UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR MATHEMATIK

Direktor: Prof. Dr. rer. nat. Jürgen Prestin

Schnelle Fourier-Transformationen für sphärische Gauß-Laguerresche Basisfunktionen

Inauguraldissertation zur Erlangung der Doktorwürde
der Universität zu Lübeck

– aus der Sektion Informatik / Technik –

vorgelegt von

Christian Wülker, M.Sc.

aus Hannover

Lübeck, den 18. Januar 2018

Vorsitzender: Prof. Dr. Thorsten M. Buzug, Universität zu Lübeck

1. Berichterstatter: Prof. Dr. Jürgen Prestin, Universität zu Lübeck

2. Berichterstatter: Prof. Dr. Daniel Potts, Technische Universität Chemnitz

Tag der mündlichen Prüfung: 28. März 2018

Zum Druck genehmigt. Lübeck, den 3. April 2018

Zusammenfassung

Sphärische Gauß-Laguerresche Basisfunktionen, kurz SGL-Basisfunktionen, sind Funktionen auf $\mathbb{R}^3$ von der Form $N_{nl} L_{n-l-1}^{(l+1/2)}(r^2) r^l Y_{lm}(\vartheta, \varphi)$. Sie bilden eine Orthonormalbasis des gewichteten L^2-Raums mit radialem Gewicht $\exp(-r^2)$. In dieser Arbeit werden auf drei verschiedene Weisen verallgemeinerte schnelle Fourier-Transformationen (FFTs) für die SGL-Basisfunktionen entwickelt. Die erste Klasse wird ausgehend von einer Quadraturformel für die SGL-Fourier-Koeffizienten bandbegrenzter Funktionen hergeleitet. Mittels einer Separation der Variablen wird die Transformation dabei in eine radiale und eine sphärische Teiltransformation aufgeteilt, welche anschließend beschleunigt werden. Die zweite und dritte Klasse sind iterative Verfahren, die sich aus der jeweiligen schnellen inversen Transformation und ihrer Adjungierten ergeben. Im Unterschied zur ersten Klasse müssen die Punkte, an denen die Daten hier vorliegen, nicht notwendigerweise auf einem speziellen Gitter liegen und ihre Anzahl ist nicht fest vorgegeben. Bei der zweiten Klasse wird die bandbegrenzte Funktion von Interesse an den gegebenen Punkten in die Form eines dreidimensionalen trigonometrischen Polynoms gebracht, welches anschließend mit der inversen nichtäquidistanten FFT (NFFT) von Potts et al. effizient ausgewertet wird. Bei der dritten Klasse wird demgegenüber die Idee der NFFT unmittelbar auf den SGL-Fall übertragen. Konkret bedeutet dies, dass ein Approximant an die Funktion bestimmt wird, der sich mit weniger Aufwand auswerten lässt, als die Funktion selbst. Die SGL-Fourier-Koeffizienten des Approximanten stimmen dabei bis zu einer vorgegebenen Bandbreite mit den SGL-Fourier-Koeffizienten der Funktion überein. Die schnellen Algorithmen wurden in der Programmiersprache C++ implementiert und ausführlich getestet. Die Ergebnisse werden hier vorgestellt.

In Erinnerung an

LENNART CLAUS

$\star$13.12.1988 $\dagger$28.12.2012

ehemaliger Student der Mathematik
an der Humboldt-Universität zu Berlin

Inhaltsverzeichnis

Einleitung **1**

1 Sphärische Gauß-Laguerresche Basisfunktionen **9**
1.1 Die Kugelflächenfunktionen (*spherical harmonics*) 9
1.2 Die hypergeometrische Reihe 15
1.3 Die (sphärischen) Bessel-Funktionen 16
1.4 Die verallgemeinerten Laguerre-Polynome 19
1.5 Konstruktion der SGL-Funktionen und Nachweis der Basiseigenschaft 21
1.6 Spektralverhalten unter Rotationen und Translationen 23

2 Basale Algorithmen und schnelle SGL-Fourier-Transformationen für reguläre Daten **29**
2.1 Ein SGL-Abtasttheorem 29
2.1.1 Die sphärische Quadraturformel von Driscoll und Healy 30
2.1.2 Die Gauß-Quadraturformel für den Typ $\int_0^\infty f(\xi)\exp(-\xi^2)\,\mathrm{d}\xi$ 32
2.1.3 Herleitung des Abtasttheorems 37
2.2 Schnelle SGL-Fourier-Transformation (FSGLFT) 37
2.2.1 Schnelle Fourier-Transformation (FFT) 39
2.2.2 Schnelle Kosinus-Transformation (DCT) 40
2.2.3 Schnelle Legendre-Transformation (FLT) 42
2.2.4 Schnelle sphärische Fourier-Transformation (FSFT) 45
2.2.5 Der Clenshaw-Algorithmus 48
2.2.6 Herleitung der schnellen Algorithmen 51
2.3 Numerische Tests 54

3 Schnelle SGL-Fourier-Transformationen für gestreute Daten **59**
3.1 Schnelle nichtäquidistante Fourier-Transformationen (NFFT) 60
3.2 NFSGLFT-A: Trigonometrisierung und NFFT 67
3.2.1 Herleitung der schnellen Algorithmen 67
3.2.2 Fehlerabschätzung 73
3.2.3 Numerische Tests 76
3.3 NFSGLFT-B: direkte Verallgemeinerung der NFFT 81
3.3.1 Schnelle SO(3)-Fourier-Transformation (FSOFT) 82
3.3.2 Schnelle Gauß-Transformation (FGT) 87
3.3.3 Herleitung der schnellen Algorithmen 90
3.3.4 Numerische Tests und Vergleich mit Variante A 95
3.3.5 Fehlerabschätzung 100

Abkürzungsverzeichnis ix

Abbildungsverzeichnis xi

Tabellenverzeichnis xiii

Literaturverzeichnis xv

Lebenslauf xix

Einleitung

Seit ihrer Aufarbeitung durch Cooley und Tukey [1965] hat die schnelle Fourier-Transformation[1] (FFT) auf dem Einheitskreis $\mathbb{T}$ viele Verallgemeinerungen auf andere Definitionsbereiche mit zugehörigen Basisfunktionen erfahren. Anwendungen in der Signalverarbeitung etwa profitieren heutzutage von der Erweiterung der FFT auf den d-dimensionalen Torus $\mathbb{T}^d$, wobei multivariate trigonometrische Polynome in Analogie zum eindimensionalen Fall verwendet werden (s. z. B. [Dudgeon und Mersereau, 1984, Abschn. 2]). Ein weiteres Beispiel ist die zweidimensionale Einheitssphäre $\mathbb{S}^2$ (s. [Driscoll und Healy, 1994; Healy et al., 2003; Kunis und Potts, 2003; McEwen und Wiaux, 2011]). Hier werden die Kugelflächenfunktionen (engl.: *spherical harmonics*) als orthonormale Basisfunktionen des Raums $L^2(\mathbb{S}^2)$ der über $\mathbb{S}^2$ quadratintegrierbaren Funktionen benutzt. Dies hat auch die Entwicklung schneller Fourier-Transformationen auf der dreidimensionalen Rotationsgruppe SO(3) eingeleitet (s. [Kostelec und Rockmore, 2008; Potts et al., 2009; McEwen et al., 2015]). Dabei übernehmen die sogenannten Wigner-D-Funktionen[2] die Rolle der Kugelflächenfunktionen. Kürzlich wurden spezielle Produkte von Kugelflächenfunktionen und verallgemeinerten Laguerre-Polynomen[3] mit einem exponentiell abfallenden radialen Gewicht als Orthonormalbasis für den Raum $L^2(\mathbb{B}^3)$ der über der dreidimensionalen Einheitskugel $\mathbb{B}^3$ quadratintegrierbaren Funktionen verwendet. Auch in diesem Fall wurde eine schnelle Fourier-Transformation entwickelt (s. [Leistedt und McEwen, 2012] für weitere Informationen).

In dieser Arbeit stellen wir schnelle Fourier-Transformationen auf dem gesamten Anschauungsraum $\mathbb{R}^3$ vor. Auf der einen Seite erweitert dies die obige Familie von Definitionsbereichen auf natürliche Weise. Aufgrund der Nichtkompaktheit von $\mathbb{R}^3$ befinden wir uns andererseits in einer neuartigen Situation. Aktuelle Beispiele für verallgemeinerte FFTs auf nichtkompakten Räumen finden sich des Weiteren in der kürzlich erschienenen Monographie [Chirikjian und Kyatkin, 2016].

Natürlich muss die fehlende Kompaktheit des Definitionsbereichs kompensiert werden. Während es denkbar ist, Basisfunktionen zu verwenden, die ein geeignetes Abklingverhalten zeigen, statten wir den Raum $L^2(\mathbb{R}^3)$ in dieser Arbeit mit der *Gaußschen*[4] *Gewichtsfunktion* $\exp(-\|x\|_2^2)$ aus. Diese Gewichtsfunktion wird auch als multivariates *Hermite-Gewicht*[5] bezeichnet. Hierbei und im Folgenden ist $\|\cdot\|_2$ die 2-Norm auf $\mathbb{R}^3$. Genauer betrachten wir den gewichteten L^2-Raum

$$\text{(E.1)} \qquad H \coloneqq \left\{ f : \mathbb{R}^3 \to \mathbb{C} \,:\, f \text{ Lebesgue-messbar}^6 \text{ und } \int_{\mathbb{R}^3} |f(x)|^2 \, \mathrm{e}^{-\|x\|_2^2} \, \mathrm{d}x < \infty \right\},$$

[1] Jean Baptiste Joseph Fourier (⋆1768 bei Auxerre, Frankreich, †1830 in Paris, ebd.)
[2] Eugene Paul Wigner (⋆1902 in Budapest, Ungarn, †1995 in Princeton, NJ, USA)
[3] Edmond Nicolas Laguerre (⋆1834 in Bar-le-Duc, Frankreich, †1886, ebd.)
[4] Johann Carl Friedrich Gauß (⋆1777 in Braunschweig, †1855 in Göttingen)
[5] Charles Hermite (⋆1822 in Dieuze in der Region Lothringen, Frankreich, †1901 in Paris, ebd.)

ausgestattet mit dem inneren Produkt

$$\langle f, g\rangle_H := \int_{\mathbb{R}^3} f(x)\overline{g(x)}\mathrm{e}^{-\|x\|_2^2}\,\mathrm{d}x, \qquad f, g \in H, \tag{E.2}$$

und der induzierten Norm $\|\cdot\|_H := \sqrt{\langle\cdot,\cdot\rangle_H}$. In diesem Kontext werden Funktionen als gleich angesehen, wenn sie fast überall identisch sind.

Ein großer Vorteil des Raums H besteht darin, dass er es uns erlaubt, mit strukturell so einfachen Funktionen wie *Polynomen* zu arbeiten. Genauer machen wir uns das folgende, kürzlich von Maizlish und Prymak [2015, Abschn. 1] bemerkte Resultat zunutze, welches entscheidend für alles Weitere ist:

Satz E.1. *Der Raum $\Pi(\mathbb{R}^3)$ der komplexwertigen Polynome auf $\mathbb{R}^3$ ist dicht in H, d. h. jede Funktion $f \in H$ lässt sich bezüglich $\|\cdot\|_H$ beliebig genau durch Polynome annähern.*

Der obige Satz motiviert den Einsatz von Orthogonalpolynomen als Basis für den Hilbert-Raum[7] H. Es ist dabei zu beachten, dass solche Polynome im multivariaten Fall nicht eindeutig bestimmt sind. Tatsächlich findet man in der Literatur verschiedene Sätze von Orthogonalpolynomen im Raum H, die sich aus unterschiedlichen Konstruktionsansätzen ergeben (vgl. [Dunkl und Xu, 2001, Abschn. 5.1.3]). Durch eine *Separation der Variablen* wurde von Ritchie und Kemp [2000] ein vollständiger Satz von Orthogonalpolynomen in H aus den Kugelflächenfunktionen Y_{lm} (Abschn. 1.1) und den verallgemeinerten Laguerre-Polynomen $L_n^{(\alpha)}$ (Abschn. 1.4) konstruiert. Diese sind von der Form

$$H_{nlm}(r, \vartheta, \varphi) \;=\; N_{nl}\, L_{n-l-1}^{(l+1/2)}(r^2)\, r^l\, Y_{lm}(\vartheta, \varphi)$$

und werden von uns *sphärische Gauß-Laguerresche Basisfunktionen*, kurz *SGL-Basisfunktionen*, genannt. In ihrer ursprünglichen Form, d. h. mit dem zusätzlichen radialen Abklingfaktor $\exp(-r^2)$ und im ungewichteten Raum $L^2(\mathbb{R}^3)$, heißen die Funktionen *GTO-Basisfunktionen* (GTO für *Gaussian-type orbital*). Die SGL-Basisfunktionen und ihre Eigenschaften untersuchen wir im ersten Kapitel dieser Arbeit.

Das Hauptanliegen dieser Dissertation ist neben der Untersuchung der SGL-Basisfunktionen die effiziente numerische Berechnung der *SGL-Fourier-Koeffizienten* $\hat{f}_{nlm} := \langle f, H_{nlm}\rangle_H$ sogenannter *bandbegrenzter* Funktionen $f \in H$, d. h. endlicher Linearkombinationen der SGL-Basisfunktionen. Mit anderen Worten entwickeln wir in dieser Arbeit *schnelle SGL-Fourier-Transformationen*, also verallgemeinerte FFTs für die SGL-Basisfunktionen. Wir unterscheiden dabei zwei verschiedene Klassen: schnelle Transformationen für *reguläre* Daten, deren Entwicklung das zweite Kapitel gewidmet ist, und solche für *gestreute* Daten, die wir im dritten Kapitel vorstellen. Alle diese diskreten Transformationen haben gemeinsam, dass sie die SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ bandbegrenzter Funktionen f aus gegebenen *Daten* $f(x_i)$, d. h. aus endlich vielen Abtastwerten der jeweiligen Funktion f, berechnen.

Die SGL-Fourier-Transformationen für reguläre Daten (FSGLFTs) entwickeln wir, wie dies bei verallgemeinerten FFTs häufig getan wird, ausgehend von einer Quadraturformel für die Fourier-Koeffizienten bandbegrenzter Funktionen. Diese Funktionen haben

[6] Henri Léon Lebesgue (⋆1875 in Beauvais, Frankreich, †1941 in Paris, ebd.)
[7] David Hilbert (⋆1862 in Königsberg, heute Kaliningrad in Russland, †1943 in Göttingen)

den entscheidenden Vorteil, dass sie nur endlich viele von Null verschiedene SGL-Fourier-Koeffizienten besitzen (vgl. Def. 2.1.1). Dies erlaubt es uns, die SGL-Fourier-Koeffizienten bandbegrenzter Funktionen *exakt*, d. h. fehlerfrei bei angenommener exakter Arithmetik, zu berechnen. Unser dabei zum Einsatz kommendes *SGL-Abtasttheorem* (Satz 2.1.12) mit der oben erwähnten Quadraturformel hat den Vorteil, dass die Quadraturgewichte positiv sind (Folg. 2.1.13). Regulär bedeutet nun, dass die Punkte x_i, an denen die Daten $f(x_i)$ gegeben sind, die Knotenpunkte eines Gitters von speziellem Typ sein müssen, der unserer Quadraturformel zugrunde liegt. Dieser Gittertyp ergibt sich, wie die Quadraturformel selbst, durch eine Separation der Variablen, bei der der Radius r von den Winkeln ϑ und φ getrennt wird (vgl. Abb. 1.1). Die besondere Gitterstruktur nutzen wir algorithmisch aus, indem wir die sich ergebende naive Transformation in eine *radiale* und *sphärische Teiltransformation* aufteilen. Die radiale Teiltransformation beschleunigen wir dann mithilfe des Clenshaw-Algorithmus[8] (Abschn. 2.2.5), während wir für die sphärische Teiltransformation eine schnelle sphärische Fourier-Transformation (FSFT, Abschn. 2.2.4) verwenden. Auf natürliche Weise ergeben sich so auch schnelle inverse Transformationen.

Bei den schnellen SGL-Fourier-Transformationen für gestreute Daten (NFSGLFTs) müssen die Punkte x_i, an denen die Daten $f(x_i)$ hier vorliegen, nicht notwendigerweise auf einem speziellen Gitter liegen, und ihre Anzahl ist nicht fest vorgegeben. Schnelle Fourier-Transformationen dieses Typs wurden im Fall der klassischen FFT von Potts et al. entwickelt (s. [Potts et al., 2001; Potts, 2003; Kunis, 2006]). Die nichtäquidistante FFT (NFFT, Abschn. 3.1) ist heute in der Praxis fest etabliert. Wir stellen in dieser Arbeit zwei verschiedene Varianten von schnellen SGL-Fourier-Transformationen für gestreute Daten vor, wobei wir jeweils mit der inversen Transformation beginnen. Bei der ersten Variante A wird die bandbegrenzte Funktion f an den gegebenen Punkten x_i in die Form eines dreidimensionalen trigonometrischen Polynoms gebracht, welches anschließend mit der inversen NFFT effizient ausgewertet werden kann. Diese Strategie wurde im Fall der Einheitssphäre $\mathbb{S}^2$ und der Rotationsgruppe SO(3) bereits erfolgreich angewandt (s. [Kunis und Potts, 2003] bzw. [Potts et al., 2009]). Bei der zweiten Variante B übertragen wir die Idee der NFFT von Potts et al. unmittelbar auf den SGL-Fall. Konkret bedeutet dies, dass wir einen Approximanten an die bandbegrenzte Funktion f bestimmen, der sich mit weniger Aufwand auswerten lässt, als die Funktion f selbst. Als Ansatz wählen wir hier eine Linearkombination von Translaten einer Gaußschen Ansatzfunktion, da eine solche mithilfe einer schnellen Gauß-Transformation (FGT, Abschn. 3.3.2) effizient ausgewertet werden kann. Die Skalierungskoeffizienten für die Translate werden so bestimmt, dass die SGL-Fourier-Koeffizienten des Approximanten bis zu einer vorgegebenen Bandbreite mit den SGL-Fourier-Koeffizienten von f übereinstimmen. Dies erreichen wir, indem wir spezielle Ansatzpunkte für die Translate der Gaußschen Ansatzfunktion wählen, wofür wir uns insbesondere die Theorie der schnellen Fourier-Transformation auf der Rotationsgruppe SO(3) zunutze machen (FSOFT, Abschn. 3.3.1).

Die schnellen SGL-Fourier-Transformationen für gestreute Daten sind, im Gegensatz zu den exakten Transformationen für reguläre Daten, *approximative* Algorithmen, deren Ergebnissen selbst in exakter Arithmetik ein Fehler innewohnt. Daher spielt bei unseren

[8] Charles William Clenshaw (⋆1926 in Southend-on-Sea, Großbritannien, †2004 in Preston in der Grafschaft Lancashire, ebd.)

Untersuchungen der Zusammenhang zwischen dem Fehler der schnellen Algorithmen und deren Rechenaufwand eine besonders wichtige Rolle.

Bevor wir mit der Entwicklung der schnellen SGL-Fourier-Transformationen beginnen, motivieren wir die in dieser Arbeit vorgestellten Algorithmen anhand eines typischen Anwendungsbeispiels: Für zwei geeignete Funktionen $f, g : \mathbb{R}^3 \to \mathbb{C}$ maximiere man den Absolutbetrag des gewichteten *Überlappungsintegrals*

$$I(R,t) \coloneqq \int_{\mathbb{R}^3} f(R^{-1}x - t)\overline{g(x)}\,\mathrm{e}^{-\|x\|_2^2}\,\mathrm{d}x \tag{E.3}$$

in Abhängigkeit von der Rotation $R \in \mathrm{SO}(3)$ und dem Translationsvektor $t \in \mathbb{R}^3$. Dieses Problem kann auf folgende Art und Weise angegangen werden. Nehmen wir an, dass die Funktionen f und g sich als endliche Linearkombination der SGL-Basisfunktionen – also als bandbegrenzte Funktionen – darstellen lassen,

$$\begin{aligned} f &= \sum\nolimits_{nlm} \hat{f}_{nlm} H_{nlm}, \\ g &= \sum\nolimits_{nlm} \hat{g}_{nlm} H_{nlm}, \end{aligned}$$

wobei jeweils über die entsprechenden beschränkten Indexbereiche der SGL-Basisfunktionen summiert wird. Die Koeffizienten $\hat{f}_{nlm} = \langle f, H_{nlm}\rangle_H$ und $\hat{g}_{nlm} = \langle g, H_{nlm}\rangle_H$ sind die SGL-Fourier-Koeffizienten von f bzw. g. Indem wir die obigen Darstellungen von f und g in (E.3) einsetzen, nimmt das Überlappungsintegral I die Form

$$I(R,t) = \sum\nolimits_{nlm} \sum\nolimits_{n'l'm'} \hat{f}_{nlm}\,\overline{\hat{g}_{n'l'm'}} \int_{\mathbb{R}^3} H_{nlm}(R^{-1}x - t)\overline{H_{n'l'm'}(x)}\,\mathrm{e}^{-\|x\|_2^2}\,\mathrm{d}x \tag{E.4}$$

an. Dieser Ansatz hat den entscheidenden Vorteil, dass die SGL-Fourier-Koeffizienten von f und g nur einmal berechnet werden müssen, um das Integral I für viele verschiedene Rotationen und Translationen auswerten zu können. Als zentrales Ergebnis dieser Arbeit stehen zu diesem Zweck nunmehr die hier entwickelten schnellen SGL-Fourier-Transformationen zur Verfügung. Darüber hinaus ist es möglich, das von f und g unabhängige Integral auf der rechten Seite von (E.4) auf effiziente Art und Weise auszuwerten. Dies liegt daran, dass das Verhalten der SGL-Basisfunktionen unter Rotationen besonders einfach ist; es wird durch die oben bereits erwähnten Wigner-D-Funktionen beschrieben (Satz 1.6.1). Mit Satz 1.6.2 werden wir in dieser Arbeit als wichtiges Resultat auch eine einfache Beschreibung des Verhaltens der SGL-Basisfunktionen unter Translationen herleiten. Damit steht von nun an eine effiziente algorithmische Maschinerie zur Verfügung, um Probleme der obigen Art in der Praxis effektiv zu lösen.

Abbildung E.1 zeigt eine Übersicht über die in dieser Arbeit verwendeten Klassen von schnellen Algorithmen. Diese schematische Darstellung zeigt insbesondere auch die Beziehungen dieser Klassen zueinander, welche den Aufbau und die Struktur der Arbeit bestimmen. Eine Liste mit den in dieser Arbeit verwendeten Abkürzungen findet sich auf Seite ix direkt nach Kapitel 3.

Konventionen

Zur qualitativen Beurteilung der in dieser Arbeit vorgestellten Algorithmen ziehen wir nicht nur die in den numerischen Tests gemessene Laufzeit und den Fehler zu Rate, sondern bestimmen insbesondere auch die *Rechen-* und *Speicherkomplexität* der Algorithmen. Diese charakterisieren das Wachstum des Rechenaufwands bzw. Speicherbedarfs in Abhängigkeit von Problemgröße. Wie üblich geben wir die Komplexität jeweils in *Landau-Notation*[9] an, wie sie etwa in [Berg, 1968, Kap. 1] eingeführt wird. Die $\mathcal{O}$-Ausdrücke besitzen dabei stets für alle natürlichen Zahlen (m, n, B, M, etc.) Gültigkeit. Der Rechenaufwand ist die Anzahl an benötigten Rechenschritten, wobei ein *Rechenschritt* bzw. eine *Operation* aus einer komplexen Multiplikation und anschließenden Addition besteht. Der Speicherbedarf ist die Anzahl an benötigten *Bits*; acht Bits bilden zusammen ein *Byte*. Wir geben diese Einheiten in der $\mathcal{O}$-Notation nicht explizit an. Bei der Speicherkomplexität wird jeweils weder der Speicherplatz für die Eingabedaten, noch der für die Ausgabedaten berücksichtigt, da dieser Bedarf unabhängig von dem verwendeten Algorithmus ist. Wir unterscheiden im Text nicht explizit, ob eine Klasse von Algorithmen gemeint ist, oder spezielle Vertreter dieser Klasse; dies ist jeweils direkt aus dem Kontext zu entnehmen. So schreiben wir etwa „Die FFT lässt sich direkt auf den mehrdimensionalen Fall verallgemeinern [...]“, aber auch „Die heute in der Praxis verwendeten FFTs [...]“. In dieser Arbeit werden des Öfteren zwei- und dreidimensionale Indexbereiche linearisiert. Wir geben dann stets auch die Formeln an, mit deren Hilfe die ursprünglichen Indizes aus dem linearen zurückgewonnen werden können (z. B. Gln. 2.20 oder Gln. 3.34). Auf eine Darstellung der recht länglichen Herleitung dieser Formeln, die ausnahmslos mit elementaren Mitteln erfolgt, verzichten wir aber. Die Zahlenbereiche $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Z}$, $\mathbb{R}$ und $\mathbb{C}$ sind wie üblich definiert. Für $n \in \mathbb{N}_0$ ist $\Pi_n(\mathbb{R}^3)$ der Raum der komplexwertigen Polynome auf $\mathbb{R}^3$ vom Grad höchstens n. Die Räume $\Pi_n([-1,1])$ und $\Pi_n([0,\infty))$ sind analog definiert. Unter dem *Grad* eines dreidimensionalen Polynoms verstehen wir in dieser Arbeit den Totalgrad. Die nicht explizit angegebenen Einträge in Matrizen sind stets gleich Null. Mit δ_{jk} bezeichnen wir wie üblich das *Kronecker-Symbol*[10],

$$\delta_{jk} := \begin{cases} 1, & \text{wenn } j = k, \\ 0 & \text{sonst.} \end{cases}$$

Danksagung

Mein Dank gebührt Prof. Dr. Jürgen Prestin für sein großes Vertrauen und die vielen motivierenden Gespräche. Von ihm habe ich in den letzten Jahren viel gelernt, nicht nur über die Analysis, sondern auch über akademische Strukturen. Ein wichtiger Ansprechpartner während der Anfertigung dieser Arbeit war für mich auch Prof. Dr. Daniel Potts von der Technischen Universität Chemnitz. Er hat ebenfalls sehr wertvolle fachliche Impulse gegeben und damit insbesondere das Kapitel 3 maßgeblich beeinflusst.

Den Mitarbeiterinnen und Mitarbeitern des Instituts für Mathematik der Universität zu Lübeck danke ich für die produktive Arbeitsatmosphäre und die menschliche Wärme.

[9] Edmund Landau (⋆1877 in Berlin, †1938, ebd.)

[10] Leopold Kronecker (⋆1823 in Liegnitz, heute Legnica in Polen, †1891 in Berlin)

In besonders guter Erinnerung sind mir die vielen Gespräche mit Dr. Jörn Schnieder, Dr. Sabrina Kombrink, Inga Stolz und Kevin Schober, sowie die professionelle technische Unterstützung von Maike Wolf. Den Studenten Thomas Buddenkotte, Denis-Michael Lux, Franziska Prüß und Anna Christine Louise Kuptz danke ich für ihr Engagement und ihr Interesse am Thema. Herr Lux und Frau Prüß haben wertvolle Vorarbeiten bei der Implementierung der iNFSGLFT-A und -B geleistet. Frau Kuptz hat eine kleine Korrektur an der Matrix-Faktorisierung in Satz 2.2.21 angebracht.

Für die biographischen Daten von Charles William Clenshaw bin ich Prof. Dr. Peter Turner von der Clarkson University (Potsdam, NY, USA) zu Dank verpflichtet.

Zutiefst verbunden bin ich meiner Familie und meiner Freundin Tabea, die mich während meines Studiums und der Anfertigung dieser Dissertation stets unterstützt haben. Für den Rückhalt und den Glauben an mich bin ich von Herzen dankbar.

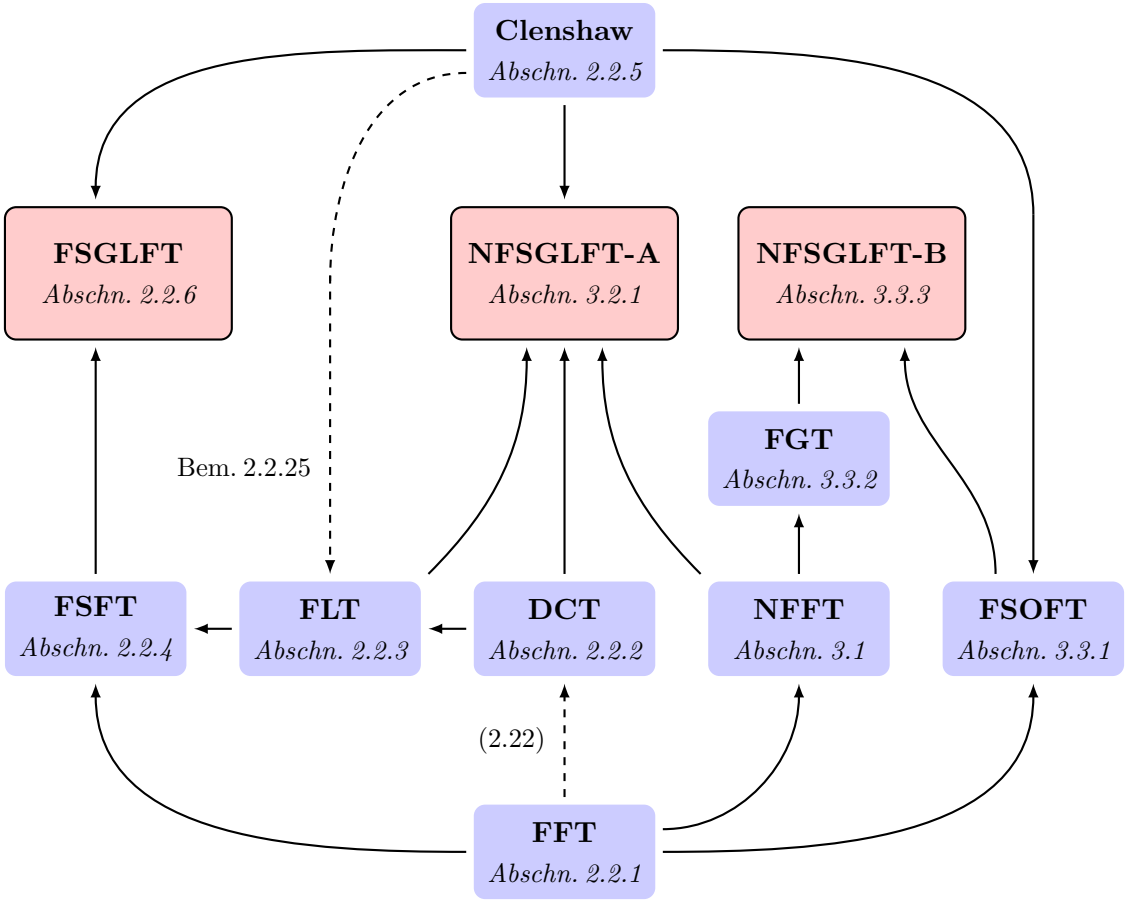

Abbildung E.1. Übersicht über die in dieser Arbeit behandelten Klassen von schnellen Algorithmen mit ihren Beziehungen zueinander. Die gestrichelten Pfeile zeigen alternative Herleitungsmöglichkeiten. Die rot unterlegten, umrandeten Klassen sind neu und bilden das Zentrum dieser Arbeit.

1 Sphärische Gauß-Laguerresche Basisfunktionen

In diesem Kapitel führen wir die SGL-Basisfunktionen ein (Def. 1.5.1) und untersuchen sie auf ihre Eigenschaften. Es handelt sich bei diesen Funktionen um einen vollständigen Satz von Orthogonalpolynomen in dem in (E.1) definierten gewichteten Hilbert-Raum H. Diese Orthogonalpolynome gehen aus einem speziellen Konstruktionsansatz hervor, bei dem der Anschauungsraum $\mathbb{R}^3$ geometrisch in das Produkt aus Radius und Sphäre zerlegt wird. Die SGL-Basisfunktionen setzen sich aus zwei wesentlichen Bestandteilen zusammen: einem verallgemeinerten Laguerre-Polynom im Radialteil und einer Kugelflächenfunktion im Winkelteil. Im Folgenden behandeln wir in Abschnitt 1.1 zunächst die Kugelflächenfunktionen. Es folgt ein kurzer Abschnitt über die hypergeometrische Reihe (Abschn. 1.2) und eine Besprechung der sphärischen Bessel-Funktionen (Abschn. 1.3). Danach führen wir in Abschnitt 1.4 die verallgemeinerten Laguerre-Polynome ein. In Abschnitt 1.5 konstruieren wir dann die SGL-Basisfunktionen und weisen in Satz 1.5.2 ihre Basiseigenschaft nach. Im letzten Abschnitt dieses Kapitels untersuchen wir das Spektralverhalten der SGL-Basisfunktionen unter Rotationen (Satz 1.6.1) und Translationen (Satz 1.6.2), da dies wie eingangs gezeigt für die Anwendung von besonderer Bedeutung ist. Im verhältnismäßig aufwendigen Beweis von Satz 1.6.2 kommen dabei die in den Abschnitten 1.2 und 1.3 vorgestellten analytischen Hilfsmittel erstmals zum Einsatz.

1.1 Die Kugelflächenfunktionen (*spherical harmonics*)

Die Kugelflächenfunktionen spielen eine wichtige Rolle in der Approximationstheorie auf sphärischen Geometrien im Anschauungsraum $\mathbb{R}^3$. Auch in dieser Arbeit sind die Kugelflächenfunktionen von besonderer Bedeutung. In diesem Abschnitt sollen daher die wichtigsten Eigenschaften zusammengefasst werden. Eine ausführliche Darstellung der Theorie findet sich in [Freeden et al., 1998] und [Dai und Xu, 2013].

Definition 1.1.1. Wir definieren die *zweidimensionale Einheitssphäre* als

$$\mathbb{S}^2 := \{x \in \mathbb{R}^3 : \|x\|_2 = 1\}.$$

Jedem Punkt $x = [\xi_0, \xi_1, \xi_2] \in \mathbb{R}^3$ ordnen wir die *Kugelkoordinaten* r, ϑ und φ zu, sodass

$$\begin{aligned} \xi_0 &= r \sin\vartheta \cos\varphi, \\ \xi_1 &= r \sin\vartheta \sin\varphi, \\ \xi_2 &= r \cos\vartheta. \end{aligned}$$

Der *Radius* $r \in [0, \infty)$ entspricht der Norm $\|x\|_2$ des Vektors x, der Winkel $\vartheta \in [0, \pi]$ ist der sogenannte *Polarwinkel* und $\varphi \in [0, 2\pi)$ der *Azimutalwinkel* (vgl. Abb. 1.1). Man beachte, dass diese Zuordnung nicht überall eindeutig ist, was in dieser Arbeit aber nicht

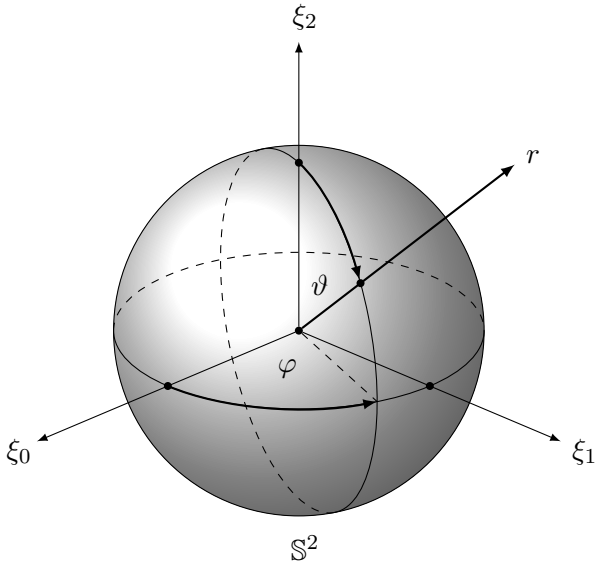

Abbildung 1.1. Kugelkoordinaten im $\mathbb{R}^3$.

weiter von Belang ist. Für $x \in \mathbb{S}^2$ gilt offensichtlich $r = 1$. Stets ohne Verwechslungsgefahr schreiben wir in dieser Arbeit $x = [r, \vartheta, \varphi]$, wenn r, ϑ und φ die Kugelkoordinaten des Punktes x sind, und identifizieren im Folgenden r mit $\|\cdot\|_2$.

Wir betrachten nun den Raum der über $\mathbb{S}^2$ quadratintegrierbaren Funktionen,

$$L^2(\mathbb{S}^2) \coloneqq \left\{ f : \mathbb{S}^2 \to \mathbb{C} \,:\, f \ \sigma\text{-messbar und} \int_{\mathbb{S}^2} |f(w)|^2 \,\mathrm{d}\sigma(w) < \infty \right\},$$

wobei σ das Oberflächenmaß auf $\mathbb{S}^2$ bezeichne. Dieser Raum trägt das Skalarprodukt $\langle \cdot, \cdot \rangle_{L^2(\mathbb{S}^2)}$, welches sich mithilfe der oben eingeführten Kugelkoordinaten als

$$\langle f, g \rangle_{L^2(\mathbb{S}^2)} = \int_0^{\pi} \int_0^{2\pi} f(\vartheta, \varphi) \overline{g(\vartheta, \varphi)} \,\mathrm{d}\varphi \sin\vartheta \,\mathrm{d}\vartheta, \qquad f, g \in L^2(\mathbb{S}^2), \tag{1.1}$$

darstellen lässt. Der Raum $L^2(\mathbb{S}^2)$ ist ein Hilbert-Raum mit induzierter Norm $\|\cdot\|_{L^2(\mathbb{S}^2)}$, innerhalb dessen σ-fast überall identische Funktionen als gleich angesehen werden.

Unter einem Polynom auf $\mathbb{S}^2$ bzw. einem sphärischen Polynom verstehen wir ein Polynom auf $\mathbb{R}^3$, das auf $\mathbb{S}^2$ eingeschränkt wurde. Für $n \in \mathbb{N}_0$ setzen wir $\Pi_n(\mathbb{S}^2) \coloneqq \Pi_n(\mathbb{R}^3)|_{\mathbb{S}^2}$ und bezeichnen die Elemente von $\Pi_n(\mathbb{S}^2)$ als Polynome vom Grad höchstens n auf $\mathbb{S}^2$. Die Gesamtheit aller sphärischen Polynome bezeichnen wir mit $\Pi(\mathbb{S}^2) \coloneqq \Pi(\mathbb{R}^3)|_{\mathbb{S}^2}$. Ohne zu sehr ins Detail zu gehen, halten wir fest, dass die Zuordnung $p \mapsto p|_{\mathbb{S}^2}$, $p \in \Pi(\mathbb{R}^3)$, nicht injektiv ist: für ein gegebenes Polynom p auf $\mathbb{R}^3$ ist nämlich $r^{2k}p$, $k \in \mathbb{N}$, stets ein von p verschiedenes Polynom, das auf $\mathbb{S}^2$ mit p zusammenfällt. Wir können jedoch ein gegebenes Polynom p auf $\mathbb{S}^2$ eindeutig als dreidimensionales Polynom in den Variablen $\sin\vartheta\cos\varphi$, $\sin\vartheta\sin\varphi$ und $\cos\vartheta$ schreiben. Den Grad von p definieren wir dementsprechend als den Grad dieses eindeutig bestimmten Polynoms. Da die Einheitssphäre $\mathbb{S}^2$ kompakt ist, folgt mit dem bekannten Satz von Stone-Weierstraß[1], dass der Raum $\Pi(\mathbb{S}^2)$ dicht in $L^2(\mathbb{S}^2)$ ist.

[1] Marshall Harvey Stone ($\star$1903 in New York, NY, USA, †1989 in Madras, heute Chennai in Indien), Karl Weierstraß ($\star$1815 in Ostenfelde, heute ein Stadtteil von Ennigerloh im Münsterland, †1897 in Berlin)

Wir kommen nun zu den Kugelflächenfunktionen. Um diese zu definieren, führen wir erst die Legendre-Polynome[2] und anschließend die zugeordneten Legendre-Polynome ein.

Definition 1.1.2. Das *Legendre-Polynom* vom Grad $l \in \mathbb{N}_0$ ist definiert als

$$P_l : [-1,1] \to \mathbb{R}, \qquad P_l(\xi) := \frac{1}{2^l l!} \frac{\mathrm{d}^l}{\mathrm{d}\xi^l} (\xi^2 - 1)^l.$$

Definition 1.1.3. Das *zugeordnete* (oder *assoziierte*) *Legendre-Polynom* vom Grad $l \in \mathbb{N}_0$ und der Ordnung $m \in \{-l, \dots, l\}$ wird definiert als

$$P_{lm} : [-1,1] \to \mathbb{R}, \qquad P_{lm}(\xi) := \frac{(-1)^m}{2^l l!} (1-\xi^2)^{m/2} \frac{\mathrm{d}^{l+m}}{\mathrm{d}\xi^{l+m}} (\xi^2 - 1)^l.$$

Man beachte, dass die zugeordneten Legendre-Polynome nur für gerade Ordnung m wirklich Polynome sind (vgl. auch Abschn. 2.2.3). Sie werden daher bisweilen als zugeordnete bzw. assoziierte Legendre-*Funktionen* bezeichnet. Es gilt offenbar $P_{l,0} = P_l$. Die zugeordneten Legendre-Polynome genügen der speziellen Orthogonalitätsrelation [Abramowitz und Stegun, 1972, Gln. 8.14.11 u. 8.14.13]

$$\int_{-1}^{1} P_{km}(\xi) P_{lm}(\xi) \mathrm{d}\xi = \frac{2}{2l+1} \frac{(l+m)!}{(l-m)!} \delta_{kl}, \qquad k, l \in \mathbb{N}_0,\ |m| \le \min\{k, l\}. \tag{1.2}$$

Darüber hinaus genügen die zugeordneten Legendre-Polynome einer speziellen *Drei-Term-Rekursion*: Es gilt für $\xi \in [-1,1]$

$$(l+1-m) P_{l+1,m}(\xi) = (2l+1)\xi P_{lm}(\xi) - (l+m) P_{l-1,m}(\xi), \qquad |m| \le l \in \mathbb{N} \tag{1.3}$$

[Abramowitz und Stegun, 1972, Gl. 8.5.3]. Die zugeordneten Legendre-Polynome sind global beschränkt:

Lemma 1.1.4. *Es gilt*

$$\sqrt{\frac{(l-m)!}{(l+m)!}} |P_{lm}(\xi)| \le \begin{cases} 1, & \textit{wenn } m = 0, \\ 1/\sqrt{2}, & \textit{wenn } |m| \ge 1. \end{cases}$$

Beweis. Im Fall $m = 0$ folgt die Aussage aus der bekannten Abschätzung $|P_l(\xi)| \le 1$ [Freeden et al., 1998, Gl. 3.2.2]. Für den Fall $|m| \ge 1$ vergleiche man die Aussage mit [Lohöfer, 1998, Gl. 5]. □

Im Verlauf dieser Arbeit benötigen wir ferner eine Abschätzung für den Betrag der Funktionen $(1-\xi^2)^{-1/2} P_{lm}(\xi)$. Um eine solche herzuleiten, verwenden wir das folgende Lemma, das zwar bekannt ist, dessen Beweis wir aber in der Literatur nicht finden konnten.

Lemma 1.1.5. *Für* $1 \le |m| \le l$ *gilt*

$$\frac{P_{lm}(\xi)}{\sqrt{1-\xi^2}} = -\frac{1}{2m} \{ P_{l+1,m+1}(\xi) + (l-m+1)(l-m+2) P_{l+1,m-1}(\xi) \}. \tag{1.4}$$

[2] Adrien-Marie Legendre (⋆1752 in Paris, Frankreich, †1833, ebd.)

Beweis. Wir zeigen dies mit [Edmonds, 1996, Gl. 2.5.24]

$$\frac{P_{lm}(\xi)}{\sqrt{1-\xi^2}} = -\frac{1}{2m\xi}\{P_{l,m+1}(\xi) + (l+m)(l-m+1)P_{l,m-1}(\xi)\} \tag{1.5}$$

und [Arfken und Weber 2005, Gl. 12.94; Edmonds 1996, Gl. 2.5.25]

$$\sqrt{1-\xi^2}\,\frac{\mathrm{d}P_{lm}}{\mathrm{d}\xi}(\xi) = -\frac{1}{2}\{P_{l,m+1}(\xi) - (l+m)(l-m+1)P_{l,m-1}(\xi)\}, \tag{1.6}$$

$$(1-\xi^2)\,\frac{\mathrm{d}P_{lm}}{\mathrm{d}\xi}(\xi) = (l+m)P_{l-1,m}(\xi) - l\xi P_{lm}(\xi). \tag{1.7}$$

Aus (1.6) und (1.7) ergibt sich nach Elimination der Ableitung mit $l \mapsto l+1$

$$\frac{P_{lm}(\xi)}{\sqrt{1-\xi^2}} = -\frac{P_{l+1,m+1}(\xi)}{2(l+m+1)} + \frac{l+1}{l+m+1}\frac{\xi P_{l+1,m}(\xi)}{\sqrt{1-\xi^2}} + \frac{l-m+2}{2}P_{l+1,m-1}(\xi). \tag{1.8}$$

Wieder mit $l \mapsto l+1$ folgt aus (1.5)

$$\frac{\xi P_{l+1,m}(\xi)}{\sqrt{1-\xi^2}} = -\frac{1}{2m}\{P_{l+1,m+1}(\xi) + (l+m+1)(l-m+2)P_{l+1,m-1}(\xi)\}. \tag{1.9}$$

Die Formel (1.4) ergibt sich nun durch Einsetzen von (1.9) in (1.8) und Vereinfachen. □

Lemma 1.1.6. *Für* $1 \le |m| \le l < n$ *gilt*

$$\sqrt{\frac{(l-m)!}{(l+m)!}}\,\frac{|P_{lm}(\xi)|}{\sqrt{1-\xi^2}} \le \frac{1}{\sqrt{2}}\frac{l+2}{|m|} \le \frac{n+1}{\sqrt{2}}.$$

Beweis. Mit der Dreiecksungleichung und den Lemmata 1.1.4 und 1.1.5 folgt für $|m| \ge 2$

$$\begin{aligned}
\frac{|P_{lm}(\xi)|}{\sqrt{1-\xi^2}} &\le \frac{1}{2|m|}\left\{\left(\frac{1}{2}\frac{(l+m+2)!}{(l-m)!}\right)^{1/2} + (l-m+1)\,(l-m+2)\left(\frac{1}{2}\frac{(l+m)!}{(l-m+2)!}\right)^{1/2}\right\} \\
&= \frac{1}{2|m|}\left(\frac{1}{2}\frac{(l+m)!}{(l-m)!}\right)^{1/2}\{\sqrt{(l+m+1)(l+m+2)} + \sqrt{(l-m+1)(l-m+2)}\} \\
&\le \frac{l+2}{|m|}\left(\frac{1}{2}\frac{(l+m)!}{(l-m)!}\right)^{1/2},
\end{aligned}$$

wobei sich die zweite Abschätzung aus der bekannten Ungleichung vom arithmetischen und geometrischen Mittel ergibt. Der Fall $|m| = 1$ ist gesondert zu untersuchen, z. B. mithilfe von [Freeden et al., 1998, Gln. 3.2.3 u. 3.2.4]. □

Mithilfe der zugeordneten Legendre-Polynome definieren wir nun die Kugelflächenfunktionen, den zentralen Gegenstand dieses Abschnitts.

