}\right) \right)$ , siehe z.B. [51, S. 108]. Wir erhalten die Ergebnisse in Abbildung 14.

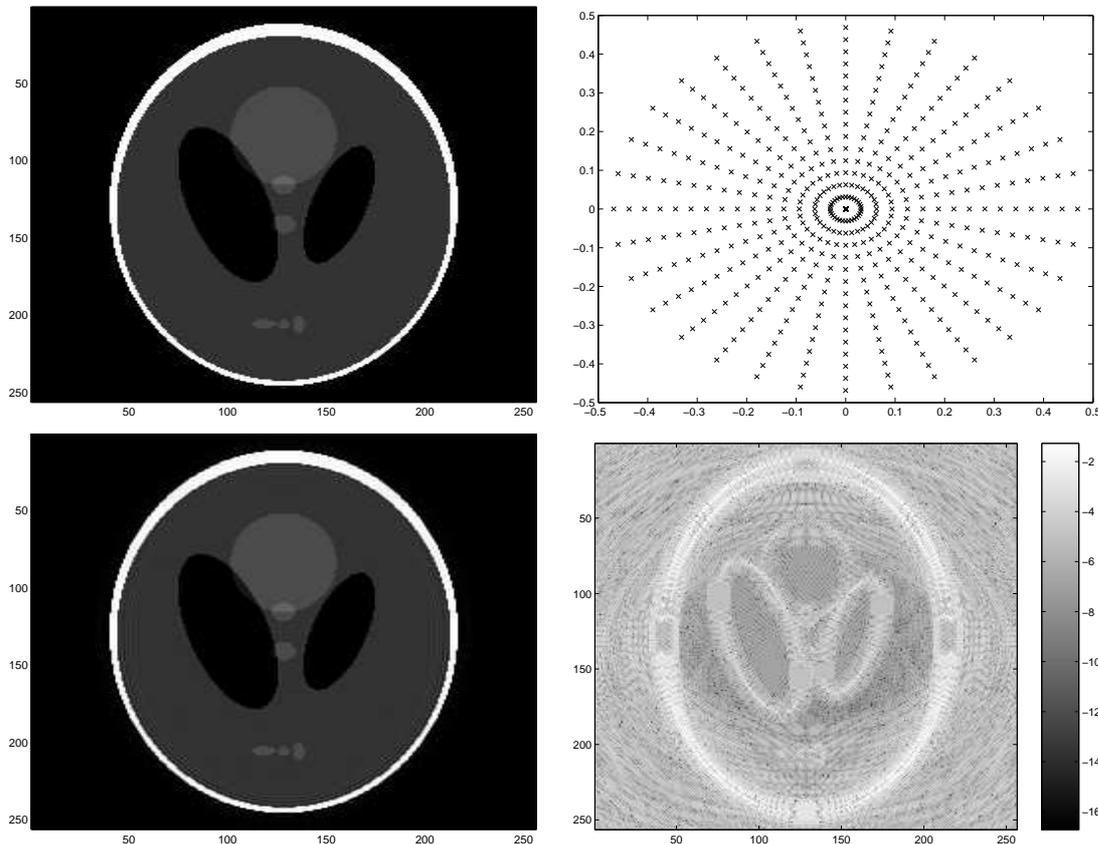


Abbildung 14: Rekonstruktion, ausgehend von einem Polargitter; oben, links: Shepp-Logan-Phantom, rechts: Polargitter im Ortsfrequenzraum, Ausschnitt; unten, links: Rekonstruktion nach zehn Iterationen, rechts: relativer Fehler, logarithmisch.

□

## 7 Ausblick

In der vorliegenden Arbeit haben wir Algorithmen zur inversen Fourier–Transformation für beliebige Knoten vorgeschlagen. Nach einer Einführung der wesentlichen Begriffe konnten wir in Kapitel 2 die zentralen Problemstellungen der inversen Fourier–Transformation für beliebige Knoten formulieren. Diese sind im Einzelnen: die Wahl eines Polynomraumes und die Formulierung geeigneter Interpolations– beziehungsweise diskreter Approximationsbedingungen. Letztere konnten in die gewichteten und gedämpften Normalgleichungen erster und zweiter Art überführt werden, siehe Abschnitt 2.6.

Mit der Nutzung schneller Algorithmen zur Matrix–Vektor–Multiplikation für Vandermonde–ähnliche–Matrizen, siehe Kapitel 3, haben wir in Abschnitt 4.3 leistungsstarke CG–Typ–Verfahren zur näherungsweisen Berechnung der inversen Fourier–Transformation für beliebige Knoten entwickelt. Für die Implementierung der Iterationsverfahren, siehe Kapitel 5, haben wir uns auf den Fall der trigonometrischen Polynome zurückgezogen, da hierfür effiziente Implementierungen für das direkte Problem vorliegen. Numerische Tests in Kapitel 6 untermauern die Leistungsfähigkeit der vorgeschlagenen Algorithmen.