Definition 1.1.7. Die *Kugelflächenfunktion* vom Grad $l \in \mathbb{N}_0$ und der Ordnung $m \in \{-l, \dots, l\}$ wird definiert als

$$Y_{lm} : \mathbb{S}^2 \to \mathbb{C}, \qquad Y_{lm}(\vartheta, \varphi) := M_{lm}\,P_{lm}(\cos\vartheta)\,\mathrm{e}^{\mathrm{i}m\varphi}$$

mit der Normierungskonstanten

$$M_{lm} := \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-m)!}{(l+m)!}}.$$

Die besondere Bedeutung der Kugelflächenfunktionen liegt darin, dass Y_{lm} stets ein Polynom vom Grad l auf $\mathbb{S}^2$ ist, es gilt also $Y_{lm} \in \Pi_n(\mathbb{S}^2)$ für $l \leq n$. Tatsächlich bilden die Kugelflächenfunktionen vom Grad höchstens n stets eine Basis von $\Pi_n(\mathbb{S}^2)$ (vgl. [Freeden et al., 1998, Abschn. 1.3]). Außerdem ergibt sich durch die Substitution $\xi = \cos\vartheta$ in der Orthogonalitätsrelation (1.2) zusammen mit Orthonormalität der trigonometrischen Monome auf $[0, 2\pi)$, dass die Kugelflächenfunktionen ein *Orthonormalsystem* in $L^2(\mathbb{S}^2)$ bilden (vgl. Gl. 1.1). Es gilt daher mit dem oben Gesagten der wichtige

Satz 1.1.8. *Für $n \in \mathbb{N}_0$ bilden die Kugelflächenfunktionen Y_{lm}, $|m| \leq l \leq n$, eine Orthonormalbasis von $\Pi_n(\mathbb{S}^2)$. Insbesondere bilden die Kugelflächenfunktionen ein vollständiges Orthonormalsystem in $L^2(\mathbb{S}^2)$.*

Der zweite Teil des Satzes sagt aus, dass für $f \in L^2(\mathbb{S}^2)$ die Fourier-Partialsummen

$$\sum\nolimits_{|m| \leq l < n} \langle f, Y_{lm} \rangle_{L^2(\mathbb{S}^2)} Y_{lm}, \qquad n \in \mathbb{N},$$

bezüglich $\|\cdot\|_{L^2(\mathbb{S}^2)}$ für $n \to \infty$ gegen f konvergieren. Äquivalent dazu gilt die Parsevalsche[3] Gleichung

$$\|f\|^2_{L^2(\mathbb{S}^2)} = \sum\nolimits_{|m| \leq l \in \mathbb{N}_0} |\langle f, Y_{lm} \rangle_{L^2(\mathbb{S}^2)}|^2, \qquad f \in L^2(\mathbb{S}^2).$$

Auch in dieser Arbeit wird sich das bekannte Additionstheorem für die Kugelflächenfunktionen als nützliches Werkzeug erweisen. Es ist hierbei und im Folgenden $\langle \cdot, \cdot \rangle_2$ das Standard-Skalarprodukt auf $\mathbb{R}^3$.

Lemma 1.1.9 (Sphärisches Additionstheorem). *Für festes $l \in \mathbb{N}_0$ und $v, w \in \mathbb{S}^2$ gilt*

$$\frac{4\pi}{2l+1} \sum\nolimits_{|m| \leq l} Y_{lm}(v) \overline{Y_{lm}(w)} = P_l(\langle v, w \rangle_2).$$

Beweis. Siehe [Freeden et al., 1998, Satz 3.1.3]. □

Zwei weitere Eigenschaften der Kugelflächenfunktionen sind für uns von besonderer Bedeutung: ihr relativ einfaches Verhalten unter Rotationen, das sich direkt auf die SGL-Basisfunktionen überträgt (vgl. Abschn. 1.6), sowie ihr enger Zusammenhang mit den harmonischen Polynomen auf $\mathbb{R}^3$. Diese beiden Eigenschaften sollen nun besprochen werden. Wir beginnen mit der letzteren.

Ein Polynom p auf $\mathbb{R}^3$ heißt *homogen vom Grad n*, wenn

$$p(\lambda x) = \lambda^n p(x), \qquad \lambda \in \mathbb{R},\ x \in \mathbb{R}^3.$$

Den Raum der Polynome auf $\mathbb{R}^3$, die homogen vom Grad n sind, bezeichnen wir als $\mathrm{Hom}_n(\mathbb{R}^3)$. Es sei weiter

$$\Delta := \frac{\partial^2}{\partial \xi_0^2} + \frac{\partial^2}{\partial \xi_1^2} + \frac{\partial^2}{\partial \xi_2^2}$$

der bekannte *Laplace-Operator*[4]. Wir nennen ein Polynom $p \in \mathrm{Hom}_n(\mathbb{R}^3)$ *harmonisch*, wenn

$$\Delta p \equiv 0.$$

[3] Marc-Antoine Parseval (⋆1755 in Rosières-aux-Salines, Frankreich, †1836 in Paris, ebd.)
[4] Pierre-Simon Laplace (⋆1749 in Beaumont-en-Auge, Frankreich, †1827 in Paris, ebd.)

Den Raum der Polynome auf $\mathbb{R}^3$, die homogen vom Grad n und zudem harmonisch sind, bezeichnen wir als $\mathrm{Harm}_n(\mathbb{R}^3)$. In Analogie zu $\Pi_n(\mathbb{S}^2)$ schreiben wir $\mathrm{Harm}_n(\mathbb{S}^2) := \mathrm{Harm}_n(\mathbb{R}^3)|_{\mathbb{S}^2}$. Das folgende Lemma zeigt uns, dass sich die homogenen Polynome auf bestimmte Art und Weise in ihre harmonischen Bestandteile zerlegen lassen.

Lemma 1.1.10. *Es sei $n \in \mathbb{N}_0$. Der Raum $\mathrm{Hom}_n(\mathbb{R}^3)$ lässt sich als direkte innere Summe*

$$\mathrm{Hom}_n(\mathbb{R}^3) \;=\; \bigoplus_{j=0}^{\lfloor n/2 \rfloor} r^{2j}\,\mathrm{Harm}_{n-2j}(\mathbb{R}^3)$$

darstellen.

Beweis. Siehe [Freeden et al., 1998, Satz 2.2.3]. □

In der Monomdarstellung sieht man, dass sich jedes Polynom auf $\mathbb{R}^3$ in seine homogenen Bestandteile zerlegen lässt. Es folgt daraus unmittelbar:

Folgerung 1.1.11. *Es sei $n \in \mathbb{N}_0$. Der Raum $\Pi_n(\mathbb{R}^3)$ besitzt die Zerlegung*

$$\Pi_n(\mathbb{R}^3) \;=\; \bigoplus_{i=0}^{n} \mathrm{Hom}_i(\mathbb{R}^3) \;=\; \bigoplus_{i=0}^{n} \bigoplus_{j=0}^{\lfloor i/2 \rfloor} r^{2j}\,\mathrm{Harm}_{i-2j}(\mathbb{R}^3).$$

An dieser Stelle kommen die Kugelflächenfunktionen ins Spiel. Es gilt nämlich:

Lemma 1.1.12. *Für festes $l \in \mathbb{N}_0$ ist*

$$\mathrm{span}\{r^l Y_{lm} : |m| \le l\} \;=\; \mathrm{Harm}_l(\mathbb{R}^3).$$

Beweis. Wir halten fest, dass stets

$$\mathrm{Harm}_l(\mathbb{R}^3) \;=\; r^l\,\mathrm{Harm}_l(\mathbb{S}^2)$$

gilt, denn die Polynome in $\mathrm{Harm}_l(\mathbb{R}^3)$ sind nach Definition homogen. Die Aussage des Lemmas folgt daher mit [Dai und Xu, 2013, Abschn. 1.6.2]

$$\mathrm{span}\{Y_{lm} : |m| \le l\} \;=\; \mathrm{Harm}_l(\mathbb{S}^2).$$ □

Bemerkung 1.1.13. Die Funktionen $r^l Y_{lm}$ werden im Englischen als *solid harmonics* bezeichnet. Mit Folgerung 1.1.11 und Lemma 1.1.12 weisen wir in Abschnitt 1.5 die Vollständigkeit der SGL-Basisfunktionen nach.

Wir besprechen nun noch das Rotationsverhalten der Kugelflächenfunktionen. Dazu definieren wir an dieser Stelle die Rotationsgruppe SO(3) (SO für *speziell*[*e*] *orthogonal*[*e*] [Gruppe] bzw. *special orthogonal* [*group*]), deren Elemente bekanntlich die Rotationen im Anschauungsraum $\mathbb{R}^3$ sind.

Definition 1.1.14. Die *dreidimensionale Rotationsgruppe* SO(3) wird definiert als die Menge der orthogonalen Matrizen der Größe 3×3 mit Determinante gleich Eins,

$$\mathrm{SO}(3) \;:=\; \{R \in \mathbb{R}^{3\times 3} : R^{-1} = R^{\mathrm{T}},\ \det(R) = 1\}.$$

Für eine Rotation $R \in \mathrm{SO}(3)$ definieren wir den *Rotationsoperator*

$$Rf \coloneqq f \circ R^{-1}, \qquad f : \mathbb{R}^3 \to \mathbb{C}. \tag{1.10}$$

Die besondere Bedeutung der Kugelflächenfunktionen im Zusammenhang mit Rotationen ergibt sich aus der *Rotationsinvarianz* der Räume $\mathrm{Harm}_l(\mathbb{R}^3)$:

Lemma 1.1.15. *Für festes $l \in \mathbb{N}_0$ und $R \in \mathrm{SO}(3)$ gilt*

$$\mathrm{span}\{RY_{lm} : |m| \leq l\} = \mathrm{Harm}_l(\mathbb{S}^2).$$

Beweis. Siehe [Freeden et al., 1998, S. 47 ff.]. □

Insbesondere lehrt uns dieses Lemma, dass sich eine rotierte Kugelflächenfunktion RY_{lm} stets nach Kugelflächenfunktionen vom gleichen Grad entwickeln lässt:

$$RY_{lm} = \sum\nolimits_{|m'| \leq l} \langle RY_{lm}, Y_{lm'} \rangle_{L^2(\mathbb{S}^2)} Y_{lm'}. \tag{1.11}$$

Die Koeffizienten $\langle RY_{lm}, Y_{lm'} \rangle_{L^2(\mathbb{S}^2)}$ sind für uns von besonderem Interesse.

Definition 1.1.16. Die Funktion $D^{(l)}_{mm'} : \mathrm{SO}(3) \to \mathbb{C}$, $D^{(l)}_{mm'}(R) \coloneqq \langle RY_{lm}, Y_{lm'} \rangle_{L^2(\mathbb{S}^2)}$, heißt *Wigner-D-Funktion* der Ordnungen l, m und m'.

Den Wigner-D-Funktionen begegnen wir in Abschnitt 1.6 bei der Untersuchung des Rotationsverhaltens der SGL-Basisfunktionen, sowie bei der Besprechung der schnellen Fourier-Transformation auf SO(3) in Abschnitt 3.3.1. Die Wigner-D-Funktionen werden bisweilen als *verallgemeinerte Kugelflächenfunktionen* bezeichnet. Zum Abschluss definieren wir noch die Wigner-3j-Symbole, denen wir in Abschnitt 1.3 begegnen. Eine ausführliche Darstellung der Theorie der Wigner-D-Funktionen und des Wigner-3j-Symbols findet sich in [Biedenharn und Louck, 1981, Kap. 3].

Definition 1.1.17. Es seien $|m| \leq l$, $|m'| \leq l'$, sowie $|m''| \leq l''$. Wir definieren das zugehörige *Wigner-3j-Symbol* als

$$\begin{pmatrix} l & l' & l'' \\ m & m' & m'' \end{pmatrix} \coloneqq \sqrt{\frac{8\pi}{(2l+1)(2l'+1)(2l''+1)}} \frac{\int_{\mathbb{S}^2} Y_{lm} Y_{l'm'} Y_{l''m''} \,\mathrm{d}\sigma}{\sqrt{\int_{-1}^{1} P_l(\xi) P_{l'}(\xi) P_{l''}(\xi) \mathrm{d}\xi}},$$

sofern folgende Bedingungen erfüllt sind:

(i) $m + m' + m'' = 0$,

(ii) $|l - l'| \leq l'' \leq l + l'$,

(iii) $l + l' + l''$ gerade, wenn $m = m' = m'' = 0$.

Andernfalls setzen wir das Wigner-3j-Symbol gleich Null.

1.2 Die hypergeometrische Reihe

Die hypergeometrische Reihe und die daraus abgeleiteten Funktionen ${}_1F_1$ und ${}_2F_1$ sind in dieser Arbeit in erster Linie ein technisches Hilfsmittel. Wir nehmen hier nur die Definition und die für uns wichtigen Eigenschaften mit auf.

Definition 1.2.1. Für $p, q \in \mathbb{N}$ mit $p \leq q+1$ und Parameter $\alpha_0, ..., \alpha_{p-1} \in \mathbb{C}$; $\beta_0, ..., \beta_{q-1} \in \mathbb{C} \setminus \{0, -1, -2, \dots\}$, definiert man die *hypergeometrische Reihe* ${}_pF_q$ als die spezielle Potenzreihe

$$ {}_pF_q(\alpha_0, \dots, \alpha_{p-1}; \beta_0, \dots, \beta_{q-1}; \xi) \coloneqq \sum_{k=0}^{\infty} \frac{(\alpha_0)_k \cdots (\alpha_{p-1})_k}{(\beta_0)_k \cdots (\beta_{q-1})_k} \frac{\xi^k}{k!}, \qquad \xi \in \mathbb{C}, $$

mit dem *Pochhammer-Symbol*[5]

$$ (\zeta)_k \coloneqq \zeta(\zeta+1)(\zeta+2)\cdots(\zeta+k-1), \qquad \zeta \in \mathbb{C},\ k \in \mathbb{N}_0, \tag{1.12} $$

das auch *aufsteigende Fakultät* genannt wird.

Bemerkung 1.2.2. Entsprechend der Definition der hypergeometrischen Reihe spielt die Reihenfolge der Parameter $\alpha_0, ..., \alpha_{p-1}$ und $\beta_0, ..., \beta_{q-1}$ keine Rolle. Ist einer der Parameter $\alpha_0, ..., \alpha_{p-1}$ nichtpositiv und ganzzahlig, so reduziert sich die hypergeometrische Reihe nach der Definition des Pochhammer-Symbols auf ein Polynom.

In dieser Arbeit begegnen wir nur den Spezialfällen ${}_1F_1$ und ${}_2F_1$. Die Reihe ${}_1F_1$ wird *Kummersche*[6] *(konfluente hypergeometrische) Funktion* genannt. Diese Reihe konvergiert absolut und gleichmäßig auf beschränkten Gebieten [Andrews et al., 1999, Satz 2.1.1]. Die Funktion ${}_2F_1$, oft als *die* (*gewöhnliche* oder *Gaußsche*) *hypergeometrische Funktion* bezeichnet, tritt in dieser Arbeit nur als Polynom auf. Als wichtige Eigenschaft der Funktion ${}_1F_1$ halten wir fest:

Lemma 1.2.3 (Kummer-Transformation). *Es gilt stets*

$$ {}_1F_1(\alpha; \beta; \xi) = \mathrm{e}^{\xi}\, {}_1F_1(\beta - \alpha; \beta; -\xi). $$

Beweis. Siehe [Andrews et al., 1999, Abschn. 4.1]. □

1.3 Die (sphärischen) Bessel-Funktionen

Wie die hypergeometrischen Funktionen definieren wir auch die Bessel-Funktionen[7] über eine spezielle Potenzreihe. Auch sie spielen in dieser Arbeit eine eher technische Rolle. Für eine sehr ausführliche Darstellung der Theorie sei auf [Watson, 1995] verwiesen.

Definition 1.3.1. Die *Bessel-Funktion* (*erster Art*) der Ordnung $\nu \in \mathbb{C} \setminus \{-1, -2, \dots\}$ wird definiert als

$$ J_\nu(\xi) \coloneqq \sum_{m=0}^{\infty} \frac{(-1)^m}{m!\,\Gamma(m+\nu+1)} \left(\frac{\xi}{2}\right)^{2m+\nu}, \qquad \xi \in \mathbb{C}. $$

Bemerkung 1.3.2. Für $\mathfrak{Re}(\nu) \geq 0$ konvergiert die obige Reihe absolut und gleichmäßig auf beschränkten Gebieten in $\mathbb{C}$, für $\mathfrak{Re}(\nu) < 0$ divergiert sie im Ursprung [Watson, 1995, Abschn. 3.1.3].

[5] Leo August Pochhammer (⋆1841 in Stendal, †1920 in Kiel)
[6] Ernst Eduard Kummer (⋆1810 in Sorau, heute Żary in Polen, †1893 in Berlin)
[7] Friedrich Wilhelm Bessel (⋆1784 in Minden, †1846 in Königsberg, heute Kaliningrad in Russland)

Die Bessel-Funktionen hängen über eine für uns nützliche Integraltransformation mit der konfluenten hypergeometrischen Funktion ${}_1F_1$ zusammen. Es handelt sich hierbei um einen Spezialfall der *Hankel-Transformation*[8] [Andrews et al., 1999, Abschn. 4.11].

Lemma 1.3.3 (Hankelsche Formel). *Es seien* $\alpha, \mu \in \mathbb{C}$, $\nu \in \mathbb{C} \setminus \{-1, -2, \dots\}$ *und* $\beta > 0$ *gegeben. Unter der Voraussetzung* $\mathfrak{Re}(\mu + \nu) > 0$ *gilt*

$$\int_0^\infty J_\nu(\alpha\xi)\,\xi^{\mu-1}\,\mathrm{e}^{-\beta^2\xi^2}\mathrm{d}\xi \;=\; \frac{\Gamma(\mu/2+\nu/2)\,(\alpha/2\beta)^\nu}{2\,\beta^\mu\,\Gamma(\nu+1)}\,{}_1F_1\!\left(\frac{\mu+\nu}{2};\nu+1;-\frac{\alpha^2}{4\beta^2}\right).$$

Beweis. Siehe [Watson, 1995, S. 393 ff.]. □

Mithilfe der Bessel-Funktionen definieren wir nun die sphärischen Bessel-Funktionen.

Definition 1.3.4. Die *sphärische Bessel-Funktion* (*erster Art*) der Ordnung $n \in \mathbb{N}_0$ wird definiert als

$$j_n : (0,\infty) \to \mathbb{R}, \qquad j_n(\xi) \;\coloneqq\; \sqrt{\frac{\pi}{2\xi}}\,J_{n+1/2}(\xi).$$

Bemerkung 1.3.5. Vermöge des Grenzwerts [Abramowitz und Stegun, 1972, Gl. 10.1.4]

$$\lim_{\xi\to 0} j_n(\xi) \;=\; \begin{cases} 1, & \text{wenn } n = 0, \\ 0 & \text{sonst,} \end{cases}$$

lassen sich die sphärischen Bessel-Funktionen im Ursprung stetig fortsetzen.

Der Betrag der sphärischen Bessel-Funktionen lässt sich wie folgt abschätzen:

Lemma 1.3.6 (vgl. [Watson, 1995, S. 53]). *Es gilt für* $l \in \mathbb{N}_0$

$$|j_l(\xi)| \;\le\; \frac{(\xi/2)^l}{l!}, \qquad \xi \in [0,\infty).$$

Als wichtiges Werkzeug benutzen wir in dieser Arbeit die folgende Integraltransformation und die dazugehörige Inversionsformel. Es handelt sich auch bei dieser Integraltransformation um eine spezielle Klasse von Hankel-Transformationen.

Definition 1.3.7. Es seien $n \in \mathbb{N}_0$ und $\gamma > 0$ gegeben. Für geeignete Funktionen $f : (0,\infty) \to \mathbb{R}$ definieren wir die *gewichtete sphärische Bessel-Transformation* über

$$\tilde{f}_n(\gamma;\beta) \;\coloneqq\; \sqrt{\frac{2}{\pi}}\int_0^\infty f(\xi)\,j_n(\beta\xi)\,\xi^2\,\mathrm{e}^{-\gamma\xi^2}\mathrm{d}\xi, \qquad \beta \in (0,\infty).$$

Eine große Klasse von Funktionen, inklusive des radialen Teils R_{nl} der SGL-Basisfunktionen (Def. 1.5.1), lässt sich mithilfe der folgenden Inversionsformel aus der jeweiligen Transformierten rekonstruieren.

[8] Hermann Hankel (⋆1839 in Halle an der Saale, †1873 in Schramberg)

Lemma 1.3.8. *Es seien $n \in \mathbb{N}_0$ und $\gamma > 0$ gegeben. Weiter sei $f : (0, \infty) \to \mathbb{R}$ stetig und von beschränkter Variation auf jedem endlichen Teilintervall von $(0, \infty)$. Gilt nun*

$$\int_0^\infty |f(\xi)| \xi \, \mathrm{e}^{-\gamma \xi^2} \mathrm{d}\xi \; < \; \infty,$$

so kann f punktweise aus der Transformierten $\tilde{f}_n(\gamma; \cdot)$ zurückgewonnen werden durch

$$f(\xi) \; = \; \mathrm{e}^{\gamma \xi^2} \sqrt{\frac{2}{\pi}} \int_0^\infty \tilde{f}_n(\gamma; \beta) j_n(\beta \xi) \beta^2 \mathrm{d}\beta, \qquad \xi \in (0, \infty).$$

Beweis. Siehe [Watson, 1995, Abschn. 14.4]. □

Das folgende Lemma stellt über die Exponentialfunktion einen Zusammenhang zwischen den sphärischen Bessel-Funktionen und den Kugelflächenfunktionen her. Dieses Resultat wird im Englischen als *plane wave expansion* bezeichnet.

Lemma 1.3.9 (Sphärisches Bessel-Additionstheorem). *Für $v, w \in \mathbb{S}^2$ und $\xi \in (0, \infty)$ gilt*

$$\mathrm{e}^{\mathrm{i}\xi \langle v, w \rangle} \; = \; 4\pi \sum_{l=0}^{\infty} \sum\nolimits_{|m| \le l} \mathrm{i}^l \, j_l(\xi) \, Y_{lm}(v) \overline{Y_{lm}(w)}.$$

Die Konvergenz ist bei festem ξ gleichmäßig in v und w.

Beweis. Siehe [Watson, 1995, Abschn. 11.5] für eine Herleitung der Formel. Die gleichmäßige Konvergenz folgt aus dem Additionstheorem für die Kugelflächenfunktionen (Lem. 1.1.9) zusammen mit der Abschätzung $|P_l| \le 1$ (vgl. Lem. 1.1.4) und Lemma 1.3.6. □

Nützlich ist für uns auch das nächste Lemma, wenn wir in Abschnitt 1.6 das Verhalten der SGL-Basisfunktionen unter Translationen untersuchen. Es beschreibt das Verhalten der Funktionen $j_l(r) Y_{lm}(\vartheta, \varphi)$ unter Translationen entlang der ξ_2-Achse (vgl. Abb. 1.1). Man beachte, dass solche Translationen keinen Einfluss auf den Azimutalwinkel φ haben. Wir definieren an dieser Stelle für $\nu > 0$ den *Translationsoperator*

$$T_2(\nu) f(\xi_0, \xi_1, \xi_2) \; := \; f(\xi_0, \xi_1, \xi_2 - \nu), \qquad f : \mathbb{R}^3 \to \mathbb{C}. \tag{1.13}$$

Es gilt nun:

Lemma 1.3.10 (Sphärisches Bessel-Additionstheorem II). *Es seien $|m| \le l \in \mathbb{N}_0$ und $\nu > 0$ gegeben. Dann gilt für $\beta \in (0, \infty)$*

$$j_l(\beta r) Y_{lm}(\vartheta, \varphi) \; = \; \sum_{l'=0}^{\infty} \sum_{k=|l-l'|}^{l+l'} A_k^{(ll'm)} \, j_k(\beta \nu) \, T_2(\nu) [j_{l'}(\beta \, \cdot) Y_{l'm}](r, \vartheta, \varphi), \tag{1.14}$$

wobei die Koeffizienten $A_k^{(ll'm)}$ durch

$$A_k^{(ll'm)} \; := \; (-1)^{(k-l+l')/2+m} \sqrt{(2l+1)(2l'+1)} \, (2k+1) \begin{pmatrix} l & l' & k \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l & l' & k \\ m & -m & 0 \end{pmatrix}$$

gegeben sind. Für festes r und β konvergiert die Reihe in (1.14) gleichmäßig auf $\mathbb{S}^2$.

Beweis. Für die Herleitung der Formel siehe [Ritchie, 2005, Anh. A]. Um die gleichmäßige Konvergenz einzusehen, zeigt man mit Lemma 1.1.4 unter Berücksichtigung von Definition 1.1.17 zunächst die Abschätzungen

$$\begin{aligned} |A_k^{(ll'm)}| &\leq \sqrt{(2l+1)(2l'+1)}(2k+1), \\ |Y_{l'm}| &\leq \sqrt{\frac{2l'+1}{4\pi}}, \end{aligned}$$

und nutzt dann wieder Lemma 1.3.6. □

Bemerkung 1.3.11. Bei einer Translation entlang der *negativen* ξ_2-Achse, d. h. beim Übergang zu $-\nu < 0$, ergibt sich auf der rechten Seite von (1.14) der zusätzliche Faktor $(-1)^k$. Dies erkennt man, indem man in der Herleitung von Ritchie die Identität

$$Y_{lm}(\vartheta + \pi, \varphi) = (-1)^{l+m} Y_{lm}(\vartheta, \varphi)$$

beachtet, die direkt aus der Definition 1.1.3 der zugeordneten Legendre-Polynome folgt.

Bemerkung 1.3.12. Ritchie bemerkt, dass die Koeffizienten $A_k^{(ll'm)}$ tatsächlich nicht von dem Vorzeichen von m abhängen. Darüber hinaus wird der Umgang mit diesen Koeffizienten dadurch vereinfacht, dass $A_k^{(ll'm)} = 0$ gilt, wenn $l - l' + k$ ungerade ist. Dies folgt daraus, dass $l - l' + k = (l + l' + k) - 2l'$ genau dann ungerade ist, wenn $l + l' + k$ ungerade ist. Dann aber gilt (vgl. Def. 1.1.17, iii)

$$\begin{pmatrix} l & l' & k \\ 0 & 0 & 0 \end{pmatrix} = 0.$$

1.4 Die verallgemeinerten Laguerre-Polynome

Wir betrachten nun bei festem reellen $\alpha > -1$ die spezielle Gewichtsfunktion $\xi^\alpha \exp(-\xi)$ auf der positiven Halbachse $[0, \infty)$. Aus der Theorie der univariaten Orthogonalpolynome ist bekannt, dass es genau einen Satz von reellwertigen Polynomen $L_n^{(\alpha)}$ jeweils vom Grad n gibt, die der Orthogonalitätsrelation

$$\int_0^\infty L_m^{(\alpha)}(\xi)\, L_n^{(\alpha)}(\xi)\, \xi^\alpha \mathrm{e}^{-\xi}\, \mathrm{d}\xi = \frac{\Gamma(n+\alpha+1)}{n!}\, \delta_{mn}, \qquad m, n \in \mathbb{N}_0, \tag{1.15}$$

genügen [Mhaskar, 1996, Satz 1.1.2]. Dies sind die *verallgemeinerten Laguerre-Polynome*, zu denen der geschlossene Ausdruck

$$L_n^{(\alpha)}(\xi) = \sum_{k=0}^{n} \frac{(-1)^k}{k!} \binom{n+\alpha}{n-k} \xi^k, \qquad \xi \in [0, \infty), \tag{1.16}$$

angegeben werden kann [Andrews et al., 1999, Gl. 6.2.2]. Hierbei wird für $\nu \in \mathbb{R}$ und $k \in \mathbb{Z}$ der *verallgemeinerte Binomialkoeffizient*

$$\binom{\nu}{k} := \begin{cases} \nu(\nu-1)(\nu-2)\cdots(\nu-k+1)/k!, & \text{wenn } k > 0, \\ 1, & \text{wenn } k = 0, \\ 0 & \text{sonst}, \end{cases}$$

verwendet. Die verallgemeinerten Laguerre-Polynome lassen sich auch mithilfe der konfluenten hypergeometrischen Funktion ${}_1F_1$ darstellen:

Lemma 1.4.1 (s. [Abramowitz und Stegun, 1972, Gl. 13.6.9]). *Es seien $n \in \mathbb{N}_0$ und $\alpha > -1$. Dann gilt*

$$L_n^{(\alpha)}(\xi) \;=\; \binom{n+\alpha}{n} {}_1F_1(-n; 1+\alpha; \xi).$$

Aus der Definition folgt unmittelbar, dass für jedes $n \in \mathbb{N}_0$ bei festem $\alpha > -1$ die ersten $n+1$ Polynome $L_0^{(\alpha)}, \ldots, L_n^{(\alpha)}$ eine Basis des $\Pi_n([0,\infty))$ bilden. Von besonderer Bedeutung ist für uns ferner die Tatsache, dass die verallgemeinerten Laguerre-Polynome der speziellen *Drei-Term-Rekursionsformel*

$$(1.17)\quad (k+1)\,L_{k+1}^{(\alpha)}(\xi) = (2k+\alpha+1-\xi)\,L_k^{(\alpha)}(\xi) - (k+\alpha)\,L_{k-1}^{(\alpha)}(\xi), \quad \xi \in [0,\infty),\ k \in \mathbb{N},$$

genügen [Andrews et al., 1999, Gl. 6.2.5]; diese Rekursionsformel ermöglicht uns nämlich eine schnelle und genaue Berechnung der verallgemeinerten Laguerre-Polynome. Der Betrag dieser Polynome lässt sich wie folgt abschätzen:

Lemma 1.4.2 (vgl. [Abramowitz und Stegun, 1972, Gl. 22.14.13]). *Für $\alpha \geq 0$ gilt*

$$\left|L_n^{(\alpha)}(\xi)\right| \;\leq\; \binom{n+\alpha}{n} \mathrm{e}^{\xi/2}, \qquad \xi \in [0,\infty).$$

Mithilfe von (1.16) können wir die verallgemeinerten Laguerre-Polynome zwanglos auf die gesamte reelle Achse fortsetzen. Die folgenden zwei Hilfssätze sind für uns nützlich.

Lemma 1.4.3 (s. [Abramowitz und Stegun, 1972, Gl. 22.12.7]). *Es gilt stets*

$$\sum_{n=0}^{m} \binom{m+\alpha}{m-n} L_n^{(\alpha)}(\xi)\,\psi^n\,(1-\psi)^{m-n} \;=\; L_m^{(\alpha)}(\xi\psi).$$

Lemma 1.4.4 (vgl. [Andrews et al., 1999, Gl. 6.2.4]). *Es gilt für $|\psi| < 1$ die Formel*

$$\sum_{n=0}^{\infty} L_n^{(\alpha)}(\xi)\,\psi^n \;=\; (1-\psi)^{-\alpha-1}\,\mathrm{e}^{-\frac{\psi}{1-\psi}\xi}.$$

Die Konvergenz ist hierbei absolut.

Das folgende Resultat, das wir später brauchen, zeigt uns abschließend, dass die verallgemeinerten Laguerre-Polynome bei festem Parameter und negativem Argument ein *subexponentielles* Wachstum bezüglich des Grades n aufweisen.

Lemma 1.4.5 (vgl. [Szegő, 1975, Satz 8.22.3]). *Bei festem α und $\xi > 0$ gilt*

$$L_n^{(\alpha)}(-\xi) \;=\; \frac{n^{\alpha/2-1/4}}{2\sqrt{\pi}} \frac{\mathrm{e}^{-\xi/2}}{\xi^{\alpha/2+1/4}} \mathrm{e}^{2\sqrt{n\xi}} \left(1 + \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)\right) \quad (n \to \infty).$$

Diese Abschätzung ist auf endlichen, abgeschlossenen Intervallen gleichmäßig erfüllt.[9]

[9] Zum Sprachgebrauch vgl. [Berg, 1968, S. 5].

1.5 Konstruktion der SGL-Funktionen und Nachweis der Basiseigenschaft

Bevor wir in Definition 1.5.1 die SGL-Basisfunktionen formell einführen, vollziehen wir auf Grundlage von Ritchie [2011, Abschn. 2.1.13] deren Konstruktion aus den Kugelflächenfunktionen und den verallgemeinerten Laguerre-Polynomen nach. Zu diesem Zweck schreiben wir das Skalarprodukt (E.2) des in (E.1) definierten Hilbert-Raums H mithilfe der in Abschnitt 1.1 eingeführten Kugelkoordinaten als

$$\langle f, g\rangle_H = \int_0^\infty \int_0^\pi \int_0^{2\pi} f(r,\vartheta,\varphi)\overline{g(r,\vartheta,\varphi)}\,\mathrm{d}\varphi\,\sin\vartheta\,\mathrm{d}\vartheta\,r^2\,\mathrm{e}^{-r^2}\,\mathrm{d}r. \tag{1.18}$$

Das Integrationsgebiet der zwei inneren Integrale identifizieren wir mit der Einheitssphäre $\mathbb{S}^2$ (vgl. Gl. 1.1). Für die Konstruktion von Orthogonalpolynomen in H bietet sich eine Separation des Radius r von den Winkeln ϑ und φ an: Wir machen den Produktansatz

$$P_j(r,\vartheta,\varphi) := R_j(r)\,S_j(\vartheta,\varphi)$$

für eine jede SGL-Basisfunktion P_j. Der radiale Teil R_j und der sphärische Teil S_j von P_j sollen dabei selbst wieder Polynome auf der positiven Halbachse $[0,\infty)$ bzw. der Sphäre $\mathbb{S}^2$ sein. Darüber hinaus ist es wünschenswert, dass die R_j und S_j voneinander unabhängige Orthogonalitätsrelationen erfüllen: Wir fordern

$$\int_0^\infty R_j(r)\overline{R_k(r)}\,r^2\,\mathrm{e}^{-r^2}\,\mathrm{d}r = \delta_{jk}, \tag{1.19}$$

$$\int_0^\pi \int_0^{2\pi} S_j(\vartheta,\varphi)\overline{S_k(\vartheta,\varphi)}\,\mathrm{d}\varphi\,\sin\vartheta\,\mathrm{d}\vartheta = \delta_{jk}. \tag{1.20}$$

Entsprechend (1.18) erhalten wir auf diese Weise ein Orthonormalsystem in H.

Ein Vergleich von (1.20) mit (1.1) macht sofort klar, dass die Kugelflächenfunktionen eine gute Wahl für den sphärischen Part S_j sind. Wir setzen also formal $S_j := Y_{lm}$. Damit ist (1.20) erfüllt. In einem nächsten Schritt wird die Kugelflächenfunktion Y_{lm} radial erweitert. Die Lösung der Schrödinger-Gleichung[10] für das Wasserstoff-Atom motiviert hier den Ansatz [Biedenharn und Louck, 1981, Abschn. 7.4]

$$R_j(r) = R_j^{(\alpha)}(r) := N_j^{(\alpha)}\,r^\alpha\,L_j^{(\alpha+1/2)}(r^2).$$

Indem wir $\alpha := l$ setzen, erhalten wir mit der Substitution $\xi = r^2$ in (1.15) die Orthogonalitätsrelation

$$\int_0^\infty R_j^{(l)}(r)\,R_k^{(l)}(r)\,r^2\,\mathrm{e}^{-r^2}\,\mathrm{d}r = \frac{\Gamma(j+l+3/2)}{2j!}\left\{N_j^{(l)}\right\}^2\delta_{jk}.$$

Wählen wir als Normierungskonstante also

$$N_j^{(l)} := \sqrt{\frac{2k!}{\Gamma(k+l+3/2)}},$$

so ist (1.19) erfüllt. Man beachte, dass die Polynome $R_j^{(l)}$ reellwertig sind.

[10] Erwin Rudolf Josef Alexander Schrödinger (⋆1887 in Wien, Österreich, †1961, ebd.)

An dieser Stelle ist es wichtig zu bemerken, dass man nicht erwarten kann, ein Polynom auf $\mathbb{R}^3$ zu erhalten, indem man ein Polynom auf $\mathbb{S}^2$ mit einem Polynom in r erweitert. Tatsächlich sind die Funktionen $R_j^{(l)}(r)Y_{lm}(\vartheta,\varphi)$ aber Polynome vom Grad $2j+l$ auf $\mathbb{R}^3$, wie man sich leicht überlegt (vgl. Lemma 1.1.12 und beachte, dass $L_j^{(l+1/2)}(r^2)$ stets ein Polynom vom Grad $2j$ auf $\mathbb{R}^3$ ist). Durch die Wahl $j := n-l-1$, $n > l$, die durch Satz 1.5.2 nachträglich motiviert wird, erhalten die SGL-Basisfunktionen nun ihre endgültige Gestalt:

Definition 1.5.1 (SGL-Basisfunktionen). Die *sphärische Gauß-Laguerresche Basisfunktion* der Ordnungen $n \in \mathbb{N}$, $l \in \{0,\dots,n-1\}$ und $m \in \{-l,\dots,l\}$ wird definiert als

$$H_{nlm} : \mathbb{R}^3 \to \mathbb{C}, \qquad H_{nlm}(r,\vartheta,\varphi) \;:=\; N_{nl}\, R_{nl}(r)\, Y_{lm}(\vartheta,\varphi)$$

mit der Normierungskonstanten

$$N_{nl} \;:=\; \sqrt{\frac{2(n-l-1)!}{\Gamma(n+1/2)}}$$

und dem Radialteil

$$R_{nl}(r) \;:=\; L_{n-l-1}^{(l+1/2)}(r^2)\, r^l.$$

In Satz E.1 haben wir gesehen, dass der Raum $\Pi(\mathbb{R}^3)$ der Polynome auf $\mathbb{R}^3$ dicht im Raum H ist. Die SGL-Basisfunktionen bilden ein polynomielles Orthonormalsystem in H. Der nächste Satz zeigt uns, dass es sich bei den SGL-Basisfunktionen sogar um eine *Orthonormalbasis* von H handelt, was die Namensgebung im Nachhinein plausibel macht.

Satz 1.5.2 (Basiseigenschaft). *Die SGL-Basisfunktionen bilden ein vollständiges Orthonormalsystem in H.*

Beweis. Die Orthonormalität der SGL-Basisfunktionen,

$$\langle H_{nlm}, H_{n'l'm'}\rangle_H \;=\; \delta_{nn'}\,\delta_{ll'}\,\delta_{mm'},$$

ergibt sich wie oben gezeigt aus deren Konstruktion. Wir weisen jetzt nach, dass die SGL-Basisfunktionen den Raum $\Pi(\mathbb{R}^3)$ der Polynome auf $\mathbb{R}^3$ voll ausschöpfen. Dazu definieren wir als Hilfsmittel die speziellen Untervektorräume

$$V_{nl} \;:=\; \operatorname{span}\{H_{nlm} : |m| \le l\}, \qquad l < n \in \mathbb{N},$$

von H. Induktiv zeigt man mit Lemma 1.1.12

$$r^{2(n-l-1)}\operatorname{Harm}_l(\mathbb{R}^3) \;\subset\; V_{l+1,l} \oplus \cdots \oplus V_{nl}.$$

Es seien nun $i,j \in \mathbb{N}_0$ mit $j \le i/2$ beliebig. Indem wir $l := i-2j$ und $n := i-j+1 > i-2j = l$ wählen, erhalten wir speziell

$$r^{2j}\operatorname{Harm}_{i-2j}(\mathbb{R}^3) \;\subset\; V_{i-2j+1,i-2j} \oplus \cdots \oplus V_{i-j+1,i-2j}.$$

Bilden wir nun für beliebiges $n \in \mathbb{N}_0$ die direkte innere Summe über $i \leq n$, $j \leq i/2$, so erhalten wir mit Folgerung 1.1.11

$$\Pi_n(\mathbb{R}^3) \subseteq \bigoplus_{i=0}^{n} \bigoplus_{j=0}^{\lfloor i/2 \rfloor} \bigoplus_{k=0}^{j} V_{i-2j+k+1,i-2j} \tag{1.21}$$

und damit die Aussage des Satzes. □

Bemerkung 1.5.3. In der Tat gilt in (1.21) die Gleichheit, da die Polynome auf der rechten Seite stets vom Grad kleiner oder gleich n sind.

Dieser zentrale Satz sagt aus, dass wir eine jede Funktion in H bezüglich $\|\cdot\|_H$ beliebig genau durch Linearkombinationen der SGL-Basisfunktionen approximieren können. Insbesondere konvergieren für $f \in H$ die SGL-Fourier-Partialsummen

$$\sum\nolimits_{|m|\leq l<n\leq B} \langle f, H_{nlm}\rangle_H \, H_{nlm}, \qquad B \in \mathbb{N},$$

bezüglich $\|\cdot\|_H$ gegen f, wenn $B \to \infty$. Äquivalent dazu gilt auch hier die entsprechende Parsevalsche Gleichung

$$\|f\|_H^2 \;=\; \sum\nolimits_{|m|\leq l<n\in\mathbb{N}} |\langle f, H_{nlm}\rangle_H|^2, \qquad f \in H.$$

1.6 Spektralverhalten unter Rotationen und Translationen

Zum Abschluss dieses Kapitels untersuchen wir das Spektralverhalten der SGL-Basisfunktionen unter Rotationen und Translationen. Das Rotationsverhalten ist einfacher zu beschreiben als das Translationsverhalten: Da eine Rotation $R \in \mathrm{SO}(3)$ offensichtlich keinen Einfluss auf den Radius r hat, erhalten wir unter Berücksichtigung der Definition 1.5.1 der SGL-Basisfunktionen und der Definition 1.1.16 der Wigner-D-Funktionen aus (1.11) sofort den folgenden ersten

Satz 1.6.1 (Spektralverhalten unter Rotationen). *Es sei $R \in \mathrm{SO}(3)$. Dann gilt mit dem in (1.10) definierten Rotationsoperator R die Formel*

$$\langle RH_{nlm}, H_{n'l'm'}\rangle_H \;=\; \delta_{nn'}\,\delta_{ll'}\,D^{(l)}_{mm'}(R).$$

Hinsichtlich des Translationsverhaltens halten wir zunächst fest, dass sich jede Translation im $\mathbb{R}^3$ zerlegen lässt in eine Rotation um den Ursprung, eine anschließende Translation entlang der ξ_2-Achse (vgl. Abb. 1.1) und eine abschließende Rückrotation. Dabei wird die Translationsachse zunächst so rotiert, dass sie mit der ξ_2-Achse übereinstimmt. Dann wird die Translation entlang der ξ_2-Achse vorgenommen. Abschließend wird mit der inversen Rotation wieder zurück rotiert. Wir können uns daher bei der folgenden Untersuchung auf Translationen entlang der ξ_2-Achse beschränken. Dies hat den Vorteil, dass solche Translationen keinen Einfluss auf den Azimutalwinkel φ haben (vgl. Abschn. 1.3 u. Abb. 1.1).

Satz 1.6.2 (Spektralverhalten unter Translationen). *Es gilt mit dem in (1.13) definierten Translationsoperator T_2 für $\nu > 0$ die Formel*

$$\begin{aligned}\langle T_2(\nu)H_{nlm}, H_{n'l'm'}\rangle_H \;=\; \delta_{mm'}(-1)^{n-l-1}\frac{\sqrt{\pi}}{4}\frac{N_{nl}N_{n'l'}}{(n'-l'-1)!}\sum_{k=|l-l'|}^{l+l'}(-1)^k A_k^{(ll'm)}\nu^k \\ \times\sum_{j=0}^{n-l-1}\frac{(-1)^j}{j!}\binom{n-1/2}{n-l-1-j}\frac{\Gamma(\mu+j+1/2)}{\Gamma(k+3/2)}C_j^{(nn'll'k)}(\nu)\end{aligned}$$

mit den in Lemma 1.3.10 angegebenen Koeffizienten $A_k^{(ll'm)}$, der Hilfsvariablen $\mu := n' + (l - l' + k)/2$, sowie den Hilfsfunktionen $C_j^{(nn'll'k)}$, welche gegeben sind durch

$$C_j^{(nn'll'k)}(\nu) := \sum_{p=0}^{n-\mu}\frac{{}_2F_1(j,\mu-n+p;\mu-n-j+1;-1)(-\mu+k-j+1)_p}{p!(k+3/2)_p}\binom{-j-p}{n-\mu-p}\nu^{2p}$$

wenn $\mu \in \mathbb{N}$, bzw. $C_j^{(nn'll'k)}(\nu) := 0$ sonst.

Man beachte, dass im Fall $\mu = n$ speziell $C_j^{(nn'll'k)} \equiv 1$ gilt. Für den Beweis dieses Satzes benötigen wir zwei technische Lemmata. Das erste stammt aus der Theorie der endlichen Differenzengleichungen.

Lemma 1.6.3. *Es sei p ein Polynom vom Grad echt kleiner $n \in \mathbb{N}$. Dann gilt*

$$\sum_{k=0}^{n}(-1)^{n-k}\binom{n}{k}p(k) \;=\; 0.$$

Beweis. Für ein beliebiges Polynom vom Grad höchstens $n \in \mathbb{N}_0$ stimmt der obige Ausdruck als Vorwärtsdifferenz n-ter Ordnung mit der n-ten Ableitung des Polynoms überein (vgl. [Levy und Lessman, 1992, Abschn. 1.4]). □

Lemma 1.6.4. *Es sei $n \in \mathbb{N}_0$. Dann gilt*

$$\Gamma(n+1/2) \;=\; \sqrt{\pi}(n+1)_n 4^{-n}.$$

Beweis. Dies folgt aus der Funktionalgleichung $\Gamma(z+1) = z\Gamma(z)$, $z \in \mathbb{C}\setminus\{0,-1,-2,\dots\}$, mit $\Gamma(1/2) = \sqrt{\pi}$ [Andrews et al., 1999, Gln. 1.1.6 u. 1.1.22]. □

Beweis von Satz 1.6.2. Für $x = [r,\vartheta,\varphi]$ schreiben wir im Folgenden stets $x - \nu e_2 = [r',\vartheta',\varphi]$, wobei e_2 der kanonische Einheitsvektor entlang der ξ_2-Achse sei. Weiter setzen wir abkürzend $T_{nn'll'}^{(mm')}(\nu) := \langle T_2(\nu)H_{nlm}, H_{n'l'm'}\rangle_H$. Mit dem bekannten Satz von der majorisierten Konvergenz erhalten wir zunächst

$$T_{nn'll'}^{(mm')}(\nu) = \lim_{\gamma\to 0+}\int_0^\infty\int_0^\pi\int_0^{2\pi}R_{nl}(r')Y_{lm}(\vartheta',\varphi)\mathrm{e}^{-\gamma(r')^2}R_{n'l'}(r)\overline{Y_{l'm'}(\vartheta,\varphi)}\mathrm{e}^{-r^2}\mathrm{d}\varphi\,\sin\vartheta\,\mathrm{d}\vartheta\,r^2\mathrm{d}r.$$

Mit Lemma 1.3.8 erhalten wir für $r' > 0$ hierin

$$R_{nl}(r')Y_{lm}(\vartheta',\varphi)\,\mathrm{e}^{-\gamma(r')^2} \;=\; \sqrt{\frac{2}{\pi}}\int_0^\infty\tilde{R}_{nl}(\gamma;\beta)j_l(\beta r')Y_{lm}(\vartheta',\varphi)\beta^2\mathrm{d}\beta. \tag{1.22}$$

Der erste Teil von Lemma 1.3.10 liefert mit Bemerkung 1.3.11 auf der rechten Seite

$$j_l(\beta r')Y_{lm}(\vartheta',\varphi) \;=\; \sum_{l''=0}^{\infty}\sum_{k=|l-l''|}^{l+l''}(-1)^k A_k^{(ll''|m|)} j_k(\beta\nu) j_{l''}(\beta r) Y_{l''m}(\vartheta,\varphi). \tag{1.23}$$

Darüber hinaus ergibt sich auf der rechten Seite von (1.22) aus [Erdélyi et al., 1954, Gl. 8.9.5] für $\gamma \in (0,1)$

$$\tilde{R}_{nl}(\gamma;\beta) \;=\; N_{nl}\,\frac{(\gamma-1)^{n-l-1}}{\gamma^{n+1/2}}\,\beta^l L_{n-l-1}^{(l+1/2)}\left(\frac{\beta^2}{4\gamma(1-\gamma)}\right)\mathrm{e}^{-\beta^2/4\gamma}\,2^{-l-3/2}. \tag{1.24}$$

Für $\gamma = 1$ lautet die Formel [Erdélyi et al., 1954, Gl. 8.9.2]

$$\tilde{R}_{nl}(1;\beta) \;=\; \frac{N_{nl}}{(n-l-1)!}\,\beta^{2n-l-2}\,\mathrm{e}^{-\beta^2/4}\,2^{-2n+l+1/2}. \tag{1.25}$$

In der obigen Formel für $T_{nn'll'}^{(mm')}(\nu)$ können die Integrale bezüglich r, ϑ und φ vertauscht werden. Zusammen mit Lemma 1.3.6 zeigt uns (1.24), dass nach Einsetzen von (1.22) in die Formel auch eine Vertauschung der Integrale bezüglich r und β erlaubt ist. Mit dem Satz von Tonelli[11] erhalten wir mit (1.22) und (1.23) durch zweimaliges Vertauschen der Integrationsreihenfolge unter Berücksichtigung des zweiten Teils von Lemma 1.3.10 und Definition 1.3.7 somit

$$\begin{aligned} T_{nn'll'}^{(mm')}(\nu) \;&=\; \lim_{\gamma\to 0+}\sqrt{\frac{2}{\pi}}\int_0^\infty\!\!\int_0^\infty \tilde{R}_{nl}(\gamma;\beta)\sum_{l''=0}^{\infty}\sum_{k=|l-l''|}^{l+l''}(-1)^k A_k^{(ll''m)}\delta_{l'l''}\,\delta_{mm'} \\ &\qquad\qquad \times\, j_k(\beta\nu)\,j_{l''}(\beta r)\,\beta^2\,\mathrm{d}\beta\,R_{n'l'}(r)\,r^2\,\mathrm{e}^{-r^2}\,\mathrm{d}r \\ &=\; \delta_{mm'}\sum_{k=|l-l'|}^{l+l'}(-1)^k A_k^{(ll'm)}\lim_{\gamma\to 0+}\underbrace{\int_0^\infty \tilde{R}_{nl}(\gamma;\beta)\,\tilde{R}_{n'l'}(1;\beta)\,j_k(\beta\nu)\,\beta^2\,\mathrm{d}\beta}_{=:\,I_k^{(nn'll')}(\gamma;\nu)}, \end{aligned} \tag{1.26}$$

wobei der Grenzwert $\lim_{\gamma\to 0+} I_k^{(nn'll')}(\gamma;\nu)$ zu untersuchen bleibt. Die beiden Kronecker-Symbole $\delta_{l'l''}$ und $\delta_{mm'}$ ergeben sich hierbei aus der Orthonormalität der Kugelflächenfunktionen. Wegen Bemerkung 1.3.12 können wir uns im Folgenden auf den Fall $l-l'+k$ gerade zurückziehen. Es gilt dann $\mu = n'+(l-l'+k)/2 \in \mathbb{N}$ und

$$\mu \;\geq\; 1+l'+\frac{l-l'+k}{2} \;\geq\; 1+\frac{l+l'+|l-l'|}{2} \;\geq\; 1+l. \tag{1.27}$$

Wir zeigen jetzt

$$\begin{aligned} \lim_{\gamma\to 0+} I_k^{(nn'll')}(\gamma;\nu) \;&=\; (-1)^{n-l-1}\nu^k\,\frac{\sqrt{\pi}}{4}\,\frac{N_{nl}\,N_{n'l'}}{(n'-l'-1)!} \\ &\qquad \times \sum_{j=0}^{n-l-1}\frac{(-1)^j}{j!}\binom{n-1/2}{n-l-1-j}\frac{\Gamma(\mu+j+1/2)}{\Gamma(k+3/2)}\,C_j^{(nn'll'k)}(\nu) \end{aligned} \tag{1.28}$$

[11] Leonida Tonelli (⋆1885 in Gallipoli, Italien, †1946 in Pisa, ebd.)

mit den im Satz angegebenen Hilfsfunktionen $C_j^{(nn'll'k)}$. Indem wir (1.24) und (1.25) in das Integral $I_k^{(nn'll')}(\gamma;\nu)$ einsetzen, erhalten wir

$$\begin{aligned} I_k^{(nn'll')}(\gamma;\nu) \;=\; & (-1)^{n-l-1}\frac{N_{nl}\,N_{n'l'}}{(n'-l'-1)!}\sum_{j=0}^{n-l-1}\frac{(-1)^j}{j!}\binom{n-1/2}{n-l-1-j}2^{-2n'-l+l'-2j-1} \\ & \times\gamma^{-n-j-1/2}\,(1-\gamma)^{n-l-j-1}\underbrace{\int_0^\infty \mathrm{e}^{-\frac{1+1/\gamma}{4}\beta^2}\,j_k(\beta\nu)\,\beta^{2n'+l-l'+2j}\,\mathrm{d}\beta}_{=:\,J_j^{(n'll'k)}(\gamma;\nu)}, \end{aligned}$$

wobei wir zusätzlich die geschlossene Darstellung (1.16) der verallgemeinerten Laguerre-Polynome benutzt haben. Die Integrale $J_j^{(n'll'k)}(\gamma;\nu)$ auf der rechten Seite lösen wir mit [Erdélyi et al., 1954, Gl. 8.6.14] zu

$$\begin{aligned} J_j^{(n'll'k)}(\gamma;\nu) \;=\; & \sqrt{\frac{\pi}{2}}\,\frac{\Gamma(\mu+j+1/2)}{\Gamma(k+3/2)}\left(\frac{\gamma}{1+\gamma}\right)^{\mu+j+1/2}2^{2n'+l-l'+2j-1/2} \\ & \times\nu^k\,\mathrm{e}^{-\frac{\gamma}{1+\gamma}\nu^2}\,{}_1F_1\left(-\mu+k-j+1;\,k+3/2;\,\frac{\gamma}{1+\gamma}\nu^2\right). \end{aligned}$$

Wir untersuchen nun das Verhalten von

$$\begin{aligned} I_k^{(nn'll')}(\gamma;\nu) \;=\; & (-1)^{n-l-1}\nu^k\,\frac{\sqrt{\pi}}{4}\,\frac{N_{nl}\,N_{n'l'}}{(n'-l'-1)!}\,\gamma^{\mu-n}\,\mathrm{e}^{-\frac{\gamma}{1+\gamma}\nu^2} \\ & \times\sum_{j=0}^{n-l-1}\frac{(-1)^j}{j!}\binom{n-1/2}{n-l-1-j}\frac{\Gamma(\mu+j+1/2)}{\Gamma(k+3/2)}(1+\gamma)^{-\mu-j-1/2} \\ & \times(1-\gamma)^{n-l-j-1}\,{}_1F_1\left(-\mu+k-j+1;\,k+3/2;\,\frac{\gamma}{1+\gamma}\nu^2\right) \end{aligned} \tag{1.29}$$

im Grenzwert $\gamma\to 0+$. Da die rechte Seite den Faktor $\gamma^{\mu-n}$ enthält, unterscheiden wir drei verschiedene Fälle. Gilt $k>2(n-n')-l+l'$, so ist $\mu-n>0$ und $\lim_{\gamma\to 0+} I_k^{(nn'll')}(\gamma;\nu)=0$, da aus der Definition 1.2.1 der hypergeometrischen Funktion ${}_1F_1$

$$\lim_{\gamma\to 0+}{}_1F_1\left(-\mu+k-j+1;\,k+3/2;\,\frac{\gamma}{1+\gamma}\nu^2\right) \;=\; 1 \tag{1.30}$$

folgt. Ist andererseits $k=2(n-n')-l+l'$, so gilt $\mu-n=0$ und

$$\begin{aligned} \lim_{\gamma\to 0+} I_k^{(nn'll')}(\gamma;\nu) \;=\; & (-1)^{n-l-1}\nu^k\,\frac{\sqrt{\pi}}{4}\,\frac{N_{nl}\,N_{n'l'}}{(n'-l'-1)!} \\ & \times\sum_{j=0}^{n-l-1}\frac{(-1)^j}{j!}\binom{n-1/2}{n-l-1-j}\frac{\Gamma(\mu+j+1/2)}{\Gamma(k+3/2)}, \end{aligned}$$

was sich ebenfalls aus (1.30) ergibt.

Gilt nun $k < 2(n-n') - l + l'$, so ist $\mu - n < 0$. In diesem Fall können wir (1.30) nicht ohne Weiteres verwenden. Stattdessen nehmen wir die Potenzreihendarstellung

$$(1.31) \qquad (1+\gamma)^{-j-p}(1-\gamma)^{-j} \;=\; \sum_{q=0}^{\infty} \binom{-j-p}{q} {}_2F_1(j,-q;1-q-p-j;-1)\,\gamma^q$$

als Ausgangspunkt, die sich ergibt, indem man die Cauchysche[12] Produktformel auf die zu $(1+\gamma)^{-j-p}$ und $(1-\gamma)^{-j}$ gehörigen Binomialreihen anwendet. Die Potenzreihe auf der rechten Seite von (1.31) besitzt den Konvergenzradius Eins, konvergiert für $|\gamma| < 1$ also absolut. Durch Einsetzen von (1.31) in (1.29) erhalten wir mit der Definition 1.2.1 der hypergeometrischen Funktion ${}_1F_1$ nach Umsortieren der Potenzen von γ

$$(1.32) \qquad I_k^{(nn'll')}(\gamma;\nu) = (-1)^{n-l-1}\nu^k \frac{\sqrt{\pi}}{4} \frac{N_{nl}N_{n'l'}}{(n'-l'-1)!} \mathrm{e}^{-\frac{\gamma}{1+\gamma}\nu^2} \Gamma(k+3/2)^{-1}$$
$$\times (1+\gamma)^{-\mu-1/2}(1-\gamma)^{n-l-1} \sum_{p,q=0}^{\infty} \frac{\nu^{2p} E_{pq}^{(nn'll'k)}}{p!(k+3/2)_p} \gamma^{p+q+\mu-n}$$

mit

$$(1.33) \qquad E_{pq}^{(nn'll'k)} := \sum_{j=0}^{n-l-1} \frac{(-1)^j}{j!} \binom{n-1/2}{n-l-1-j} \Gamma(\mu+j+1/2)$$
$$\times (-\mu+k-j+1)_p \binom{-j-p}{q} {}_2F_1(j,-q;1-q-p-j;-1).$$

Bevor wir nun den Grenzwert $\gamma \to 0+$ bilden, weisen wir $E_{pq}^{(nn'll'k)} = 0$ für $p+q < n-\mu$ nach. Zu diesem Zweck zeigen wir mit Lemma 1.6.4 zunächst

$$\binom{n-1/2}{n-l-1-j} = \frac{\Gamma(n+1/2)}{(n-l-1-j)!\Gamma(l+j+3/2)}$$
$$= \frac{4^{-n+l+j+1}(n+1)_n}{(n-l-1-j)!(l+j+2)_{l+j+1}}.$$

Wieder mit Lemma 1.6.4 ergibt sich, da $\mu \in \mathbb{N}$,

$$\Gamma(\mu+j+1/2) \;=\; \sqrt{\pi}\,(\mu+j+1)_{\mu+j}\,4^{-\mu-j}.$$

Die Abschätzung (1.27) zeigt uns damit

$$(1.34) \qquad \binom{n-1/2}{n-l-1-j}\Gamma(\mu+j+1/2) \;=\; \frac{\sqrt{\pi}\,(n+1)_n}{(n-l-1-j)!} \frac{(\mu+j+1)_{\mu+j}}{(l+j+2)_{l+j+1}} 4^{\mu-n+l+1}$$
$$= \frac{\sqrt{\pi}\,(n+1)_n}{(n-l-1-j)!}(l+j+3/2)_{\mu-l-1}\,4^{2\mu-n}.$$

Aus der Definition 1.2.1 der hypergeometrischen Funktion ${}_2F_1$ erhalten wir weiter

$$(1.35) \quad \binom{-j-p}{q} {}_2F_1(j,-q;1-j-q-p;-1) = \frac{1}{q!}\sum_{s=0}^{q}(-1)^s(-q)_s(1-p-q+s-j)_{q-s}(j)_s.$$

[12] Augustin-Louis Cauchy (⋆1789 in Paris, Frankreich, †1857 in Sceaux, ebd.)

Ein Einsetzen von (1.34) und (1.35) in (1.33), anschließendes Vertauschen der Summationsreihenfolge und Erweitern mit $(n-l-1)!$ ergibt schließlich

$$E_{pq}^{(nn'll'k)} = 4^{2\mu-n}\,\frac{\sqrt{\pi}(n+1)_n}{(n-l-1)!}\,\frac{1}{q!}\sum_{s=0}^{q}(-1)^s\,(-q)_s$$
$$\times\sum_{j=0}^{n-l-1}(-1)^j\binom{n-l-1}{j}(l+j+3/2)_{\mu-l-1}(-\mu+k-j+1)_p(1-p-q+s-j)_{q-s}(j)_s.$$

Es kann $(l+j+3/2)_{\mu-l-1}(-\mu+k-j+1)_p(1-p-q+s-j)_{q-s}(j)_s$ als Polynom vom Grad $p+q+\mu-l-1$ ausgewertet an der Stelle j aufgefasst werden. Daher erhalten wir mit Lemma 1.6.3 nun wie behauptet $E_{pq}^{(nn'll'k)} = 0$ für $p+q < n-\mu$.

Insgesamt haben wir damit gezeigt, dass die Doppelreihe in (1.32) als Potenzreihe in γ mit Konvergenzradius Eins verstanden werden kann, die als solche auf kompakten Teilmengen von $[0,1)$ gleichmäßig konvergiert. Dies erlaubt uns eine gliedweise Grenzwertbildung $\gamma \to 0+$. Da alle Summanden mit $p+q > n-\mu$ bei dieser Grenzwertbildung verschwinden, bleiben nur die Summanden erhalten, bei denen $p+q = n-\mu$ gilt. Dies bedeutet

$$\lim_{\gamma\to 0+} I_k^{(nn'll')}(\gamma;\nu) \;=\; (-1)^{n-l-1}\,\nu^k\,\frac{\sqrt{\pi}}{4}\,\frac{N_{nl}\,N_{n'l'}}{(n'-l'-1)!}\sum_{p=0}^{n-\mu}\frac{\nu^{2p}}{p!\,(k+3/2)_p}\,E_{p,(n-\mu)-p}^{(nn'll'k)}.$$

Ein Einsetzen von (1.33) und Umstellen zeigt auch in diesem Fall (1.28). Die Behauptung des Satzes ergibt sich nun durch Einsetzen von (1.28) in (1.26). □

Bemerkung 1.6.5. Ritchie [2005] leitet mit der gleichen Strategie das Analogon des obigen Satzes für die GTO-Basisfunktionen her (vgl. auch [Ritchie, 2011, Abschn. 3.4.2]). In unserem Fall ist die Herleitung allerdings technisch deutlich aufwendiger, da an entscheidender Stelle nichtkonvergente Integrale entstehen. Aus diesem Grund haben wir in Definition 1.3.7 das konvergenzerzeugende Gauß-Weierstraß-Gewicht $\exp(-\gamma\xi^2)$ eingeführt. Für diese Änderung kommen wir durch die nachträgliche Grenzwertbildung $\gamma \to 0+$ auf.

Bemerkung 1.6.6. Wegen des Kronecker-Symbols $\delta_{mm'}$ in (1.26) und Bemerkung 1.3.12 hängt $T_{nn'll'}^{(mm')}(\nu)$ nur von $|m|$ und nicht von m' oder dem Vorzeichen von m ab. Dies erleichtert den Umgang mit diesen Koeffizienten. Man beachte, dass die im Satz auftretenden hypergeometrischen Funktionen ${}_2F_1$ tatsächlich nur Polynome sind (vgl. Bem. 1.2.2).

2 Basale Algorithmen und schnelle SGL-Fourier-Transformationen für reguläre Daten

In diesem Kapitel wenden wir uns erstmals dem Hauptziel dieser Arbeit zu, der Entwicklung schneller Fourier-Transformationen für die SGL-Basisfunktionen. Diese Algorithmen berechnen die SGL-Fourier-Koeffizienten $\hat{f}_{nlm} := \langle f, H_{nlm} \rangle_H$ sogenannter bandbegrenzter Funktionen $f \in H$ (Def. 2.1.1) auf *diskrete* Art und Weise, d. h. in endlich vielen Schritten, aus abgetasteten Funktionswerten der Funktion f. Die Algorithmen dieses Kapitels sind dadurch charakterisiert, dass f hierbei an den Knoten eines speziellen Gitters abgetastet werden muss. In dieser Hinsicht unterscheiden sich die Algorithmen grundlegend von denen in Kapitel 3. Das Arbeiten mit speziellen Gittern hat auf der anderen Seite den Vorteil, dass die Algorithmen dieses Kapitels *exakt* sind, d. h. fehlerfrei bei angenommener exakter Arithmetik. Wir gehen wie folgt vor: In Abschnitt 2.1 leiten wir eine Quadraturformel für die SGL-Fourier-Koeffizienten bandbegrenzter Funktionen her. Das darauf basierende SGL-Abtasttheorem (Satz 2.1.12) bildet die Grundlage der diskreten SGL-Fourier-Transformation (Def. 2.2.1). In Abschnitt 2.2 entwickeln wir die schnellen SGL-Fourier-Transformationen (Satz 2.2.26, Alg. 2.8) und deren Inverse (Flg. 2.2.27, Alg. 2.9). Die Grundidee ist dabei eine Separation des Radius r von den Winkeln ϑ und φ, sowie die anschließende Verwendung einer schnellen sphärischen Fourier-Transformation (Abschn. 2.2.4) und des Clenshaw-Algorithmus (Abschn. 2.2.5). Abschließend demonstrieren wir in Abschnitt 2.3 die Praxistauglichkeit der schnellen Algorithmen anhand von numerischen Tests. Große Teile dieses Kapitels wurden in [Prestin und Wülker, 2017] bereits veröffentlicht.

2.1 Ein SGL-Abtasttheorem

Wir beginnen direkt mit der Definition der oben erwähnten bandbegrenzten Funktionen, für die wir im Folgenden ein Abtasttheorem herleiten.

Definition 2.1.1 (Bandbegrenzte Funktionen). Es sei $f \in H$. Die Funktion f heißt *bandbegrenzt* mit der *Bandbreite* $B \in \mathbb{N}$, wenn $\hat{f}_{nlm} = 0$ für $n > B$ gilt.

Bandbegrenzte Funktionen haben also, wie eingangs erwähnt, stets nur endlich viele von Null verschiedene SGL-Fourier-Koeffizienten; es handelt sich demnach um endliche Linearkombinationen der SGL-Basisfunktionen. Dies erlaubt es uns, die SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$, $|m| \leq l < n \leq B$, einer Funktion f mit Bandbreite B auf diskrete Weise *exakt* zu berechnen. Die besondere Struktur der SGL-Basisfunktionen bringt es mit sich, dass wir dazu eine Quadraturformel auf der Sphäre mit einer Quadraturformel für das radiale Gewicht $\exp(-r^2)$ auf der positiven Halbachse $[0, \infty)$ kombinieren können. Auf diesem Wege erhalten wir eine Quadraturformel für die SGL-Fourier-Koeffizienten von f.

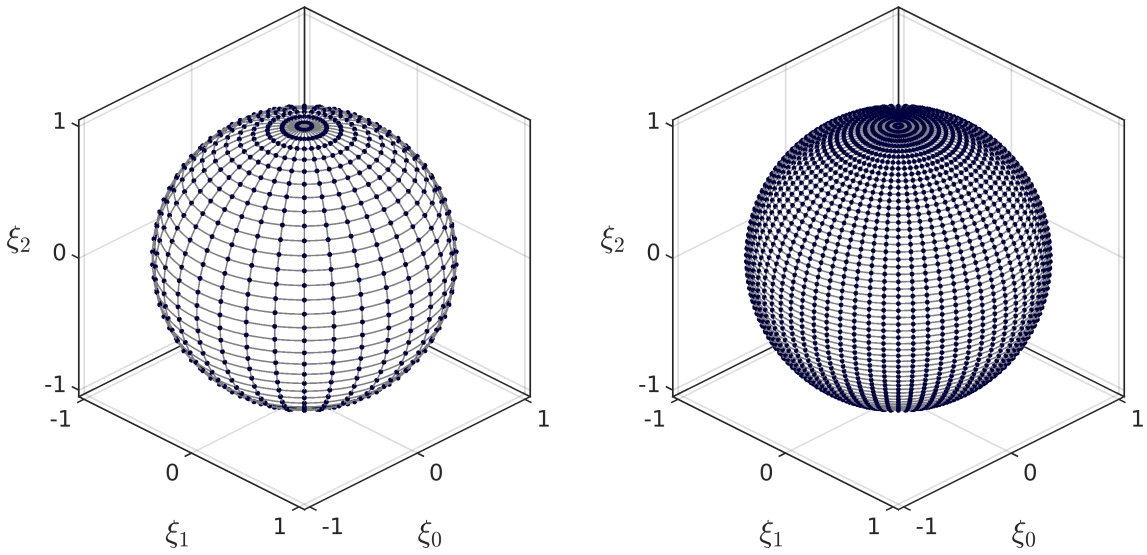

Abbildung 2.1. Abtastwinkel ϑ_j und φ_k des sphärischen Abtasttheorems von Driscoll und Healy (Satz 2.1.2) für **(links)** $n = 16$ und **(rechts)** $n = 32$, dargestellt als Punkte auf $\mathbb{S}^2$ (vgl. Abb. 1.1).

Im Folgenden besprechen wir zuerst eine spezielle sphärische Quadraturformel, die heute in der Praxis fest etabliert ist (Abschn. 2.1.1); sie bildet den Ausgangspunkt für die in dieser Arbeit verwendeten schnellen sphärischen Fourier-Transformationen. Für den radialen Teil eignet sich besonders eine Gauß-Quadraturformel; diese soll anschließend besprochen werden (Abschn. 2.1.2). In Abschnitt 2.1.3 leiten wir dann das SGL-Abtasttheorem her.

2.1.1 Die sphärische Quadraturformel von Driscoll und Healy

In ihrer mittlerweile als klassisch anzusehenden Arbeit wurde von Driscoll und Healy [1994] das folgende Abtasttheorem hergeleitet. Es lehrt uns, wie man die Fourier-Koeffizienten $\langle p, Y_{lm}\rangle_{\mathbb{S}^2}$ eines sphärischen Polynoms p auf diskrete Weise berechnen kann.

Satz 2.1.2 [Driscoll und Healy, 1994, Satz 3]. *Es seien $n \in \mathbb{N}$ und p ein Polynom vom Grad höchstens $n-1$ auf $\mathbb{S}^2$, d. h. $p \in \operatorname{span}\{Y_{lm} : |m| \le l < n\}$. Dann genügen die sphärischen Fourier-Koeffizienten von p der Quadraturformel*

$$\langle p, Y_{lm}\rangle_{\mathbb{S}^2} = \sum_{j,k=0}^{2n-1} \beta_j\, p(\vartheta_j, \varphi_k)\, \overline{Y_{lm}(\vartheta_j, \varphi_k)}, \qquad |m| \le l < n, \tag{2.1}$$

mit den Abtastwinkeln $\vartheta_j = \vartheta_{j,n} := (2j+1)\pi/4n$ und $\varphi_k = \varphi_{k,n} := k\pi/n$, sowie den Quadraturgewichten

$$\beta_j = \beta_{j,n} := \frac{2\pi \sin\vartheta_j}{n^2} \sum_{i=0}^{n-1} \frac{\sin((2i+1)\vartheta_j)}{2i+1}. \tag{2.2}$$

In Abbildung 2.1 sind die Abtastwinkel ϑ_j und φ_k für $n = 16$ und $n = 32$ dargestellt. Abbildung 2.2 zeigt die dazugehörigen Gewichte β_j. Wir wiederholen den mit funktionalanalytischen Mitteln geführten Beweis von Satz 2.1.2 hier nicht. Stattdessen nehmen

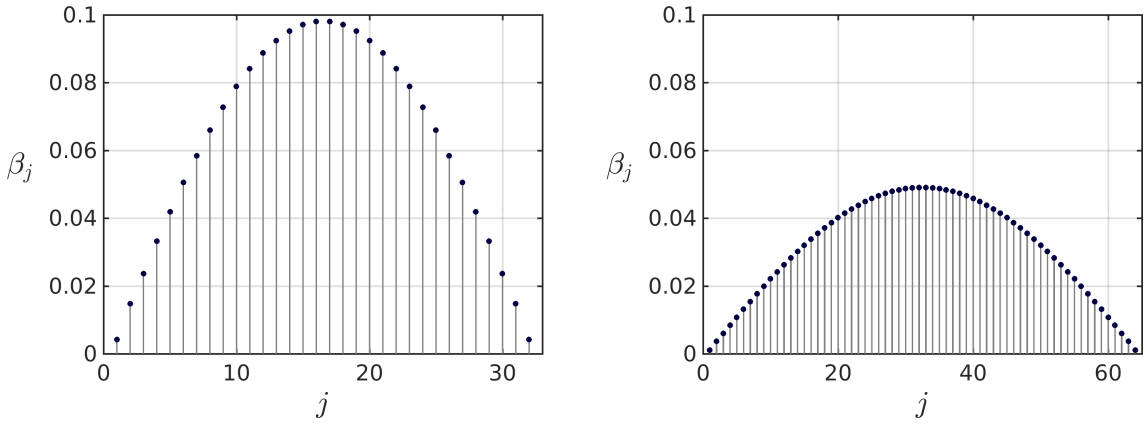

Abbildung 2.2. Quadraturgewichte β_j des sphärischen Abtasttheorems von Driscoll und Healy (Satz 2.1.2) für **(links)** $n = 16$ und **(rechts)** $n = 32$. Man beachte, wie die Gewichte die höhere Dichte von Abtastwinkeln an den Polen von $\mathbb{S}^2$ kompensieren (vgl. Abb. 2.1): Je dichter die Abtastwinkel liegen, desto kleiner werden die dazugehörigen Gewichte.

wir das folgende Resultat mit auf, das die Praxistauglichkeit der Quadraturformel (2.1) nachträglich unterstreicht.

Lemma 2.1.3. *Die durch (2.2) gegebenen Quadraturgewichte sind positiv, d. h. es gilt stets* $\beta_j > 0$.

Beweis. Wir führen hier einen elementaren Beweis dieser bekannten Aussage. Es sei $n \in \mathbb{N}$ gegeben. Zunächst halten wir fest, dass $0 < \vartheta_j < \pi$ und dementsprechend $\sin \vartheta_j > 0$ gilt ($j = 0, \ldots, 2n-1$). Weiter zeigen wir

$$\begin{aligned}
\sum_{k=0}^{n-1} \frac{\sin((2k+1)\vartheta_j)}{2k+1} &= \mathfrak{Im} \sum_{k=0}^{n-1} \frac{1}{2k+1} \mathrm{e}^{\mathrm{i}(2j+1)(2k+1)\pi/4n} \\
&= \mathfrak{Im} \sum_{k=0}^{n-1} \left(\frac{\mathrm{i}}{n} \int_0^{(2j+1)\pi/4} \mathrm{e}^{\mathrm{i}(2k+1)\omega/n} \,\mathrm{d}\omega + \frac{1}{2k+1} \right) \\
&= n^{-1} \,\mathfrak{Re} \int_0^{(2j+1)\pi/4} \mathrm{e}^{\mathrm{i}\omega/n} \sum_{k=0}^{n-1} \mathrm{e}^{\mathrm{i}2k\omega/n} \,\mathrm{d}\omega \\
&= n^{-1} \,\mathfrak{Re} \int_0^{(2j+1)\pi/4} \frac{\mathrm{e}^{2\mathrm{i}\omega} - 1}{\mathrm{e}^{\mathrm{i}\omega/n} - \mathrm{e}^{-\mathrm{i}\omega/n}} \,\mathrm{d}\omega \\
&= (2n)^{-1} \int_0^{(2j+1)\pi/4} \frac{\sin(2\omega)}{\sin(\omega/n)} \,\mathrm{d}\omega.
\end{aligned}$$

Indem wir auf der rechten Seite $\zeta/2$ für ω substituieren, erhalten wir die Integraldarstellung

$$\sum_{k=0}^{n-1} \frac{\sin((2k+1)\vartheta_j)}{2k+1} = (4n)^{-1} \int_0^{(j+1/2)\pi} \frac{\sin \zeta}{\sin(\zeta/2n)} \,\mathrm{d}\zeta. \tag{2.3}$$

Um die Positivität der rechten Seite von (2.3) zu zeigen, unterscheiden wir vier verschiedene Fälle: $j < n$ oder $j \geq n$, j gerade oder ungerade. Der Grund für die erste

Unterscheidung ist, dass der Nenner $\sin(\zeta/2n)$ auf dem Intervall $[0, \pi n)$ streng monoton wachsend und auf dem Intervall $(\pi n, 2\pi n]$ streng monoton fallend ist. Darüber hinaus ist $\sin(\zeta/2n)$ auf dem ganzen Intervall $[0, (j+1/2)\pi] \subset [0, 2\pi n]$ nichtnegativ, was die Untersuchung vereinfacht.

Es sei nun $j < n$. Wir setzen $\kappa_j := 0$, wenn j gerade ist, und $\kappa_j := 1$, wenn j ungerade ist. Zwei einfache Abschätzungen zeigen

$$\begin{aligned}\int_0^{(j+1/2)\pi} \frac{\sin\zeta}{\sin(\zeta/2n)}\,\mathrm{d}\zeta &> \int_0^{(j+\kappa_j)\pi} \frac{\sin\zeta}{\sin(\zeta/2n)}\,\mathrm{d}\zeta \\ &> \sum_{k=0}^{(j+\kappa_j)/2-1} \frac{1}{\sin((2k+1)\pi/2n)} \left\{ \int_{2k\pi}^{(2k+1)\pi} + \int_{(2k+1)\pi}^{2(k+1)\pi} \right\} \sin\zeta\,\mathrm{d}\zeta \;=\; 0.\end{aligned}$$

Ist andererseits $j \geq n$, so erhalten wir mit der Substitution $\zeta = 2\pi n - \xi$ unter Berücksichtigung der Identität $\sin((2\pi n - \xi)/2n) = \sin(\xi/2n)$

$$\int_0^{(j+1/2)\pi} \frac{\sin\zeta}{\sin(\zeta/2n)}\,\mathrm{d}\zeta = \left\{ \underbrace{\int_0^{2\pi n}}_{=0} - \int_{(2n-j-1/2)\pi}^{2\pi n} \right\} \frac{\sin\xi}{\sin(\xi/2n)}\,\mathrm{d}\xi = \int_0^{(2n-j-1/2)\pi} \frac{\sin\xi}{\sin(\xi/2n)}\,\mathrm{d}\xi.$$

Wir können dann wie oben argumentieren. □

Bemerkung 2.1.4. In der Vergangenheit sind viele alternative sphärische Abtasttheoreme vorgestellt worden, die zum Teil mit einer geringeren Anzahl an Abtastwinkeln auskommen (s. z. B. [McEwen und Wiaux, 2011]). In der Regel sind solche Abtasttheoreme direkt mit einer Klasse von schnellen sphärischen Fourier-Transformationen verbunden. Das Verhältnis der Anzahl an Abtastwinkeln zur Anzahl an sphärischen Fourier-Koeffizienten bei Driscoll und Healy lässt sich jedoch nicht substantiell verbessern: beide wachsen wie n^2.

2.1.2 Die Gauß-Quadraturformel für den Typ $\int_0^\infty f(\xi)\exp(-\xi^2)\,\mathrm{d}\xi$

Wir betrachten im Folgenden das klassische Hermite-Gewicht $\exp(-\xi^2)$ auf der positiven Halbachse $[0, \infty)$, anstatt wie üblich auf ganz $\mathbb{R}$. Aus der Theorie der univariaten Orthogonalpolynome ist bekannt, dass es genau eine Familie von Orthogonalpolynomen

$$p_n(\xi) \;=\; \gamma_n \xi^n + \gamma_{n-1}\xi^{n-1} + \cdots + \gamma_0, \qquad \xi \in [0,\infty),\ \gamma_n > 0,\ n \in \mathbb{N}_0, \tag{2.4}$$

gibt, die der Orthogonalitätsrelation

$$\int_0^\infty p_m(\xi)\,p_n(\xi)\,\mathrm{e}^{-\xi^2}\,\mathrm{d}\xi \;=\; \delta_{mn}, \qquad m, n \in \mathbb{N}_0, \tag{2.5}$$

genügen (s. [Mhaskar, 1996, Satz 1.1.3]). Für $n \in \mathbb{N}_0$ ist das Orthogonalpolynom p_n also vom Grad n. Als weitere wichtige Eigenschaft zeigt man, dass das Polynom p_n genau n einfache Nullstellen im offenen Intervall $(0, \infty)$ besitzt (s. z. B. [Mhaskar, 1996, Satz 1.2.1]). Insbesondere halten wir fest:

Bemerkung 2.1.5. Man beachte, dass die Polynome p_n sich grundlegend von den klassischen Hermite-Polynomen unterscheiden. Dies erkennt man bereits an der unterschiedlichen Anzahl an Nullstellen des Polynoms p_n und des n-ten Hermite-Polynoms in $[0, \infty)$.

Um eine Quadraturformel für die SGL-Fourier-Koeffizienten bandbegrenzter Funktionen herzuleiten, benötigen wir in dieser Arbeit eine Darstellung von Integralen der Form $\int_0^\infty f(\xi)\exp(-\xi^2)\mathrm{d}\xi$ durch endliche Summen. Die Funktion f sei dabei ein geeigneter Integrand. Dies führt zu der klassischen Frage, welche Stützstellen ξ_k in $[0,\infty)$ und zugehörige Gewichte ν_k für festes $n \in \mathbb{N}$ zu wählen sind, sodass die Gleichung

$$\int_0^\infty p(\xi)\,\mathrm{e}^{-\xi^2}\mathrm{d}\xi \;=\; \sum_{k=0}^{n-1} \nu_k\, p(\xi_k) \tag{2.6}$$

für alle Polynome p vom Grad kleiner oder gleich einem möglichst großen $d \in \mathbb{N}$ Gültigkeit besitzt. Die Zahl d nennt man den *Genauigkeitsgrad* der Quadraturformel (2.6).

Da die rechte Seite von (2.6) genau $2n$ Freiheitsgrade hat, ist zu erwarten, dass diese Gleichung durch geschickte Wahl der Stützstellen ξ_k und Gewichte ν_k für alle Polynome vom Grad kleiner oder gleich $2n-1$ erfüllt werden kann. Besonders vorteilhaft wären dabei durchweg positive Gewichte ν_k. Es zeigt sich nun, dass diese gewünschten Eigenschaften mittels einer *Gauß-Quadratur* auf elegante Weise zu erreichen sind. Dazu definieren wir den *Christoffel-Darboux-Kern*[1]

$$K_n(\psi,\omega) \;\coloneqq\; \sum_{k=0}^{n-1} p_k(\psi)\,p_k(\omega), \qquad \psi,\omega \in [0,\infty), \tag{2.7}$$

sowie die *Christoffel-Funktion* $\lambda_n(\xi) \coloneqq K_n(\xi,\xi)^{-1}$, $\xi \in [0,\infty)$, die überall wohldefiniert und positiv ist. Letzteres folgt aus der Tatsache, dass $p_0 > 0$. Es ergibt sich nun als im Grunde klassisches Resultat (vgl. [Mhaskar, 1996, Satz 1.2.2]):

Satz 2.1.6. *Es sei $n \in \mathbb{N}$. Wählt man als Stützstellen $\xi_k = \xi_{k,n}$ die Nullstellen des n-ten Orthogonalpolynoms p_n und als zugehörige Gewichte $\nu_k = \nu_{k,n} \coloneqq \lambda_n(\xi_k)$, so gilt die Quadraturformel (2.6) für alle Polynome p vom Grad höchstens $2n-1$.*

Folgerung 2.1.7. *Unter den Voraussetzungen von Satz 2.1.6 gilt stets $\xi_k, \nu_k > 0$.*

Beweis. Wie oben erwähnt liegen die Nullstellen ξ_k des n-ten Orthogonalpolynoms p_n sämtlich in $(0,\infty)$, und die Christoffel-Funktion λ_n ist strikt positiv. □

Bemerkung 2.1.8. Es gibt bekanntlich keine Quadraturformel der Form (2.6), die für alle Polynome vom Grad $2n$ Gültigkeit besitzt. Die durch Satz 2.1.6 gegebene Gaußsche Quadraturformel ist in dieser Hinsicht optimal.

In Abbildung 2.3 sind die Abtastpunkte ξ_k und Gewichte ν_k für $n = 32$ und $n = 64$ dargestellt. Um die durch Satz 2.1.6 gegebene Quadraturformel in der Praxis nutzen zu können, müssen wir die Abtastpunkte ξ_k und Gewichte ν_k für verschiedene n berechnen können. Dazu benötigen wir mehr Wissen über die Orthogonalpolynome p_n, das jetzt

[1] Elwin Bruno Christoffel (⋆1829 in Monschau, †1900 in Straßburg, heute Strasbourg in Frankreich), Jean Gaston Darboux (⋆1842 in Nîmes, Frankreich, †1917 in Paris, ebd.)

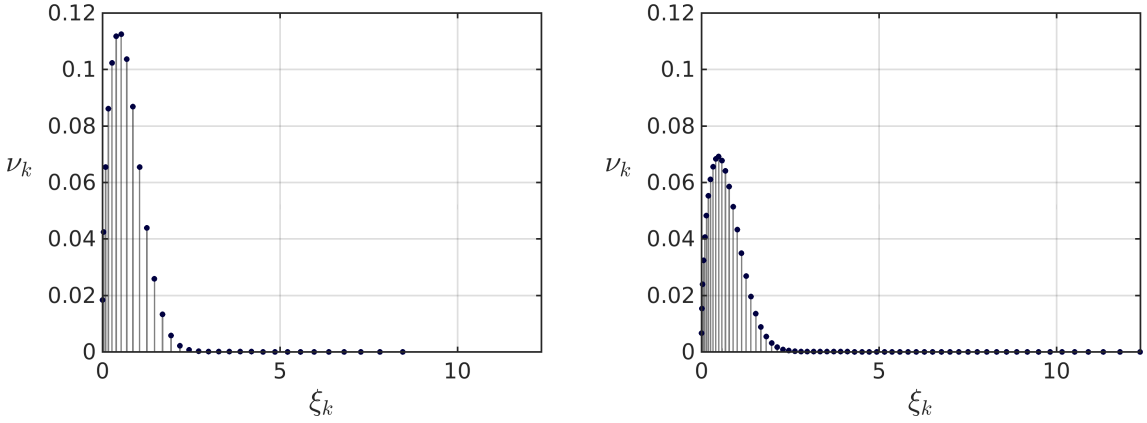

Abbildung 2.3. Abtastpunkte ξ_k und Gewichte ν_k der Gauß-Quadratur (Satz 2.1.6) für das Hermite-Gewicht $\exp(-\xi^2)$ auf der positiven Halbachse $[0,\infty)$ mit **(links)** $n = 32$ und **(rechts)** $n = 64$.

erarbeitet werden soll. Wir bemerken zunächst, dass die Polynome p_n eine spezielle Drei-Term-Rekursion erfüllen. Um diese anzugeben, normieren wir die Orthogonalpolynome *monisch*, d. h. wir definieren $P_k := p_k/\gamma_k$ für $k \in \mathbb{N}_0$, und setzen

$$\tau_k := \int_0^\infty P_k^2(\xi)\,\mathrm{e}^{-\xi^2}\mathrm{d}\xi, \qquad k \in \mathbb{N}_0, \tag{2.8}$$

$$\alpha_k := \tau_k^{-1}\int_0^\infty P_k^2(\xi)\,\xi\,\mathrm{e}^{-\xi^2}\mathrm{d}\xi, \qquad k \in \mathbb{N}_0, \tag{2.9}$$

$$\beta_k := \tau_k/\tau_{k-1}, \qquad k \in \mathbb{N}. \tag{2.10}$$

Setzt man zusätzlich $P_{-1} := 0$ und $\beta_0 := 0$, so lautet die Drei-Term-Rekursion [Steen et al., 1969, Gl. 2.5]

$$P_{k+1}(\xi) = (\xi - \alpha_k)P_k(\xi) - \beta_k P_{k-1}(\xi), \qquad \xi \in [0,\infty),\ k \in \mathbb{N}_0. \tag{2.11}$$

Ein Vorteil der Rekursionsgleichung (2.11) besteht darin, dass man mit ihrer Hilfe aus den ersten beiden Orthogonalpolynomen P_0 und P_1 sukzessiv die weiteren Orthogonalpolynome P_k bestimmen kann. Außerdem erlauben es die darin auftretenden Koeffizienten α_k und β_k, die Stützstellen ξ_k und Gewichte ν_k der durch Satz 2.1.6 gegebenen Quadraturformel zu berechnen: Für fest gewähltes $n \in \mathbb{N}$ sind die Stützstellen ξ_k nämlich genau die Eigenwerte der tridiagonalen *Jacobi-Matrix*[2]

$$\begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & & \\ \sqrt{\beta_1} & \ddots & \ddots & \\ & \ddots & \ddots & \sqrt{\beta_{n-1}} \\ & & \sqrt{\beta_{n-1}} & \alpha_{n-1} \end{bmatrix} \tag{2.12}$$

(s. [Gautschi, 1997, Abschn. 2.2.5]); sie können als solche z. B. mit dem QR-Algorithmus effizient berechnet werden (s. [Kress, 1998, Abschn. 7.4]). Die zugehörigen Gewichte ν_k

[2] Carl Gustav Jacob Jacobi (⋆1804 in Potsdam, †1851 in Berlin)

können anschließend mithilfe der Formel

$$\nu_k \;=\; \frac{\tau_{n-1}}{P_{n-1}(\xi_k)} \left(\frac{\mathrm{d}P_n}{\mathrm{d}\xi}(\xi_k) \right)^{-1} \tag{2.13}$$

berechnet werden (vgl. [Steen et al., 1969, Gln. 2.1 u. 2.2]).

Wir bestimmen die ersten beiden Polynome P_0 und P_1 zu

$$P_0 \;\equiv\; 1, \tag{2.14}$$

$$P_1(\xi) \;=\; \xi - \pi^{-1/2}, \qquad \xi \in [0, \infty). \tag{2.15}$$

Im nun beginnenden zweiten Teil dieses Unterabschnitts leiten wir nach Vorlage von Steen et al. [1969] ein effizientes iteratives Schema zur Berechnung der Stützstellen ξ_k und Gewichte ν_k her. Als entscheidendes Hilfsmittel benutzen wir dazu die nun folgende *Christoffel-Darboux-Formel* für den Christoffel-Darboux-Kern (2.7), die auch in der allgemeinen Theorie der Orthogonalpolynome eine wichtige Rolle spielt.

Lemma 2.1.9 (s. [Mhaskar, 1996, Satz 1.1.4]). *Für $n \in \mathbb{N}$ gilt*

$$K_n(\xi, \psi) \;=\; \frac{\gamma_{n-1}}{\gamma_n} \frac{p_n(\xi) p_{n-1}(\psi) - p_{n-1}(\xi) p_n(\psi)}{\xi - \psi}, \qquad \xi, \psi \in [0, \infty),\ \xi \neq \psi.$$

Wir formulieren dazu die folgende, durch Grenzwertbildung $\psi \to \xi$ leicht zu beweisende

Folgerung 2.1.10 (vgl. [Steen et al., 1969, Gl. 2.14]). *Für $n \in \mathbb{N}$ gilt*

$$\frac{\mathrm{d}P_n}{\mathrm{d}\xi}(\xi)\, P_{n-1}(\xi) \;-\; P_n(\xi) \frac{\mathrm{d}P_{n-1}}{\mathrm{d}\xi}(\xi) \;=\; \tau_{n-1} \sum_{j=0}^{n-1} \frac{P_j^2(\xi)}{\tau_j}, \qquad \xi \in [0, \infty).$$

In einem ersten Schritt halten wir fest, dass in der Drei-Term-Rekursion (2.11) speziell

$$P_{k+1}(0) \;=\; -\,\alpha_k P_k(0) \;-\; \beta_k P_{k-1}(0), \qquad k \in \mathbb{N}_0, \tag{2.16}$$

gilt. Außerdem erhalten wir nach Multiplikation von (2.11) mit $P_{k-1}(\xi) \exp(-\xi^2)$ und anschließender Integration unter Beachtung der Orthogonalitätsrelation (2.5) mit (2.10)

$$\tau_k \;=\; \int_0^\infty P_k(\xi)\, P_{k-1}(\xi)\, \xi\, \mathrm{e}^{-\xi^2} \mathrm{d}\xi, \qquad k \in \mathbb{N}.$$

Mit partieller Integration folgt hieraus

$$\tau_k \;=\; \frac{P_k(0)\, P_{k-1}(0)}{2} \;+\; \frac{1}{2} \int_0^\infty \frac{\mathrm{d}P_k}{\mathrm{d}\xi}(\xi)\, P_{k-1}(\xi)\, \mathrm{e}^{-\xi^2} \mathrm{d}\xi.$$

Das Integral auf der rechten Seite lösen wir unter Berücksichtigung der Orthogonalitätsrelation (2.5) mit Folgerung 2.1.10 und (2.8) zu

$$\begin{aligned}
\int_0^\infty \frac{\mathrm{d}P_k}{\mathrm{d}\xi}(\xi)\, P_{k-1}(\xi)\, \mathrm{e}^{-\xi^2} \mathrm{d}\xi \;&=\; \int_0^\infty \frac{\mathrm{d}P_k}{\mathrm{d}\xi}(\xi)\, P_{k-1}(\xi) - P_k(\xi) \frac{\mathrm{d}P_{k-1}}{\mathrm{d}\xi}(\xi)\, \mathrm{e}^{-\xi^2} \mathrm{d}\xi \\
&=\; \tau_{k-1} \sum_{j=0}^{k-1} \tau_j^{-1} \int_0^\infty P_j^2(\xi)\, \mathrm{e}^{-\xi^2} \mathrm{d}\xi \\
&=\; k\, \tau_{k-1}.
\end{aligned}$$

Algorithmus 2.1. Vorberechnungen für die Gauß-Hermite-Quadratur auf $[0, \infty)$

Eingabe: $n \in \mathbb{N}$
Ausgabe: Stützstellen $\xi_k > 0$ und Gewichte $\nu_k > 0$, $k = 0, \ldots, n-1$

for $k = 1$ **to** $n-1$ **do**
  Berechne α_k und β_k mittels (2.18) bzw. (2.10) aus τ_{k-1}, τ_k und $P_k(0)$;
  Für $k < n-1$ berechne $P_{k+1}(0)$ mittels (2.16) aus α_k, β_k, $P_k(0)$ und $P_{k-1}(0)$;
  Für $k < n-1$ berechne τ_{k+1} mittels (2.17) aus $P_{k+1}(0)$, $P_k(0)$ und τ_k;
end
Berechne ξ_k, $k = 0, \ldots, n-1$, als Eigenwerte der Jacobi-Matrix (2.12);
Berechne ν_k, $k = 0, \ldots, n-1$, mittels (2.13).