In Verbindung mit dieser Arbeit sind wir an einer Reihe weiterer Fragestellungen interessiert:

1. Die Implementierung der schnellen Matrix–Vektor–Multiplikation mit  $S^H$  ist für den sphärischen Fall noch offen.
2. Die Erweiterung der CG–Typ–Verfahren um eine von der aktuellen Iterierten abhängige Dämpfung scheint nach ersten Tests erfolgversprechend. Theoretische Ergebnisse zur Konvergenz sind uns jedoch nicht bekannt.
3. Die enge Verbindung der Normalgleichung zweiter Art zur Lösung von Approximationsproblemen mit radialen Basisfunktionen ist sehr interessant.
4. Die Lösung von Operatorgleichungen mit gitterfreien Fourier–Methoden ist von Bedeutung.



# A Index der wichtigsten definierten Symbole

## Kapitel 2

$E$	Einheitsmatrix
$F$	Fourier-Matrix
$\mathbb{D}$	(allgemeiner) Urbildraum, Zeit- beziehungsweise Ortsraum
$\mathbb{K}$	skalarer Bildraum
$L$	Hilbertraum der quadratintegrierbaren Funktionen
$N$	Polynomgrad, Abschneidefrequenz
$L_N$	Polynomraum
$N_L$	Dimension des Polynomraumes, Gesamtanzahl der Fourier-Koeffizienten
$e^{-2\pi i \mathbf{k}^T \cdot}$	Element der trigonometrischen Standardorthonormalbasis
$\mathbb{T}^d$	$d$ -Torus, Urbildraum der trigonometrischen Polynome
$I_N^d$	Indexmenge der Fourier-Koeffizienten für trigonometrische Polynome
$S^2$	2-Sphäre, Urbildraum der sphärischen Polynome
$J_N^2$	Indexbereich der Fourier-Koeffizienten für sphärische Polynome
$P_k$	Legendre-Polynom
$P_k^{[n]}$	zugeordnete Legendre-Funktion
$Y_{\mathbf{k}}$	Element der Standardorthonormalbasis auf der Sphäre
$[-1, 1]$	Intervall der reellen Achse, Urbildraum der (algebraischen) Polynome
$\hat{f}_k$	Fourier-Koeffizient
$\hat{\mathbf{f}}$	Vektor der Fourier-Koeffizienten
$S$	Synthese
$G$	Gram-Matrix
$A$	Analyse
$\mathcal{P}$	orthogonaler Projektor
$M$	Anzahl der Abtaststellen
$x_j$	Abtaststelle, Knoten
$X$	Abtastmenge
$f_j$	Abtastwert, Funktionswert
$\mathcal{D}$	ideale Abtastung
$\mathbf{f}$	Vektor der Abtastwerte
$S$	Vandermonde-ähnliche-Matrix
$\omega_j$	Gewichte
$W$	Gewichtsmatrix
$\hat{\omega}_k$	Dämpfungsfaktoren
$\hat{W}$	Dämpfungsmatrix
$K_j$	polynomiale Kerne
$\mathcal{K}$	Linearkombination der Kerne
$\delta_X$	Abtastmengen-Norm
$R$	(erlaubte) Partition

**Kapitel 3**

$\varphi$	gut lokalisierte Funktion
$\tilde{\varphi}$	gut lokalisierte periodische Funktion
$\tilde{\psi}$	gut lokalisierte periodische Funktion mit kompaktem Träger in $\mathbb{T}^d$
$\mathbf{R}$	Diagonalmatrix der skalierten Fourier-Koeffizienten von $\tilde{\varphi}$
$\mathbf{B}$	schwach besetzte Matrix mit den Funktionswerten von $\tilde{\psi}$
$p_k$	orthogonale Polynome
$p_c(\cdot, k)$	Familie der zugeordneten Polynome
$T_k$	Chebyshev-Polynome erster Art

**Kapitel 4**

$\mathbf{f}^\omega$	Substitution für $\mathbf{W}^{\frac{1}{2}} \mathbf{f}$
$\hat{\mathbf{f}}^\omega$	Substitution für $\hat{\mathbf{W}}^{\frac{1}{2}} \hat{\mathbf{f}}$
$\mathbf{S}^\omega$	Substitution für $\mathbf{W}^{\frac{1}{2}} \mathbf{S} \hat{\mathbf{W}}^{-\frac{1}{2}}$
$\hat{\mathbf{f}}_l$	$l$ -te Iterierte
$\mathbf{r}_l$	Originalresiduenvektor
$\hat{\mathbf{z}}_l$	Residuenvektor der gewichteten Normalgleichung erster Art
$\tilde{\mathbf{z}}_l$	Residuenvektor der gedämpften Normalgleichung zweiter Art
$\mathcal{K}_l^\omega(\mathbf{S}, \hat{\mathbf{r}}_0)$	Krylov-Raum