Daraus folgt die nützliche Rekursionsgleichung

$$\tau_k = \frac{P_k(0)\,P_{k-1}(0)}{2} + \frac{k}{2}\,\tau_{k-1}, \qquad k \in \mathbb{N}. \tag{2.17}$$

Partielle Integration in (2.9) liefert mit der Orthogonalitätsrelation (2.5) noch

$$\alpha_k = \frac{P_k^2(0)}{2}\,\tau_k^{-1}, \qquad k \in \mathbb{N}_0. \tag{2.18}$$

Aus den Gleichungen (2.8), (2.10), (2.14), (2.15), (2.16), (2.17) und (2.18) ergibt sich nun folgendes iteratives Vorgehen: Im ersten Schritt bestimmt man $\tau_0 = \sqrt{\pi}/2$ mittels (2.8) und (2.14), woraus mit (2.18) und (2.14) sofort $\alpha_0 = 1/\sqrt{\pi}$ folgt. Im nächsten Schritt ergibt sich mit (2.17) aus (2.14), (2.15) und dem Wert τ_0 der Wert $\tau_1 = (\sqrt{\pi}/4) - 1/(2\sqrt{\pi})$. Aus den Werten

$$\tau_0,\ \tau_1,\ P_1(0)$$

bestimmt man nun mittels (2.18) und (2.10) die Rekursionskoeffizienten α_1 und β_1. Diese ergeben mit (2.16) den Wert $P_2(0)$. Mit den Werten $P_1(0)$, $P_2(0)$ und τ_1 erhält man mit (2.17) den Wert τ_2. Aus den Werten

$$\tau_1,\ \tau_2,\ P_2(0)$$

bestimmt man mittels (2.18) und (2.10) nun wiederum den Wert der Rekursionskoeffizienten α_2 und β_2. So fortfahrend erhält man die Werte aller Rekursionskoeffizienten $\alpha_0, \ldots, \alpha_{n-1}$; $\beta_1, \ldots, \beta_{n-1}$, und damit die Polynome P_{n-1} und P_n. Wie oben beschrieben können daraus die Stützstellen ξ_k und Gewichte ν_k der durch Satz 2.1.6 gegebenen Gaußschen Quadraturformel berechnet werden. Das komplette Vorgehen ist in Algorithmus 2.1 noch einmal zusammengefasst. Wir erwähnen noch:

Bemerkung 2.1.11. Bei genauerer Betrachtung des oben beschriebenen Verfahrens zeigt sich dessen unmittelbare Übertragbarkeit auf andere Familien von Orthogonalpolynomen. Tatsächlich wird das Vorgehen in allgemeiner Form z. B. schon in [Hildebrand, 1956, Abschn. 8.4] angedeutet.

2.1.3 Herleitung des Abtasttheorems

Indem wir die Sätze 2.1.2 und 2.1.6 miteinander kombinieren, erhalten wir nun das folgende SGL-Abtasttheorem, das für die Algorithmen dieses Kapitels von grundlegender Bedeutung ist.

Satz 2.1.12 [Prestin und Wülker, 2017, Satz 3.4]. *Es sei $f \in H$ eine bandbegrenzte Funktion mit Bandbreite B. Dann genügen die SGL-Fourier-Koeffizienten von f der Quadraturformel*

$$\hat{f}_{nlm} = \sum_{i,j,k=0}^{2B-1} \alpha_i r_i^2 \beta_j f(r_i, \vartheta_j, \varphi_k) \overline{H_{nlm}(r_i, \vartheta_j, \varphi_k)}, \qquad |m| \le l < n \le B, \tag{2.19}$$

wobei $\alpha_i := \nu_{i,2B}$ und $r_i := \xi_{i,2B}$ die Gewichte bzw. Stützstellen der durch Satz 2.1.6 gegebenen Gauß-Quadratur bezeichnen, während $\vartheta_j = \vartheta_{j,B}$, $\varphi_k = \varphi_{k,B}$ und $\beta_j = \beta_{j,B}$ die Abtastwinkel bzw. Gewichte der sphärischen Quadraturformel in Satz 2.1.2 sind.

Beweis. Es ist nur zu beachten, dass bei festem r_i die eingeschränkte Funktion $f(r_i, \cdot, \cdot)$ ein Polynom vom Grad echt kleiner als B auf $\mathbb{S}^2$ ist, während bei festen Winkeln ϑ_j und φ_k die Funktionen $f(\cdot, \vartheta_j, \varphi_k)$ und $H_{nlm}(\cdot, \vartheta_j, \varphi_k)$ Polynome vom Grad höchstens $2B-2$ auf $[0, \infty)$ sind. □

Für die Bandbreiten $B = 16$ und $B = 32$ ergeben sich die Abtastpunkte $[r_i, \vartheta_j, \varphi_k]$ aus den in Abbildung 2.1 gezeigten Abtastwinkeln ϑ_j und φ_k zusammen mit den in Abbildung 2.3 gezeigten Abtastradien r_i. Ein besonderer Vorteil der Quadraturformel (2.19) ist die Tatsache, dass die darin auftretenden Gewichte positiv sind.

Folgerung 2.1.13. *In Satz 2.1.12 gilt stets $\alpha_i r_i^2 \beta_j > 0$.*

Beweis. Dies folgt sofort aus Lemma 2.1.3 und Folgerung 2.1.7. □

2.2 Schnelle SGL-Fourier-Transformation (FSGLFT)

Ausgehend von unserem SGL-Abtasttheorem (Satz 2.1.12) definieren wir nun die diskrete SGL-Fourier-Transformation. Dazu identifizieren wir das Tripel $[n, l, m]$, $|m| \le l < n \le B$, mit $\mu \in \{0, \ldots, B(B+1)(2B+1)/6 - 1\}$ vermöge des eineindeutigen Zusammenhangs

$$\mu = \frac{n(n-1)(2n-1)}{6} + l(l+1) + m.$$

Dies erlaubt es uns, die Indizes n, l und m als Funktionen von μ aufzufassen:

$$\begin{cases} n(\mu) &= \left\lfloor \frac{1}{\sqrt{3}} \cosh\left(\frac{1}{3} \operatorname{arcosh}\left(36\sqrt{3}\mu\right)\right) + \frac{1}{2} \right\rfloor, \\ l(\mu) &= \left\lfloor \sqrt{\mu - \frac{n(\mu)(n(\mu)-1)(2n(\mu)-1)}{6}} \right\rfloor, \\ m(\mu) &= \mu - \frac{n(\mu)(n(\mu)-1)(2n(\mu)-1)}{6} - l(\mu)(l(\mu)+1). \end{cases} \tag{2.20}$$

Darüber hinaus identifizieren wir $[i,j,k] \in \{0,\dots,2B-1\}^3$ mit $\nu \in \{0,\dots,(2B)^3-1\}$ mithilfe des bijektiven Zusammenhangs

$$\nu \;=\; (2B)^2 i + 2Bj + k.$$

Wir können die Indizes i, j und k dann als Funktionen von ν verstehen:

$$\begin{aligned}
i(\nu) \;&=\; \frac{1}{(2B)^2}\left(\nu - \nu \bmod 2B - 2B\left(\frac{\nu - \nu \bmod 2B}{2B} \bmod 2B\right)\right),\\
j(\nu) \;&=\; \frac{\nu - \nu \bmod 2B}{2B} \bmod 2B,\\
k(\nu) \;&=\; \nu - (2B)^2 i(\nu) - 2Bj(\nu) \;=\; \nu \bmod 2B.
\end{aligned}$$

Es sind im Folgenden $[r_i, \vartheta_j, \varphi_k]$ die Abtastpunkte aus Satz 2.1.12 und $\alpha_i r_i^2 \beta_j$ die dazugehörigen Quadraturgewichte.

Definition 2.2.1 (DSGLFT). Es seien $B \in \mathbb{N}$ und

$$\begin{aligned}
W_B \;&:=\; \operatorname{diag}\left[\alpha_{i(\nu)} r_{i(\nu)}^2 \beta_{j(\nu)}\right]_{\nu=0,\dots,(2B)^3-1} \in \mathbb{R}^{(2B)^3\times(2B)^3},\\
H_B \;&:=\; \left[\overline{H_{n(\mu),l(\mu),m(\mu)}\left(r_{i(\nu)}, \vartheta_{j(\nu)}, \varphi_{k(\nu)}\right)}\right]_{\substack{\mu=0,\dots,B(B+1)(2B+1)/6-1\\ \nu=0,\dots,(2B)^3-1}} \in \mathbb{C}^{B(B+1)(2B+1)/6\times(2B)^3}.
\end{aligned}$$

Die Abbildung $H_B W_B : \mathbb{C}^{(2B)^3} \to \mathbb{C}^{B(B+1)(2B+1)/6}$ heißt *diskrete SGL-Fourier-Transformation* (DSGLFT für *discrete SGL Fourier transform*).

Wir können das Prinzip der DSGLFT auf folgende Art und Weise veranschaulichen. Ohne Verwechslungsgefahr definieren wir für eine Funktion $f \in H$ mit Bandbreite B den zugehörigen „Ortsvektor“ $f := [f(r_{i(\nu)}, \vartheta_{j(\nu)}, \varphi_{k(\nu)})]_{\nu=0,\dots,(2B)^3-1}$ und den entsprechenden Fourier-Vektor $\hat{f} := [\hat{f}_{n(\mu),l(\mu),m(\mu)}]_{\mu=0,\dots,B(B+1)(2B+1)/6-1}$. Es gilt damit

$$H_B W_B f \;=\; \hat{f}.$$

Mit der DSGLFT können also die SGL-Fourier-Koeffizienten bandbegrenzter Funktionen auf diskrete Art und Weise berechnet werden. In exakter Arithmetik ist diese Berechnung fehlerfrei. Mit dem offensichtlichen Zusammenhang $H_B^{\mathrm{H}} \hat{f} = f$ erhalten wir das

Lemma 2.2.2. *Es seien $B \in \mathbb{N}$ und $f \in H$ eine bandbegrenzte Funktion mit Bandbreite B. Weiter seien die zu f gehörigen Vektoren f und $\hat{f}$ definiert wie oben. Dann gilt*

$$\begin{aligned}
H_B^{\mathrm{H}} (H_B W_B) f \;&=\; f,\\
(H_B W_B) H_B^{\mathrm{H}} \hat{f} \;&=\; \hat{f}.
\end{aligned}$$

Es gibt verschiedene Möglichkeiten, die DSGLFT algorithmisch umzusetzen. Im einfachsten Fall berechnet man für alle $[n,l,m]$, $|m| \le l < n \le B$, den jeweiligen Fourier-Koeffizienten $\hat{f}_{nlm}$ direkt mittels Auswertung der Dreifachsumme in (2.19). Nimmt man dafür die Drei-Term-Rekursionsformeln (1.3) und (1.17) zu Hilfe, so ergibt sich ein Algorithmus mit einer Rechenkomplexität von $\mathcal{O}(B^7)$ und einer Speicherkomplexität von $\mathcal{O}(1)$;

wir bezeichnen diesen als *naive* DSGLFT. Der Rechenaufwand kann auf $\mathcal{O}(B^6)$ reduziert werden, indem man die Transformationsmatrix H_B vorberechnet; die Speicherkomplexität von $\mathcal{O}(B^6)$ kann hierbei aber schnell zum Problem werden. In diesem Abschnitt wollen wir daher auf geeignetere Weise *schnelle* SGL-Fourier-Transformationen (FSGLFTs für *fast SGL Fourier transforms*) entwickeln. Dazu besprechen wir im Folgenden zunächst eine Reihe von Hilfsresultaten, die wir für die Herleitung unserer FSGLFTs benötigen (Abschnitte 2.2.1–2.2.5). Diese Hilfsmittel werden wir zum großen Teil auch in Kapitel 3 verwenden (vgl. Abb. E.1). In Abschnitt 2.2.6 leiten wir dann die FSGLFTs her. Jeder dieser Algorithmen korrespondiert mit einer speziellen Faktorisierung der Matrix H_B. Das Lemma 2.2.2 zeigt uns dementsprechend, dass wir mit jeder FSGLFT auch über eine schnelle inverse Transformation (iFSGLFT) verfügen. Das Hauptresultat dieses Abschnitts ist der Satz 2.2.26 und seine Folgerung 2.2.27.

2.2.1 Schnelle Fourier-Transformation (FFT)

Die *schnelle Fourier-Transformation* (FFT für *fast Fourier transform*) ist zweifelsohne eine der am besten untersuchten Klassen von schnellen Algorithmen aller Zeiten; sie kommt heute in einer unüberschaubaren Vielzahl von Anwendungen zum Einsatz. Wir wiederholen hier noch einmal die wichtigsten Eigenschaften, um später darauf verweisen zu können.

Definition 2.2.3 (DFT). Für $n \in \mathbb{N}$ betrachten wir die primitive n-te Einheitswurzel $\chi_n := \exp(-2\pi\mathrm{i}/n)$ und definieren die *Fourier-Matrix*

$$F_n := \left[\chi_n^{jk}\right]_{j,k=0,\ldots,n-1} \in \mathbb{C}^{n\times n}.$$

Die Abbildung $F_n : \mathbb{C}^n \to \mathbb{C}^n$ wird als *diskrete Fourier-Transformation* (DFT für *discrete Fourier transform*) bezeichnet.

Um eine DFT durchzuführen, benötigen wir also einen Algorithmus zur Berechnung der speziellen Summen

$$\left[F_n x\right]_j = \sum_{k=0}^{n-1} \xi_k \,\mathrm{e}^{-2\pi\mathrm{i}jk/n}, \qquad j = 0, \ldots, n-1, \tag{2.21}$$

für beliebige Eingabedaten $x = [\xi_0, \ldots, \xi_{n-1}]^{\mathrm{T}}$. Eine direkte Berechnung dieser Summen benötigt offensichtlich $\mathcal{O}(n^2)$ Operationen. Die heute in der Praxis verwendeten FFTs haben eine geringere Komplexität von $\mathcal{O}(n \log n)$. Für eine ausführliche Darstellung solcher Algorithmen siehe z. B. [van Loan, 1992]. Schnelle Algorithmen zur Multiplikation mit der inversen Fourier-Matrix F_n^{-1} werden als inverse FFTs (iFFTs) bezeichnet; sie besitzen ebenfalls die Komplexität $\mathcal{O}(n \log n)$. Die Existenz von F_n^{-1} zeigt uns das folgende

Lemma 2.2.4 (s. [van Loan, 1992, Satz 1.1.2]). *Es gilt stets* $F_n^{-1} = n^{-1} F_n^{\mathrm{H}}$.

Bemerkung 2.2.5. Die FFT lässt sich direkt auf den mehrdimensionalen Fall verallgemeinern, indem man das Prinzip der FFT auf jede Dimension anwendet (vgl. [van Loan, 1992, Abschn. 3.4]). Wir begegnen der mehrdimensionalen FFT in Abschnitt 3.1 bei der Behandlung der nichtäquidistanten FFT (NFFT).

2.2.2 Schnelle Kosinus-Transformation (DCT)

Im Grunde kann man selbst vereinfachend nicht von *der* schnellen Kosinus-Transformation sprechen, denn es sind mehrere verschiedene Klassen von zugrunde liegenden diskreten Transformationen zu unterscheiden (vgl. Bem. 2.2.7). Wir betrachten hier eine spezielle diskrete Transformation und bezeichnen diese als *die* diskrete Kosinus-Transformation (DCT für *discrete cosine transform*). Für festes $n \in \mathbb{N}$ wird die DCT mithilfe der Matrix

$$C_n \coloneqq \Big[\cos(i\omega_j)\Big]_{i,j=0,\ldots,n-1} \in \mathbb{R}^{n\times n}$$

mit $\omega_j \coloneqq (2j+1)\pi/2n$ definiert.

Definition 2.2.6 (DCT). Es seien $n \in \mathbb{N}$ und

$$D_n \coloneqq \operatorname{diag}\Big[1/\sqrt{n}, \sqrt{2/n}, \ldots, \sqrt{2/n}\Big] \in \mathbb{R}^{n\times n}.$$

Die Abbildung $\tilde{C}_n \coloneqq D_n C_n : \mathbb{C}^n \to \mathbb{C}^n$ heißt *diskrete Kosinus-Transformation.*

Bemerkung 2.2.7. In der Literatur wird üblicherweise zwischen vier verschiedenen Varianten DCT-I bis DCT-IV der diskreten Kosinus-Transformation unterschieden (s. z. B. [Plonka und Tasche, 2005, Abschn. 2]). Diese stellen die in der Praxis etablierten von acht theoretisch denkbaren Versionen dar [Strang, 1999]. Die DCT in Definition 2.2.6 ist eng mit der DCT-II verwandt, die oft als *die* diskrete Kosinus-Transformation bezeichnet wird.

Für einen gegebenen Vektor der Länge n benötigt man für eine Multiplikation mit $\tilde{C}_n$ offenbar $\mathcal{O}(n^2)$ Schritte. Im Kontext der Kosinus-Transformationen werden die schnellen Algorithmen zur Durchführung der DCT in der Regel auch mit DCT abgekürzt (anstelle von FCT für *fast cosine transform*). Wir gehen hier nicht näher auf die schnellen Algorithmen ein (s. dazu etwa [Plonka und Tasche, 2005]), sondern weisen auf den engen Zusammenhang zwischen der DCT und der klassischen iFFT der Länge $2n$ hin, der in leicht abgewandelter Form von Ahmed et al. [1974] bemerkt wurde (der Grundgedanke findet sich schon in [Cooley et al., 1970, Abschn. 2.6]); setzt man nämlich

$$Q_n \coloneqq \left[\begin{array}{c|ccc|c|ccc} 2 & & & & 0 & & & \\ \hline & \mathrm{e}^{\pi\mathrm{i}/2n} & & & & & & \mathrm{e}^{-\pi\mathrm{i}/2n} \\ & & \ddots & & & & \iddots & \\ & & & \mathrm{e}^{\pi\mathrm{i}(n-1)/2n} & & \mathrm{e}^{-\pi\mathrm{i}(n-1)/2n} & & \end{array}\right] \in \mathbb{C}^{n\times 2n},$$

so erhält man

$$C_n = 2^{-1} Q_n \cdot F_{2n}^{\mathrm{H}} \cdot \Big[E_n \,\Big|\, 0_n\Big]^{\mathrm{T}} \tag{2.22}$$

mit der Nullmatrix $0_n \in \mathbb{R}^{n\times n}$. Die in der Praxis verwendeten DCTs haben dementsprechend eine Komplexität von $\mathcal{O}(n \log n)$. Es ist jedoch anzumerken, dass der Zusammenhang (2.22) zwischen der DCT und der iFFT der Länge $2n$ keineswegs der einzige ist (s. Makhoul [1980]; vgl. auch [van Loan, 1992, Abschn. 4.4]).

Als erste wichtige Eigenschaft der DCT halten wir fest, dass es sich hierbei um eine *orthogonale* Transformation handelt, womit gleichzeitig auch die Frage nach der entsprechenden inversen Transformation (iDCT) beantwortet wird:

Lemma 2.2.8. *Es gilt stets* $\tilde{C}_n^{-1} = \tilde{C}_n^{\mathrm{T}}$.

Beweis. Dies kann mit einigem Rechenaufwand unter Zuhilfenahme von Lemma 2.2.4 anhand von (2.22) nachgewiesen werden. Eine alternative Herleitung mit Kosinus-Additionstheoremen findet sich bei Baszenski und Tasche [1997]. Strang [1999, Abschn. 2] zeigt die Aussage ausschließlich mit Eigenwerttheorie: In der Tat sind die Zeilen der Matrix C_n genau die (nicht normierten) Eigenvektoren der Zweite-Differenzen-Matrix

$$\begin{bmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{bmatrix}$$

zu den Eigenwerten $4\sin^2(i\pi/2n)$, $i = 0, \ldots, n-1$. □

Insbesondere folgt aus Lemma 2.2.8 für zwei Vektoren x und y der Länge n stets

$$\langle x, y\rangle_2 \;=\; \langle \tilde{C}_n x, \tilde{C}_n y\rangle_2. \tag{2.23}$$

Bezüglich der 2-Norm ist die DCT also ein *isometrischer Isomorphismus*, denn es folgt aus (2.23) sofort $\|\tilde{C}_n x\|_2 = \|x\|_2$. In Kapitel 3 werden wir die folgende Abschätzung für die 1-Norm $\|\cdot\|_1$ benötigen:

Lemma 2.2.9. *Es seien* $n \in \mathbb{N}$ *und* $x \in \mathbb{C}^n$. *Dann gilt*

$$\|\tilde{C}_n x\|_1 \;\le\; \sqrt{n}\,\|x\|_1.$$

Beweis. Aus der Isometrieeigenschaft der DCT bezüglich der 2-Norm folgt mit der Cauchy-Schwarzschen[3] Ungleichung

$$\|\tilde{C}_n x\|_1 \;\le\; \sqrt{n}\,\|\tilde{C}_n x\|_2 \;=\; \sqrt{n}\,\|x\|_2 \;\le\; \sqrt{n}\,\|x\|_1. \qquad \square$$

Zwei weitere Eigenschaften der DCT sind für uns von besonderer Bedeutung, da sie das Arbeiten mit (trigonometrischen) Polynomen stark vereinfachen. Um dies einzusehen, führen wir zunächst die Tschebyschow-Polynome[4] ein.

Definition 2.2.10. Das k-te *Tschebyschow-Polynom* (*erster Art*) T_k wird auf $[-1, 1]$ vermöge $T_k(\cos\omega) := \cos(k\omega)$ definiert ($k \in \mathbb{N}_0$).

Aus dem bekannten Kosinus-Additionstheorem folgt, dass T_k stets ein Polynom vom Grad k ist. Dementsprechend bilden für jedes $n \in \mathbb{N}_0$ die ersten $n+1$ Tschebyschow-Polynome $T_0, \ldots, T_n$ eine Basis des $\Pi_n([-1, 1])$. Das nächste Lemma zeigt uns nun, wie man Polynome mithilfe der DCT schnell in die Tschebyschow-Basis übersetzen kann.

[3] Hermann Amandus Schwarz (⋆1843 in Hermsdorf, heute Sobieszów in Polen, †1921 in Berlin)

[4] Pafnuti Lwowitsch Tschebyschow (⋆1821 in Okatowo im Kreis Borowsk im Oblast Kaluga, Russland, †1894 in St. Petersburg, ebd.)

Lemma 2.2.11 (vgl. [Kunis und Potts, 2003, Abschn. 3]). *Es sei*

$$p = \sum_{k=0}^{n-1} \alpha_k T_k$$

ein Polynom vom Grad höchstens $n-1$ *auf* $[-1,1]$ *und* $\omega_j = (2j+1)\pi/2n$. *Dann gilt*

$$\Big[\alpha_k\Big]_{k=0,\dots,n-1} = D_n \tilde{C}_n \Big[p(\cos\omega_j)\Big]_{j=0,\dots,n-1}.$$

Man beachte hierbei, dass $D_n\tilde{C}_n = D_n^2 C_n$. Der nächste Hilfssatz, der direkt aus Lemma 2.2.11 folgt, wird uns zusammen mit (2.23) im nächsten Abschnitt helfen, den Rechenaufwand der diskreten Legendre-Transformation zu reduzieren.

Folgerung 2.2.12 (vgl. [Healy et al., 2003, Abschn. 2.4]). *Es sei* p *ein Polynom vom Grad höchstens* $n-1$ *auf* $[-1,1]$ *und* $\omega_j = (2j+1)\pi/2n$. *Mit* $v := [p(\cos\omega_j)]_{j=0,\dots,n-1}$ *gilt*

$$\Big[\tilde{C}_n v\Big]_i = 0, \qquad i > \operatorname{grad}(p).$$

2.2.3 Schnelle Legendre-Transformation (FLT)

Die assoziierten Legendre-Polynome wurden in Abschnitt 1.1 eingeführt (Def. 1.1.2). Es sei nun $n \in \mathbb{N}$ gegeben. Wir setzen $\vartheta_j := (2j+1)\pi/4n$ für $j = 0,\dots,2n-1$ und definieren die *Legendre-Matrizen*

$$L_m = L_{m,n} := \Big[P_{lm}(\cos\vartheta_j)\Big]_{\substack{l=|m|,\dots,n-1\\ j=0,\dots,2n-1}} \in \mathbb{R}^{(n-|m|)\times 2n}, \qquad |m| < n. \tag{2.24}$$

Definition 2.2.13 (DLT). Es sei $n \in \mathbb{N}$. Die Folge der Abbildungen $L_m : \mathbb{C}^{2n} \to \mathbb{C}^{n-|m|}$, $m = 1-n,\dots n-1$, heißt *diskrete Legendre-Transformation* (DLT für *discrete Legendre transform*).

Es seien nun Daten $x_m \in \mathbb{C}^{2n}$, $|m| < n$, gegeben. Bei einer Vorberechnung und Speicherung der Legendre-Matrizen L_m sind $\mathcal{O}(n^2)$ Schritte für die Berechnung jedes einzelnen Matrix-Vektor-Produkts $L_m x_m$ notwendig. Es ergibt sich ein Algorithmus mit einer Rechen- und Speicherkomplexität von $\mathcal{O}(n^3)$ zur Durchführung der DLT. In den vergangenen Jahren sind viele, teilweise sehr unterschiedliche FLTs (FLT für *fast Legendre transform*) beschrieben worden (s. etwa [Driscoll und Healy, 1994; Healy et al., 2003; Kunis und Potts, 2003]; vgl. dazu Bem. 2.2.16). Dies liegt hauptsächlich daran, dass die FLT einen festen Bestandteil der schnellen sphärischen Fourier-Transformation darstellt, die wir im nächsten Abschnitt besprechen. Wir nehmen hier die Herleitung der sogenannten *seminaiven* FLT von Healy et al. [2003] mit auf, da wir diese in den späteren numerischen Tests verwenden. Mit dem Clenshaw-Algorithmus (Abschn. 2.2.5) lernen wir noch eine weitere Klasse von FLTs kennen (vgl. Bem. 2.2.25).

Algorithmus 2.2. Seminaive FLT

Eingabe: Daten $x_m \in \mathbb{C}^{2n}$, $|m| < n \in \mathbb{N}$
Ausgabe: Transformierte Daten $L_m x_m \in \mathbb{C}^{n-|m|}$, $|m| < n$

for $m = 1 - n$ **to** $n - 1$ **do**
  if m gerade **then**
    Berechne $\tilde{x}_m := \tilde{C}_{2n} x_m$ mittels DCT;
    Berechne $\left[\tilde{L}_m^{(|m|)} \middle| \cdots \middle| \tilde{L}_m^{(n-1)}\right]^{\mathrm{T}} \tilde{x}_m$ mit $\tilde{L}_m^{(l)} := \tilde{C}_{2n} L_m^{(l)}$;
  else
    Berechne $\tilde{x}_m := \tilde{C}_{2n} W_n x_m$ mittels DCT;
    Berechne $\left[\tilde{L}_m^{(|m|)} \middle| \cdots \middle| \tilde{L}_m^{(n-1)}\right]^{\mathrm{T}} \tilde{x}_m$ mit $\tilde{L}_m^{(l)} := \tilde{C}_{2n} V_n L_m^{(l)}$;
  end
end

Rechen- und Speicherkomplexität: $\mathcal{O}(n^3)$

Es sei zu n oben ein m mit $|m| < n$ fest gewählt. Weiter bezeichne

$$L_m^{(l)} := \Big[P_{lm}(\cos\vartheta_j)\Big]_{j=0,\dots,2n-1}$$

die l-te Zeile von L_m, geschrieben als Spaltenvektor ($|m| \le l < n$). Das Ziel besteht in der Berechnung der Skalarprodukte $\langle L_m^{(l)}, x_m\rangle_2$. Mit Lemma 2.2.8 erhalten wir zunächst (vgl. Gl. 2.23)

$$\langle L_m^{(l)}, x_m\rangle_2 = \langle \tilde{C}_{2n} L_m^{(l)}, \tilde{C}_{2n} x_m\rangle_2. \tag{2.25}$$

Der Vektor $\tilde{C}_{2n} x_m$ kann mithilfe der DCT in $\mathcal{O}(n \log n)$ Schritten berechnet werden.

Es sei m gerade. Da P_{lm} in diesem Fall ein Polynom vom Grad l ist, gilt nach Folgerung 2.2.12

$$\Big[\tilde{C}_{2n} L_m^{(l)}\Big]_i = 0, \qquad i > l.$$

Somit lässt sich das Skalarprodukt auf der rechten Seite von (2.25) in $l+1$ statt n Schritten berechnen, sofern die Vektoren $\tilde{C}_{2n} L_m^{(l)}$ und $\tilde{C}_{2n} x_m$ bereits vorliegen. Insbesondere fällt bei einer Vorberechnung und Speicherung der transformierten Daten $\tilde{C}_{2n} L_m^{(l)}$ eine viel geringere Datenmenge an, als bei einer Vorberechnung und Speicherung der Legendre-Matrizen L_m; die Speicherkomplexität von $\mathcal{O}(n^3)$ bleibt dabei allerdings dieselbe.

Der Fall m ungerade kann auf den Fall m gerade zurückgeführt werden: In diesem Fall ist $(1-\xi^2)^{-1/2} P_{lm}(\xi)$ ein Polynom vom Grad $l-1$. Mit entsprechender Gewichtung der Daten x_m lässt sich das oben beschriebene Verfahren also auch hier anwenden; es gilt mit $V_n := \operatorname{diag}[(\sin\vartheta_j)^{-1}]_{j=0,\dots,2n-1}$ und $W_n := \operatorname{diag}[\sin\vartheta_j]_{j=0,\dots,2n-1}$ nämlich

$$\langle L_m^{(l)}, x_m\rangle_2 = \langle V_n L_m^{(l)}, W_n x_m\rangle_2 = \langle \tilde{C}_{2n} V_n L_m^{(l)}, \tilde{C}_{2n} W_n x_m\rangle_2.$$

Aus den obigen Überlegungen erhalten wir insgesamt die folgende Charakterisierung der seminaiven FLT, die in Pseudocode als Algorithmus 2.2 dargestellt ist.

Algorithmus 2.3. Adjungierte seminaive FLT

Eingabe: Daten $x_m \in \mathbb{C}^{n-|m|}$, $|m| < n \in \mathbb{N}$
Ausgabe: Transformierte Daten $L_m^{\mathrm{T}} x_m \in \mathbb{C}^{2n}$, $|m| < n$

for $m = 1 - n$ **to** $n - 1$ **do**
  if m gerade **then**
    Berechne $\tilde{x}_m := \left[\tilde{L}_m^{(|m|)} \middle| \cdots \middle| \tilde{L}_m^{(n-1)}\right] x_m$ mit $\tilde{L}_m^{(l)} := \tilde{C}_{2n} L_m^{(l)}$;
    Berechne $L_m^{\mathrm{T}} x_m = \tilde{C}_{2n}^{-1} \tilde{x}_m$ mittels iDCT;
  else
    Berechne $\tilde{x}_m := \left[\tilde{L}_m^{(|m|)} \middle| \cdots \middle| \tilde{L}_m^{(n-1)}\right] x_m$ mit $\tilde{L}_m^{(l)} := \tilde{C}_{2n} V_n L_m^{(l)}$;
    Berechne $L_m^{\mathrm{T}} x_m = W_n \tilde{C}_{2n}^{-1} \tilde{x}_m$ mittels iDCT;
  end
end

Rechen- und Speicherkomplexität: $\mathcal{O}(n^3)$

Satz 2.2.14. *Es seien $n \in \mathbb{N}$ und $\tilde{L}_m^{(l)} := \tilde{C}_{2n} L_m^{(l)}$ für m gerade bzw. $\tilde{L}_m^{(l)} := \tilde{C}_{2n} V_n L_m^{(l)}$ für m ungerade. Die Legendre-Matrizen L_m besitzen die Faktorisierung*

$$L_m = \left[\tilde{L}_m^{(|m|)} \,\middle|\, \cdots \,\middle|\, \tilde{L}_m^{(n-1)}\right]^{\mathrm{T}} \cdot \tilde{C}_{2n}$$

für m gerade, bzw.

$$L_m = \left[\tilde{L}_m^{(|m|)} \,\middle|\, \cdots \,\middle|\, \tilde{L}_m^{(n-1)}\right]^{\mathrm{T}} \cdot \tilde{C}_{2n} \cdot W_n$$

für m ungerade.

Durch Umkehren der Produktreihenfolge und Transponieren der einzelnen Faktoren erhalten wir unter Berücksichtigung von Lemma 2.2.8:

Folgerung 2.2.15. *Mit den Bezeichnungen in Satz 2.2.14 gilt*

$$L_m^{\mathrm{T}} = \tilde{C}_{2n}^{-1} \cdot \left[\tilde{L}_m^{(|m|)} \,\middle|\, \cdots \,\middle|\, \tilde{L}_m^{(n-1)}\right]$$

für m gerade, bzw.

$$L_m^{\mathrm{T}} = W_n \cdot \tilde{C}_{2n}^{-1} \cdot \left[\tilde{L}_m^{(|m|)} \,\middle|\, \cdots \,\middle|\, \tilde{L}_m^{(n-1)}\right]$$

für m ungerade.

Insbesondere erhalten wir daraus die *adjungierte* seminaive FLT, d. h. einen schnellen Algorithmus zur Auswertung der Abbildungen $L_m^{\mathrm{T}} : \mathbb{C}^{n-|m|} \to \mathbb{C}^{2n}$, $m = 1 - n, \ldots, n - 1$. Die adjungierte seminaive FLT, die in Pseudocode als Algorithmus 2.3 dargestellt ist, hat ebenfalls eine Komplexität von $\mathcal{O}(n^3)$. Diese Klasse von schnellen Algorithmen benötigen wir im nächsten Abschnitt bei der Herleitung der schnellen inversen sphärischen Fourier-Transformation (iFSFT); in Abschnitt 3.2.1 kommt sie dann noch einmal zum Einsatz.

Bemerkung 2.2.16. Nach dem bekannten *Teile-und-Herrsche*-Prinzip (s. z. B. [Cormen et al., 2001, Abschn. 2.3.1]) entwickeln Healy et al. die seminaive FLT weiter und erhalten verschiedene FLTs mit einer Rechen- und Speicherkomplexität von $\mathcal{O}(n^2 \log^2 n)$.

Man vergleiche hierzu [Healy et al., 2003, Satz 3] und beachte, dass die „*precomputed data structure*“ dort nur für die FLT benötigt wird. Kunis und Potts [2003, Abschn. 4] bieten ebenfalls FLTs mit der Komplexität $\mathcal{O}(n^2 \log^2 n)$; hier wurde sogar eine Stabilisierung für große Problemgrößen vorgenommen. Diese elaborierten FLTs sind für uns jedoch nur von theoretischem Interesse: Für die in dieser Arbeit behandelten Problemgrößen ist die seminaive FLT eine geeignete Wahl (vgl. dazu [Healy et al., 2003, Abschn. 6]).

2.2.4 Schnelle sphärische Fourier-Transformation (FSFT)

In Abschnitt 2.1.1 (Satz 2.1.2) haben wir gesehen, wie man die Fourier-Koeffizienten $\langle p, Y_{lm} \rangle_{\mathbb{S}^2}$ eines sphärischen Polynoms p auf diskrete Art und Weise berechnen kann. Die *schnelle sphärische Fourier-Transformation* (FSFT für *fast spherical Fourier transform*), die nun besprochen werden soll, ist die Klasse von schnellen Algorithmen für genau diesen Zweck. Algorithmen dieser Art haben sich in vielen Anwendungen als außerordentlich nützlich erwiesen.

Wir identifizieren das Paar $[l, m]$, $|m| \leq l < n$, mit $\sigma \in \{0, \ldots, n^2 - 1\}$ vermöge des bijektiven Zusammenhangs

$$\sigma = l(l+1) + m.$$

Dies erlaubt es uns, die Indizes l und m in diesem Abschnitt als Funktionen von σ aufzufassen:

$$\begin{cases} l(\sigma) &= \lfloor \sqrt{\sigma} \rfloor, \\ m(\sigma) &= \sigma - l(\sigma)(l(\sigma)+1). \end{cases} \tag{2.26}$$

Darüber hinaus identifizieren wir das Paar $[j, k] \in \{0, \ldots, 2n-1\}^2$ mit $\tau \in \{0, \ldots, 4n^2-1\}$ vermittels des eineindeutigen Zusammenhangs

$$\tau = 2nj + k.$$

Wir können die Indizes j und k in diesem Abschnitt dann als Funktionen von τ ansehen:

$$\begin{aligned} j(\tau) &= \frac{\tau - (\tau \bmod 2n)}{2n}, \\ k(\tau) &= \tau \bmod 2n. \end{aligned}$$

Es sind im Folgenden ϑ_j und φ_k die Abtastwinkel aus Satz 2.1.2 und β_j die dazugehörigen sphärischen Quadraturgewichte.

Definition 2.2.17 (DSFT). Es seien $n \in \mathbb{N}$ und

$$\begin{aligned} V_n &:= \operatorname{diag}\left[\beta_{j(\tau)}\right]_{\tau=0,\ldots,4n^2-1} \in \mathbb{R}^{4n^2 \times 4n^2}, \\ Y_n &:= \left[\overline{Y_{l(\sigma),m(\sigma)}\left(\vartheta_{j(\tau)}, \varphi_{k(\tau)}\right)}\right]_{\substack{\sigma=0,\ldots,n^2-1 \\ \tau=0,\ldots,4n^2-1}} \in \mathbb{C}^{n^2 \times 4n^2}. \end{aligned}$$

Die Abbildung $Y_n V_n : \mathbb{C}^{4n^2} \to \mathbb{C}^{n^2}$ heißt *diskrete sphärische Fourier-Transformation* (DSFT für *discrete spherical Fourier transform*).

Zur Durchführung der DSFT bedarf es eines Algorithmus zur Berechnung der Doppelsummen in (2.1). Eine direkte Berechnung dieser Summen benötigt offenbar $\mathcal{O}(n^4)$ Schritte. Im Folgenden beschreiben wir das Grundgerüst der FSFTs von Driscoll und Healy [1994] und Healy et al. [2003], da eine solche FSFT in den späteren numerischen Tests zum Einsatz kommt. Für eine weitere Klasse von FSFTs, die auf der nichtäquidistanten FFT (NFFT) basieren, siehe [Kunis und Potts, 2003] (vgl. auch Bem. 3.2.1).

Ein Umsortieren der Doppelsummen in (2.1) liefert zunächst

$$\langle p, Y_{lm}\rangle_{\mathbb{S}^2} = M_{lm} \sum_{j=0}^{2n-1} \beta_j P_{lm}(\cos\vartheta_j) \sum_{k=0}^{2n-1} p(\vartheta_j, \varphi_k)\,\mathrm{e}^{-\mathrm{i}m\varphi_k}, \qquad |m| \le l < n, \tag{2.27}$$

wobei M_{lm} die Normierungskonstante der Kugelflächenfunktion Y_{lm} ist (Def. 1.1.7). Diese Separation der Variablen resultiert in einer Klasse von FSFTs mit einer Rechen- und Speicherkomplexität von $\mathcal{O}(n^3)$ oder sogar $\mathcal{O}(n^2 \log^2 n)$: Die inneren Summen in (2.27) können bei festem $j = 0, \ldots, 2n-1$ unter Verwendung der FFT der Länge $2n$ in $\mathcal{O}(n \log n)$ Schritten für alle $m = 1-n, \ldots, n-1$ berechnet werden (vgl. Gl. 2.21). Es ergibt sich eine Rechenkomplexität von $\mathcal{O}(n^2 \log n)$ und eine Speicherkomplexität von $\mathcal{O}(n^2)$ für diesen ersten Schritt. Anschließend wird die äußere Summe in (2.27) bei jeweils festem $m = 1-n, \ldots, n-1$ für alle $l = |m|, \ldots, n-1$ ausgewertet. Dies leistet die FLT in insgesamt $\mathcal{O}(n^3)$ Schritten bei Verwendung der seminaiven Variante (Alg. 2.2), bzw. in $\mathcal{O}(n^2 \log^2 n)$ Schritten bei Einsatz einer der daraus abgeleiteten Varianten (vgl. Bem. 2.2.16). Dieser zweite Schritt hat dann eine Speicherkomplexität von $\mathcal{O}(n^3)$ bzw. $\mathcal{O}(n^2 \log^2 n)$. Wir bezeichnen die FSFT mit seminaiver FLT als seminaive FSFT.

Wir stellen die oben beschriebene Klasse von FSFTs hier als spezielle Faktorisierung der Matrix Y_n in Definition 2.2.17 dar. Dazu definieren wir die Hilfsmatrizen

$$\begin{cases} A_n := \left[\delta_{jk}\right]_{\substack{j=0,\ldots,2n-2\\ k=0,\ldots,2n-1}} \in \mathbb{R}^{(2n-1)\times 2n}, \\ B_n := \operatorname{diag}\left[\mathrm{e}^{2\pi \mathrm{i} k(n-1)/2n}\right]_{k=0,\ldots,2n-1} \in \mathbb{C}^{2n\times 2n}, \end{cases} \tag{2.28}$$

welche die Berechnung der inneren Summen in (2.27) der FFT formal zugänglich machen (man beachte den Indexbereich in Gleichung 2.21). Zusätzlich dazu definieren wir die Permutationsmatrizen

$$\begin{aligned} P_n &:= \left[e_{\psi(\sigma)}^{\mathrm{T}}\right]_{\sigma=0,\ldots,n^2-1} \in \mathbb{R}^{n^2\times n^2}, \\ Q_n &:= \left[e_{\omega(\zeta)}^{\mathrm{T}}\right]_{\zeta=0,\ldots,2n(2n-1)-1} \in \mathbb{R}^{2n(2n-1)\times 2n(2n-1)}, \end{aligned}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und (s. Gln. 2.26)

$$\begin{aligned} \psi(\sigma) &:= \frac{n^2 - n - m(\sigma)^2 \operatorname{sgn} m(\sigma)}{2} + \left(n - \frac{\operatorname{sgn} m(\sigma)}{2}\right) m(\sigma) + l(\sigma), \\ \omega(\zeta) &:= (2n-1)(\zeta \bmod 2n) + \frac{\zeta - (\zeta \bmod 2n)}{2n}. \end{aligned}$$

Die Matrix Q_n sorgt für eine Umsortierung von $j = 0, \ldots, 2n-1$; $m = 1-n, \ldots, n-1$, hin zu $m = 1-n, \ldots, n-1$; $j = 0, \ldots, 2n-1$. Die Matrix P_n erledigt die Umsortierung von

Algorithmus 2.4. FSFT

Eingabe: Abtastwerte $p(\vartheta_j, \varphi_k)$; $j, k = 0, \ldots, 2n-1$, eines sphärischen Polynoms p vom Grad höchstens $n-1 \in \mathbb{N}_0$

Ausgabe: Sphärische Fourier-Koeffizienten $\tilde{p}_{lm} := \langle p, Y_{lm} \rangle_{\mathbb{S}^2}$, $|m| \leq l < n$

for $j = 0$ **to** $2n-1$ **do**

  Berechne mithilfe der FFT $[T_{jm}]_{m=1-n,\ldots,n-1} := A_n F_{2n} B_n \, [p(\vartheta_j, \varphi_k)]_{k=0,\ldots,2n-1}$;

end

Berechne mithilfe der FLT

$$[\tilde{p}_{lm}]_{l=|m|,\ldots,n-1} = \operatorname{diag}[M_{lm}]_{l=|m|,\ldots,n-1} \cdot L_m \cdot \operatorname{diag}[\beta_j]_{j=0,\ldots,2n-1} \cdot [T_{jm}]_{j=0,\ldots,2n-1}$$

für alle $m = 1-n, \ldots, n-1$.

Rechen- und Speicherkomplexität: $\mathcal{O}(n^3)$ (seminaiv), $\mathcal{O}(n^2 \log^2 n)$ (elaboriert)

$m = 1-n, \ldots, n-1$; $l = |m|, \ldots, n-1$, hin zu $l = 0, \ldots, n-1$; $m = -l, \ldots, l$. Es ist F_{2n} die Fourier-Matrix der Länge $2n$ (Def. 2.2.3), die Matrizen $L_m = L_{m,n}$ sind die in (2.24) definierten Legendre-Matrizen. Wir erhalten mit den obigen Bestandteilen insgesamt:

Satz 2.2.18. *Die Matrix Y_n in Definition 2.2.17 besitzt die Faktorisierung*

$$Y_n = \operatorname{diag}\Big[M_{l(\sigma),m(\sigma)}\Big]_{\sigma=0,\ldots,n^2-1} \cdot P_n \cdot \underbrace{\begin{bmatrix} L_{1-n} & & \\ & \ddots & \\ & & L_{n-1} \end{bmatrix}}_{2n-1 \text{ Blöcke } L_m \text{ der Größe } (n-|m|)\times 2n} \cdot Q_n \cdot \underbrace{\{E_{2n} \otimes (A_n F_{2n} B_n)\}}_{2n \text{ Blöcke der Größe } (2n-1)\times 2n}.$$

Die Legendre-Matrizen L_m können nun wie im vorigen Abschnitt beschrieben weiter faktorisiert werden; Satz 2.2.14 etwa liefert die seminaive FSFT. Die obige Klasse von FSFTs ist in Pseudocode als Algorithmus 2.4 dargestellt. Durch Umkehren der Produktreihenfolge und hermitesches Transponieren der einzelnen Faktoren erhalten wir zusammen mit Lemma 2.2.4 zusätzlich die folgende Faktorisierung der Matrix Y_n^{H}:

Folgerung 2.2.19. *Die Matrix Y_n^{H} besitzt die Faktorisierung*

$$Y_n^{\mathrm{H}} = 2n \left\{E_{2n} \otimes \left(B_n^{\mathrm{H}} F_{2n}^{-1} A_n^{\mathrm{T}}\right)\right\} \cdot Q_n^{\mathrm{T}} \cdot \begin{bmatrix} L_{1-n}^{\mathrm{T}} & & \\ & \ddots & \\ & & L_{n-1}^{\mathrm{T}} \end{bmatrix} \cdot P_n^{\mathrm{T}} \cdot \operatorname{diag}\Big[M_{l(\sigma),m(\sigma)}\Big]_{\sigma=0,\ldots,n^2-1}.$$

Die Matrizen L_m^{T} können nun ebenfalls weiter faktorisiert werden; Folgerung 2.2.15 liefert hier die seminaive Variante. Das folgende Lemma, das sofort aus Satz 2.1.2 folgt, zeigt uns mit Folgerung 2.2.19 abschließend, dass wir mit den oben beschriebenen FSFTs auch stets eine schnelle inverse Transformation (iFSFT) zur Hand haben (Alg. 2.5).