## Literatur

- [1] R.F. Bass and K. Gröchenig. Random sampling of multivariate trigonometric polynomials, 2003. <http://www.math.uconn.edu/~bass/sample.pdf>.
- [2] Å. Björck. *Numerical methods for least squares problems*. SIAM, Philadelphia, 1996.
- [3] D. Braess. *Nonlinear Approximation Theory*. Springer, Berlin, 1986.
- [4] M. Conrad. Approximation und Multiskalenzerlegung auf der Sphäre. Diplomarbeit, Universität Hamburg, 2001.
- [5] J.W. Cooley and J.W. Tukey. An algorithm for machine calculation of complex Fourier series. *Math. Comput.*, 19:297 – 301, 1965.
- [6] G. Dahlquist and Å. Björck. *Numerical Mathematics and Scientific Computation*. SIAM, Philadelphia, 2003. to appear, <http://www.mai.liu.se/~akbjo/>.
- [7] J.R. Driscoll and D.M. Healy. Computing Fourier transforms and convolutions on the 2–sphere. *Adv. Appl. Math.*, 15:202 – 240, 1994.
- [8] J.R. Driscoll, D.M. Healy, and D.N. Rockmore. Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs. *SIAM J. Comput.*, 26:1066 – 1099, 1996.
- [9] J.O. Droese. Verfahren zur schnellen Fourier–Transformation mit nichtäquidistanten Knoten. Diplomarbeit, TU Darmstadt, 1996.
- [10] R.J. Duffin and A.C. Schaeffer. A class of nonharmonic Fourier series. *Trans. Am. Math. Soc.*, 72:341 – 366, 1952.
- [11] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data II. *Appl. Comput. Harmon. Anal.*, 2:85 – 100, 1995.
- [12] H.G. Feichtinger, K. Gröchenig, and T. Strohmer. Efficient numerical methods in non-uniform sampling theory. *Numer. Math.*, 69:423 – 440, 1995.
- [13] J.A. Fessler and B.P. Sutton. Nonuniform fast Fourier transforms using min-max interpolation. *IEEE Tr. Sig. Proc.*, 51:560 – 574, 2003.
- [14] W. Freeden, T. Gervens, and M. Schreiner. *Constructive approximation on the sphere*. Clarendon Press, 1998. With applications to geomathematics.
- [15] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. SIAM, Philadelphia, 1997.
- [16] K. Gröchenig. A discrete theory of irregular sampling. *Lin. Alg. Appl.*, 193:129 – 150, 1993.
- [17] K. Gröchenig and T. Strohmer. Numerical and theoretical aspects of non–uniform sampling of band–limited images. In F. Marvasti, editor, *Theory and Practice of Nonuniform Sampling*, pages 283 – 324. Kluwer/Plenum, 2001.
- [18] R. Hampel. Sampling Theorie und Interpolierende Skalierungsfunktionen in höheren Dimensionen. Dissertation, Philipps-Universität Marburg, 2001.
- [19] M. Hanke. *Conjugate gradient type method for ill–posed problems*. Wiley, New York, 1995.
- [20] M. Hanke-Bourgeois. *Grundlagen der Numerischen Mathematik und des Wissenschaftlichen Rechnens*. Teubner, Wiesbaden, 2002.