Lemma 2.2.20. *Es sei $p \in \Pi_{n-1}(\mathbb{S}^2)$, $n \in \mathbb{N}$. Mit dem gleichnamigen Vektor $p := [p(\vartheta_{j(\tau)}, \varphi_{k(\tau)})]_{\tau=0,\ldots,4n^2-1}$ und dem Vektor $\tilde{p} := [\langle p, Y_{l(\sigma),m(\sigma)} \rangle_{\mathbb{S}^2}]_{\sigma=0,\ldots,n^2-1}$ gilt*

$$Y_n^{\mathrm{H}} (Y_n V_n) p = p,$$

$$(Y_n V_n) Y_n^{\mathrm{H}} \tilde{p} = \tilde{p}.$$

Algorithmus 2.5. iFSFT

Eingabe: Sphärische Fourier-Koeffizienten $\tilde{p}_{lm} := \langle p, Y_{lm} \rangle_{\mathbb{S}^2}$, $|m| \leq l < n$, eines sphärischen Polynoms p vom Grad höchstens $n-1 \in \mathbb{N}_0$
Ausgabe: Funktionswerte $p(\vartheta_j, \varphi_k)$; $j, k = 0, \ldots, 2n-1$

Berechne mithilfe der adjungierten FLT

$$[U_{jm}]_{j=0,\ldots,2n-1} := L_m^{\mathrm{T}} \cdot \operatorname{diag}[M_{lm}]_{l=|m|,\ldots,n-1} \cdot [\tilde{p}_{lm}]_{l=|m|,\ldots,n-1}$$

für alle $m = 1-n, \ldots, n-1$;

for $j = 0$ **to** $2n-1$ **do**
  Berechne mithilfe der iFFT $[p(\vartheta_j, \varphi_k)]_{k=0,\ldots,2n-1} = 2n\, B_n^{\mathrm{H}} F_{2n}^{-1} A_n^{\mathrm{T}} [U_{jm}]_{m=1-n,\ldots,n-1}$;
end

Rechen- und Speicherkomplexität: $\mathcal{O}(n^3)$ (seminaiv), $\mathcal{O}(n^2 \log^2 n)$ (elaboriert)

2.2.5 Der Clenshaw-Algorithmus

Wir betrachten im Folgenden allgemein ein Funktionensystem $\{f_k : k \in \mathbb{N}_0\}$, das einer Drei-Term-Rekursion der Form

$$f_{k+1}(\xi) = \alpha_k(\xi) f_k(\xi) + \beta_k(\xi) f_{k-1}(\xi), \qquad \xi \in \mathbb{R},\ k \in \mathbb{N}, \tag{2.29}$$

genügt. Dabei nehmen wir an, dass sich die Koeffizientenfunktionen α_k und β_k in $\mathcal{O}(1)$ Schritten auswerten lassen. Wir suchen nach einer effizienten Methode, um für gegebene Daten $[\gamma_0, \ldots, \gamma_{n-1}]$ die Summen

$$S_j := \sum_{k=0}^{n-1} \gamma_k f_k(\xi_j), \qquad j = 0, \ldots, m-1, \tag{2.30}$$

an gegebenen Punkten $[\xi_0, \ldots, \xi_{m-1}]$ zu berechnen ($m, n \in \mathbb{N}$). Dieses Problem lässt sich in Matrix-Vektor-Schreibweise als

$$\begin{bmatrix} S_0 \\ \vdots \\ S_{m-1} \end{bmatrix} = \underbrace{\begin{bmatrix} f_0(\xi_0) & \cdots & f_{n-1}(\xi_0) \\ \vdots & & \vdots \\ f_0(\xi_{m-1}) & \cdots & f_{n-1}(\xi_{m-1}) \end{bmatrix}}_{=:\, A} \cdot \begin{bmatrix} \gamma_0 \\ \vdots \\ \gamma_{n-1} \end{bmatrix} \tag{2.31}$$

darstellen.

Berechnet man die Einträge der Matrix $A = A(\xi_0, \ldots, \xi_{m-1}) \in \mathbb{C}^{m \times n}$ jeweils einzeln mit der Drei-Term-Rekursion (2.29), so benötigt die Auswertung von (2.31) offensichtlich $\mathcal{O}(mn^2)$ Operationen. Mithilfe des *Clenshaw-Algorithmus*, der erstmals im Zusammenhang mit den Tschebyschow-Polynomen (Def. 2.2.10) von Clenshaw [1955] vorgestellt wurde, lässt sich die Komplexität des Problems auf $\mathcal{O}(mn)$ reduzieren. Dazu führt man zunächst die von den Daten abhängigen Koeffizientenfunktionen

$$\begin{aligned} \delta_{n+1} &:= \delta_n := 0, \\ \delta_k(\xi) &:= \alpha_k(\xi)\delta_{k+1}(\xi) + \beta_{k+1}(\xi)\delta_{k+2}(\xi) + \gamma_k, \qquad \xi \in \mathbb{R},\ 0 \leq k < n, \end{aligned} \tag{2.32}$$

ein. Es gilt damit

$$\gamma_k \;\equiv\; \delta_k - \alpha_k\delta_{k+1} - \beta_{k+1}\delta_{k+2}, \qquad 0 \le k < n. \tag{2.33}$$

Durch Einsetzen von (2.33) in (2.30) ergibt sich nun für festes $j \in \{0, \dots, m-1\}$

$$\begin{aligned} S_j \;&=\; \gamma_0 f_0(\xi_j) + \sum_{k=1}^{n-1} \{\delta_k(\xi_j) - \alpha_k(\xi_j)\delta_{k+1}(\xi_j) - \beta_{k+1}(\xi_j)\delta_{k+2}(\xi_j)\} f_k(\xi_j) \\ &=\; \gamma_0 f_0(\xi_j) + \delta_1(\xi_j) f_1(\xi_j) \;+\; \delta_2(\xi_j) \underbrace{\{f_2(\xi_j) - \alpha_1(\xi_j) f_1(\xi_j)\}}_{=\,\beta_1(\xi_j) f_0(\xi_j)} \\ &\qquad + \sum_{k=3}^{n-1} \delta_k(\xi_j) \underbrace{\{f_k(\xi_j) - \alpha_{k-1}(\xi_j) f_{k-1}(\xi_j) - \beta_{k-1}(\xi_j) f_{k-2}(\xi_j)\}}_{=\,0} \\ &=\; \{\gamma_0 + \beta_1(\xi_j)\delta_2(\xi_j)\} f_0(\xi_j) + \delta_1(\xi_j) f_1(\xi_j). \end{aligned} \tag{2.34}$$

Dabei wurde an den bezeichneten Stellen die Drei-Term-Rekursion (2.29) berücksichtigt.

Wendet man das obige Ergebnis für jedes $j = 0, \dots, m-1$ an, so erhält man unter Berücksichtigung der Rekursionsgleichung (2.32) den bekannten Clenshaw-Algorithmus. Wir charakterisieren diesen durch folgende Faktorisierung der Matrix A:

Satz 2.2.21. *Für $m, n \in \mathbb{N}$ mit $n \ge 3$ besitzt die Matrix A in (2.31) die Faktorisierung*

$$A \;=\; V_0 \cdots V_{n-2} \tag{2.35}$$

mit

$$V_0 \;:=\; \left[\begin{array}{c|ccc|ccc} f_0(\xi_0) & f_1(\xi_0) & & & \beta_1(\xi_0) f_0(\xi_0) & & \\ \vdots & & \ddots & & & \ddots & \\ f_0(\xi_{m-1}) & & & f_1(\xi_{m-1}) & & & \beta_1(\xi_{m-1}) f_0(\xi_{m-1}) \end{array}\right],$$

$$V_j \;:=\; \left[\begin{array}{c|c|ccc|ccc} E_j & & & & & & & \\ \hline & & \alpha_j(\xi_0) & & & \beta_{j+1}(\xi_0) & & \\ & 1_m & & \ddots & & & \ddots & \\ & & & & \alpha_j(\xi_{m-1}) & & & \beta_{j+1}(\xi_{m-1}) \\ \hline & & & E_m & & & & \end{array}\right], \quad 1 \le j < n-2,$$

$$V_{n-2} \;:=\; \left[\begin{array}{c|c|c} E_{n-2} & & \\ \hline & & \alpha_{n-2}(\xi_0) \\ & 1_m & \vdots \\ & & \alpha_{n-2}(\xi_{m-1}) \\ \hline & & 1_m \end{array}\right],$$

wobei 1_m den Einsvektor der Länge m bezeichne.

Für $n = 1$ und $n = 2$ ist eine Faktorisierung der Matrix A nicht von Belang. Das genaue Vorgehen zur Auswertung von (2.31) ist in Pseudocode als Algorithmus 2.6 dargestellt. Jeder Wert S_j lässt sich in $\mathcal{O}(1)$ Schritten aus den Werten $\beta_1(\xi_j)$, γ_0, $\delta_1(\xi_j)$ und $\delta_2(\xi_j)$ berechnen, da auf der rechten Seite der Gleichung (2.34) nur die beiden Funktionswerte

Algorithmus 2.6. Clenshaw-Algorithmus

Eingabe: Punkte $[\xi_0, \ldots, \xi_{m-1}]$, $m \in \mathbb{N}$, und Daten $[\gamma_0, \ldots, \gamma_{n-1}]$, $n \in \mathbb{N}$
Ausgabe: Summen $S_0, \ldots, S_{m-1}$

for $l = 0$ **to** $n-2$ **do**
  Berechne $\delta_{n-l-1}(\xi_j)$ für $j = 0, \ldots, m-1$ mittels (2.32) aus $\delta_{n-l}(\xi_j)$ und $\delta_{n-l+1}(\xi_j)$;
end
Berechne S_j für $j = 0, \ldots, m-1$ mittels (2.34) aus $\delta_1(\xi_j)$ und $\delta_2(\xi_j)$.

Rechen- und Speicherkomplexität: $\mathcal{O}(mn)$ bzw. $\mathcal{O}(m)$

Algorithmus 2.7. Adjungierter Clenshaw-Algorithmus

Eingabe: Punkte $[\xi_0, \ldots, \xi_{m-1}]$ und Daten $[\gamma_0, \ldots, \gamma_{m-1}]$, $m \in \mathbb{N}$; $n \in \mathbb{N}$
Ausgabe: Summen $\tilde{S}_0, \ldots, \tilde{S}_{n-1}$

for $k = 0$ **to** $n-2$ **do**
  Berechne $\tilde{S}_k$ und $f_{k+1}(\xi_j)$, $j = 0, \ldots, m-1$, für $k \geq 1$ mittels (2.29) aus $f_k(\xi_j)$ und $f_{k-1}(\xi_j)$, $j = 0, \ldots, m-1$;
end
Berechne $\tilde{S}_{n-1}$, für $n \geq 2$ mittels (2.29) aus $f_{n-1}(\xi_j)$ und $f_{n-2}(\xi_j)$, $j = 0, \ldots, m-1$.

Rechen- und Speicherkomplexität: $\mathcal{O}(mn)$ bzw. $\mathcal{O}(m)$

$f_0(\xi_j)$ und $f_1(\xi_j)$ auftreten. Da die Berechnung der Werte $\delta_1(\xi_j)$ und $\delta_2(\xi_j)$ mittels der Rekursionsgleichung (2.32) zusammen nur $\mathcal{O}(n)$ Schritte benötigt, ergibt dies einen Rechenaufwand von $\mathcal{O}(n)$ für die Berechnung einer jeden Summe S_j. Insgesamt ergibt sich so ein Rechenaufwand von $\mathcal{O}(mn)$, wie behauptet.

Bemerkung 2.2.22. Hinsichtlich der Rechenkomplexität kommt der Clenshaw-Algorithmus einer Vorberechnung der Matrix A in (2.31) gleich. Eine solche Vorberechnung besitzt aber eine Speicherkomplexität von $\mathcal{O}(mn)$, wohingegen der Clenshaw-Algorithmus eine deutlich geringere Speicherkomplexität von lediglich $\mathcal{O}(m)$ aufweist.

Aus der Matrix-Faktorisierung (2.35) ergibt sich unmittelbar der *adjungierte* Clenshaw-Algorithmus. Dieser erlaubt für gegebene Punkte $[\xi_0, \ldots, \xi_{m-1}]$ und zugehörige Daten $[\gamma_0, \ldots, \gamma_{m-1}]$ eine Berechnung der Summen

$$\tilde{S}_k := \sum_{j=0}^{m-1} \gamma_j \, f_k(\xi_j), \qquad k = 0, \ldots, n-1,$$

in $\mathcal{O}(mn)$ Schritten ($m, n \in \mathbb{N}$). Der adjungierte Clenshaw-Algorithmus (Alg. 2.7) wird nämlich von der Matrix A^{H} repräsentiert, und es folgt aus Satz 2.2.21:

Folgerung 2.2.23. *Unter den Voraussetzungen von Satz 2.2.21 besitzt die Matrix A^{H} die Faktorisierung*

$$A^{\mathrm{H}} = V_{n-2}^{\mathrm{H}} \cdots V_0^{\mathrm{H}}.$$

Bemerkung 2.2.24. Eine Alternative zum adjungierten Clenshaw-Algorithmus ist durch die *schnelle diskrete Polynom-Transformation* (FDPT für *fast discrete polynomial transform*) von Driscoll et al. [1997] gegeben. Diese Klasse von schnellen Algorithmen hat im Fall $m = n$ eine Komplexität von $\mathcal{O}(n \log^2 n)$ und ist in dieser Hinsicht besser als der Clenshaw-Algorithmus. Sie basiert auf einer speziellen Faktorisierung der Matrix A in (2.31), bei der Matrizen mit Toeplitz-Struktur[5] auftreten. Diese Struktur erlaubt eine schnelle Berechnung der entsprechenden Matrix-Vektor-Produkte mithilfe der FFT (s. [van Loan, 1992, Abschn. 4.2.4]). Die Speicherkomplexität der FDPTs von Driscoll et al. ist $\mathcal{O}(n \log n)$; diese Algorithmen schneiden hier schlechter ab als der Clenshaw-Algorithmus (vgl. [Driscoll et al., 1997, Abschn. 2.2, Bem. 1]). In dieser Arbeit ist die FDPT nur von theoretischem Interesse, da bei den behandelten Problemgrößen kein wesentlicher Vorteil gegenüber dem Clenshaw-Algorithmus zu erwarten ist. Eine weitere, DCT-basierte Variante der FDPT ist in [Potts et al., 1998] beschrieben.

Bemerkung 2.2.25. Da die assoziierten Legendre-Polynome (Def. 1.1.3) einer Drei-Term-Rekursion der Form (2.29) genügen (vgl. Gl. 1.3), lässt sich aus dem adjungierten Clenshaw-Algorithmus auch eine FLT mit einer Rechenkomplexität von $\mathcal{O}(n^3)$ und einer Speicherkomplexität von $\mathcal{O}(n)$ ableiten. Es ergibt sich so auch eine FSFT mit einer Rechenkomplexität von $\mathcal{O}(n^3)$ und einer Speicherkomplexität von $\mathcal{O}(n^2)$. Wenn man statt des adjungierten Clenshaw-Algorithmus eine FDPT von Driscoll et al. verwendet, so ergibt sich eine FLT mit einer Rechenkomplexität von lediglich $\mathcal{O}(n^2 \log^2 n)$ und einer Speicherkomplexität von $\mathcal{O}(n^2 \log n)$ oder sogar $\mathcal{O}(n \log n)$ (vgl. [Driscoll et al., 1997, Abschn. 2.2, Bem. 2]). Daraus resultiert eine FSFT mit einer Rechenkomplexität von $\mathcal{O}(n^2 \log^2 n)$ und einer Speicherkomplexität von $\mathcal{O}(n^2 \log n)$ bzw. $\mathcal{O}(n^2)$.

2.2.6 Herleitung der schnellen Algorithmen

Es sei im Folgenden $f \in H$ eine beliebige bandbegrenzte Funktion mit Bandbreite B. Unser Vorgehen wird von dem Beweis unseres SGL-Abtasttheorems (Satz 2.1.12) motiviert. Ähnlich wie dort separieren wir zunächst den Radius r von den Winkeln ϑ und φ, hier durch ein Umsortieren der Dreifachsumme in (2.19):

$$\hat{f}_{nlm} = N_{nl} \sum_{i=0}^{2B-1} \alpha_i r_i^2 R_{nl}(r_i) \sum_{j,k=0}^{2B-1} \beta_j f(r_i, \vartheta_j, \varphi_k) \overline{Y_{lm}(\vartheta_j, \varphi_k)}, \quad |m| \leq l < n \leq B. \tag{2.36}$$

Diese Separation der Variablen unterteilt die naive DSGLFT in eine *radiale* und eine *sphärische Teiltransformation*. Der sphärische Teil besteht in der Berechnung der inneren Doppelsumme für alle $[l, m]$, $|m| \leq l < B$, jeweils bei festem $i = 0, \ldots, 2B - 1$. Anschließend wird im radialen Teil bei festem $[l, m]$, $|m| \leq l < B$, die äußere Summe für alle $n = l + 1, \ldots, B$ ausgewertet. Die obige Separation der Variablen allein reduziert die Rechenkomplexität potentiell auf $\mathcal{O}(B^5)$, die Speicherkomplexität steigt durch die Vorberechnung der inneren Doppelsummen auf lediglich $\mathcal{O}(B^3)$.

Die Komplexität der radialen und sphärischen Teiltransformation können wir weiter reduzieren. Wie im Beweis von Satz 2.1.12 stellen wir fest, dass es sich bei der eingeschränkten Funktion $f(r_i, \cdot, \cdot)$ um ein sphärisches Polynom vom Grad höchstens $B - 1$ handelt.

[5] Otto Toeplitz (⋆1881 in Breslau, heute Wrocław in Polen, †1940 in Jerusalem)

Das Abtasttheorem von Driscoll und Healy (Satz 2.1.2) zeigt uns dementsprechend, dass es sich bei der inneren Doppelsumme jeweils um einen speziellen sphärischen Fourier-Koeffizienten, nämlich $\langle f(r_i,\cdot,\cdot), Y_{lm}\rangle_{\mathbb{S}^2}$, handelt. Für die sphärische Teiltransformation eignet sich also eine FSFT, mit deren Hilfe bei festem $i = 0,\dots,2B-1$ die Gesamtheit dieser sphärischen Fourier-Koeffizienten effizient berechnet werden kann. Nutzt man an dieser Stelle die seminaive FSFT, so ergibt sich hierfür eine Rechenkomplexität von insgesamt $\mathcal{O}(B^4)$, die Speicherkomplexität von $\mathcal{O}(B^3)$ bleibt erhalten. Bei Verwendung einer FSFT mit der Komplexität $\mathcal{O}(B^2\log^2 B)$ reduziert sich die Rechenkomplexität der sphärischen Teiltransformation bei gleichbleibender Speicherkomplexität auf $\mathcal{O}(B^3\log^2 B)$.

Die Komplexität der radialen Teiltransformation können wir reduzieren, indem wir die Drei-Term-Rekursionsformel (1.17) auf effizientere Weise nutzen, als nur für die direkte Berechnung der Funktionswerte $R_{nl}(r_i)$. Bei genauerer Betrachtung der äußeren Summe in (2.36) stellen wir fest, dass diese Summen bei jeweils festem $[l,m]$, $|m| \le l < B$, für alle $n = l+1,\dots,B$ mithilfe des adjungierten Clenshaw-Algorithmus (Alg. 2.7) berechnet werden können. Dadurch reduziert sich die Komplexität der radialen Teiltransformation auf $\mathcal{O}(B^4)$. Es ergibt sich so eine FSGLFT mit einer Rechenkomplexität von $\mathcal{O}(B^4)$ und einer Speicherkomplexität von $\mathcal{O}(B^3)$. Verwendet man statt des adjungierten Clenshaw-Algorithmus eine FDPT (Bem. 2.2.24) und für die sphärische Teiltransformation eine FSFT mit der Komplexität $\mathcal{O}(B^2\log^2 B)$, so reduziert sich die Rechenkomplexität der FSGLFT bei gleichbleibender Speicherkomplexität sogar auf $\mathcal{O}(B^3\log^2 B)$.

Als Hauptresultat dieses Abschnitts wollen wir die oben beschriebene FSGLFT als Matrix-Faktorisierung darstellen. Dazu definieren wir die Permutationsmatrizen

$$\begin{aligned} S_B &:= \left[e^{\mathrm{T}}_{\iota(\mu)}\right]_{\mu=0,\dots,B(B+1)(2B+1)/6-1} \in \mathbb{R}^{B(B+1)(2B+1)/6\times B(B+1)(2B+1)/6}, \\ T_B &:= \left[e^{\mathrm{T}}_{\kappa(\eta)}\right]_{\eta=0,\dots,2B^3-1} \in \mathbb{R}^{2B^3\times 2B^3}, \end{aligned} \tag{2.37}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und (s. Gln. 2.20)

$$\begin{aligned} \iota(\mu) &:= n(\mu) + \left(B + \left(\frac{2B-1}{2} - \frac{2l(\mu)-1}{3}\right)(l(\mu)-1) - 1\right) l(\mu) \\ &\quad + (B - l(\mu))(l(\mu) + m(\mu)) - 1, \\ \kappa(\eta) &:= (\eta \bmod 2B)B^2 + \frac{\eta - (\eta \bmod 2B)}{2B}. \end{aligned}$$

Die Matrix T_B sorgt für eine Umsortierung von $i = 0,\dots,2B-1$; $l = 0,\dots,B-1$; $m = -l,\dots,l$, hin zu $l = 0,\dots,B-1$; $m = -l,\dots,l$; $i = 0,\dots,2B-1$. Die Matrix S_B erledigt die Umsortierung von $l = 0,\dots,B-1$; $m = -l,\dots,l$; $n = l+1,\dots,B$, hin zu $n = 1,\dots,B$; $l = 0,\dots,n-1$; $m = -l,\dots,l$. Weiter definieren wir die zur radialen Teiltransformation gehörigen Matrizen

$$R_l := \left[N_{nl}\,R_{nl}(r_i)\right]_{\substack{i=0,\dots,2B-1\\ n=l+1,\dots,B}} \in \mathbb{R}^{2B\times(B-l)}, \qquad l < B.$$

Es ist Y_B die Matrix der DSFT in Definition 2.2.17 mit $n := B$. Mit den obigen Bestandteilen ergibt sich nun aus den Überlegungen dieses Abschnitts insgesamt:

Algorithmus 2.8. FSGLFT

Eingabe: Abtastwerte $f(r_i, \vartheta_j, \varphi_k)$; $i, j, k = 0, \ldots, 2B-1$, einer bandbegrenzten Funktion mit Bandbreite B

Ausgabe: SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$, $|m| \leq l < n \leq B$

for $i = 0$ **to** $2B-1$ **do**

  Berechne $[\langle f(r_i, \cdot, \cdot), Y_{l(\sigma),m(\sigma)} \rangle_{\mathbb{S}^2}]_{\sigma=0,\ldots,B^2-1} = Y_B V_B \, [f(r_i, \vartheta_{j(\tau)}, \varphi_{k(\tau)})]_{\tau=0,\ldots,(2B)^2-1}$ mittels FSFT, mit σ und τ wie in Abschnitt 2.2.4;

end

for $l = 0$ **to** $B-1$ **do**

  for $m = -l$ **to** l **do**

    Berechne mittels adjungiertem Clenshaw-Algorithmus oder FDPT

    $$[\hat{f}_{nlm}]_{n=l+1,\ldots,B} \;=\; R_l^{\mathrm{T}} \cdot \mathrm{diag}[\alpha_i r_i^2]_{i=0,\ldots,2B-1} \cdot [\langle f(r_i, \cdot, \cdot), Y_{lm} \rangle_{\mathbb{S}^2}]_{i=0,\ldots,2B-1};$$

  end

end

Rechenkomplexität: $\mathcal{O}(B^4)$ (seminaive FSFT, adjungierter Clenshaw-Algorithmus), $\mathcal{O}(B^3 \log^2 B)$ (mit FDPT)

Speicherkomplexität: $\mathcal{O}(B^3)$

Satz 2.2.26. *Die Matrix H_B in Definition 2.2.1 besitzt die Faktorisierung*

$$H_B \;=\; S_B \cdot \underbrace{\begin{bmatrix} \tilde{R}_0 & & \\ & \ddots & \\ & & \tilde{R}_{B-1} \end{bmatrix}}_{B \text{ Blöcke (s. u.)}} \cdot T_B \cdot \underbrace{\{E_{2B} \otimes Y_B\}}_{2B \text{ Blöcke der Größe } B^2 \times (2B)^2},$$

wobei die Matrizen

$$\tilde{R}_l \;:=\; \underbrace{E_{2l+1} \otimes R_l^{\mathrm{T}}}_{2l+1 \text{ Blöcke der Größe } (B-l)\times 2B}, \qquad l < B,$$

selbst wieder eine Blockstruktur besitzen.

Die radiale und sphärische Teiltransformation sind hierbei als einzelne Blockmatrizen strukturell gut erkennbar. Die Matrizen Y_B und R_l^{T} können nun wie in Abschnitt 2.2.4 bzw. 2.2.5 beschrieben selbst wieder faktorisiert werden: Eine Faktorisierung von Y_B ist durch Satz 2.2.18 gegeben; bei Verwendung des adjungierten Clenshaw-Algorithmus für den radialen Teil ergibt sich die Faktorisierung von R_l^{T} jeweils aus Folgerung 2.2.23. Die oben beschriebene FSGLFT ist in Pseudocode als Algorithmus 2.8 dargestellt. Die abschließende Folgerung aus Satz 2.2.26, die wir unter Berücksichtigung der Rechenregeln für das Kronecker-Produkt durch Umkehren der Produktreihenfolge und hermitesches Transponieren der einzelnen Faktoren erhalten, zeigt uns mit Lemma 2.2.2, dass wir stets auch über eine schnelle inverse Transformation verfügen (iFSGLFT, Alg. 2.9).

Algorithmus 2.9. iFSGLFT

Eingabe: SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$, $|m| \leq l < n \leq B$, einer bandbegrenzten Funktion mit Bandbreite B

Ausgabe: Funktionswerte $f(r_i, \vartheta_j, \varphi_k)$; $i, j, k = 0, \ldots, 2B-1$

for $l = 0$ **to** $B-1$ **do**
  for $m = -l$ **to** l **do**
    Berechne mittels Clenshaw-Algorithmus oder adjungierter FDPT
    $$[\langle f(r_i, \cdot, \cdot), Y_{lm} \rangle_{\mathbb{S}^2}]_{i=0,\ldots,2B-1} \;=\; R_l\,[\hat{f}_{nlm}]_{n=l+1,\ldots,B};$$
  end
end

for $i = 0$ **to** $2B-1$ **do**
  Berechne $[f(r_i, \vartheta_{j(\tau)}, \varphi_{k(\tau)})]_{\tau=0,\ldots,(2B)^2-1} = Y_B^{\mathrm{H}}\,[\langle f(r_i, \cdot, \cdot), Y_{lm(\sigma)} \rangle_{\mathbb{S}^2}]_{\sigma=0,\ldots,B^2-1}$ mittels iFSFT, mit τ und σ wie in Abschnitt 2.2.4;
end

Rechenkomplexität: $\mathcal{O}(B^4)$ (seminaive iFSFT, Clenshaw-Algorithmus), $\mathcal{O}(B^3 \log^2 B)$ (mit adjungierter FDPT)

Speicherkomplexität: $\mathcal{O}(B^3)$

Folgerung 2.2.27. *Die Matrix H_B^{H} besitzt die Faktorisierung*

$$H_B^{\mathrm{H}} \;=\; \left\{E_{2B} \otimes Y_B^{\mathrm{H}}\right\} \cdot T_B^{\mathrm{T}} \cdot \begin{bmatrix} \tilde{R}_0^{\mathrm{T}} & & \\ & \ddots & \\ & & \tilde{R}_{B-1}^{\mathrm{T}} \end{bmatrix} \cdot S_B^{\mathrm{T}}.$$

Auch hier sind der radiale und der sphärische Teil strukturell gut voneinander unterscheidbar. Die Matrizen Y_B^{H} und R_l können nun selbst wieder faktorisiert werden (s. Flg. 2.2.19 bzw. Satz 2.2.21).

2.3 Numerische Tests

Die naive DSGLFT (s. S. 39 unten) und deren Inverse, sowie die FSGLFT (Alg. 2.8) und iFSGLFT (Alg. 2.9) wurden in der Programmiersprache C++ implementiert[6]. Für die radiale Teiltransformation der FSGLFT und iFSGLFT wurde der adjungierte Clenshaw-Algorithmus (Alg. 2.7) bzw. der Clenshaw-Algorithmus (Alg. 2.6) benutzt. Für die sphärische Teiltransformation wurde die seminaive FSFT (Algorithmen 2.2 und 2.4) bzw. iFSFT (Algorithmen 2.3 und 2.5) des Softwarepakets *SpharmonicKit*[7] verwendet (Vers. 2.7).

[6] Die schnellen Algorithmen stehen unter `https://github.com/cwuelker/SGLPack` zur freien Verfügung (GNU General Public License, Vers. 3.0).

[7] Siehe `http://www.cs.dartmouth.edu/~geelong/sphere`

Die Abtastradien r_i und radialen Quadraturgewichte α_i aus dem SGL-Abtasttheorem (Satz 2.1.12) wurden in *Mathematica* (Wolfram Research, Vers. 10) mit einer hohen Genauigkeit vorberechnet und mit 100 bzw. 1000 Nachkommastellen gespeichert. Die sphärischen Quadraturgewichte β_j wurden mit hoher Genauigkeit vorberechnet und mit 100 Nachkommastellen gespeichert; diese wurden nur für die naive DSGLFT benötigt.

Die Testrechnungen wurden auf einem x86-64-System mit 3.4 GHz durchgeführt (CPU: Intel Core i7-3770). Es wurde keine Parallelisierung vorgenommen. Der größte Teil der Berechnungen wurde mit doppelter Genauigkeit (`double`) durchgeführt; der Clenshaw-Algorithmus und dessen Adjungierter wurden mit erweiterter doppelter Genauigkeit (`long double`, 80 Bit, davon 64 Bit Mantisse) durchgeführt.

Im Testlauf wurde über die Bandbreiten $B = 2^k$, $k = 3, \ldots, 8$, iteriert. Für jede Bandbreite wurden zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$, $|m| \leq l < n \leq B$, generiert, wobei der Real- und Imaginärteil jeweils gleichverteilt zwischen -1 und 1 lag. Aus diesen Testdaten wurden dann mittels iFSGLFT und naiver iDSGLFT jeweils die Funktionswerte $f(r_i, \vartheta_j, \varphi_k)$; $i, j, k = 0, \ldots, 2B-1$, berechnet. Anschließend wurden mittels FSGLFT und naiver DSGLFT jeweils die Fourier-Koeffizienten $\hat{f}^{\circ}_{nlm}$, $|m| \leq l < n \leq B$, rekonstruiert. Es wurde die Laufzeit gemessen und der *maximale absolute* und *maximale relative Rekonstruktionsfehler*

$$\max_{|m| \leq l < n \leq B} |\hat{f}_{nlm} - \hat{f}^{\circ}_{nlm}| \qquad \text{bzw.} \qquad \max_{|m| \leq l < n \leq B} \frac{|\hat{f}_{nlm} - \hat{f}^{\circ}_{nlm}|}{|\hat{f}_{nlm}|} \tag{2.38}$$

berechnet. Das obige Vorgehen wurde zehn mal wiederholt und so die durchschnittliche Laufzeit und der durchschnittliche Rekonstruktionsfehler bestimmt.

Tabelle 2.1 zeigt die Ergebnisse der Fehlermessung. In Tabelle 2.2 sind die Ergebnisse der Laufzeitmessung aufgelistet. Bei allen Bandbreiten war die Laufzeit der schnellen Transformationen erheblich kürzer als bei den naiven Algorithmen. Bei den Bandbreiten $B = 64, 128$ und 256 mussten die naiven Algorithmen aufgrund der langen Laufzeit sogar abgebrochen werden. Für die in der Praxis derzeit relevanteste Bandbreite $B = 32$ benötigt eine schnelle Hin- und Rücktransformation deutlich weniger als eine Sekunde, gegenüber etwas mehr als eineinhalb Stunden bei den naiven Algorithmen. Bei der größten behandelten Bandbreite $B = 256$ beträgt die Laufzeit der schnellen Algorithmen insgesamt noch weniger als zehn Minuten. Hinsichtlich des gemessenen Transformationsfehlers lässt sich festhalten, dass dieser unter Berücksichtigung der Maschinengenauigkeit sehr klein ist. Zu bemerken ist, dass der Fehler der schnellen Algorithmen signifikant größer ist, als der der naiven Algorithmen. Dieser Effekt wurde in [Prestin und Wülker, 2017, Abschn. 4] nicht beobachtet; dort war der Fehler der schnellen Algorithmen umgekehrt kleiner als der der naiven Algorithmen (dies ist dadurch zu erklären, dass die geringere Anzahl an Rechenschritten bei den schnellen Algorithmen in einem kleineren Rundungsfehler resultiert). Eine genaue Überprüfung ergab, dass der größere Fehler im hier beschriebenen Test von der speziellen Implementierung der seminaiven FLT im Softwarepaket SpharmonicKit verursacht wurde. Nichtsdestoweniger belegen die Ergebnisse die Praxistauglichkeit der vorgestellten FSGLFTs eindeutig. Die entwickelten schnellen Algorithmen erlauben es zudem, Problemgrößen zu behandeln, deren Behandlung vormals ganz undenkbar war.

B	iDSGLFT und DSGLFT (naiv)	iFSGLFT und FSGLFT
8	$(6{,}86 \pm 1{,}26)\,\mathrm{E}\,{-15}$	$(1{,}42 \pm 0{,}07)\,\mathrm{E}\,{-12}$
	$(2{,}45 \pm 2{,}77)\,\mathrm{E}\,{-14}$	$(6{,}17 \pm 2{,}60)\,\mathrm{E}\,{-12}$
16	$(2{,}44 \pm 0{,}29)\,\mathrm{E}\,{-14}$	$(4{,}75 \pm 0{,}54)\,\mathrm{E}\,{-12}$
	$(1{,}96 \pm 1{,}74)\,\mathrm{E}\,{-13}$	$(4{,}25 \pm 1{,}91)\,\mathrm{E}\,{-11}$
32	$(7{,}63 \pm 0{,}77)\,\mathrm{E}\,{-14}$	$(1{,}37 \pm 0{,}14)\,\mathrm{E}\,{-11}$
	$(1{,}04 \pm 1{,}12)\,\mathrm{E}\,{-12}$	$(4{,}98 \pm 4{,}21)\,\mathrm{E}\,{-10}$
64	-	$(5{,}32 \pm 0{,}65)\,\mathrm{E}\,{-11}$
	-	$(3{,}66 \pm 3{,}98)\,\mathrm{E}\,{-09}$
128	-	$(1{,}99 \pm 0{,}08)\,\mathrm{E}\,{-10}$
	-	$(3{,}18 \pm 1{,}91)\,\mathrm{E}\,{-08}$
256	-	$(7{,}55 \pm 0{,}67)\,\mathrm{E}\,{-10}$
	-	$(2{,}38 \pm 1{,}40)\,\mathrm{E}\,{-07}$

Tabelle 2.1. **(Obere Zeile)** maximaler absoluter und **(untere Zeile)** relativer Rekonstruktionsfehler einer naiven iDSGLFT bzw. iFSGLFT und anschließenden naiven DSGLFT bzw. FSGLFT in Abhängigkeit von der Bandbreite B.

B	iDSGLFT und DSGLFT (naiv)	iFSGLFT und FSGLFT
8	$6{,}54\,\mathrm{E}\,{-1}\,\mathrm{s}$	$2{,}05\,\mathrm{E}\,{-3}\,\mathrm{s}$
16	$5{,}30\,\mathrm{E}\,{+1}\,\mathrm{s}$	$1{,}46\,\mathrm{E}\,{-2}\,\mathrm{s}$
32	$5{,}63\,\mathrm{E}\,{+3}\,\mathrm{s}$	$1{,}62\,\mathrm{E}\,{-1}\,\mathrm{s}$
64	-	$2{,}13\,\mathrm{E}\,{+0}\,\mathrm{s}$
128	-	$3{,}12\,\mathrm{E}\,{+1}\,\mathrm{s}$
256	-	$5{,}22\,\mathrm{E}\,{+2}\,\mathrm{s}$

Tabelle 2.2. Durchschnittliche Laufzeit einer naiven iDSGLFT bzw. iFSGLFT und anschließenden naiven DSGLFT bzw. FSGLFT in Abhängigkeit von der Bandbreite B.

3 Schnelle SGL-Fourier-Transformationen für gestreute Daten

Es sind viele Anwendungen denkbar, bei denen die gegebenen Abtastwerte $f(x_i)$ einer bandbegrenzten Funktion f nicht auf dem entsprechenden durch Satz 2.1.12 gegebenen Gitter liegen. So kann es sein, dass die Punkte x_i ein kartesisches[1] Gitter in $\mathbb{R}^3$ bilden, oder auf ganz andere Art *gestreut* (engl.: *scattered*) sind. In solchen Fällen ist eine Berechnung der SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ von f mittels der im letzten Kapitel vorgestellten FSGLFT nach Interpolation der Daten $f(x_i)$ zwar denkbar, jedoch in der Regel nicht ratsam. Um auch solcher Situationen Herr zu werden, widmen wir uns in diesem Kapitel der Entwicklung *schneller SGL-Fourier-Transformationen für gestreute Daten* (NFSGLFTs für *nonlattice fast SGL Fourier transforms*). Im Falle der klassischen FFT wurde von Potts et al. die *schnelle nichtäquidistante Fourier-Transformation* (NFFT für *nonequispaced fast Fourier transform*) entwickelt (s. [Potts et al., 2001; Potts, 2003; Kunis, 2006]). Die NFFT ist heute in der Praxis fest etabliert und hat sich in vielen Anwendungen als äußerst nützlich erwiesen. Durch einen direkten Einsatz der NFFT ergeben sich auch im Fall der Sphäre $\mathbb{S}^2$ und der Rotationsgruppe SO(3) schnelle Fourier-Transformationen für gestreute Daten (s. [Kunis und Potts, 2003] bzw. [Potts et al., 2009]). Wir wenden die NFFT in dieser Arbeit nicht nur mit der gleichen Strategie innerhalb einer ersten Klasse von NFSGFLTs an (Variante A, Abschn. 3.2), sondern übertragen das Prinzip der NFFT auch direkt auf den SGL-Fall, woraus sich eine weitere Klasse von NFSGLFTs ergibt (Variante B, Abschn. 3.3). Bei der Variante A kommen neben der NFFT die bereits besprochene adjungierte FLT, der Clenshaw-Algorithmus und die DCT zum Einsatz (vgl. Abb. E.1). Bei der Variante B benutzen wir als entscheidende Hilfsmittel die in Abschnitt 3.3.1 besprochene schnelle FFT auf SO(3) und die in Abschnitt 3.3.2 behandelte schnelle Gauß-Transformation. Die Herleitung beider Klassen von NFSGLFTs beginnen wir wie bei der NFFT jeweils mit der inversen Transformation, der folgende Definition zugrunde liegt:

Definition 3.1 (iNDSGLFT). Es seien Punkte $x_i \in \mathbb{R}^3$, $i = 0, \ldots, M-1$, und eine Bandbreite B vorgegeben. Unter Zuhilfenahme von (2.20) setzen wir

$$\tilde{H} := \Big[H_{n(\mu),l(\mu),m(\mu)}(x_i)\Big]_{\substack{i=0,\ldots,M-1\\ \mu=0,\ldots,B(B+1)(2B+1)/6-1}} \in \mathbb{C}^{M\times B(B+1)(2B+1)/6}.$$

Die Abbildung $\tilde{H} = \tilde{H}(B; x_0, \ldots, x_{M-1}) : \mathbb{C}^{B(B+1)(2B+1)/6} \to \mathbb{C}^M$ heißt *inverse diskrete SGL-Fourier-Transformation für gestreute Daten* (iNDSGLFT für *inverse nonlattice discrete SGL Fourier transform*).

[1] nach René Descartes (⋆1596 in La Haye en Touraine, Frankreich, †1650 in Stockholm, Schweden)

Die direkte Multiplikation eines Vektors mit der Matrix $\tilde{H}$ benötigt offenbar $\mathcal{O}(MB^3)$ Operationen. Die in diesem Kapitel vorgestellten Algorithmen haben demgegenüber eine Komplexität von $\mathcal{O}(\Phi(B) + \Psi(B)M)$, wobei Φ und Ψ jeweils von der Bandbreite B abhängige Funktionen sind: Bei der ersten Variante erhalten wir formal

$$\text{(iNFSGLFT-A)} \qquad \begin{cases} \Phi(B) \;=\; \Phi(B,\sigma) \;=\; B^4 + (\sigma B)^3 \log(\sigma B), \\ \Psi(B) \;=\; q^3, \end{cases}$$

wobei σ den Überabtastungsfaktor und q den Abschneideparameter der hier zum Einsatz kommenden iNFFT bezeichnen (vgl. Abschn. 3.1). Bei Verwendung einer adjungierten FPDT anstatt des Clenshaw-Algorithmus kann $\Phi(B) = B^3 \log^2 B + (\sigma B)^3 \log(\sigma B)$ erreicht werden. Bei der zweiten Variante erhalten wir

$$\text{(iNFSGLFT-B)} \qquad \begin{cases} \Phi(B) \;=\; \Phi(B,\tau,n,\sigma,q) \;=\; (\tau B)^4 + (\sigma n)^3 \log(\sigma n) + (q\tau B)^3, \\ \Psi(B) \;=\; q^3, \end{cases}$$

wobei τ den Überabtastungsfaktor der iNFSGLFT-B bezeichnet (vgl. Abschn. 3.3.3), während n der Abschneideparameter der von uns verwendeten schnellen Gauß-Transformation ist und σ und q den Überabtastungsfaktor bzw. Abschneideparameter der darin auftretenden iNFFT und ihrer Adjungierten bezeichnen (Abschn. 3.3.2). Man beachte, dass die angegebenen Funktionen Φ und Ψ stellvertretend für ganze Klassen zu verstehen sind. Da die obigen Klassen von iNFSGLFTs jeweils durch eine Faktorisierung der Matrix $\tilde{H}$ in Definition 3.1 charakterisiert werden, haben wir stets auch eine schnelle adjungierte Transformation mit gleicher Komplexität zur Hand. Daraus konstruieren wir nach dem Vorbild der NFFT die schnelle Vorwärtstransformation, d. h. die jeweilige NFSGLFT.

Es ist wichtig zu bemerken, dass es sich bei allen vorgestellten Algorithmen wie im Fall der NFFT um *approximative* Algorithmen handelt, deren Ergebnissen selbst in exakter Arithmetik ein Fehler inhärent ist. Daher spielt bei unseren Untersuchungen der Zusammenhang zwischen dem Fehler und den Funktionen Φ und Ψ eine besonders wichtige Rolle. So werden wir im Fall der iNFSGLFT-A zeigen, dass $\sigma =$ konst. und q auf geeignete Art und Weise als $q = \mathrm{o}(B)$ gewählt werden können, um den Fehler bei wachsender Bandbreite B zu kontrollieren.

3.1 Schnelle nichtäquidistante Fourier-Transformationen (NFFT)

In diesem Abschnitt wiederholen wir die Funktionsweise der NFFT von Potts et al. aufgrund ihrer besonderen Bedeutung innerhalb dieses Kapitels. Wir orientieren uns dabei an der Darstellung in [Potts, 2003]. Darüber hinaus leiten wir in Satz 3.1.9 als wichtiges Resultat auch eine Fehlerabschätzung für die d-dimensionale iNFFT her, die wir später brauchen. Obwohl es sich bei der NFFT im Grunde um eine gitterfreie Transformation handelt, kommen im Hintergrund spezielle Gitter vor, die gleichzeitig die Indexbereiche der Fourier-Koeffizienten beschreiben. Wir beginnen mit der Definition dieser Gitter.

Definition 3.1.1. Es seien $d, n \in \mathbb{N}$, n gerade. Wir setzen

$$I_n^d \coloneqq \{z = [\zeta_0, \ldots, \zeta_{d-1}] \in \mathbb{Z}^d : -n/2 \leq \zeta_i < n/2,\ i = 0, \ldots, d-1\}.$$

Definition 3.1.2. Für $d \in \mathbb{N}$ definieren wir den *d-dimensionalen Torus* als Faktorgruppe

$$\mathbb{T}^d \coloneqq \mathbb{R}^d / 2\pi\mathbb{Z}^d.$$

Bemerkung 3.1.3. Für jede beliebige Nebenklasse $[t] \in \mathbb{T}^d$ kann der Vertreter t natürlich aus $[0, 2\pi)^d$ gewählt werden. Daher identifizieren wir den Torus $\mathbb{T}^d$ im Folgenden der Einfachheit halber mit $[0, 2\pi)^d$.

Anders als im Fall der klassischen FFT beginnt die Herleitung der NFFT mit der inversen Transformation, d. h. mit der iNFFT. Es seien dazu die Fourier-Koeffizienten $\omega_k \in \mathbb{C}$, $k \in I_n^d$, eines d-dimensionalen trigonometrischen Polynoms

$$p(t) \coloneqq \sum\nolimits_{k \in I_n^d} \omega_k \, \mathrm{e}^{\mathrm{i}\langle k, t\rangle_2}, \qquad t \in \mathbb{T}^d, \tag{3.1}$$

vom Grad höchstens n gegeben (n gerade). Die iNFFT ist ein schneller Algorithmus zur Auswertung des trigonometrischen Polynoms p an beliebigen Punkten $t_0, \ldots, t_{m-1} \in \mathbb{T}^d$, $m \in \mathbb{N}$. Der Unterschied zur klassischen iFFT besteht darin, dass die Punkte $t_0, \ldots, t_{m-1}$ nicht notwendigerweise auf einem Gitter liegen müssen, und dass deren Anzahl m nicht fest vorgegeben ist.

Um das Problem in Matrix-Vektor-Schreibweise bringen zu können, linearisieren wir den Indexbereich der Fourier-Koeffizienten von p (vgl. [Potts, 2003, S. 11]): Wir identifizieren $k = [\kappa_0, \ldots, \kappa_{d-1}] \in I_n^d$ mit $\chi \in \{0, \ldots, n^d - 1\}$ vermöge des bijektiven Zusammenhangs

$$\chi = \left(\kappa_0 + \frac{n}{2}\right) + n\left(\kappa_1 + \frac{n}{2}\right) + \cdots + n^{d-1}\left(\kappa_{d-1} + \frac{n}{2}\right). \tag{3.2}$$

Dies erlaubt es uns, k als Funktion von χ aufzufassen,

$$k(\chi) = \left[\left\lfloor \frac{\chi - (\chi \bmod n^{j+1})}{n^j} \right\rfloor - \frac{n}{2}\right]_{j=0,\ldots,d-1}.$$

Nun lässt sich die Berechnung der Polynomwerte $p(t_0), \ldots, p(t_{m-1})$ als Auswertung des Matrix-Vektor-Produkts

$$\Big[p(t_i)\Big]_{i=0,\ldots,m-1} = \underbrace{\Big[\mathrm{e}^{\mathrm{i}\langle k(\chi),\, t_i\rangle_2}\Big]_{\substack{i=0,\ldots,m-1 \\ \chi=0,\ldots,n^d-1}}}_{=:\, N} \cdot \Big[\omega_{k(\chi)}\Big]_{\chi=0,\ldots,n^d-1} \tag{3.3}$$

formulieren.

Definition 3.1.4 (iNDFT). Die Abbildung $N = N(d; n; t_0, \ldots, t_{m-1}) : \mathbb{C}^{n^d} \to \mathbb{C}^m$ wird als *inverse nichtäquidistante diskrete Fourier-Transformation* (iNDFT für *inverse nonequispaced discrete Fourier transform*) bezeichnet.

Eine direkte Auswertung des Produkts (3.3) benötigt offenbar $\mathcal{O}(mn^d)$ Operationen, entsprechend der Größe der Matrix N. Die heute fest etablierte iNFFT von Potts et al. ist ein approximativer Algorithmus mit einer geringeren Komplexität. Sie basiert auf

folgender Idee: Man bestimme einen Approximanten s an p, der sich im „Ortsraum“ schnell auswerten lässt, sodass p und s im „Frequenzraum“ möglichst genau übereinstimmen. Letztere Forderung ist bestmöglich erfüllt, wenn

$$(2\pi)^{-d} \int_{\mathbb{T}^d} s(t)\,\mathrm{e}^{-\mathrm{i}\langle k,t\rangle_2}\,\mathrm{d}t \;=\; \omega_k, \qquad k \in \mathbb{Z}^d, \tag{3.4}$$

gilt, wobei wir $\omega_k := 0$ für $k \in \mathbb{Z}^d \setminus I_n^d$ setzen.

Es sei nun $\varphi \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ eine Ansatzfunktion, die sich an einem beliebigen Punkt in $\mathcal{O}(1)$ Schritten auswerten lässt. Es wird vorausgesetzt, dass die 2π-periodisierte Version

$$\tilde{\varphi}(t) \;:=\; \sum\nolimits_{z\in\mathbb{Z}^d} \varphi(t - 2\pi z), \qquad t \in \mathbb{T}^d,$$

eine gleichmäßig konvergente Fourier-Reihe besitzt. Diese ist gegeben durch

$$\tilde{\varphi}(t) \;=\; \sum\nolimits_{k\in\mathbb{Z}^d} \tilde{\varphi}_k\,\mathrm{e}^{\mathrm{i}\langle k,t\rangle_2}, \qquad t \in \mathbb{T}^d, \tag{3.5}$$

mit den Fourier-Koeffizienten

$$\tilde{\varphi}_k \;=\; (2\pi)^{-d} \int_{\mathbb{T}^d} \tilde{\varphi}(t)\,\mathrm{e}^{-\mathrm{i}\langle k,t\rangle_2}\,\mathrm{d}t, \qquad k \in \mathbb{Z}^d.$$

Die Gleichheit in (3.5) ist im L^2-Sinne zu verstehen; dies erwähnen wir bei den folgenden Fourier-Reihen nicht mehr. Die Fourier-Koeffizienten $\tilde{\varphi}_k$ hängen nach der Poissonschen[2] Summenformel direkt mit der Fourier-Transformierten der Ansatzfunktion φ zusammen:

$$\tilde{\varphi}_k \;=\; (2\pi)^{-d} \int_{\mathbb{R}^d} \varphi(x)\,\mathrm{e}^{-\mathrm{i}\langle k,x\rangle_2}\,\mathrm{d}x, \qquad k \in \mathbb{Z}^d. \tag{3.6}$$

Als ersten Approximanten $\tilde{p}$ an das trigonometrische Polynom p wird nun eine Linearkombination von Translaten der 2π-periodisierten Ansatzfunktion $\tilde{\varphi}$ gewählt,

$$\tilde{p}(t) \;:=\; \sum\nolimits_{l\in I_{\sigma n}^d} \alpha_l\,\tilde{\varphi}\left(t - \frac{2\pi}{\sigma n}\,l\right), \qquad t \in \mathbb{T}^d, \tag{3.7}$$

wobei $\sigma \in \mathbb{N}$ ein *Überabtastungsfaktor* ist, und die Skalierungskoeffizienten α_l der Translate so zu bestimmen sind, dass (3.4) näherungsweise erfüllt ist.

Die Entwicklung von $\tilde{p}$ in eine Fourier-Reihe liefert

$$\begin{aligned} \tilde{p}(t) \;&=\; \sum\nolimits_{k\in\mathbb{Z}^d} \beta_k\,\tilde{\varphi}_k\,\mathrm{e}^{\mathrm{i}\langle k,t\rangle_2} \\ &=\; \sum\nolimits_{k\in I_{\sigma n}^d} \beta_k\,\tilde{\varphi}_k\,\mathrm{e}^{\mathrm{i}\langle k,t\rangle_2} \;+\; \sum\nolimits_{z\in\mathbb{Z}^d\setminus\{0\}} \sum\nolimits_{k\in I_{\sigma n}^d} \beta_k\,\tilde{\varphi}_{k+\sigma n z}\,\mathrm{e}^{\mathrm{i}\langle k+\sigma n z,t\rangle_2} \end{aligned} \tag{3.8}$$

mit den σn-periodischen Koeffizienten

$$\beta_k \;=\; \sum\nolimits_{l\in I_{\sigma n}^d} \alpha_l\,\mathrm{e}^{-2\pi\mathrm{i}\langle k,l\rangle_2/\sigma n}. \tag{3.9}$$

Bemerkung 3.1.5. Man beachte hierbei das einfache Spektralverhalten der periodisierten Ansatzfunktion $\tilde{\varphi}$ unter Translationen, das in der ersten Gleichheit in (3.8) zum Ausdruck kommt. Dies ist ein besonderer Vorteil der klassischen Fourier-Basis auf $\mathbb{T}^d$. Eine der Hauptschwierigkeiten bei der Übertragung auf den SGL-Fall in Abschnitt 3.3.3 liegt in dem vergleichsweise komplizierten Spektralverhalten der Ansatzfunktion unter Translationen begründet (vgl. dazu auch Abschn. 1.6).

[2] Siméon Denis Poisson (⋆1781 in Pithiviers im Département Loiret, Frankreich, †1840 in Paris, ebd.)

Unter der Annahme, dass der Betrag der Fourier-Koeffizienten $\tilde{\varphi}_{k+\sigma n z}$ mit $k \in I_{\sigma n}^d$ und $z \in \mathbb{Z}^d \setminus \{0\}$ sehr klein ist, kann die Doppelsumme auf der rechten Seite von (3.8) vernachlässigt werden. Geht man zusätzlich davon aus, dass der Betrag der Fourier-Koeffizienten $\tilde{\varphi}_k$ für $k \in I_n^d$ jeweils ungleich Null ist, motiviert ein Vergleich der ersten Summe auf der rechten Seite von (3.8) mit der rechten Seite von (3.1) die Wahl

$$\beta_k := \begin{cases} \omega_k / \tilde{\varphi}_k, & \text{wenn } k \in I_n^d, \\ 0, & \text{wenn } k \in I_{\sigma n}^d \setminus I_n^d, \end{cases}$$

sodass die Gleichheit in (3.4) für alle $k \in I_{\sigma n}^d$ gilt.

Nun werden die Skalierungskoeffizienten α_l in (3.7) aus den Koeffizienten β_k bestimmt. Dazu bringen wir (3.9) in Matrix-Vektor-Schreibweise:

$$\left[\beta_{k(\chi)}\right]_{\chi=0,\ldots,(\sigma n)^d-1} = \underbrace{\left[\mathrm{e}^{-2\pi \mathrm{i} \langle k(\chi),\, l(\nu)\rangle_2 / \sigma n}\right]_{\chi,\nu=0,\ldots,(\sigma n)^d-1}}_{=:\, F} \cdot \left[\alpha_{l(\nu)}\right]_{\nu=0,\ldots,(\sigma n)^d-1}. \tag{3.10}$$

Die Matrix $F = F(d, n, \sigma)$ ist eine Fourier-Matrix in d Dimensionen (mit spezieller Anordnung der Indizes). Ihre Inverse ist durch $(\sigma n)^{-d} F^{\mathrm{H}}$ gegeben (dies folgt unmittelbar aus Lemma 2.2.4). Insbesondere lassen sich also mithilfe der d-dimensionalen iFFT die Skalierungskoeffizienten α_l in $\mathcal{O}((\sigma n)^d \log(\sigma n))$ Schritten aus den Koeffizienten β_k berechnen. Um einen besonders schnellen Algorithmus zu erhalten, sollte der Überabtastungsfaktor σ so gewählt werden, dass σn eine Zweierpotenz ist.

Bemerkung 3.1.6. Auch dieser Schritt gestaltet sich im SGL-Fall komplizierter, da wir es an dieser Stelle nicht mehr mit einer klassischen Fourier-Matrix zu tun haben.

In einem letzten Schritt wird der Approximant $\tilde{p}$ durch einen weiteren Approximanten s angenähert, der sich im Ortsraum schnell auswerten lässt. Unter der Annahme, dass die Ansatzfunktion φ im Ortsraum schnell abklingt, kann φ durch eine Funktion ψ ersetzt werden, deren Träger vollständig in einem Quader $[-2\pi q/\sigma n, 2\pi q/\sigma n]^d$ enthalten ist ($q \in \mathbb{N}$, $q < \sigma n$). Genauer wird

$$\psi(x) := \begin{cases} \varphi(x) & \text{für } x \in [-2\pi q/\sigma n, 2\pi q/\sigma n]^d, \\ 0 & \text{sonst}, \end{cases} \tag{3.11}$$

gewählt und wieder die 2π-periodische Funktion $\tilde{\psi} := \sum_{z \in \mathbb{Z}^d} \psi(\cdot - 2\pi z)$ gebildet. Die Zahl q wird *Abschneideparameter* genannt. Als Approximant an $\tilde{p}$ und damit an p bietet sich

$$s(t) := \sum\nolimits_{l \in I_{\sigma n}^d} \alpha_l \, \tilde{\psi}\left(t - \frac{2\pi}{\sigma n} l\right), \qquad t \in \mathbb{T}^d,$$

an (vgl. Gl. 3.7). Gemäß (3.11) kann der Indexbereich $I_{\sigma n}^d$ für festes $t = [\tau_0, \ldots, \tau_{d-1}]$ hier auf den Bereich $\{l = [\iota_0, \ldots, \iota_{d-1}] \in I_{\sigma n}^d : \sigma n \tau_i / 2\pi - q \leq \iota_i \leq \sigma n \tau_i / 2\pi + q,\ i = 0, \ldots, d-1\}$ eingeschränkt werden, wie man sich geometrisch leicht überlegt. Dies erlaubt eine Auswertung von s in $\mathcal{O}(q^d)$ anstatt $\mathcal{O}((\sigma n)^d)$ Schritten. Insgesamt ergibt sich so eine Rechenkomplexität von $\mathcal{O}((\sigma n)^d \log(\sigma n) + m q^d)$ für die oben beschriebene iNFFT. Die Speicherkomplexität ist $\mathcal{O}((\sigma n)^d)$.

Der obige Algorithmus korrespondiert mit einer näherungsweisen Faktorisierung der Matrix N in (3.3). Wir verweisen hier auf [Potts, 2003, Abschn. 1.2]. Es folgt daraus insbesondere, dass wir auch über eine *adjungierte* iNFFT, d. h. einen schnellen Algorithmus für die Multiplikation mit der Matrix N^{H}, verfügen. Diese hat die gleiche Rechen- und Speicherkomplexität wie die iNFFT.

In dieser Arbeit werden wir als Ansatzfunktion φ stets eine Gaußsche Ansatzfunktion verwenden (vgl. [Potts, 2003, Abschn. 1.3.2]):

Definition 3.1.7. Es seien $\sigma, n \in \mathbb{N}$, n gerade, sowie $\lambda > 0$. Wir definieren die Gaußsche Ansatzfunktion für die oben beschriebene iNFFT als

$$\varphi(x) \coloneqq \prod_{j=0}^{d-1} \phi(\xi_j), \qquad x = [\xi_0, \ldots, \xi_{d-1}] \in \mathbb{R}^d,$$

mit

$$\phi(\xi) \coloneqq \frac{1}{\sqrt{2\pi\lambda}} \left(\frac{\sigma n}{2\pi}\right) \mathrm{e}^{-(\sigma n/2\pi)^2 \xi^2/2\lambda}.$$

Wir können einen geschlossenen Ausdruck für die Fourier-Transformierte der Gaußschen Ansatzfunktion an Stellen in $\mathbb{Z}^d$ angeben, indem wir [Potts, 2003, Gl. 1.34] auf den d-dimensionalen Fall verallgemeinern. Dies ermöglicht uns nach (3.6) die Verwendung der Gaußschen Ansatzfunktion in der iNFFT.

Lemma 3.1.8. *Es seien σ, n und λ wie in Definition 3.1.7. Dann gilt für die Fourier-Transformierte der Gaußschen Ansatzfunktion φ die Formel*

$$\int_{\mathbb{R}^d} \varphi(x) \mathrm{e}^{-\mathrm{i}\langle k,x\rangle_2} \,\mathrm{d}x = \prod_{j=0}^{d-1} \phi_{\kappa_j}, \qquad k = [\kappa_0, \ldots, \kappa_{d-1}] \in \mathbb{Z}^d,$$

mit $\phi_\kappa \coloneqq \exp(-2\lambda(\pi\kappa/\sigma n)^2)$.

Wir wollen nun eine Abschätzung für den Fehler der iNFFT mit der Gaußschen Ansatzfunktion herleiten. Genauer untersuchen wir den *maximalen absoluten Fehler*

$$E_\infty = E_\infty(w; d, n, \sigma, q; t_0, \ldots, t_{m-1}) \coloneqq \max_{i \in \{0,\ldots,m-1\}} |p(t_i) - s(t_i)|$$

mit den gegebenen Fourier-Koeffizienten $w \coloneqq [\omega_{k(\chi)}]_{\chi=0,\ldots,n^d-1}$. Hierzu verallgemeinern wir [Potts, 2003, Satz 1.7] unter Verwendung von Lemma 3.1.8 auf den d-dimensionalen Fall (man beachte die Normierung der Gaußschen Ansatzfunktion bei Potts). Diese Abschätzung brauchen wir im Fall $d = 3$ in Abschnitt 3.2.2.

Satz 3.1.9. *Der maximale absolute Fehler der oben beschriebenen iNFFT mit der Gaußschen Ansatzfunktion und der speziellen Wahl $\lambda \coloneqq \frac{\sigma}{2\sigma-1}\frac{q}{\pi}$ lässt sich wie folgt abschätzen:*

$$E_\infty \leq \|w\|_1 \left((2^d - 1)\left(2 + \frac{1}{\pi q}\right)^d + \frac{3^d - 1}{q^{d/2}} \left(\sqrt{\frac{2\sigma - 1}{2\sigma}} + \frac{1}{2\pi}\sqrt{\frac{2\sigma}{2\sigma - 1}} \right)^d \right) \mathrm{e}^{-q\pi\left(1 - \frac{1}{2\sigma}\left(1 + \frac{d}{2\sigma-1}\right)\right)}.$$

Im Fall $d = 3$ ergibt sich für $\sigma \geq 2$ aus Monotoniegründen speziell

$$E_\infty \leq 136 \|w\|_1 \mathrm{e}^{-q\pi/2}.$$

Beweis. Aus der Dreiecksungleichung folgt zunächst für beliebiges $i \in \{0, \dots, m-1\}$

$$|p(t_i) - s(t_i)| \;\le\; \underbrace{|p(t_i) - \tilde{p}(t_i)|}_{=:\, E_{\ddot{\mathrm{U}}}(t_i)} + \underbrace{|\tilde{p}(t_i) - s(t_i)|}_{=:\, E_{\mathrm{A}}(t_i)}.$$

Der *Überlappungsfehler* $E_{\ddot{\mathrm{U}}}(t_i)$, der durch das Abschneiden von $\tilde{p}$ im Frequenzraum entsteht, kann durch

$$E_{\ddot{\mathrm{U}}}(t_i) \;\le\; \|w\|_1 \max_{k \in I_n^d} \sum\nolimits_{z \in \mathbb{Z}^d \setminus \{0\}} \left| \frac{\tilde{\varphi}_{k+\sigma n z}}{\tilde{\varphi}_k} \right|$$

abgeschätzt werden [Potts, 2003, Gl. 1.23]. Auf der rechten Seite unterscheiden wir solche Summanden, bei denen genau eine der d Komponenten von z ungleich Null ist, von solchen, bei denen genau zwei der d Komponenten von z ungleich Null sind, usw. Es folgt mit (3.6) und Lemma 3.1.8, dass

$$E_{\ddot{\mathrm{U}}}(t_i) \;\le\; \|w\|_1 \sum_{j=1}^{d} \binom{d}{j} \Bigg(\underbrace{\max_{\kappa \in I_n^1} \sum\nolimits_{\zeta \in \mathbb{Z} \setminus \{0\}} \left| \frac{\phi_{\kappa+\sigma n \zeta}}{\phi_\kappa} \right|}_{=:\, L} \Bigg)^j.$$

Der Beweis von [Potts, 2003, Satz 1.7] zeigt

$$\begin{aligned} L \;=\; L(\sigma, n, q) \;&\le\; \mathrm{e}^{-2\pi^2\lambda(1-1/\sigma)} \left(1 + \frac{\sigma}{2\pi^2\lambda(2\sigma-1)} + \mathrm{e}^{-4\lambda\pi^2/\sigma} \left(1 + \frac{\sigma}{2\pi^2\lambda(2\sigma+1)} \right) \right) \\ &\le\; \mathrm{e}^{-q\pi\left(1 - \frac{1}{2\sigma}\left(1 + \frac{d}{2\sigma-1}\right)\right)} \left(1 + \frac{1}{2\pi q} + \mathrm{e}^{-4\pi q/(2\sigma-1)} \left(1 + \frac{1}{2\pi q} \frac{2\sigma-1}{2\sigma+1} \right) \right) \\ &\le\; \mathrm{e}^{-q\pi\left(1 - \frac{1}{2\sigma}\left(1 + \frac{d}{2\sigma-1}\right)\right)} \left(2 + \frac{1}{\pi q} \right), \end{aligned}$$

entsprechend der speziellen Wahl von λ. Es folgt mit der binomischen Formel

$$E_{\ddot{\mathrm{U}}}(t_i) \;\le\; \|w\|_1 \, (2^d - 1) \left(2 + \frac{1}{\pi q} \right)^d \mathrm{e}^{-q\pi\left(1 - \frac{1}{2\sigma}\left(1 + \frac{d}{2\sigma-1}\right)\right)}.$$

Der *Abschneidefehler* $E_{\mathrm{A}}(t_i)$, der durch das Abschneiden von φ im Ortsraum entsteht, kann mit der Supremumsnorm $\|\cdot\|_\infty$ durch

$$E_{\mathrm{A}}(t_i) \;\le\; \|w\|_1 \, (\sigma n)^{-d} \max_{k \in I_n^d} |\tilde{\varphi}_k|^{-1} \sum\nolimits_{l \in \mathbb{Z}^d \,:\, \left\| t_i + \frac{2\pi}{\sigma n} l \right\|_\infty \ge \frac{2\pi}{\sigma n} q} \left| \varphi\left(t_i + \frac{2\pi}{\sigma n} l \right) \right| \tag{3.12}$$

abgeschätzt werden [Potts, 2003, Gl. 1.27]. Es gilt nach (3.6) und Lemma 3.1.8

$$\max_{k \in I_n^d} |\tilde{\varphi}_k|^{-1} \;=\; (2\pi)^d \left(\max_{\kappa \in I_n^1} |\phi_\kappa|^{-1} \right)^d \;=\; (2\pi)^d \mathrm{e}^{\lambda(\pi/\sigma)^2 d/2}.$$

Es sei o. B. d. A. $t_i = [\tau_0, \dots, \tau_{d-1}] \in [0, 2\pi/\sigma n)^d$. Auf der rechten Seite von (3.12) unterscheiden wir Summanden mit $l = [\iota_j]_{j=0,\dots,d-1}$, bei denen die Bedingung $|\tau_j + 2\pi\iota_j/\sigma n| \ge 2\pi q/\sigma n$ für genau ein j von d erfüllt ist, für genau zwei j von d erfüllt ist, usw. Es gilt

$$\left| \phi\left(\tau_j + \frac{2\pi}{\sigma n} \iota \right) \right| \;\le\; \begin{cases} \left| \phi\left(\frac{2\pi}{\sigma n} \iota \right) \right|, & \text{wenn } \iota \ge 0, \\ \left| \phi\left(\frac{2\pi}{\sigma n} (\iota + 1) \right) \right|, & \text{wenn } \iota < 0. \end{cases}$$

Ähnlich wie oben ergibt sich damit

$$\begin{aligned} E_{\mathrm{A}}(t_i) &\leq \|w\|_1 \left(\frac{2\pi}{\sigma n}\right)^d \mathrm{e}^{\lambda(\pi/\sigma)^2 d/2} \sum_{j=1}^{d} \binom{d}{j} \left(\frac{1}{\sqrt{2\pi\lambda}} \left(\frac{\sigma n}{2\pi}\right)\right)^{d-j} \left(2 \sum\nolimits_{\iota \geq q} \left|\phi\left(\frac{2\pi}{\sigma n}\iota\right)\right|\right)^j \\ &\leq \|w\|_1 (2\pi\lambda)^{-d/2} \mathrm{e}^{\lambda(\pi/\sigma)^2 d/2} \sum_{j=1}^{d} \binom{d}{j} 2^j \Bigg(\underbrace{\sum\nolimits_{\iota \geq q} \mathrm{e}^{-\iota^2/2\lambda}}_{=:\, M}\Bigg)^j. \end{aligned}$$

Der Beweis von [Potts, 2003, Satz 1.7] zeigt

$$M = M(\sigma, q) \leq \left(1 + \frac{\lambda}{q}\right) \mathrm{e}^{-q^2/2\lambda}.$$

Es ergibt sich mit der speziellen Wahl von λ und der binomischen Formel

$$\begin{aligned} E_{\mathrm{A}}(t_i) &\leq \|w\|_1 (3^d - 1)(2\pi\lambda)^{-d/2} \left(1 + \frac{\lambda}{q}\right)^d \mathrm{e}^{\lambda(\pi/\sigma)^2 d/2 - q^2/2\lambda} \\ &= \|w\|_1 \frac{3^d - 1}{q^{d/2}} \left(\sqrt{\frac{2\sigma - 1}{2\sigma}} + \frac{1}{2\pi}\sqrt{\frac{2\sigma}{2\sigma - 1}}\right)^d \mathrm{e}^{-q\pi\left(1 - \frac{1}{2\sigma}\left(1 + \frac{d}{2\sigma - 1}\right)\right)}. \end{aligned}$$

Damit ist die Aussage des Lemmas bewiesen. □

Bemerkung 3.1.10. Die obige Fehlerabschätzung besitzt mit den entsprechenden Änderungen auch für die adjungierte iNFFT Gültigkeit, wie man mithilfe von [Werner, 2011, Satz 3.4.2] und der Abschätzung $\|\cdot\|_\infty \leq \|\cdot\|_1$ zeigen kann (vgl. auch [Potts, 2003, S. 19]).

Man beachte, dass die Schranke für den Fehler E_∞ zwar nicht direkt von n abhängt, dass aber $q < \sigma n$ gewählt werden muss. Bei geeigneter Wahl des Überabtastungsfaktors σ fällt der Fehler *exponentiell* ab, wenn wir den Abschneideparameter q vergrößern.

Abschließend beschreiben wir noch, wie sich aus der iNFFT und ihrer Adjungierten die NFFT herleiten lässt, d. h. die schnelle Vorwärtstransformation, die aus den Polynomwerten $p(t_i)$ die Fourier-Koeffizienten w_k berechnet. Dazu gibt es verschiedene Möglichkeiten, von denen wir hier nur eine festhalten. Für eine ausführliche Diskussion und Weiterentwicklungen des hier besprochenen Verfahrens verweisen wir auf [Kunis, 2006, Kap. 5] (vgl. auch [Potts, 2003, Abschn. 1.7]).

Es seien $N \in \mathbb{C}^{m \times n^d}$ und $f := [p(t_i)]_{i=0,\ldots,m-1} \in \mathbb{C}^m$ wie in (3.3). Wir suchen nach einem Fourier-Vektor $\tilde{f} \in \mathbb{C}^{n^d}$, der das lineare Gleichungssystem

$$N\tilde{f} = f \tag{3.13}$$

löst. Unter den obigen Voraussetzungen besitzt dieses Gleichungssystem mindestens eine Lösung. Um eine solche zu berechnen, formulieren wir (3.13) als *Kleinste-Quadrate-Problem*: Man bestimme $\tilde{f}$ so, dass

$$\|f - N\tilde{f}\|_2^2 \leq \|f - Ng\|_2^2 \qquad \forall g \in \mathbb{C}^{n^d}. \tag{3.14}$$

Man unterscheidet drei Fälle: Ist die Anzahl m an Punkten größer als die Anzahl n^d an gesuchten Fourier-Koeffizienten ($n^d < m$), so erhält man die Lösungen $\tilde{f}$ von (3.14) durch Lösen der *Normalengleichungen erster Art* des überbestimmten Systems (3.13),

$$N^{\mathrm{H}} N\tilde{f} = N^{\mathrm{H}} f. \tag{3.15}$$

Gehen wir zusätzlich davon aus, dass die Spalten von N linear unabhängig sind (d. h. $\operatorname{rang}(N) = n^d$), so ist die Lösung $\tilde{f}$ sogar eindeutig bestimmt. Zur numerischen Lösung von (3.15) bietet sich das CGNR-Verfahren an (CGNR für *conjugate-gradient normal-equation residual*, vgl. [Golub und van Loan, 1996, Alg. 10.4.1]). Der Vorteil dieser Methode besteht darin, dass sie im Wesentlichen auf Multiplikationen mit N und N^{H} basiert, wofür die iNFFT und ihre Adjungierte als schnelle Algorithmen zur Verfügung stehen.

Ist die Anzahl an Punkten kleiner als die Anzahl an Fourier-Koeffizienten ($m < n^d$), so wird das Kleinste-Quadrate-Problem (3.14) zu einem speziellen Optimierungsproblem erweitert: Man bestimme $\tilde{f}$ so, dass

$$\|f - N\tilde{f}\|_2^2 \;\leq\; \|f - Ng\|_2^2 \quad \forall g \in \mathbb{C}^{n^d}, \qquad \|\tilde{f}\|_2 = \min\,. \tag{3.16}$$

Durch die Zusatzbedingung ist die Lösung $\tilde{f}$ in diesem Fall unabhängig vom Rang der Matrix N eindeutig bestimmt. Gehen wir davon aus, dass die Zeilen von N linear unabhängig sind (d. h. $\operatorname{rang}(N) = m$), so erhalten wir $\tilde{f}$ durch Lösen der *Normalengleichungen zweiter Art* des unterbestimmten Systems (3.13),

$$NN^{\mathrm{H}}g \;=\; f, \qquad \tilde{f} \;=\; N^{\mathrm{H}}g. \tag{3.17}$$

Zur numerischen Lösung von (3.17) eignet sich die CGNE-Methode (CGNE für *conjugate-gradient normal-equation error*, vgl. [Golub und van Loan, 1996, Alg. 10.4.2]). Auch hier können die iNFFT und ihre Adjungierte für die darin auftretenden Multiplikationen mit N und N^{H} verwendet werden.

Ist die Anzahl an Punkten gleich der Anzahl an Fourier-Koeffizienten ($m = n^d$), so kann wieder die CGNR-Methode auf die Normalengleichungen (3.15) angewendet werden; man erhält unter Zuhilfenahme der iNFFT und ihrer Adjungierten also auch hier einen schnellen Algorithmus.

3.2 NFSGLFT-A: Trigonometrisierung und NFFT

Wir kommen nun zur ersten Klasse von NFSGLFTs, der Variante A. Bei der inversen Transformation kommt dabei die iNFFT aus dem vorigen Abschnitt direkt zum Einsatz, nachdem die auszuwertende bandbegrenzte Funktion $f \in H$ an den gegebenen Punkten x_i in die Form eines dreidimensionalen trigonometrischen Polynoms gebracht wurde. Für diese Übersetzung, die Ähnlichkeit mit einer Trigonometrisierung hat, verwenden wir den Clenshaw-Algorithmus bzw. eine adjungierte FDPT, eine adjungierte FLT, sowie die DCT.

3.2.1 Herleitung der schnellen Algorithmen

Es seien die SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ einer bandbegrenzte Funktion $f \in H$ mit Bandbreite $B \geq 2$, sowie Punkte $x_i = [r_i, \vartheta_i, \varphi_i]$, $i = 0, \ldots, M-1$, gegeben. Weiter sei $\rho > 0$ so gewählt, dass $r_i \leq \rho$ für alle $i \in \{0, \ldots, M-1\}$ gilt. Wir definieren die Hilfsfunktion

$$\gamma(r) \;=\; \gamma(\rho; r) \;:=\; \frac{2r - \rho}{\rho}, \qquad r \in [0, \rho].$$

Die Funktion γ ist ein Polynom vom Grad Eins, welches das Intervall $[0, \rho]$ bijektiv auf das Intervall $[-1, 1]$ abbildet. Die Umkehrabbildung bezeichnen wir mit γ^{-1}.

Wir betrachten nun

$$\begin{aligned} f(r,\vartheta,\varphi) &= \sum\nolimits_{|m|\le l<n\le B} \hat{f}_{nlm} H_{nlm}(r,\vartheta,\varphi) \\ &= \sum\nolimits_{|m|\le l<B} \underbrace{\left(\sum_{n=l+1}^{B} \hat{f}_{nlm} N_{nl} R_{nl}(r)\right)}_{=:\, g_{lm}(r)} Y_{lm}(\vartheta,\varphi). \end{aligned}$$

Die Funktion $g_{lm}(r)$ ist jeweils ein Polynom vom Grad höchstens $2B-2$ auf $[0,\rho]$. Wir schreiben dieses mithilfe der Funktion γ als

$$g_{lm} = g_{lm} \circ \gamma^{-1} \circ \gamma.$$

Die Funktion $\tilde{g}_{lm} := g_{lm} \circ \gamma^{-1}$ ist dementsprechend jeweils ein Polynom vom Grad höchstens $2B-2$ auf $[-1,1]$. Dieses übersetzen wir mithilfe der DCT (Abschn. 2.2.2) zunächst in die Tschebyschow-Basis (Def. 2.2.10). Dazu berechnen wir für $j = 0,\dots,2B-1$

$$\tilde{g}_{lm}(\cos\omega_j) = g_{lm}(\gamma^{-1}(\cos\omega_j)) = g_{lm}\left(\frac{\rho}{2}(1+\cos\omega_j)\right)$$

mit $\omega_j := (2j+1)\pi/4B$. Der Clenshaw-Algorithmus (Alg. 2.6) leistet dies für jedes Paar $[l,m]$, $|m| \le l < B$, in jeweils $\mathcal{O}(B^2)$ Schritten. Es ergibt sich eine Komplexität von $\mathcal{O}(B^4)$ für den ersten Schritt. Nun berechnen wir mit der DCT für jedes Paar $[l,m]$

$$\left[\alpha_{\kappa lm}\right]_{\kappa=0,\dots,2B-1} = D_{2B}\tilde{C}_{2B}\left[\tilde{g}_{lm}(\cos\omega_j)\right]_{j=0,\dots,2B-1}. \tag{3.18}$$

Dieser zweite Schritt besitzt eine Komplexität von $\mathcal{O}(B^3\log B)$. Es gilt nach Lemma 2.2.11

$$(\tilde{g}_{lm}\circ\cos)(\omega) = \sum_{\kappa=0}^{2B-1} \alpha_{\kappa lm}(T_\kappa\circ\cos)(\omega) = \sum_{\kappa=0}^{2B-1} \alpha_{\kappa lm}\cos(\kappa\omega) = \sum_{\kappa=0}^{2B-1} \frac{\alpha_{\kappa lm}}{2}\left(\mathrm{e}^{\mathrm{i}\kappa\omega} + \mathrm{e}^{-\mathrm{i}\kappa\omega}\right).$$

Daraus ergibt sich

$$g_{lm}(r_i) = \sum\nolimits_{\kappa\in I_{4B}^1} \beta_{\kappa lm}\,\mathrm{e}^{\mathrm{i}\kappa\arccos\gamma(r_i)}, \qquad i = 0,\dots,M-1,$$

mit

$$\beta_{\kappa lm} := \begin{cases} \alpha_{0,lm} & \text{für } \kappa = 0, \\ \alpha_{|\kappa|,lm}/2 & \text{für } 0 < |\kappa| < 2B, \\ 0 & \text{sonst.} \end{cases} \tag{3.19}$$

Die oben beschriebene radiale Teiltransformation hat eine Rechenkomplexität von $\mathcal{O}(B^4)$ und eine Speicherkomplexität von $\mathcal{O}(B^3)$, wie man sich schnell klar macht. Nutzt man statt des Clenshaw-Algorithmus eine adjungierte FDPT, so reduziert sich die Rechenkomplexität bei gleichbleibender Speicherkomplexität auf $\mathcal{O}(B^3\log^2 B)$.

Wir betrachten nun bei festem $i \in \{0, \dots, M-1\}$ die Funktion

$$\begin{aligned} f(r_i, \vartheta, \varphi) &= \sum\nolimits_{|m| \le l < B} g_{lm}(r_i) Y_{lm}(\vartheta, \varphi) \\ &= \sum\nolimits_{\kappa_0 \in I_{4B}^1} \sum\nolimits_{|m| < B} \underbrace{\left(\sum_{l=|m|}^{B-1} \beta_{\kappa_0, lm} M_{lm} P_{lm}(\cos\vartheta) \right)}_{=:\, h_{\kappa_0, m}(\cos\vartheta)} \mathrm{e}^{\mathrm{i}(\kappa_0 \arccos\gamma(r_i) + m\varphi)}. \end{aligned}$$

Für gerades m ist die Funktion $h_{\kappa_0,m}$ jeweils ein Polynom vom Grad höchstens $B-1$ auf $[-1,1]$. Es folgt

$$h_{\kappa_0,m} = \sum_{\kappa=0}^{2B-1} \varepsilon_{\kappa_0,\kappa m} T_\kappa$$

mit

$$\left[\varepsilon_{\kappa_0,\kappa m}\right]_{\kappa=0,\dots,2B-1} = D_{2B} \tilde{C}_{2B} \left[h_{\kappa_0,m}(\cos\omega_j)\right]_{j=0,\dots,2B-1}. \tag{3.20}$$

Wie oben ergibt sich nun

$$h_{\kappa_0,m}(\cos\vartheta_i) = \sum\nolimits_{\kappa \in I_{4B}^1} \zeta_{\kappa_0,\kappa m} \mathrm{e}^{\mathrm{i}\kappa\vartheta_i}$$

mit

$$\zeta_{\kappa_0,\kappa m} := \begin{cases} \varepsilon_{\kappa_0,0,m} & \text{für } \kappa = 0, \\ \varepsilon_{\kappa_0,|\kappa|,m}/2 & \text{für } 0 < |\kappa| < 2B, \\ 0 & \text{für } \kappa = -2B. \end{cases} \tag{3.21}$$

Der Fall m ungerade kann auf den Fall m gerade zurückgespielt werden: Für m ungerade ist $(1-\xi^2)^{-1/2} h_{\kappa_0,m}(\xi)$ jeweils ein Polynom vom Grad höchstens $B-2$ auf $[-1,1]$. Es folgt

$$h_{\kappa_0,m}(\cos\omega) = \sum_{\kappa=0}^{2B-2} \varepsilon_{\kappa_0,\kappa m} \sin(\omega) T_\kappa(\cos\omega)$$

mit

$$\left[\varepsilon_{\kappa_0,\kappa m}\right]_{\kappa=0,\dots,2B-1} = D_{2B} \tilde{C}_{2B} V_B \left[h_{\kappa_0,m}(\cos\omega_j)\right]_{j=0,\dots,2B-1}, \tag{3.22}$$

wobei V_B die in Abschnitt 2.2.3 für $n := B$ definierte diagonale Hilfsmatrix ist. Da nun

$$\sin(\omega) T_\kappa(\cos\omega) = \sin(\omega)\cos(\kappa\omega) = \frac{1}{4\mathrm{i}}\left(\mathrm{e}^{\mathrm{i}(\kappa+1)\omega} - \mathrm{e}^{-\mathrm{i}(\kappa+1)\omega} - \mathrm{e}^{\mathrm{i}(\kappa-1)\omega} + \mathrm{e}^{-\mathrm{i}(\kappa-1)\omega}\right),$$

gilt

$$h_{\kappa_0,m}(\cos\vartheta_i) = \sum\nolimits_{\kappa \in I_{4B}^1} \zeta_{\kappa_0,\kappa m} \mathrm{e}^{\mathrm{i}\kappa\vartheta_i}$$

mit

$$\zeta_{\kappa_0,\kappa m} := \frac{\operatorname{sgn}\kappa}{4\mathrm{i}} \begin{cases} 0 & \text{für } \kappa = 0, \\ 2\varepsilon_{\kappa_0,0,m} - \varepsilon_{\kappa_0,2,m} & \text{für } |\kappa| = 1, \\ \varepsilon_{\kappa_0,|\kappa|-1,m} - \varepsilon_{\kappa_0,|\kappa|+1,m} & \text{für } 1 < |\kappa| < 2B-2, \\ \varepsilon_{\kappa_0,|\kappa|-1,m} & \text{für } 2B-2 \le |\kappa| < 2B, \\ 0 & \text{für } \kappa = -2B. \end{cases} \tag{3.23}$$

Die Funktionswerte $h_{\kappa_0,m}(\cos\omega_j)$ können mit der adjungierten FLT (Abschn. 2.2.3) bei festem κ_0 in $\mathcal{O}(B^3)$ Schritten für alle m berechnet werden. Dabei sollte die adjungierte

FLT so angepasst werden, dass im Fall m ungerade die Gewichtung durch V_B in (3.22) schon mit vorgenommen wird, da es andernfalls zu Stabilitätsproblemen kommen kann; bei der adjungierten seminaiven FLT erreicht man dies durch einfaches Weglassen der Gewichtungsmatrix W_n in Folgerung 2.2.15. Es ergibt sich eine Komplexität von $\mathcal{O}(B^4)$ für diesen Schritt. Anschließend können bei festem κ_0 und m die Koeffizienten $\varepsilon_{\kappa_0,\kappa m}$ mittels DCT berechnet werden. Dieser Schritt hat eine Komplexität von $\mathcal{O}(B^3 \log B)$. Die oben beschriebene sphärische Teiltransformation hat also eine Rechenkomplexität von insgesamt $\mathcal{O}(B^4)$, die Speicherkomplexität ist $\mathcal{O}(B^3)$. Nutzt man eine FLT mit der Komplexität $\mathcal{O}(B^2 \log^2 B)$, so reduziert sich die Rechenkomplexität bei gleichbleibender Speicherkomplexität auf $\mathcal{O}(B^3 \log^2 B)$.

Bemerkung 3.2.1. Die sphärische Teiltransformation bildet die Grundlage der *nichtgleichwinkligen schnellen sphärischen Fourier-Transformation* (NFSFT für *nonequiangular fast spherical Fourier transform*) von Kunis und Potts [2003]. Dort werden sphärische Polynome wie oben beschrieben in zweidimensionale trigonometrische Polynome übersetzt und anschließend mit der zweidimensionalen iNFFT ausgewertet. Man vergleiche dazu auch [Kunis, 2006, Abschn. 3.3.1].

Es ergibt sich insgesamt

$$\begin{aligned} f(x_i) &= \sum\nolimits_{\kappa_0 \in I_{4B}^1} \sum\nolimits_{|m|<B} h_{\kappa_0,m}(\cos\vartheta_i)\, \mathrm{e}^{\mathrm{i}(\kappa_0 \arccos\gamma(r_i) + m\varphi_i)} \\ &= \sum\nolimits_{\kappa_0 \in I_{4B}^1} \sum\nolimits_{\kappa_1 \in I_{4B}^1} \sum\nolimits_{|m|<B} \zeta_{\kappa_0,\kappa_1,m}\, \mathrm{e}^{\mathrm{i}(\kappa_0 \arccos\gamma(r_i) + \kappa_1\vartheta_i + m\varphi_i)} \\ &= \sum\nolimits_{k \in I_{4B}^3} \eta_k\, \mathrm{e}^{\mathrm{i}\langle k, \tilde{x}_i\rangle_2} \end{aligned} \tag{3.24}$$

mit den transformierten Punkten $\tilde{x}_i := [\arccos\gamma(r_i), \vartheta_i, \varphi_i] \in \mathbb{T}^3$ und

$$\eta_k := \begin{cases} \zeta_{\kappa_0,\kappa_1,\kappa_2} & \text{für } |\kappa_2| < B, \\ 0 & \text{sonst,} \end{cases} \qquad k = [\kappa_0, \kappa_1, \kappa_2].$$

In einem letzten Schritt wird nun die rechte Seite von (3.24) mithilfe der iNFFT in $\mathcal{O}((\sigma B)^3 \log(\sigma B) + q^3 M)$ Schritten für alle $i = 0, \ldots, M-1$ ausgewertet; dabei wählen wir in dieser Arbeit die Gaußsche Ansatzfunktion (Def. 3.1.7) und lassen den Überabtastungsfaktor σ und den Abschneideparameter $q < \sigma B$ der iNFFT zunächst variabel (vgl. Abschn. 3.1). Ingesamt ergibt sich so eine iNFSGLFT mit einer Rechenkomplexität von $\mathcal{O}(B^4 + (\sigma B)^3 \log(\sigma B) + q^3 M)$ oder sogar $\mathcal{O}(B^3 \log^2 B + (\sigma B)^3 \log(\sigma B) + q^3 M)$ und einer Speicherkomplexität von $\mathcal{O}((\sigma B)^3)$. Die Rolle von q und σ arbeiten wir im nächsten Abschnitt heraus.

Wir bezeichnen die oben beschriebene Klasse von iNFSGLFTs als Variante A. Wir verzichten hier auf eine Darstellung in Pseudocode. Stattdessen stellen wir die iNFSGLFT-A als Matrix-Faktorisierung dar. Dazu definieren wir die zur radialen Teiltransformation gehörigen Matrizen

$$R_l = R_l(\rho) := \Big[N_{nl} R_{nl}(\rho(1+\cos\omega_j)/2)\Big]_{\substack{j=0,\ldots,2B-1 \\ n=l+1,\ldots,B}} \in \mathbb{R}^{2B\times(B-l)}, \qquad l < B.$$

Weiter seien $L_m = L_{m,B}$ die in (2.24) definierten Legendre-Matrizen. Wir nutzen die Permutationsmatrix S_B aus (2.37), sowie die Permutationsmatrix

$$T_B \coloneqq \left[e_{\sigma(\psi)}^{\mathrm{T}}\right]_{\psi=0,\ldots,4B^3-1} \in \mathbb{R}^{4B^3\times 4B^3}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und

$$\sigma(\psi) \coloneqq B^2\kappa(\psi) + \frac{B^2 - B - m(\psi)^2 \operatorname{sgn} m(\psi)}{2} + \left(B - \frac{\operatorname{sgn} m(\psi)}{2}\right) m(\psi) + l(\psi),$$

wobei hier

$$\begin{aligned}
\kappa(\psi) &\coloneqq \psi \bmod 4B,\\
l(\psi) &\coloneqq \left\lfloor \sqrt{\frac{\psi-\kappa(\psi)}{4B}} \right\rfloor,\\
m(\psi) &\coloneqq \frac{\psi-\kappa(\psi)}{4B} - l(\psi)(l(\psi)+1),
\end{aligned}$$

gilt. Die Matrix S_B^{T} sorgt für eine Umsortierung von $n = 1,\ldots,B$, $l < n$, $|m| \le l$, hin zu $l < B$, $|m| \le l$, $n = l+1,\ldots,B$. Die Matrix T_B^{T} erledigt die Umsortierung von $l < B$, $|m| \le l$, $\kappa = -2B,\ldots 2B-1$, hin zu $\kappa = -2B,\ldots 2B-1$, $|m| < B$, $l = |m|,\ldots,B-1$. Als dritte Permutationsmatrix benötigen wir

$$U_B \coloneqq \left[e_{\tau(\iota)}^{\mathrm{T}}\right]_{\iota=0,\ldots,4B^3-1} \in \mathbb{R}^{4B(2B-1)\times 4B(2B-1)}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und

$$\tau(\iota) \coloneqq B + (2B+\kappa(\iota))(2B-1) + m(\iota) - 1,$$

wobei hier

$$\begin{aligned}
\kappa(\iota) &\coloneqq (\iota \bmod 4B) - 2B,\\
m(\iota) &\coloneqq \frac{\iota-\kappa(\iota)-2B}{4B} - B + 1,
\end{aligned}$$

gilt. Die Matrix U_B^{T} sorgt für die Umsortierung von $|m| < B$, $\kappa = -2B,\ldots,2B-1$, hin zu $\kappa = -2B,\ldots,2B-1$, $|m| < B$. Darüber hinaus definieren wir die speziellen Matrizen

$$A_B \coloneqq \frac{1}{2}\left[\begin{array}{c|ccc|c|ccc}
0 & & & & 2 & & & \\
\hline
 & & & 1 & & 1 & & \\
 & & \iddots & & & & \ddots & \\
 & 1 & & & & & & 1
\end{array}\right]^{\mathrm{T}} \in \mathbb{R}^{4B\times 2B},$$

sowie $I_m \coloneqq A_B$ für m gerade, bzw.

$$I_m \coloneqq \frac{1}{4\mathrm{i}}\left[\begin{array}{c|ccccc|c|ccccc}
0 & & & & & -2 & 0 & 2 & & & & \\
\hline
 & & & & -1 & 0 & & 0 & 1 & & & \\
 & & & -1 & 0 & 1 & & -1 & 0 & 1 & & \\
 & & \iddots & \iddots & \iddots & & & & \ddots & \ddots & \ddots & \\
 & -1 & 0 & 1 & & & & & & -1 & 0 & 1 \\
\hline
0 & & & & & & & & & & &
\end{array}\right]^{\mathrm{T}} \in \mathbb{R}^{4B\times 2B},$$

für m ungerade. Damit ergibt sich zwischen (3.18) und (3.19) der Zusammenhang

$$\Big[\beta_{\kappa lm}\Big]_{\kappa=-2B,\ldots,2B-1} = A_B \Big[\alpha_{\kappa lm}\Big]_{\kappa=0,\ldots,2B-1}$$

und zwischen (3.20) bzw. (3.22) und (3.21) bzw. (3.23) der Zusammenhang

$$\Big[\zeta_{\kappa_0,\kappa m}\Big]_{\kappa=-2B,\ldots,2B-1} = I_m \Big[\varepsilon_{\kappa_0,\kappa m}\Big]_{\kappa=0,\ldots,2B-1}.$$

Dies entspricht jeweils einem Wechsel von der Tschebyschow- in die Monombasis. Als weitere Hilfsmatrix definieren wir noch

$$X_B := \Big[0_{2B-1,B+1} \,\Big|\, E_{2B-1} \,\Big|\, 0_{2B-1,B}\Big] \in \mathbb{R}^{2B\times 4B}.$$

Die Matrix X_B^{T} erweitert den Bereich von $\kappa = 0,\ldots,B-1$ zu $\kappa = -2B,\ldots,2B-1$ durch Hinzufügen von Nullen. Zu guter Letzt definieren wir noch die zu den Punkten $x_0,\ldots,x_{M-1}$ gehörige spezielle NDFT-Matrix (vgl. Gl. 3.3)

$$N = N(B;x_0,\ldots,x_{M-1}) := \Big[\mathrm{e}^{\mathrm{i}\langle k(\chi),\tilde{x}_i\rangle_2}\Big]_{\substack{i=0,\ldots,M-1\\ \chi=0,\ldots,(4B)^3-1}} \in \mathbb{C}^{M\times(4B)^3}$$

mit den transformierten Punkten $\tilde{x}_i \in \mathbb{T}^3$. Es ergibt sich nun mit den obigen Bestandteilen aus den Überlegungen dieses Abschnitts insgesamt das folgende Hauptresultat:

Satz 3.2.2. *Die Matrix $\tilde{H}$ in Definition 3.1 besitzt die Faktorisierung*

$$\tilde{H} = N\cdot\big\{E_{(4B)^2}\otimes X_B^{\mathrm{T}}\big\}\cdot\big\{E_{4B}\otimes U_B^{\mathrm{T}}\big\}\cdot\left\{E_{4B}\otimes\overbrace{\begin{bmatrix} I_{1-B}\,D_{2B}\,\tilde{C}_{2B} & & \\ & \ddots & \\ & & I_{B-1}\,D_{2B}\,\tilde{C}_{2B}\end{bmatrix}}^{2B-1\text{ Blöcke der Größe }4B\times 2B}\right\}$$

$$\times\underbrace{\left\{E_{4B}\otimes\begin{bmatrix}\tilde{L}_{1-B}^{\mathrm{T}} & & \\ & \ddots & \\ & & \tilde{L}_{B-1}^{\mathrm{T}}\end{bmatrix}\right\}}_{2B-1\text{ Blöcke (s. u.)}}\cdot T_B^{\mathrm{T}}\cdot\underbrace{\Big\{E_{B^2}\otimes\Big(A_B\,D_{2B}\,\tilde{C}_{2B}\Big)\Big\}}_{B^2\text{ Blöcke der Größe }4B\times 2B}\cdot\underbrace{\begin{bmatrix}\tilde{R}_0 & & \\ & \ddots & \\ & & \tilde{R}_{B-1}\end{bmatrix}}_{B\text{ Blöcke (s. u.)}}\cdot S_B^{\mathrm{T}},$$

mit

$$\tilde{L}_m^{\mathrm{T}} := \begin{Bmatrix} E_{2B} \textit{ für } m \textit{ gerade} \\ V_B \textit{ für } m \textit{ ungerade}\end{Bmatrix}\cdot L_m^{\mathrm{T}}\cdot\operatorname{diag}\Big[M_{lm}\Big]_{l=|m|,\ldots,B-1} \in \mathbb{R}^{2B\times(B-|m|)}, \qquad |m|<B,$$

und den Blockmatrizen

$$\tilde{R}_l := \underbrace{E_{2l+1}\otimes R_l}_{2l+1\text{ Blöcke der Größe }2B\times(B-l)}, \qquad l<B.$$

Die Matrizen L_m^{T} und R_l können nun wie in Abschnitt 2.2.3 bzw. 2.2.5 beschrieben selbst wieder faktorisiert werden; bei Verwendung der adjungierten seminaiven FLT ist die Faktorisierung von L_m^{T} durch Folgerung 2.2.15 gegeben, bei Verwendung des Clenshaw-Algorithmus für den radialen Teil ergibt sich die Faktorisierung von R_l aus Satz 2.2.21.