- [21] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, 1998. Numerical Aspects of Linear Inversion.
- [22] D. Healy, D. Rockmore, P. Kostelec, and S. Moore. FFTs for the 2-sphere – improvements and variations. *J. Fourier Anal. Appl.* to appear.
- [23] E. Kreyszig. *Introductory Functional Analysis with Applications*. John Wiley & Sons, 1989.
- [24] S. Kunis. Schnelle Fourier-Algorithmen auf der Sphäre. Studienarbeit, Universität Lübeck, Preprint B-02-01, 2002.
- [25] S. Kunis and D. Potts. Fast spherical Fourier algorithms. *J. Comput. Appl. Math.*, 2002. to appear.
- [26] S. Kunis, D. Potts, and G. Steidl. Fast Fourier transform at nonequispaced knots. Technical report, Universität Lübeck, Preprint B-02-13, 2002. A user's guide to a C-library.
- [27] A.K. Louis. *Inverse und schlecht gestellte Probleme*. Teubner, Stuttgart, 1989.
- [28] A. Meister. *Numerik linearer Gleichungssysteme - Eine Einführung in moderne Verfahren*. Vieweg, Wiesbaden, 1999.
- [29] H.N. Mhaskar, F. Narcowich, and J. Ward. Quadrature formulas on spheres using scattered data. [citeseer.nj.nec.com/mhaskar98quadrature.html](http://citeseer.nj.nec.com/mhaskar98quadrature.html), 1998. Center for Approximation Theory Report 393, Department of Mathematics, Texas A & M University.
- [30] H.N. Mhaskar, F.J. Narcowich, and J. Ward. Spherical Marcinkiewicz-Zygmund inequalities and positive quadrature. *Math. Comp.*, 70:1113 – 1130, 2001.
- [31] H.N. Mhaskar and J. Prestin. On Marcinkiewicz-Zygmund-type inequalities. In N.K. Govil, R.N. Mohapatra, Z. Nashed, A. Sharma, and J. Szabados, editors, *Approximation Theory: In Memory of A.K. Varma*, pages 389–403, 1998.
- [32] F. Natterer. *The Mathematics of Computerized Tomography*. John Wiley & Sons Ltd, 1986.
- [33] D. Potts. Schnelle Polynomtransformation und Vorkonditionierer für Toeplitz-Matrizen. Dissertation, Universität Rostock, 1998.
- [34] D. Potts. Approximation of scattered data by trigonometric polynomials on the torus and the 2-sphere. *Adv. Comput. Math.*, 2002. to appear.
- [35] D. Potts. Fast algorithms for discrete polynomial transforms on arbitrary grids. *Linear Algebra Appl.*, 366:353 – 370, 2003.
- [36] D. Potts. Schnelle Fourier-Transformation für nichtäquidistante Daten und Anwendungen. Habilitation, Universität zu Lübeck, 2003. submitted.
- [37] D. Potts and G. Steidl. New Fourier reconstruction algorithms for computerized tomography. In A. Aldroubi, A.F. Laine, and M.A. Unser, editors, *Proceedings of SPIE: Wavelet Applications in Signal and Image Processing VIII*, volume 4119, pages 13 – 23, 2000.
- [38] D. Potts and G. Steidl. Fast summation at nonequispaced knots by NFFT. *SIAM J. on Sci. Comput.*, 24:2013 – 2037, 2003.
- [39] D. Potts, G. Steidl, and M. Tasche. Fast algorithms for discrete polynomial transforms. *Math. Comput.*, 67:1577 – 1590, 1998.

- [40] D. Potts, G. Steidl, and M. Tasche. Fast and stable algorithms for discrete spherical Fourier transforms. *Linear Algebra Appl.*, 275:433 – 450, 1998.
- [41] D. Potts, G. Steidl, and M. Tasche. Fast Fourier transforms for nonequispaced data: A tutorial. In John. J. Benedetto and Paulo J. S. G. Ferreira, editors, *Modern Sampling Theory: Mathematics and Applications*, pages 247 – 270, Boston, 2001.
- [42] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publ., Boston, 1996.
- [43] Y. Saad and H.A. van der Vorst. Iterative solution of linear systems in the 20-th century. *J. Comput. Appl. Math.*, 123:1 – 33, 2000.
- [44] T. Sauer. [http://www.uni-giessen.de/tomas.sauer/homepage\\_d.html](http://www.uni-giessen.de/tomas.sauer/homepage_d.html), 2002. Vorlesung Approximationstheorie.
- [45] G. Steidl. A note on fast Fourier transforms for nonequispaced grids. *Adv. Comput. Math.*, 9:337 – 353, 1998.
- [46] T. Strohmer. *Efficient Methods for Digital Signal and Image Reconstruction from Nonuniform Samples*. PhD thesis, University of Vienna, 1993.
- [47] T. Strohmer. On the estimation of the bandwidth of nonuniformly sampled signals. In *IEEE Proc. Int. Conf. Acoustic, Speech, and Signal Processing*, volume 4, pages 2047 – 2050, 1999.
- [48] R. Suda and M. Takami. A fast spherical harmonics transform algorithm. *Math. Comput.*, 71:703 – 715, 2001.
- [49] B.P. Sutton, D.C. Noll, and J.A. Fessler. Fast, iterative, field-corrected image reconstruction for MRI. *IEEE Tr. Med. Im.*, 22:178 – 188, 2003.
- [50] A.F. Timan. *Theory of Approximation of Functions of a Real Variable*. Dover, New York, 1994.
- [51] P. Toft. *The radon transform*. PhD thesis, Technical University of Denmark, 1996. Theory and implementation.
- [52] E. Zeidler, editor. *Teubner–Taschenbuch der Mathematik*. Teubner, Leipzig, 1996.