Durch Umkehren der Produktreihenfolge und hermitesches Transponieren der einzelnen Faktoren erhalten wir unter Berücksichtigung der Rechenregeln für das Kronecker-Produkt zusammen mit Lemma 2.2.8 nun abschließend noch das folgende zweite Hauptresultat. Es zeigt uns, dass wir auch über eine *adjungierte* iNFSGLFT-A mit der gleichen Komplexität wie die der iNFSGLFT-A verfügen.

Folgerung 3.2.3. *Die Matrix $\tilde{H}^{\mathrm{H}}$ besitzt die Faktorisierung*

$$\tilde{H}^{\mathrm{H}} = S_B \cdot \begin{bmatrix} \tilde{R}_0^{\mathrm{T}} & & \\ & \ddots & \\ & & \tilde{R}_{B-1}^{\mathrm{T}} \end{bmatrix} \cdot \left\{ E_{B^2} \otimes \left(\tilde{C}_{2B}^{-1} D_{2B}^{\mathrm{T}} A_B^{\mathrm{T}} \right) \right\} \cdot T_B \cdot \left\{ E_{4B} \otimes \begin{bmatrix} \tilde{L}_{1-B} & & \\ & \ddots & \\ & & \tilde{L}_{B-1} \end{bmatrix} \right\}$$
$$\times \left\{ E_{4B} \otimes \begin{bmatrix} \tilde{C}_{2B}^{-1} D_{2B}^{\mathrm{T}} I_{1-B}^{\mathrm{H}} & & \\ & \ddots & \\ & & \tilde{C}_{2B}^{-1} D_{2B}^{\mathrm{T}} I_{B-1}^{\mathrm{H}} \end{bmatrix} \right\} \cdot \{ E_{4B} \otimes U_B \} \cdot \{ E_{(4B)^2} \otimes X_B \} \cdot N^{\mathrm{H}}.$$

Die hier enthaltenen Matrizen L_m und R_l^{T} können ebenfalls selbst wieder faktorisiert werden, siehe Satz 2.2.14 für die seminaive FLT und Folgerung 2.2.23 für den adjungierten Clenshaw-Algorithmus. Auf dieselbe Art und Weise wie am Ende von Abschnitt 3.1 für die NFFT beschrieben ergibt sich aus der iNFSGLFT-A und ihrer Adjungierten auch eine schnelle Vorwärtstransformation, die wir als NFSGLFT-A bezeichnen.

3.2.2 Fehlerabschätzung

Bei näherer Betrachtung wird klar, dass der einzig approximative Teil der iNFSGLFT-A die abschließende iNFFT ist. Mithilfe der Sätze 3.1.9 und 3.2.2 können wir daher eine Abschätzung für den maximalen absoluten Fehler der iNFSGLFT-A herleiten. Letzterer wird definiert als

$$E_\infty = E_\infty(\hat{f}, x_0, \ldots, x_{M-1}; \sigma, q, B; \rho) := \max_{i \in \{0,\ldots,M-1\}} |f(x_i) - \tilde{f}(x_i)| \tag{3.25}$$

mit den gegebenen Fourier-Koeffizienten $\hat{f} := [\hat{f}_{n(\mu),l(\mu),m(\mu)}]_{\mu=0,\ldots,B(B+1)(2B+1)/6-1}$ und dem Ergebnis $[\tilde{f}(x_i)]_{i=0,\ldots,M-1}$ der iNFSGLFT-A; σ ist der Überabtastungsfaktor der iNFFT, q deren Abschneideparameter.

Satz 3.2.4. *Der maximale absolute Fehler der iNFSGLFT-A lässt sich unter Verwendung der iNFFT aus Abschnitt 3.1 für $\sigma \geq 2$ wie folgt abschätzen:*

$$E_\infty \leq C\sqrt{2B-1}(B+1)B^2 a_{\rho,B} \exp\left(b_{\rho,B} \left(B + \frac{1}{2} \right)^{1-1/\mathrm{e}} + \frac{\rho^2}{2} - \frac{q\pi}{2} \right) \|\hat{f}\|_1$$

mit einer globalen Konstanten $C \leq 132$, sowie den Koeffizienten

$$a_{\rho,B} := \begin{cases} 1, & \textit{wenn } \rho < 1 \textit{ und } B \leq \{2\mathrm{e}^{-2/\mathrm{e}} \ln(\rho)(\rho^{2/\mathrm{e}} - 1)^{-1}\}^{-1+1/\mathrm{e}} - 2^{-1}, \\ 1/\rho & \textit{sonst,} \end{cases}$$

$$b_{\rho,B} := \begin{cases} \mathrm{e}^{2/\mathrm{e}}/2, & \textit{wenn } \rho < 1 \textit{ und } B \leq \{2\mathrm{e}^{-2/\mathrm{e}} \ln(\rho)(\rho^{2/\mathrm{e}} - 1)^{-1}\}^{-1+1/\mathrm{e}} - 2^{-1}, \\ (\rho\mathrm{e})^{2/\mathrm{e}}/2 & \textit{sonst.} \end{cases}$$

Für den Beweis dieses Satzes benötigen wir drei technische Lemmata:

Lemma 3.2.5. *Für $\xi \geq \psi > 0$ gilt*

$$\ln \frac{\Gamma(\xi)}{\Gamma(\psi)} \leq (\xi - \psi) \ln \xi.$$

Beweis. Aus der Log-Konvexität der Gammafunktion auf $(0, \infty)$ (s. [Andrews et al., 1999, Flg. 1.2.6]) folgt zunächst $\ln \Gamma(\xi) - \ln \Gamma(\psi) \leq (\xi - \psi)\psi(\xi)$ mit der *Digammafunktion* $\psi := \mathrm{d} \ln \Gamma / \mathrm{d}\xi$. Die Aussage ergibt sich nun aus der Abschätzung $\psi(\xi) \leq \ln \xi$, $\xi \in (0, \infty)$ (vgl. [Abramowitz und Stegun, 1972, Gl. 6.3.21]). □

Lemma 3.2.6 [Batir, 2008, Satz 1.5]. *Für $\xi > 1$ gilt*

$$\sqrt{2\mathrm{e}} \left(\frac{\xi - 1/2}{\mathrm{e}} \right)^{\xi - 1/2} \leq \Gamma(\xi).$$

Lemma 3.2.7. *Für $\xi > 0$ gilt*

$$\ln \xi \leq \xi^{1/\mathrm{e}}.$$

Beweis. Die Funktion $f(\xi) := \xi^{1/\mathrm{e}} - \ln \xi$ besitzt die Ableitung

$$\frac{\mathrm{d}f}{\mathrm{d}\xi}(\xi) = \mathrm{e}^{-1}(\xi^{1/\mathrm{e}-1} - \mathrm{e}\xi^{-1}).$$

Es gilt $\mathrm{d}f/\mathrm{d}\xi = 0$ genau an der Stelle $\xi = \mathrm{e}^{\mathrm{e}}$. Ein Einsetzen von e^{e} in die zweite Ableitung von f zeigt, dass f an dieser Stelle ihr globales Minimum $f(\mathrm{e}^{\mathrm{e}}) = 0$ annimmt. □

Beweis von Satz 3.2.4. Um Satz 3.1.9 anwenden zu können, untersuchen wir in Satz 3.2.2 den Einfluss der Faktoren rechts der NDFT-Matrix N auf die 1-Norm des Eingabevektors. Bei den Blockmatrizen können wir uns dabei auf die einzelnen Blöcke zurückziehen.

Zunächst halten wir fest, dass die Multiplikation eines Vektors x mit einer Permutationsmatrix P keinen Einfluss auf die 1-Norm hat, es gilt $\|Px\|_1 = \|x\|_1$. Das gleiche gilt für die Matrizen A_B, I_m und X_B. Aus Lemma 2.2.9 folgt sofort $\|\tilde{C}_{2B}x\|_1 \leq \sqrt{2B}\|x\|_1$ für jeden Vektor x der Länge $2B$. Darüber hinaus gilt $\|D_{2B}x\|_1 \leq \|x\|_1/\sqrt{B}$.

Aus Lemma 1.4.2 und der Tatsache, dass $\rho(1 + \cos \omega_j)/2 \leq \rho$, folgt für den Radialteil der SGL-Basisfunktionen

$$\left| N_{nl} R_{nl}\left(\frac{\rho}{2}(1 + \cos \omega_j)\right)\right|^2 \leq \frac{2(n-l-1)!}{\Gamma(n+1/2)} \binom{n - 1/2}{n - l - 1}^2 \rho^{2l} \mathrm{e}^{\rho^2} = \frac{2\Gamma(n+1/2)\rho^{2l}\mathrm{e}^{\rho^2}}{\Gamma(n-l)\Gamma(l+3/2)^2}.$$

Mit Blick auf die rechte Seite zeigen wir

$$\begin{aligned}
\ln \frac{\Gamma(n+1/2)\rho^{2l}}{\Gamma(n-l)\Gamma(l+3/2)^2} &= \ln \frac{\Gamma(n+1/2)}{\Gamma(n-l)} + l \ln \rho^2 - 2 \ln \Gamma(l + 3/2) \\
&\leq (l + 1/2) \ln(n + 1/2) + l \ln \rho^2 - 2 \ln((l+1)^{l+1}\mathrm{e}^{-l-1}) - \ln 2 - 1 && (3.26) \\
&\leq (l+1) \ln((\rho\mathrm{e})^2(n+1/2)(l+1)^{-2}) - 2 \ln \rho - \ln 2 - 1 \\
&\leq (\rho\mathrm{e})^{2/\mathrm{e}}(n+1/2)^{1/\mathrm{e}}(l+1)^{1-2/\mathrm{e}} - 2 \ln \rho - \ln 2 - 1 && (3.27) \\
&\leq (\rho\mathrm{e})^{2/\mathrm{e}}(B+1/2)^{1-1/\mathrm{e}} - 2 \ln \rho - \ln 2 - 1. && (3.28)
\end{aligned}$$

Dabei folgt die Abschätzung (3.26) aus den Lemmata 3.2.5 und 3.2.6, die Ungleichung (3.27) ergibt sich aus Lemma 3.2.7. Im Fall $\rho \ll 1$ kann (3.28) verbessert werden: Verzichten wir in (3.26) auf den Summanden $l \ln \rho^2 \leq 0$, so erhalten wir ähnlich wie oben

$$\ln \frac{\Gamma(n+1/2)\rho^{2l}}{\Gamma(n-l)\Gamma(l+3/2)^2} \leq \mathrm{e}^{2/\mathrm{e}}(B+1/2)^{1-1/\mathrm{e}} - \ln 2 - 1. \tag{3.29}$$

Ein Vergleich der rechten Seiten von (3.28) und (3.29) zeigt, dass

$$\mathrm{e}^{2/\mathrm{e}}(B+1/2)^{1-1/\mathrm{e}} \leq (\rho\mathrm{e})^{2/\mathrm{e}}(B+1/2)^{1-1/\mathrm{e}} - 2\ln\rho$$

genau dann, wenn $B \leq \{2\mathrm{e}^{-2/\mathrm{e}}\ln(\rho)(\rho^{2/\mathrm{e}}-1)^{-1}\}^{-1+1/\mathrm{e}} - 2^{-1}$. Für festes $l < B$ und einen beliebigen Vektor $x = [\xi_0, \ldots, \xi_{B-l-1}]$ der Länge $B-l$ lassen sich die Elemente von $\tilde{R}_l^{\mathrm{T}} x$ mit den im Satz angegebenen Koeffizienten $a_{\rho,B}$ und $b_{\rho,B}$ also durch

$$\left| \sum_{n=l+1}^{B} \xi_{n-l-1} N_{nl} R_{nl}\Big(\frac{\rho}{2}(1+\cos\omega_j)\Big) \right| \leq a_{\rho,B} \exp\left(b_{\rho,B}\left(B+\frac{1}{2}\right)^{1-1/\mathrm{e}} + \frac{\rho^2}{2} - \frac{1}{2}\right)\|x\|_1$$

abschätzen. Es gilt dementsprechend für einen Vektor x der passenden Länge stets

$$\|\tilde{R}_l^{\mathrm{T}} x\|_1 \leq 2B\, a_{\rho,B} \exp\left(b_{\rho,B}\left(B+\frac{1}{2}\right)^{1-1/\mathrm{e}} + \frac{\rho^2}{2} - \frac{1}{2}\right)\|x\|_1.$$

Es seien jetzt $|m| < B$ und $x = [\xi_0, \ldots, \xi_{B-|m|}]$ ein Vektor der Länge $B-|m|$, m zunächst gerade. Mithilfe von Lemma 1.1.4 lassen sich die Elemente von $\tilde{L}_m^{\mathrm{T}} x$ durch

$$\left| \sum_{l=|m|}^{B-1} \xi_{l-|m|} M_{lm} P_{lm}(\cos\omega_j) \right| \leq \sqrt{\frac{2B-1}{4\pi}}\|x\|_1$$

abschätzen. Im Fall m ungerade folgt aus Lemma 1.1.6

$$\left| \sum_{l=|m|}^{B-1} \xi_{l-|m|} M_{lm} (\sin\omega_j)^{-1} P_{lm}(\cos\omega_j) \right| \leq \sqrt{\frac{2B-1}{8\pi}}(B+1)\|x\|_1.$$

Es gilt somit für einen Vektor x der passenden Länge stets

$$\|\tilde{L}_m^{\mathrm{T}} x\|_1 \leq \sqrt{\frac{2B-1}{2\pi}} B(B+1)\|x\|_1.$$

Nun kann der zweite Teil von Satz 3.1.9 angewendet werden. □

Satz 3.2.4 zeigt uns, dass der maximale absolute Fehler mit steigender Bandbreite B potentiell wächst. Gleiches gilt für größer werdendes ρ, was die Wahl

$$\rho := \max\{r_0, \ldots, r_{M-1}\} \tag{3.30}$$

motiviert. Wir können den Fehler bei fester Bandbreite B und festem ρ in exakter Arithmetik jedoch *beliebig klein* werden lassen, indem wir den Abschneideparameter q der iNFFT hinreichend groß wählen. Insbesondere halten wir fest, dass der Fehler der iNFSGLFT-A

(i) mit größer werdendem Abschneideparameter q mindestens *exponentiell* abfällt,

(ii) mit größer werdendem ρ potentiell *hyperexponentiell* anwächst und

(iii) mit wachsender Bandbreite B höchstens *subexponentiell* anwächst (bei höchstens subexponentiellem Wachstum von $\|\hat{f}\|_1$).

Alle drei Phänomene werden wir im nächsten Abschnitt numerisch nachweisen. Insbesondere können wir, um den Fehler zu kontrollieren, nach (iii) den Abschneideparameter q als Funktion von B auf geeignete Weise als $q = \mathrm{o}(B)$ wählen. Dies zeigt, dass wir mit der iNFSGLFT-A auch auf theoretischer Ebene einen schnellen Algorithmus zur Hand haben.

3.2.3 Numerische Tests

Die iNFSGLFT-A, ihre Adjungierte, sowie die NFSGLFT-A wurden in der Programmiersprache C++ implementiert und auf dem in Abschnitt 2.3 beschriebenen System getestet. Es wurde der Clenshaw-Algorithmus und dessen Adjungierter (Abschn. 2.2.5) verwendet. Weiter wurde die seminaive FLT und deren Adjungierte (Abschn. 2.2.3) des SpharmonicKit benutzt (vgl. Abschn. 2.3). Der Clenshaw-Algorithmus und dessen Adjungierter wurden wieder mit erweiterter doppelter Genauigkeit durchgeführt, die übrigen Berechnungen wurden mit doppelter Genauigkeit vorgenommen. Bei der iNFFT und ihrer Adjungierten wurde auf die Implementierung von Keiner et al. [2009] zurückgegriffen[3] (Vers. 3.3.1). Es wurde die DCT der *Fastest Fourier Transform in the West*[4] verwendet (FFTW, Vers. 3.3.6). In allen Tests wurde der Überabtastungsfaktor $\sigma = 2$ der iNFFT gewählt. Der Parameter ρ der iNFSGLFT-A und ihrer Adjungierten wurde stets wie in (3.30) gesetzt. Für die Fehlermessungen und zum Laufzeitvergleich wurde eine naive iNDSGLFT analog zu der im Abschnitt 2.3 genutzten naiven iDSGLFT implementiert. Es ist im Folgenden

$$\mathbb{B}^3_\kappa := \{x \in \mathbb{R}^3 : \|x\|_2 \leq \kappa\}, \qquad \kappa > 0.$$

Bemerkung 3.2.8. Bei genauer Betrachtung der Herleitung der iNFSGLFT-A und ihrer Adjungierten in Abschnitt 3.2.1 wird klar, dass in (3.24) nicht das gesamte Gitter I^3_{4B} mit den Ausmaßen $4B \times 4B \times 4B$ benötigt wird. Da die Implementierung der iNFFT und ihrer Adjungierten von Keiner et al. auch Gitter mit unterschiedlicher Ausdehnung in jeder Dimension zulässt, wurde hier ein kleineres Gitter mit den Ausmaßen $4B \times 2B \times 2B$ verwendet. Dies verbessert die Laufzeit, hat aber keinen Einfluss auf die Komplexität.

In einem ersten Test wurden für die Bandbreiten $B = 32$ und $B = 64$ jeweils zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ generiert, wobei der Real- und Imaginärteil jeweils gleichverteilt zwischen -1 und 1 lag. Zusätzlich dazu wurden jeweils $M = 10\,000$ zufällige, gleichverteilte Punkte $x_i \in \mathbb{B}^3_5$ erzeugt. Es wurden dann aus den SGL-Fourier-Koeffizienten mit dem naiven Algorithmus die Funktionswerte $f(x_i)$ und mit der iNFSGLFT-A die

[3] Siehe `https://www-user.tu-chemnitz.de/~potts/nfft`
[4] Siehe `https://www.fftw.org`

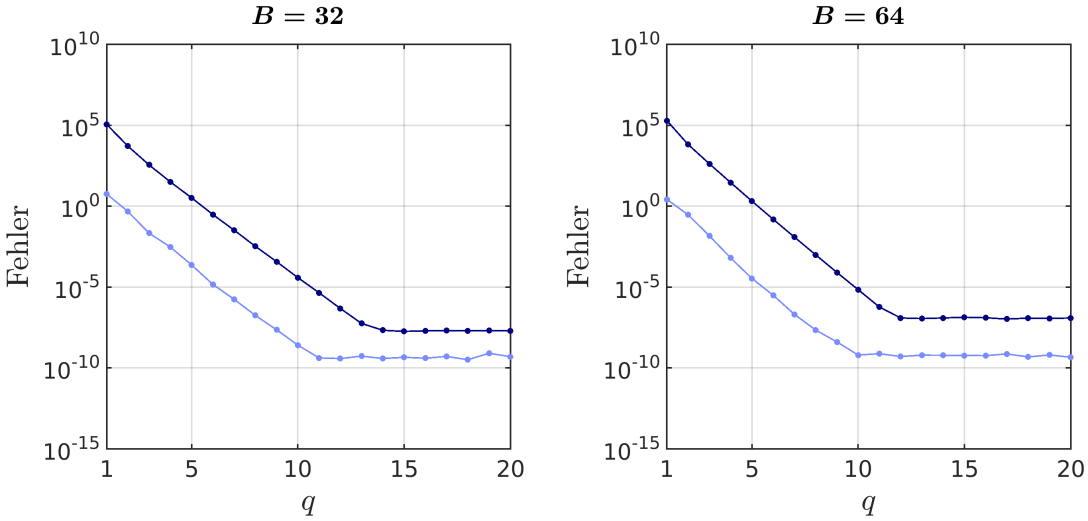

Abbildung 3.1. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-A bei $M = 10\,000$ gleichverteilten Punkten in $\mathbb{B}_5^3$ mit $\sigma = 2$ in Abhängigkeit von dem Abschneideparameter q.

näherungsweisen Funktionswerte $\tilde{f}(x_i)$ berechnet. Dies wurde jeweils für die Abschneideparameter $q = 1, \ldots, 20$ getan. Der obige Test wurde insgesamt zehnmal wiederholt und so der maximale absolute und relative Fehler

$$\max_{i=0,\ldots,M-1} |f(x_i) - \tilde{f}(x_i)| \qquad \text{bzw.} \qquad \max_{i=0,\ldots,M-1} \frac{|f(x_i) - \tilde{f}(x_i)|}{|f(x_i)|} \tag{3.31}$$

in Abhängigkeit von q bestimmt. Abbildung 3.1 zeigt die Ergebnisse dieser Fehlermessung. Die Standardabweichung der gezeigten Ergebnisse war durchgehend so klein, dass sie der Übersichtlichkeit halber nicht eingezeichnet wurde. Es ist deutlich zu erkennen, dass der Fehler wie erwartet mit wachsendem q exponentiell abfällt (vgl. S. 75, i), bevor der Rundungsfehler die Oberhand gewinnt. In beiden Fällen $B = 32$ und $B = 64$ ist der relative Fehler für die Zweierpotenz $q = 16$ lediglich von der Größenordnung 10^{-10}, also unter Berücksichtigung der Maschinengenauigkeit zufriedenstellend klein.

In einem zweiten Test wurden für die Bandbreiten $B = 32$ und $B = 64$ wie oben zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ generiert. Zusätzlich dazu wurden für $k = 1, \ldots, 6$ jeweils $M = 10^k$ zufällige, gleichverteilte Punkte $x_i \in \mathbb{B}_5^3$ erzeugt. Es wurden dann aus den SGL-Fourier-Koeffizienten die Funktionswerte $f(x_i)$ mit dem naiven Algorithmus und mit der iNFSGLFT-A rekonstruiert. Tabelle 3.1 zeigt die Ergebnisse der dabei vorgenommenen Laufzeitmessung. Es ist klar ersichtlich, dass die iNFSGLFT-A selbst bei kleinen Problemgrößen ($M \geq 100$) einen Laufzeitvorteil bietet. Mit wachsendem M gewinnt die iNFSGLFT-A weiter an Vorsprung. Im Fall $B = 32$, $M = 1\,000\,000$, etwa benötigt der naive Algorithmus ca. drei Stunden, die Laufzeit der iNFSGLFT-A betrug hingegen weniger als zwei Minuten.

Im nächsten Test wurde der Fehler der iNFSGLFT-A in Abhängigkeit von der Streuweite der Zielpunkte untersucht. Zu diesem Zweck wurden für die Bandbreite $B = 32$ wie zuvor zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ generiert. Zusätzlich dazu wurden $M =$

M	iNDSGLFT (naiv)	iNFSGLFT-A
1 E +1	9,99 E −2 s	9,27 E −1 s
1 E +2	9,95 E −1 s	9,35 E −1 s
1 E +3	9,81 E +0 s	1,03 E +0 s
1 E +4	1,04 E +2 s	2,10 E +0 s
1 E +5	1,09 E +3 s	1,22 E +1 s
1 E +6	1,11 E +4 s	1,13 E +2 s

(a)

M	iNDSGLFT (naiv)	iNFSGLFT-A
1 E +1	1,36 E +0 s	1,01 E +1 s
1 E +2	1,38 E +1 s	1,03 E +1 s
1 E +3	1,36 E +2 s	1,03 E +1 s
1 E +4	1,36 E +3 s	1,19 E +1 s
1 E +5	1,37 E +4 s	2,82 E +1 s

(b)

Tabelle 3.1. Durchschnittliche Laufzeit der naiven iNDSGLFT und der iNFSGLFT-A für **(a)** $B = 32$ und **(b)** $B = 64$, jeweils mit $\sigma = 2$ und $q = 16$, in Abhängigkeit von der Anzahl M an Zielpunkten.

1000 zufällige, gleichverteilte Punkte $x_i \in \mathbb{B}^3_\kappa$ erzeugt, wobei für κ über die Werte $i/4$, $i = 1, \ldots, 31$, iteriert wurde. Man beachte hierbei, dass nach (3.30) stets $\rho \approx \kappa$ gilt, wie man sich schnell klar macht. Der obige Test wurde zehnmal wiederholt und so der maximale absolute und relative Fehler in Abhängigkeit von κ bestimmt. In Abbildung 3.2 sind die Ergebnisse dargestellt. Man erkennt deutlich das vermutete hyperexponentielle Fehlerwachstum der iNFSGLFT-A bezüglich ρ (vgl. S. 75, ii). In diesem Test war die Standardabweichung des absoluten Fehlers wieder durchweg so klein, dass sie in Abbildung 3.2 nicht eingezeichnet wurde.

In einem vierten Test wurde der Fehler der iNFSGLFT-A in Abhängigkeit von der Bandbreite B untersucht. Dazu wurden für die Bandbreiten $B = 2^k$, $k = 3, \ldots, 7$, wie zuvor zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ generiert. Zusätzlich dazu wurden $M = 1000$ zufällige, gleichverteilte Punkte $x_i \in \mathbb{B}^3_5$ erzeugt. Es wurde der Abschneideparameter $q = 12$ gewählt. Im eigentlichen Testlauf wurden dann aus den SGL-Fourier-Koeffizienten mit dem naiven Algorithmus die Funktionswerte $f(x_i)$ und mit der iNFSGLFT-A die näherungsweisen Funktionswerte $\tilde{f}(x_i)$ berechnet. Dies wurde für jede Bandbreite zehnmal wiederholt und so der maximale absolute und relative Fehler in Abhängigkeit von B bestimmt. Abbildung 3.3 zeigt die Ergebnisse dieser Fehlermessung. Man erkennt hier das vermutete subexponentielle Fehlerwachstum der iNFSGLFT-A bezüglich B (vgl. S. 75, iii). Interessant ist, dass der absolute Fehler von der Bandbreite $B = 32$ zur Bandbreite $B = 64$ hin abfällt. Indem der obige Test noch einmal unter Verwendung der exakten

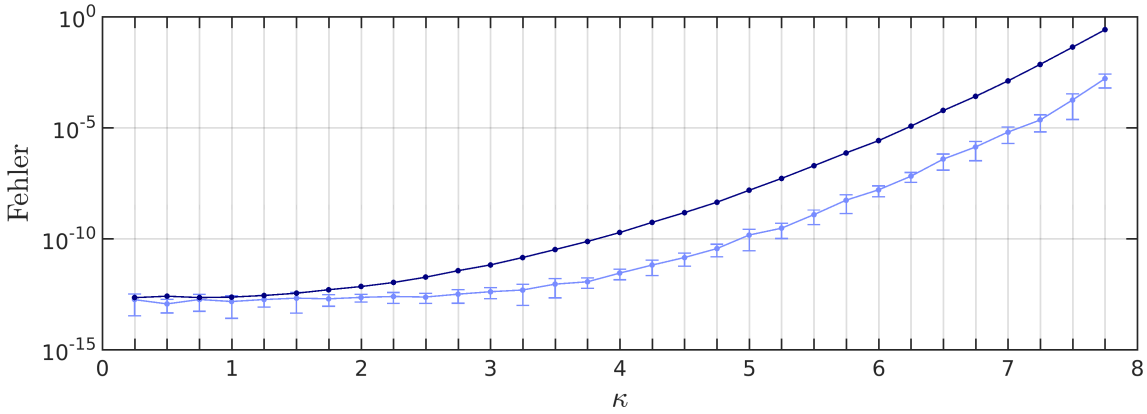

Abbildung 3.2. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-A für $B = 32$ bei $M = 1000$ gleichverteilten Punkten in $\mathbb{B}^3_\kappa$ mit $\sigma = 2$ und $q = 16$ in Abhängigkeit von κ ($\approx \rho$).

iNDFT anstelle der approximativen iNFFT durchgeführt wurde, konnte die Ursache dieses reproduzierbaren Effekts in der iNFFT gefunden werden (vgl. Abb. 3.3).

In einem letzten Test wurde die NFSGLFT-A, d. h. die Vorwärtstransformation, untersucht. Dazu wurden die kartesischen Gitter

$$I_N := \left\{ \kappa \left[\frac{2j}{N} - 1, \frac{2k}{N} - 1, \frac{2l}{N} - 1 \right] \in \mathbb{R}^3 : j, k, l = 0, \ldots, N-1 \right\}, \qquad \kappa > 0,\ N \in \mathbb{N},$$

verwendet. Es gilt hier stets $M = N^3$. Für die Bandbreiten $B = 8$ ($q = 15$) und $B = 16$ ($q = 16$) wurden bei $\kappa = 5$ die Fälle $N = 25, 50, 100$, sowie bei $N = 50$ die Fälle $\kappa = 6, 8, 10$ untersucht. Da in allen diesen Fällen M größer ist als die Anzahl an SGL-Fourier-Koeffizienten, wurde mit der iNFSGLFT-A und ihrer Adjungierten die NFSGLFT-A als CGNR-Verfahren realisiert (vgl. Abschn. 3.1). Innerhalb dieses speziellen CG-Verfahrens wurden die α_k und β_k (vgl. [Golub und van Loan, 1996, Alg. 10.4.1]) mit erweiterter doppelter Genauigkeit berechnet. In den Testläufen wurden dann zunächst wie oben zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ generiert. Daraus wurden mithilfe der exakten iNDSGLFT die Funktionswerte $f(x_i)$ rekonstruiert. Aus diesen Daten wurden dann mit der NFSGLFT-A die SGL-Fourier-Koeffizienten $\hat{f}^{\circ}_{nlm}$ berechnet. Als initiale Schätzung der SGL-Fourier-Koeffizienten für das CGNR-Verfahren wurde die *Mittelpunktsregel*

$$\hat{f}_{nlm} \approx 8N^{-1} \sum\nolimits_{x_i \in I_N} f(x_i) \overline{H_{nlm}(x_i)} \mathrm{e}^{-\|x_i\|_2^2}$$

der numerischen Quadratur verwendet. Während der Durchführung des CGNR-Verfahrens wurde dann in Abhängigkeit von der Anzahl an Iterationen der absolute und relative Rekonstruktionsfehler (2.38) der NFSGLFT-A bestimmt. In den Abbildungen 3.4 und 3.5 sind die Ergebnisse dargestellt. Für $B = 8$, $\kappa = 5$ (Abb. 3.4, links), wurde in allen Fällen von der NFSGLFT-A ein kleiner Fehler erreicht, dafür wurden allerdings relativ viele Iterationen benötigt. Mit Ausnahme der Fälle $B = 8$, $\kappa = 8, 10$ (Abb. 3.5, links), sieht man, dass selbst nach zehntausend Iterationen in keinem der untersuchten Fälle von erreichter Konvergenz gesprochen werden kann. Für $B = 16$, $\kappa = 5$ (Abb. 3.4, rechts) ist zwar gegen

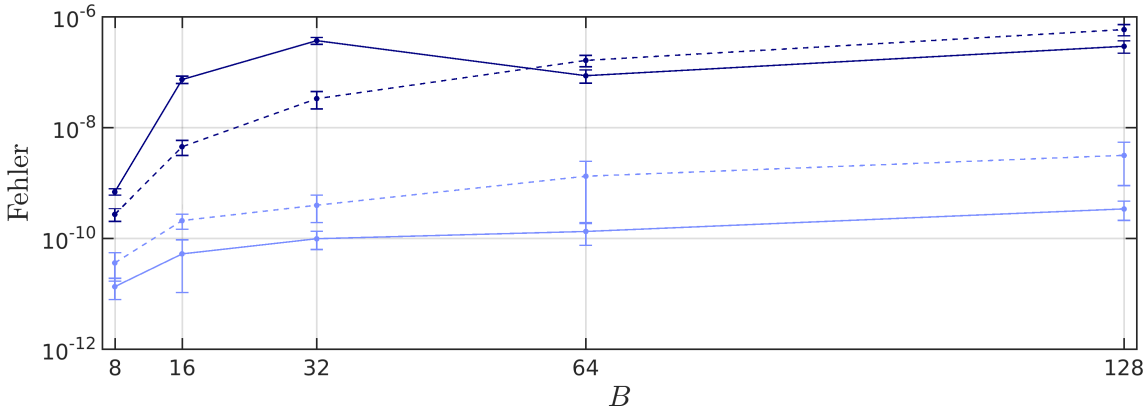

Abbildung 3.3. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-A bei $M = 1000$ gleichverteilten Punkten in $\mathbb{B}_5^3$ mit $\sigma = 2$ und $q = 12$ in Abhängigkeit von der Bandbreite B. Die gestrichelten Linien zeigen das Ergebnis unter Verwendung der iNDFT anstelle der iNFFT.

Ende hin ein Kleinerwerden des absoluten Fehlers ansatzweise zu erkennen, in keinem Fall wurde jedoch ein relativer Fehler kleiner als Eins erzielt; dazu trägt vermutlich auch die initiale Schätzung der SGL-Fourier-Koeffizienten bei, bei der dies ebensowenig der Fall war. Insgesamt scheint das Konvergenzverhalten nicht wesentlich von der Anzahl M an gegebenen Funktionswerten $f(x_i)$ abzuhängen, sofern sich die SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ aus diesen Funktionswerten rekonstruieren lassen. Die Streuweite der Punkte x_i scheint einen deutlich größeren Einfluss auf das Konvergenzverhalten zu haben (Abb. 3.5). Dies macht sich schon bei der initialen Schätzung der SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ bemerkbar: Sowohl für $B = 8$, als auch für $B = 16$ war die initiale Schätzung bei $\kappa = 8$ besser als bei $\kappa = 6$ und $\kappa = 10$; bei $\kappa = 10$ war sie immer noch deutlich besser als bei $\kappa = 6$. Interessant ist, dass im Fall $B = 8$ (Abb. 3.5, links) der Fehler bei $\kappa = 8$ und $\kappa = 10$ nach einer gewissen Zeit beginnt, größer zu werden; es ist daher wichtig zu bemerken, dass trotz dieses Größerwerdens das *Residuum* innerhalb des CGNR-Verfahrens abnahm. Dies ist ein Indiz für die Schlechtgestelltheit des Problems. Im Fall $B = 16$ (Abb. 3.5, rechts) ist dieses Verhalten nicht zu beobachten. Hier wurde im Fall $\kappa = 6$ wieder kein relativer Fehler kleiner als Eins erzielt. In den Fällen $\kappa = 8$ und $\kappa = 10$ war das Verfahren erfolgreicher.

Zusammenfassend zeigen die obigen Ergebnisse, dass es sich bei der iNFSGLFT-A und ihrer Adjungierten um praxistaugliche Algorithmen handelt, die einen deutlichen Laufzeitvorteil von weniger als einer halben Minute gegenüber fast vier Stunden bei $B = 64$ und $M = 100\,000$ bieten können. Der Fehler dieser approximativen Algorithmen ist relativ klein, sofern die Punkte x_i nicht zu weit gestreut sind. Die aus der iNFSGLFT-A und ihrer Adjungierten abgeleitete NFSGLFT-A liefert in einigen der untersuchten Fälle Ergebnisse mit einem relativ kleinen Fehler, jedoch sind hier Verbesserungen notwendig, um die Konvergenzgeschwindigkeit zu erhöhen. Dieses Problem des CGNR- und auch CGNE-Verfahrens ist bekannt (vgl. [Golub und van Loan, 1996, S. 546]). Einen Ansatzpunkt für Weiterentwicklungen stellen die Untersuchungen in [Kunis, 2006, Kap. 5] dar.

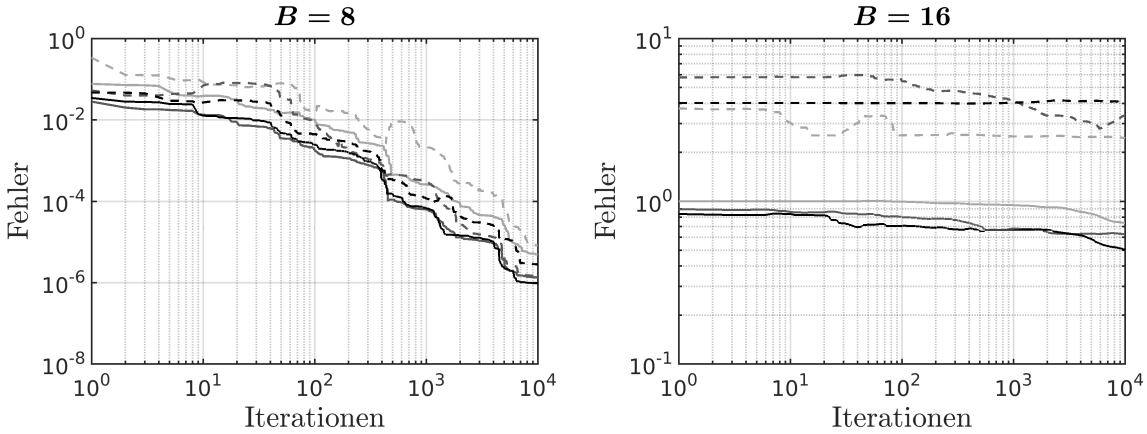

Abbildung 3.4. Maximaler **(durchgezogen)** absoluter und **(gestrichelt)** relativer Fehler der NFSGLFT-A in Abhängigkeit von der Anzahl an Iterationen mit $\kappa = 5$ und **(hellgrau)** $N = 25$, **(grau)** $N = 50$, **(schwarz)** $N = 100$.

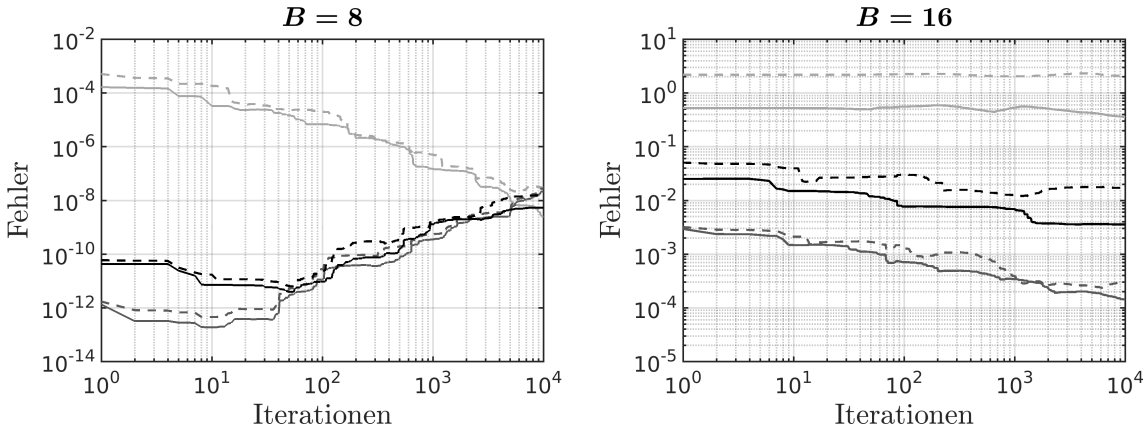

Abbildung 3.5. Maximaler **(durchgezogen)** absoluter und **(gestrichelt)** relativer Fehler der NFSGLFT-A in Abhängigkeit von der Anzahl an Iterationen mit $N = 50$ und **(hellgrau)** $\kappa = 6$, **(grau)** $\kappa = 8$, **(schwarz)** $\kappa = 10$.

3.3 NFSGLFT-B: direkte Verallgemeinerung der NFFT

Nachdem wir in den vorherigen Abschnitten die NFSGLFT-A hergeleitet, besprochen und ausführlich getestet haben, wenden wir uns jetzt der zweiten Klasse von NFSGLFTs, der Variante B, zu. Unsere Herangehensweise ist hierbei eine ganz andere als bei der Variante A: während wir dort die iNFFT und ihre Adjungierte direkt zum Einsatz gebracht haben, übertragen wir nun die Idee der NFFT unmittelbar auf den SGL-Fall. Konkret bedeutet dies, dass wir einen Approximanten s an die auszuwertende Funktion f bestimmen, der sich im „Ortsraum“ mit weniger Aufwand auswerten lässt als die Funktion f selbst. Als Ansatz für s wählen wir hier eine Linearkombination von Translaten einer Gaußschen Ansatzfunktion, da eine solche mithilfe der schnellen Gauß-Transformation (Abschn. 3.3.2) effizient ausgewertet werden kann. Die gesuchten Skalierungskoeffizienten für den Approximanten s werden – gewissermaßen „im Frequenzraum“ – so bestimmt,

dass die SGL-Fourier-Koeffizienten $\hat{s}_{nlm}$ von s bis zu einer vorgegebenen Bandbreite mit den SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ von f übereinstimmen. Dies erreichen wir, indem wir spezielle Ansatzpunkte für die Translate der Gaußschen Ansatzfunktion wählen, wofür wir uns insbesondere das Hintergrundwissen zur schnellen Fourier-Transformation auf der Rotationsgruppe SO(3) zunutze machen (Abschn. 3.3.1). Bei der Bestimmung der Skalierungskoeffizienten für s kommen dann der Clenshaw-Algorithmus bzw. eine adjungierte FDPT, sowie die klassische iFFT zum Einsatz.

3.3.1 Schnelle SO(3)-Fourier-Transformation (FSOFT)

In Abschnitt 1.6 haben wir festgestellt, dass das Spektralverhalten der SGL-Basisfunktionen unter Rotationen mithilfe der Wigner-D-Funktionen beschrieben werden kann. Die Wigner-D-Funktionen selbst haben wir im Zusammenhang mit der Rotationsgruppe SO(3) und den Kugelflächenfunktionen bereits in Abschnitt 1.1 kennengelernt (Def. 1.1.16). Eine jede Rotation $R \in \mathrm{SO}(3)$ lässt sich bekanntlich in eine Rotation um die ξ_0-Achse, eine anschließende Rotation um die ξ_1-Achse und eine Rotation um die ξ_2-Achse zerlegen (vgl. Abb. 1.1). Rotationen um die ξ_1- bzw. ξ_2-Achse werden von den Drehmatrizen

$$R_1(\omega) := \begin{bmatrix} \cos\omega & 0 & \sin\omega \\ 0 & 1 & 0 \\ -\sin\omega & 0 & \cos\omega \end{bmatrix}, \quad R_2(\omega) := \begin{bmatrix} \cos\omega & -\sin\omega & 0 \\ \sin\omega & \cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}, \qquad \omega \in [0, 2\pi),$$

repräsentiert. Eine Möglichkeit, die Rotationsgruppe SO(3) unter Zuhilfenahme nur dieser Matrizen zu parametrisieren, ist durch folgendes bekanntes Lemma gegeben:

Lemma 3.3.1 (s. [Kostelec und Rockmore, 2008, S. 148]). *Es sei $R \in \mathrm{SO}(3)$. Dann besitzt die Rotation R die Zerlegung $R = R_2(\gamma)R_1(\beta)R_2(\alpha)$ mit gewissen $\alpha \in [0, 2\pi)$, $\beta \in [0, \pi]$ und $\gamma \in [0, 2\pi)$.*

Diese Zerlegung von R nennt man *Euler-Winkel-Zerlegung*[5], die Winkel α, β und γ heißen *Euler-Winkel*. Eine Rotation $R = R(\alpha, \beta, \gamma)$ wird hierbei also zerlegt in eine Rotation um die ξ_2-Achse mit dem Winkel α, eine anschließende Rotation um die ξ_1-Achse mit dem Winkel β und eine abschließende Rotation um die ξ_2-Achse mit dem Winkel γ. Wir schreiben im Folgenden für eine Funktion f auf SO(3) der Einfachheit halber $f(R(\alpha, \beta, \gamma)) = f(\alpha, \beta, \gamma)$.

Wir betrachten nun den Raum der über SO(3) quadratintegrierbaren Funktionen,

$$L^2(\mathrm{SO}(3)) := \left\{ f : \mathrm{SO}(3) \to \mathbb{C} \,:\, f \ \rho\text{-messbar und} \int_{\mathrm{SO}(3)} |f(R)|^2 \,\mathrm{d}\rho(R) < \infty \right\},$$

mit dem bis auf die Normierung eindeutig bestimmten Haar-Maß[6] ρ auf SO(3). In diesem Hilbert-Raum werden Funktionen, die ρ-fast überall identisch sind, als gleich angesehen. Mithilfe der Euler-Winkel lässt sich das dazugehörige Skalarprodukt in der Form

$$\langle f, g \rangle_{L^2(\mathrm{SO}(3))} = \int_0^{2\pi} \int_0^{\pi} \int_0^{2\pi} f(\alpha, \beta, \gamma) \overline{g(\alpha, \beta, \gamma)} \,\mathrm{d}\alpha \sin\beta \,\mathrm{d}\beta \,\mathrm{d}\gamma, \quad f, g \in L^2(\mathrm{SO}(3)),$$

schreiben. Die induzierte Norm bezeichnen wir mit $\|\cdot\|_{L^2(\mathrm{SO}(3))}$.

[5] Leonhard Euler (⋆1707 in Basel, Schweiz, †1783 in St. Petersburg, Russland)
[6] Alfréd Haar (⋆1885 in Budapest, Ungarn, †1933 in Szeged, ebd.)

Neben dem Obenerwähnten liegt die besondere Bedeutung der Wigner-D-Funktionen in der Tatsache, dass diese Funktionen der Orthogonalitätsrelation

$$\langle D^{(l)}_{mm''}, D^{(l')}_{m'm'''}\rangle_{L^2(\mathrm{SO}(3))} = \frac{8\pi^2}{2l+1}\,\delta_{ll'}\,\delta_{mm'}\,\delta_{m''m'''} \tag{3.32}$$

genügen [Kostelec und Rockmore, 2008, S. 149]; es gilt sogar noch mehr:

Satz 3.3.2. *Die Gesamtheit der Wigner-D-Funktionen bildet ein vollständiges Orthogonalsystem in* $L^2(\mathrm{SO}(3))$.

Es lässt sich dementsprechend jede Funktion $f \in L^2(\mathrm{SO}(3))$ nach den Wigner-D-Funktionen entwickeln:

$$\lim_{n\to\infty}\left\| f - \sum\nolimits_{|m|,|m'|\le l<n} \frac{\langle f, D^{(l)}_{mm'}\rangle_{L^2(\mathrm{SO}(3))}}{\|D^{(l)}_{mm'}\|^2_{L^2(\mathrm{SO}(3))}}\, D^{(l)}_{mm'}\right\|_{L^2(\mathrm{SO}(3))} = 0.$$

Nach Normierung der Wigner-D-Funktionen erhält man auch hier die entsprechende Parsevalsche Gleichung. Es gilt ferner das folgende SO(3)-Abtasttheorem, das Satz 2.1.2 verallgemeinert.

Satz 3.3.3 (s. [Kostelec und Rockmore, 2008, Satz 1]). *Es seien* $n \in \mathbb{N}$ *und* $f \in \operatorname{span}\{D^{(l)}_{mm'} : |m|, |m'| \le l < n\}$. *Dann genügen die* SO(3)*-Fourier-Koeffizienten von* f *der Quadraturformel*

$$\langle f, D^{(l)}_{mm'}\rangle_{L^2(\mathrm{SO}(3))} = \frac{2l+1}{8\pi n}\sum_{i,j,k=0}^{2n-1} \chi_j\, f(\alpha_i,\beta_j,\gamma_k)\,\overline{D^{(l)}_{mm'}(\alpha_i,\beta_j,\gamma_k)}, \tag{3.33}$$

$|m|, |m'| \le l < n$, *mit den Abtastwinkeln* $\alpha_i := i\pi/n$, $\beta_j := (2j+1)\pi/4n$ *und* $\gamma_k := \alpha_k$, *sowie den sphärischen Quadraturgewichten* $\chi_j = \chi_{j,n}$ *aus Satz 2.1.2, die dort mit* $\beta_{j,n}$ *bezeichnet werden.*

Man beachte, dass aus Lemma 2.1.3 sofort $\chi_j > 0$ folgt. Ausgehend von Satz 3.3.3 definieren wir nun die diskrete SO(3)-Fourier-Transformation. Dazu identifizieren wir das Tripel $[l, m, m']$; $|m|, |m'| \le l < n$, mit $\sigma \in \{0, \ldots, n(4n^2-1)/3 - 1\}$ mittels des bijektiven Zusammenhangs

$$\sigma = \frac{(4l^2-1)l}{3} + (2l+1)(l+m) + l + m'.$$

Dies erlaubt es uns, die Indizes l, m, und m' als Funktionen von σ aufzufassen:

$$\begin{cases} l(\sigma) &= \left\lfloor \dfrac{1}{\sqrt{3}}\cosh\left(\dfrac{1}{3}\operatorname{arcosh}\left(9\sqrt{3}\sigma\right)\right)\right\rfloor, \\ m(\sigma) &= \left\lfloor \dfrac{1}{2l(\sigma)+1}\left(\sigma - \dfrac{(4l(\sigma)^2-1)l(\sigma)}{3}\right)\right\rfloor - l(\sigma), \\ m'(\sigma) &= \sigma - \dfrac{(4l(\sigma)^2-1)l(\sigma)}{3} - (2l(\sigma)+1)(l(\sigma)+m(\sigma)) - l(\sigma). \end{cases} \tag{3.34}$$

Darüber hinaus identifizieren wir $[i,j,k] \in \{0,\dots,2n-1\}^3$ mit $\tau \in \{0,\dots,(2n)^3-1\}$ vermöge des eineindeutigen Zusammenhangs

$$\tau \;=\; (2n)^2 i + 2nj + k.$$

Wir können die Indizes i, j und k dann als Funktionen von τ verstehen (man ersetze auf S. 38 oben B durch n und ν durch τ). In Analogie zur DSFT, DSGLFT, etc. definieren wir jetzt:

Definition 3.3.4 (DSOFT). Es seien $n \in \mathbb{N}$ und

$$\begin{aligned}
U_n &:= \operatorname{diag}\left[\tfrac{2l(\sigma)+1}{8\pi n}\right]_{\sigma=0,\dots,n(4n^2-1)/3-1} \in \mathbb{R}^{n(4n^2-1)/3\times n(4n^2-1)/3},\\
V_n &:= \operatorname{diag}\left[\chi_{j(\tau)}\right]_{\tau=0,\dots,(2n)^3-1} \in \mathbb{R}^{(2n)^3\times(2n)^3},\\
W_n &:= \left[\,\overline{D^{(l(\sigma))}_{m(\sigma),m'(\sigma)}\big(\alpha_{i(\tau)},\beta_{j(\tau)},\gamma_{k(\tau)}\big)}\,\right]_{\substack{\sigma=0,\dots,n(4n^2-1)/3-1\\ \tau=0,\dots,(2n)^3-1}} \in \mathbb{C}^{n(4n^2-1)/3\times(2n)^3}.
\end{aligned}$$

Die Abbildung $U_nW_nV_n : \mathbb{C}^{(2n)^3} \to \mathbb{C}^{n(4n^2-1)/3}$ heißt *diskrete* SO(3)*-Fourier-Transformation* (DSOFT für *discrete* SO(3) *Fourier transform*).

Ist nun $f \in \operatorname{span}\{D^{(l)}_{mm'} : |m|,|m'| \le l < n\}$ mit $n \in \mathbb{N}$, so definieren wir ohne Verwechslungsgefahr den zu f gehörigen „Ortsvektor“ $f = [f(\alpha_{i(\tau)},\beta_{j(\tau)},\gamma_{k(\tau)})]_{\tau=0,\dots,(2n)^3-1}$ und den entsprechenden Fourier-Vektor $\tilde{f} = [\langle f, D^{(l(\sigma))}_{m(\sigma),m'(\sigma)}\rangle_{L^2(\mathrm{SO}(3))}]_{\sigma=0,\dots,n(4n^2-1)/3-1}$. Es besteht dann der Zusammenhang

$$U_nW_nV_nf \;=\; \tilde{f}.$$

Die DSOFT erlaubt uns also, die SO(3)-Fourier-Koeffizienten von f auf diskrete Art und Weise zu berechnen. Setzen wir hier wieder eine fehlerfreie Arithmetik voraus, so ist diese Berechnung exakt. Es folgt aus dem offensichtlichen Zusammenhang $W_n^{\mathrm{H}}\tilde{f} = f$ außerdem:

Lemma 3.3.5. *Es seien $n \in \mathbb{N}$ und $f \in \operatorname{span}\{D^{(l)}_{mm'} : |m|,|m'| \le l < n\}$. Weiter seien die zu f gehörigen Vektoren f und $\tilde{f}$ definiert wie oben. Dann gilt*

$$\begin{aligned}
W_n^{\mathrm{H}}(U_nW_nV_n)f \;&=\; f,\\
(U_nW_nV_n)W_n^{\mathrm{H}}\tilde{f} \;&=\; \tilde{f}.
\end{aligned}$$

Die naive Durchführung einer DSOFT benötigt offenbar $\mathcal{O}(n^6)$ Schritte. Der Schlüssel zur *schnellen* SO(3)*-Fourier-Transformation* (FSOFT für *fast* SO(3) *Fourier transform*) von Kostelec und Rockmore liegt nun in folgendem Lemma.

Lemma 3.3.6 (vgl. [Biedenharn und Louck, 1981, Gln. 3.59 u. 3.72]). *Die Wigner-D-Funktionen besitzen die Darstellung*

$$D^{(l)}_{mm'}(\alpha,\beta,\gamma) \;=\; \mathrm{e}^{-\mathrm{i}m\alpha}\, d^{(l)}_{mm'}(\beta)\,\mathrm{e}^{-\mathrm{i}m'\gamma}, \tag{3.35}$$

jeweils mit einer reellwertigen Wigner-d-Funktion $d^{(l)}_{mm'} : [0,\pi] \to \mathbb{R}$,

$$d^{(l)}_{mm'}(\beta) := (-1)^{m+m'}\sqrt{\frac{(l+m')!}{(l+m)!}\frac{(l-m')!}{(l-m)!}}\left(\sin\frac{\beta}{2}\right)^{m'-m}\left(\cos\frac{\beta}{2}\right)^{m+m'} P^{(m'-m,m+m')}_{l-m'}(\cos\beta),$$

wobei $P_n^{(\alpha,\beta)}$ die klassischen Jacobi-Polynome mit $P_n^{(\alpha,\beta)}(1) = \binom{n+\alpha}{n}$ bezeichnen (vgl. [Abramowitz und Stegun, 1972, Gl. 22.2.1]).

Ein Einsetzen der Darstellung (3.35) in die Quadraturformel (3.33) ergibt nach einem Umsortieren der Summanden

$$\langle f, D_{mm'}^{(l)} \rangle_{L^2(\mathrm{SO}(3))} = \frac{2l+1}{8\pi n} \sum_{j=0}^{2n-1} \chi_j \, d_{mm'}^{(l)}(\beta_j) \sum_{i=0}^{2n-1} \mathrm{e}^{\mathrm{i}m\alpha_i} \sum_{k=0}^{2n-1} f(\alpha_i, \beta_j, \gamma_k) \mathrm{e}^{\mathrm{i}m'\gamma_k}.$$

Diese Separation der Variablen reduziert die Komplexität der Durchführung einer DSOFT auf $\mathcal{O}(n^4)$: Die innerste Summe kann bei jeweils festem i und j mithilfe der iFFT in $\mathcal{O}(n \log n)$ Schritten für alle m' berechnet werden. Es ergibt sich eine Komplexität von $\mathcal{O}(n^3 \log n)$ für diesen ersten Schritt. Anschließend wird mit der iFFT die mittlere Summe bei jeweils festem i und m' für alle m berechnet. Auch dieser Schritt hat eine Komplexität von $\mathcal{O}(n^3 \log n)$. Da die Jacobi-Polynome $P_n^{(\alpha,\beta)}$ einer Drei-Term-Rekursion in n genügen (s. [Abramowitz und Stegun, 1972, Gl. 22.7.1]), kann nun die äußere Summe bei jeweils festem m und m' mit dem adjungierten Clenshaw-Algorithmus in $\mathcal{O}(n^2)$ Schritten für alle l ausgewertet werden. Dieser letzte Schritt hat eine Komplexität von $\mathcal{O}(n^4)$. Die Speicherkomplexität dieser FSOFT ist offensichtlich $\mathcal{O}(n^3)$. Verwendet man anstelle des adjungierten Clenshaw-Algorithmus eine FDPT, so reduziert sich die Rechenkomplexität bei gleichbleibender Speicherkomplexität sogar auf $\mathcal{O}(n^3 \log^2 n)$.

Wir verzichten hier auf eine Darstellung der oben beschriebenen FSOFT in Pseudocode. Stattdessen stellen wir sie als spezielle Matrix-Faktorisierung dar. Dafür definieren wir die *Wigner-d-Matrizen*

$$D_{mm'} = D_{mm',n} := \left[d_{mm'}^{(l)}(\beta_j) \right]_{\substack{l=\max\{|m|,|m'|\},\ldots,n-1 \\ j=0,\ldots,2n-1}} \in \mathbb{R}^{(n-\max\{|m|,|m'|\})\times 2n}, \tag{3.36}$$

$|m|, |m'| < n$. Darüber hinaus verwenden wir die Fourier-Matrix F_n aus Abschnitt 2.2.1, sowie die dazugehörigen Hilfsmatrizen A_n und B_n aus Abschnitt 2.2.4. Als erste von drei Permutationsmatrizen definieren wir

$$S_n := \left[e_{\eta(\zeta)}^{\mathrm{T}} \right]_{\zeta=0,\ldots,(2n)^2(2n-1)-1} \in \mathbb{R}^{(2n)^2(2n-1)\times(2n)^2(2n-1)}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und

$$\eta(\zeta) := 2n(2n-1)i(\zeta) + (2n-1)j(\zeta) + (n + m'(\zeta) - 1),$$

wobei hier

$$\begin{aligned} m'(\zeta) &:= \zeta \bmod (2n-1) - n + 1, \\ j(\zeta) &:= \frac{\zeta - (n + m'(\zeta) - 1)}{2n-1} \bmod 2n, \\ i(\zeta) &:= \frac{\zeta - (n + m'(\zeta) - 1) - (2n-1)j(\zeta)}{2n(2n-1)}, \end{aligned}$$

gilt. Die Matrix S_n sorgt für die Umsortierung von $i, j = 0, \ldots, 2n-1$, $|m'| < n$, hin zu

$j = 0, \ldots, 2n-1$, $|m'| < n$, $j = 0, \ldots, 2n-1$. Des Weiteren benötigen wir die Permutationsmatrix

$$Q_n \coloneqq \left[e_{\varphi(\vartheta)}^{\mathrm{T}}\right]_{\vartheta=0,\ldots,2n(2n-1)^2-1} \in \mathbb{R}^{2n(2n-1)^2 \times 2n(2n-1)^2}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und

$$\varphi(\vartheta) \coloneqq (2n-1)^2 i(\vartheta) + (2n-1)(n + m'(\vartheta) - 1) + n + m(\vartheta) - 1,$$

wobei hier

$$\begin{aligned} i(\vartheta) &\coloneqq \vartheta \bmod 2n, \\ m(\vartheta) &\coloneqq \frac{\vartheta - i(\vartheta)}{2n} \bmod (2n-1) - n + 1, \\ m'(\vartheta) &\coloneqq \frac{\vartheta - i(\vartheta) - 2n(n + m(\vartheta) - 1)}{2n(2n-1)} - n + 1, \end{aligned}$$

gilt. Die Matrix Q_n leistet die Umsortierung von $j = 0, \ldots, 2n-1$; $|m'|, |m| < n$ hin zu $|m'|, |m| < n$, $j = 0, \ldots, 2n-1$. Als letzte Permutationsmatrix definieren wir noch

$$P_n \coloneqq \left[e_{\psi(\sigma)}^{\mathrm{T}}\right]_{\sigma=0,\ldots,n(4n^2-1)/3-1} \in \mathbb{R}^{n(4n^2-1)/3 \times n(4n^2-1)/3}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge und (s. Gln. 3.34)

$$\begin{aligned} \psi(\sigma) \coloneqq\ & n^2(n + m'(\sigma) - 1) + n(n + m(\sigma) - 1) - \frac{n(n-1)(2n-1)}{6} + l(\sigma) \\ & - \frac{(n - \max\{|m'(\sigma)|, 1 - m(\sigma)\})(n + \max\{|m'(\sigma)|, 1 - m(\sigma)\} - 1)}{2} \\ & - \frac{\max\{m(\sigma) - |m'(\sigma)|, 0\}(m(\sigma) + |m'(\sigma)| - 1)}{2} - \max\{|m'(\sigma)|, |m(\sigma)|\} \\ & - \max\{\min\{2|m'(\sigma)|, m(\sigma) + |m'(\sigma)|\} - 1, 0\}|m'(\sigma)| + \operatorname{sgn}(1 - m'(\sigma)) \\ & \times \left\{ \frac{|m'(\sigma)|(|m'(\sigma)| + 1)(2|m'(\sigma)| + 1)}{6} + \min\{\operatorname{sgn}(1 - m'(\sigma)), 0\}|m'(\sigma)|^2 \right\}. \end{aligned}$$

Die Matrix P_n sorgt für die Umsortierung von $|m'|, |m| < n$, $l = \max\{|m|, |m'|\}, \ldots, n-1$, hin zu $|m|, |m'| \leq l < n$. Es ergibt sich nun mit den obigen Bestandteilen aus den Überlegungen dieses Abschnitts insgesamt das folgende Resultat, das die FSOFT charakterisiert.

Satz 3.3.7. *Die Matrix W_n in Definition 3.3.4 besitzt die Faktorisierung*

$$\begin{aligned} W_n \ =\ & (2n)^2 P_n \cdot \overbrace{\begin{bmatrix} \tilde{D}_{1-n} & & \\ & \ddots & \\ & & \tilde{D}_{n-1} \end{bmatrix}}^{2n-1 \text{ Blöcke (s. u.)}} \cdot Q_n \cdot \underbrace{\left\{E_{2n(2n-1)} \otimes \left(A_n F_{2n}^{-1} B_n^{\mathrm{H}}\right)\right\}}_{2n(2n-1) \text{ Blöcke der Größe } (2n-1)\times 2n} \\ & \times S_n \cdot \underbrace{\left\{E_{(2n)^2} \otimes \left(A_n F_{2n}^{-1} B_n^{\mathrm{H}}\right)\right\}}_{(2n)^2 \text{ Blöcke der Größe } (2n-1)\times 2n} \end{aligned}$$

mit den Blockmatrizen

$$\tilde{D}_{m'} := \underbrace{\begin{bmatrix} D_{1-n,m'} & & \\ & \ddots & \\ & & D_{n-1,m'} \end{bmatrix}}_{2n-1 \text{ Blöcke } D_{mm'} \text{ der Größe } (n-\max\{|m|,|m'|\})\times 2n}, \qquad |m'| < n.$$

Die Matrizen $D_{mm'}$ können nun selbst wieder faktorisiert werden; bei Verwendung des adjungierten Clenshaw-Algorithmus ergibt sich die Faktorisierung aus Folgerung 2.2.23. Durch Umkehren der Produktreihenfolge und hermitesches Transponieren der einzelnen Faktoren erhalten wir unter Berücksichtigung der Rechenregeln für das Kronecker-Produkt zusammen mit Lemma 2.2.4 und Lemma 3.3.5 nun abschließend noch das folgende Resultat. Es zeigt uns, dass wir auch über eine iFSOFT mit der gleichen Rechen- und Speicherkomplexität wie die der FSOFT verfügen.

Folgerung 3.3.8. *Die Matrix W_n^{H} besitzt die Faktorisierung*

$$\begin{aligned} W_n^{\mathrm{H}} &= \left\{E_{(2n)^2} \otimes \left(B_n F_{2n} A_n^{\mathrm{T}}\right)\right\} \cdot S_n^{\mathrm{T}} \cdot \left\{E_{2n(2n-1)} \otimes \left(B_n F_{2n} A_n^{\mathrm{T}}\right)\right\} \\ &\quad \times Q_n^{\mathrm{T}} \cdot \begin{bmatrix} \tilde{D}_{1-n}^{\mathrm{T}} & & \\ & \ddots & \\ & & \tilde{D}_{n-1}^{\mathrm{T}} \end{bmatrix} \cdot P_n^{\mathrm{T}}. \end{aligned}$$

3.3.2 Schnelle Gauß-Transformation (FGT)

Die *schnelle Gauß-Transformation* (FGT für *fast Gauss transform*) gehört zu einer größeren Klasse von Algorithmen für die schnelle Summation radialer Funktionen. Man betrachtet dabei einen radialen Kern $K : \mathbb{R}^d \to \mathbb{C}$ ($d \in \mathbb{N}$),

$$K(x) = k(\|x\|_2),$$

mit gegebenem $k : [0,\infty) \to \mathbb{C}$. Zusätzlich zum Kern K bzw. k sind Ansatzpunkte $y_0, \ldots, y_{m-1} \in \mathbb{R}^d$ und Skalierungskoeffizienten $\alpha_0, \ldots, \alpha_{m-1} \in \mathbb{C}$, sowie Punkte $x_0, \ldots, x_{l-1} \in \mathbb{R}^d$ gegeben ($l, m \in \mathbb{N}$). Das Ziel ist die effiziente Berechnung der Summen

$$S_i := \sum_{j=0}^{m-1} \alpha_j K(x_i - y_j), \qquad i = 0, \ldots, l-1. \tag{3.37}$$

Lässt sich die Funktion k an beliebiger Stelle in $\mathcal{O}(1)$ Schritten auswerten, so benötigt eine naive Berechnung der Summen S_i offenbar $\mathcal{O}(lm)$ Operationen. Das Problem lässt sich in Matrix-Vektor-Schreibweise als

$$\begin{bmatrix} S_0 \\ \vdots \\ S_{l-1} \end{bmatrix} = \underbrace{\begin{bmatrix} K(x_0 - y_0) & \cdots & K(x_0 - y_{m-1}) \\ \vdots & & \vdots \\ K(x_{l-1} - y_0) & \cdots & K(x_{l-1} - y_{m-1}) \end{bmatrix}}_{=:\, G} \cdot \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{m-1} \end{bmatrix} \tag{3.38}$$

darstellen, wobei $G = G(x_0, \ldots, x_{l-1}; y_0, \ldots, y_{m-1}) \in \mathbb{C}^{l \times m}$.

Definition 3.3.9. Es sei $\gamma > 0$. Wir definieren den radialen *Gauß-Kern* K_γ vermöge $k_\gamma(\xi) := \exp(-\gamma\xi^2)$, $\xi \in [0,\infty)$. Die Zahl γ wird *Spreizfaktor* genannt. Die zu K_γ gehörige Abbildung $G = G(\gamma; x_0, \ldots, x_{l-1}; y_0, \ldots, y_{m-1}) : \mathbb{C}^m \to \mathbb{C}^l$ heißt *diskrete Gauß-Transformation* (DGT für *discrete Gauss transform*).

Wir wollen im Folgenden eine spezielle FGT besprechen, die in [Kunis et al., 2006] vorgestellt wurde und die auf der in Abschnitt 3.1 besprochenen iNFFT und ihrer Adjungierten basiert. Wir werden diesen schnellen Algorithmus im nächsten Abschnitt innerhalb der iNFSGLFT-B verwenden. Für weitere Informationen zur schnellen Summation radialer Funktionen sei auf [Kunis et al., 2006] und die darin enthaltenen Quellen verwiesen (vgl. auch [Potts, 2003, Kap. 2] und [Kunis, 2006, Kap. 4]).

Es sei $\rho > 0$ so gewählt, dass die Punkte x_i und y_j in einem Würfel $[0,\rho)^d + t$, $t \in \mathbb{R}^3$, enthalten sind. Periodisiert man den Gauß-Kern K_γ mit der Periode ρ durch Summation, so erhält man

$$\tilde{K}_\gamma(x) := \sum\nolimits_{z\in\mathbb{Z}^d} K_\gamma(x - \rho z).$$

Dieser periodisierte Kern besitzt nach der Poissonschen Summenformel die Fourier-Reihe

$$\tilde{K}_\gamma(x) = \sum\nolimits_{k\in\mathbb{Z}^d} \omega_k \, \mathrm{e}^{2\pi\mathrm{i}\langle k,x\rangle_2/\rho} \tag{3.39}$$

mit den Fourier-Koeffizienten

$$\omega_k = \left(\frac{\pi}{\gamma\rho^2}\right)^{d/2} \mathrm{e}^{-\pi^2\|k\|_2^2/\gamma\rho^2}. \tag{3.40}$$

Die Gleichheit in (3.39) darf punktweise verstanden werden. Dies ergibt sich hier aus der Produkt-Struktur von K_γ,

$$K_\gamma(x) = k_\gamma(\|x\|_2) = k_\gamma(\xi_0) \cdots k_\gamma(\xi_{d-1}), \qquad x = [\xi_0, \ldots, \xi_{d-1}] \in \mathbb{R}^d,$$

zusammen mit der punktweisen und absoluten Konvergenz von

$$\sum\nolimits_{\zeta\in\mathbb{Z}} k_\gamma(\xi - \rho\zeta) = \sum\nolimits_{\kappa\in\mathbb{Z}} \psi_\kappa \, \mathrm{e}^{2\pi\mathrm{i}\kappa\xi/\rho}$$

mit

$$\psi_\kappa = \frac{\sqrt{\pi}}{\sqrt{\gamma}\rho} \, \mathrm{e}^{-(\pi\kappa)^2/\gamma\rho^2} \tag{3.41}$$

[Kunis, 2006, Sätze 2.3, 5; Lem. 4.2].

Ein Approximant an $\tilde{K}_\gamma$ ist durch die Fourier-Partialsumme

$$\tilde{K}_{\gamma,n}(x) = \sum\nolimits_{k\in I_n^d} \omega_k \, \mathrm{e}^{2\pi\mathrm{i}\langle k,x\rangle_2/\rho} \tag{3.42}$$

gegeben ($n \in \mathbb{N}$ gerade). Hierbei benutzen wir wieder das Gitter aus Definition 3.1.1. Die Idee besteht nun darin, anstelle der Summen S_i in (3.37) die Summen

$$\tilde{S}_i := \sum_{j=0}^{m-1} \alpha_j \, \tilde{K}_{\gamma,n}(x_i - y_j), \qquad i = 0, \ldots, l-1. \tag{3.43}$$

zu berechnen. Setzt man dazu zunächst die Fourier-Partialsumme aus (3.42) ein, und ändert anschließend die Summationsreihenfolge, so ergibt sich

$$\tilde{S}_i \;=\; \sum\nolimits_{k\in I_n^d} \omega_k \left(\sum_{j=0}^{m-1} \alpha_j \, \mathrm{e}^{2\pi \mathrm{i}\langle k, y_j\rangle_2/\rho} \right) \mathrm{e}^{-2\pi \mathrm{i}\langle k, x_i\rangle_2/\rho}, \qquad i = 0, \ldots, l-1. \tag{3.44}$$

Die inneren Summen können unter Verwendung der adjungierten iNFFT aus Abschnitt 3.1 in $\mathcal{O}((\sigma n)^d \log(\sigma n) + mq^d)$ Schritten berechnet werden. Anschließend werden die äußeren Summen mithilfe der iNFFT in $\mathcal{O}((\sigma n)^d \log(\sigma n) + lq^d)$ Schritten ausgewertet. Man beachte dabei jeweils das Vorzeichen im Exponenten. Wir gehen hier der Einfachheit halber davon aus, dass jeweils der gleiche Überabtastungsfaktor σ und der gleiche Abschneideparameter q verwendet werden. Es ergibt sich eine Rechenkomplexität von insgesamt $\mathcal{O}((\sigma n)^d \log(\sigma n) + q^d(l+m))$. Die Speicherkomplexität ist $\mathcal{O}((\sigma n)^d)$.

Die Berechnung der Summen in (3.43) lässt sich als Auswertung des speziellen Matrix-Vektor-Produkts

$$\begin{bmatrix} \tilde{S}_0 \\ \vdots \\ \tilde{S}_{l-1} \end{bmatrix} = \underbrace{\begin{bmatrix} \tilde{K}_{\gamma,n}(x_0 - y_0) & \cdots & \tilde{K}_{\gamma,n}(x_0 - y_{m-1}) \\ \vdots & & \vdots \\ \tilde{K}_{\gamma,n}(x_{l-1} - y_0) & \cdots & \tilde{K}_{\gamma,n}(x_{l-1} - y_{m-1}) \end{bmatrix}}_{=:\,\tilde{G}} \cdot \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_{m-1} \end{bmatrix}$$

auffassen. Die Matrix $\tilde{G} = \tilde{G}(\gamma; n; x_0, \ldots, x_{l-1}; y_0, \ldots, y_{m-1})$ approximiert dabei die Matrix G in (3.38). Dem Übergang von (3.43) zu (3.44) entspricht nun die Matrix-Faktorisierung (vgl. [Kunis, 2006, Bem. 4.4])

$$\tilde{G} \;=\; V_1 \cdot D \cdot V_0$$

mit

$$\begin{aligned} V_0 \;&:=\; \left[\, \mathrm{e}^{2\pi \mathrm{i}\langle k(\chi), y_j\rangle_2/\rho} \,\right]_{\substack{\chi=0,\ldots,n^d-1 \\ j=0,\ldots,m-1}} \in \mathbb{C}^{n^d \times m}, \\ V_1 \;&:=\; \left[\, \mathrm{e}^{-2\pi \mathrm{i}\langle k(\chi), x_i\rangle_2/\rho} \,\right]_{\substack{i=0,\ldots,l-1 \\ \chi=0,\ldots,n^d-1}} \in \mathbb{C}^{l \times n^d} \end{aligned}$$

und

$$D \;:=\; \operatorname{diag}\Big[\omega_{k(\chi)}\Big]_{\chi=0,\ldots,n^d-1} \in \mathbb{C}^{n^d \times n^d},$$

wobei wir wieder die Index-Anordnung entsprechend (3.2) verwendet haben. Die Matrizen V_1 und V_0 lassen sich nun mithilfe der iNFFT bzw. ihrer Adjungierten selbst wieder faktorisieren (man beachte hier wieder jeweils das Vorzeichen im Exponenten). Insgesamt erhält man so auch eine Faktorisierung der Matrix $\tilde{G}^{\mathrm{H}}$, also eine „adjungierte FGT“, wobei es sich allerdings selbst wieder um die FGT handelt.

Wir halten abschließend fest, dass es sich bei der hier beschriebenen FGT schon aufgrund der darin vorkommenden iNFFT und ihrer Adjungierten offensichtlich um einen approximativen Algorithmus handelt. Wir untersuchen den Fehler in Abschnitt 3.3.5.

3.3.3 Herleitung der schnellen Algorithmen

Es seien die SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ einer bandbegrenzten Funktion $f \in H$ mit Bandbreite B gegeben. Darüber hinaus sei der *Überabtastungsfaktor* $\tau \in \mathbb{N}$ fest gewählt. Unter Verwendung des in Definition 3.3.9 eingeführten radialen Gauß-Kerns K_γ mit dem ebenfalls fest gewählten Spreizfaktor $\gamma > 0$ wählen wir als Ansatz für den schnell auszuwertenden Approximanten an die Funktion f speziell

$$s \;=\; s(B,\tau,\varrho,\gamma) \;:=\; \sum_{i,j,k=0}^{2\tau B-1} \varepsilon_{ijk}\, R(\alpha_j,\beta_k,0)^{-1}\, T_2(r_i)\, K_\gamma \tag{3.45}$$

mit den von $\varrho > 0$ abhängigen äquidistanten Radien $r_i = r_i(\varrho) := (2i+1)\varrho/(4\tau B - 1)$, sowie den Winkeln $\alpha_j := j\pi/\tau B$ und $\beta_k := (2k+1)\pi/4\tau B$. Hierbei ist T_2 der durch (1.13) gegebene Translationsoperator, während R der in (1.10) definierte Rotationsoperator mit den Euler-Winkeln aus Lemma 3.3.1 ist. Die Linearkombination in (3.45) entsteht also durch jeweiliges Verschieben des Gauß-Kerns K_γ entlang der ξ_2-Achse (vgl. Abb. 1.1) und anschließende Rotation um den Ursprung. Die Skalierungskoeffizienten ε_{ijk} sind aus den gegebenen SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ so zu bestimmen, dass die SGL-Fourier-Koeffizienten $\hat{s}_{nlm}$ von s möglichst gut mit den SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ von f übereinstimmen.

Für die SGL-Fourier-Koeffizienten $\hat{s}_{nlm}$ des Approximanten s ergibt sich aus der Definition der SGL-Basisfunktionen mit Lemma 3.3.6 in einem ersten Schritt

$$\begin{aligned}
\hat{s}_{nlm} \;&=\; \sum_{i,j,k=0}^{2\tau B-1} \varepsilon_{ijk}\, \langle T_2(r_i)\, K_\gamma, R(\alpha_j,\beta_k,0)\, H_{nlm}\rangle_H \\
&=\; \sum_{i,j,k=0}^{2\tau B-1} \varepsilon_{ijk} \sum\nolimits_{|m'|\le l} \overline{D^{(l)}_{mm'}(\alpha_j,\beta_k,0)}\, \langle T_2(r_i)\, K_\gamma, H_{nlm'}\rangle_H \\
&=\; \sum_{i=0}^{2\tau B-1} \langle T_2(r_i)\, K_\gamma, H_{nl,0}\rangle_H \sum_{k=0}^{2\tau B-1} d^{(l)}_{m,0}(\beta_k) \sum_{j=0}^{2\tau B-1} \varepsilon_{ijk}\, \mathrm{e}^{\mathrm{i}m\alpha_j}.
\end{aligned} \tag{3.46}$$

Dabei haben wir benutzt, dass $\langle T_2(r_i)K_\gamma, H_{nlm'}\rangle_H = \delta_{m',0}\langle T_2(r_i)K_\gamma, H_{nl,0}\rangle_H$. Als erstes wichtiges Hilfsmittel leiten wir eine geschlossene Darstellung für die Translationskoeffizienten $\langle T_2(r_i)\, K_\gamma, H_{nl,0}\rangle_H$ her. Wir behandeln hier sogar den allgemeinen Fall einer beliebigen Translation in $\mathbb{R}^3$. Dazu definieren wir für $y \in \mathbb{R}^3$ den allgemeinen Translationsoperator

$$T(y)f(x) \;:=\; f(x-y), \qquad f : \mathbb{R}^3 \to \mathbb{C}. \tag{3.47}$$

Mit dem kanonischen Einheitsvektor e_2 entlang der ξ_2-Achse erhalten wir $T_2(\nu) = T(\nu e_2)$. Damit enthält das folgende Resultat bereits den von uns benötigten Spezialfall.

Lemma 3.3.10. *Es sei $x = [r,\vartheta,\varphi] \in \mathbb{R}^3$. Dann gilt*

$$\langle T(x)K_\gamma, H_{nlm}\rangle_H \;=\; \left(\frac{\pi}{1+\gamma}\right)^{3/2} \left(\frac{\gamma}{1+\gamma}\right)^{n-1} N_{nl}\, \overline{Y_{lm}(\vartheta,\varphi)}\, r^l\, L^{(l+1/2)}_{n-l-1}\left(\frac{\gamma}{1+\gamma} r^2\right) \mathrm{e}^{-\frac{\gamma}{1+\gamma} r^2}.$$

Beweis. Aus der Definition der SGL-Basisfunktionen und K_γ erhalten wir mit dem Satz von Tonelli zunächst

$$\begin{aligned}\langle T(x)K_\gamma, H_{nlm}\rangle_H &= \int_{\mathbb{R}^3} T(x)[K_\gamma](y)\,\overline{H_{nlm}(y)}\,\mathrm{e}^{-\|y\|_2^2}\,\mathrm{d}y \\ &= N_{nl}\,\mathrm{e}^{-\gamma r^2}\int_0^\infty L_{n-l-1}^{(l+1/2)}(\rho^2)\,\rho^{l+2}\,\mathrm{e}^{-(1+\gamma)\rho^2}\int_{\mathbb{S}^2}\mathrm{e}^{2\gamma\rho\langle w,x\rangle_2}\,\overline{Y_{lm}(w)}\,\mathrm{d}\sigma(w)\,\mathrm{d}\rho\end{aligned}$$

mit dem Oberflächenmaß σ auf $\mathbb{S}^2$ (vgl. Abschn. 1.1). Aus Lemma 1.3.9 folgt mit der Orthonormalität der Kugelflächenfunktionen für das innere Integral auf der rechten Seite

$$\int_{\mathbb{S}^2}\mathrm{e}^{2\gamma\rho\langle w,x\rangle_2}\,\overline{Y_{lm}(w)}\,\mathrm{d}\sigma(w) = 4\pi\,\mathrm{i}^l\,j_l(-2\mathrm{i}\gamma\rho r)\,\overline{Y_{lm}(\vartheta,\varphi)}.$$

Dies ergibt zusammen mit der geschlossenen Darstellung (1.16) der verallgemeinerten Laguerre-Polynome und der Definition der sphärischen Bessel-Funktionen (Def. 1.3.4), wo wir nun auch komplexwertige Argumente zulassen,

$$\begin{aligned}\langle T(x)K_\gamma, H_{nlm}\rangle_H &= 2\mathrm{i}^{l+1/2}N_{nl}\,\overline{Y_{lm}(\vartheta,\varphi)}\,\pi^{3/2}(\gamma r)^{-1/2}\mathrm{e}^{-\gamma r^2} \\ &\quad\times\sum_{j=0}^{n-l-1}\frac{(-1)^j}{j!}\binom{n-1/2}{n-l-1-j}\int_0^\infty \rho^{l+2j+3/2}\mathrm{e}^{-(1+\gamma)\rho^2}J_{l+1/2}(-2\mathrm{i}\gamma\rho r)\,\mathrm{d}\rho.\end{aligned}$$

Indem wir in der Hankel-Formel (Lem. 1.3.3) $\alpha := -2\mathrm{i}\gamma r$, $\beta := \sqrt{1+\gamma}$, $\mu := l+2j+5/2$ und $\nu := l+1/2$ setzen, erhalten wir auf der rechten Seite

$$\begin{aligned}\int_0^\infty \rho^{l+2j+3/2}\mathrm{e}^{-(1+\gamma)\rho^2}J_{l+1/2}(-2\mathrm{i}\gamma\rho r)\,\mathrm{d}\rho &= \frac{\Gamma(l+j+3/2)}{2\Gamma(l+3/2)}(-\mathrm{i}\gamma r)^{l+1/2}(1+\gamma)^{-l-j-3/2} \\ &\quad\times {}_1F_1\left(l+j+\frac{3}{2}; l+\frac{3}{2}; \frac{(\gamma r)^2}{1+\gamma}\right).\end{aligned}$$

Man beachte hierbei, dass $\mathfrak{Re}(\mu+\nu) = 2l+2j+3 > 0$ gilt. Mit

$$\frac{\Gamma(l+j+3/2)}{j!\,\Gamma(l+3/2)} = \binom{l+j+1/2}{j}$$

und der Kummer-Transformation (Lem. 1.2.3) ergibt dies eingesetzt

$$\begin{aligned}\langle T(x)K_\gamma, H_{nlm}\rangle_H &= N_{nl}\,\overline{Y_{lm}(\vartheta,\varphi)}\,\pi^{3/2}(\gamma r)^l\,\mathrm{e}^{-\gamma r^2/(1+\gamma)}\sum_{j=0}^{n-l-1}(-1)^j\binom{n-1/2}{n-l-1-j} \\ &\quad\times(1+\gamma)^{-l-j-3/2}\binom{l+j+1/2}{j}{}_1F_1\left(-j; l+\frac{3}{2}; -\frac{(\gamma r)^2}{1+\gamma}\right).\end{aligned}$$

Indem wir in Lemma 1.4.1 die spezielle Wahl $n := j$ und $\alpha := l+1/2$ treffen, erhalten wir

$$\begin{aligned}\langle T(x)K_\gamma, H_{nlm}\rangle_H &= N_{nl}\,\overline{Y_{lm}(\vartheta,\varphi)}\,\pi^{3/2}(\gamma r)^l\,\mathrm{e}^{-\gamma r^2/(1+\gamma)} \\ &\quad\times\sum_{j=0}^{n-l-1}(-1)^j\binom{n-1/2}{n-l-1-j}(1+\gamma)^{-l-j-3/2}L_j^{(l+1/2)}\left(-\frac{(\gamma r)^2}{1+\gamma}\right).\end{aligned}$$

Die Aussage des Lemmas folgt nun mit Lemma 1.4.3, wo wir $m := n-l-1$, $\alpha := l+1/2$, $\xi := -(\gamma r)^2/(1+\gamma)$ und $\psi := -1/\gamma$ setzen. □

Folgerung 3.3.11. *Es sei $r > 0$. Dann gilt*

$$\langle T_2(r)K_\gamma, H_{nl,0}\rangle_H \;=\; \left(\frac{\pi}{1+\gamma}\right)^{3/2}\left(\frac{\gamma}{1+\gamma}\right)^{n-1} N_{nl}\, M_{l,0}\, r^l\, L_{n-l-1}^{(l+1/2)}\left(\frac{\gamma}{1+\gamma}r^2\right)\mathrm{e}^{-\frac{\gamma}{1+\gamma}r^2}$$

mit der Normierungskonstanten $M_{l,0}$ der Kugelflächenfunktion $Y_{l,0}$.

Im zweiten Schritt bringen wir nun die Transformation (3.46) in Matrix-Vektor-Schreibweise. Dazu identifizieren wir $[n,l,m]$, $|m| \le l < n \le \tau B$, wie in Abschnitt 2.2 beschrieben mit dem entsprechenden linearen Index $\mu \in \{0,\dots,\tau B(\tau B+1)(2\tau B+1)/6-1\}$. Darüber hinaus identifizieren wir $[i,j,k] \in \{0,\dots,2\tau B-1\}^3$ mit $\nu \in \{0,\dots,(2\tau B)^3-1\}$ vermöge des eineindeutigen Zusammenhangs

$$\nu \;=\; (2\tau B)^2 i + 2\tau B j + k.$$

Wir können die Indizes i, j und k dann als Funktionen von ν ansehen (man ersetze auf S. 38 oben B durch τB). Wir definieren die speziellen Matrizen

$$T_l = T_l(\gamma,\varrho) := \Big[\langle T_2(r_i)\,K_\gamma, H_{nl,0}\rangle_H\Big]_{\substack{n=l+1,\dots,\tau B\\ i=0,\dots,2\tau B-1}} \in \mathbb{C}^{(\tau B-l)\times 2\tau B}, \qquad l<\tau B,$$

wobei keine Verwechslungsgefahr zwischen diesen Matrizen und dem Translationsoperator T_2 besteht. Wir benötigen ferner die Permutationsmatrizen

$$\begin{aligned}
P_{\tau B} &:= \Big[e_{\psi(\sigma)}^{\mathrm{T}}\Big]_{\sigma=0,\dots,(2\tau B)^2-1} \in \mathbb{R}^{(2\tau B)^2\times(2\tau B)^2},\\
Q_{\tau B} &:= \Big[e_{\omega(\sigma)}^{\mathrm{T}}\Big]_{\sigma=0,\dots,2\tau B(2\tau B-1)-1} \in \mathbb{R}^{2\tau B(2\tau B-1)\times 2\tau B(2\tau B-1)},\\
R_{\tau B} &:= \Big[e_{\varkappa(\sigma)}^{\mathrm{T}}\Big]_{\sigma=0,\dots,2(\tau B)^3-1} \in \mathbb{R}^{2(\tau B)^3\times 2(\tau B)^3},
\end{aligned}$$

mit den kanonischen Einheitsvektoren der entsprechenden Länge, sowie

$$\begin{aligned}
\psi(\sigma) &:= 2\tau B(\sigma \bmod 2\tau B) + \frac{\sigma-(\sigma \bmod 2\tau B)}{2\tau B},\\
\omega(\sigma) &:= (2\tau B-1)(\sigma \bmod 2\tau B) + \frac{\sigma-(\sigma \bmod 2\tau B)}{2\tau B},\\
\varkappa(\sigma) &:= (\tau B)^2 i(\sigma) + \tau B(\tau B + m(\sigma)-1) - \frac{(\max\{m(\sigma),0\}-1)\max\{m(\sigma),0\}}{2}\\
&\quad - \frac{(\tau B+\min\{m(\sigma),0\}-1)(\tau B+\min\{m(\sigma),0\})}{2} + l(\sigma) - |m(\sigma)|,
\end{aligned}$$

wobei in der letzten Gleichung

$$\begin{aligned}
i(\sigma) &:= \sigma \bmod 2\tau B,\\
l(\sigma) &:= \left\lfloor \sqrt{\frac{\sigma-i(\sigma)}{2\tau B}}\right\rfloor,\\
m(\sigma) &:= \frac{\sigma-i(\sigma)}{2\tau B} - l(\sigma)(l(\sigma)+1),
\end{aligned}$$

gilt. Die Matrix $P_{\tau B}$ sorgt für eine Umsortierung von $j,k = 0,\dots,2\tau B-1$ hin zu $k,j = 0,\dots,2\tau B-1$. Die Matrix $Q_{\tau B}$ ermöglicht uns eine Umsortierung von $k = 0,\dots,2\tau B-1$;

$m = 1-\tau B,\dots,\tau B-1$, hin zu $m = 1-\tau B,\dots,\tau B-1$; $k = 0,\dots,2\tau B-1$. Die Matrix $R_{\tau B}$ erledigt die Umsortierung von $i = 0,\dots,2\tau B-1$; $m = 1-\tau B,\dots,\tau B-1$; $l = |m|,\dots,\tau B-1$, hin zu $l = 0,\dots,\tau B-1$; $m = -l,\dots,l$; $i = 0,\dots,2\tau B$. Es sind im Folgenden $S_{\tau B}$ die in (2.37) definierte Permutationsmatrix, $F_{2\tau B}$ die Fourier-Matrix der Länge $2\tau B$ (Def. 2.2.3), A_n und $I_n := B_n$ für $n \in \mathbb{N}$ die in (2.28) eingeführten Hilfsmatrizen, sowie $D_m := D_{m,0,\tau B}$ für $|m| < \tau B$ die in (3.36) definierten Wigner-d-Matrizen. Setzen wir nun für die Skalierungskoeffizienten $v := [\varepsilon_{i(\nu),j(\nu),k(\nu)}]_{\nu=0,\dots,(2\tau B)^3-1}$ und für die SGL-Fourier-Koeffizienten $\hat{s} := [\hat{s}_{n(\mu),l(\mu),m(\mu)}]_{\mu=0,\dots,\tau B(\tau B+1)(2\tau B+1)/6-1}$, so lautet (3.46) in Matrix-Vektor-Schreibweise

$$\hat{s} = S_{\tau B}\cdot\begin{bmatrix}\tilde{T}_0 & & \\ & \ddots & \\ & & \tilde{T}_{\tau B-1}\end{bmatrix}\cdot R_{\tau B}\cdot\left\{E_{2\tau B}\otimes\begin{bmatrix}D_{1-\tau B} & & \\ & \ddots & \\ & & D_{\tau B-1}\end{bmatrix}\right\} \tag{3.48}$$
$$\times\{E_{2\tau B}\otimes Q_{\tau B}\}\cdot\left\{E_{(2\tau B)^2}\otimes(A_{\tau B}F_{2\tau B}I_{\tau B})\right\}\cdot\{E_{2\tau B}\otimes P_{\tau B}\}\cdot v,$$

wobei die Matrizen

$$\tilde{T}_l := E_{2l+1}\otimes T_l, \qquad l<\tau B,$$

selbst wieder eine Blockstruktur besitzen. Die besondere Struktur von (3.48) erlaubt es uns, geeignete Skalierungskoeffizienten v aus den gegebenen SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ der Funktion f zu bestimmen (vgl. dazu Bem. 3.1.6). Hierzu definieren wir $\hat{f} := [\hat{f}_{n(\mu),l(\mu),m(\mu)}]_{\mu=0,\dots,\tau B(\tau B+1)(2\tau B+1)/6-1}$ als Fourier-Vektor. Wegen der Bandbegrenztheit von f gilt dabei $\hat{f}_{nlm} = 0$ für $n > B$. Ersetzen wir nun in (3.48) die linke Seite $\hat{s}$ durch $\hat{f}$, so erhalten wir ein Gleichungssystem für die Skalierungskoeffizienten v von s. Indem wir dieses Gleichungssystem nun lösen, sorgen wir dafür, dass die SGL-Fourier-Koeffizienten $\hat{s}_{nlm}$ von s mit den gegebenen SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ von f bis zur Bandbreite τB übereinstimmen. Hierbei kommt erstmals die Bedeutung des Überabtastungsfaktors τ zum Ausdruck.

Mit den Gewichten $\chi_j = \chi_{j,\tau B}$ der SO(3)-Quadraturformel in Satz 3.3.3 definieren wir $V_{\tau B} := \operatorname{diag}[\chi_j]_{j=0,\dots,2\tau B-1}$. Zusätzlich dazu führen wir für $|m| < \tau B$ die Matrizen $U_m := \operatorname{diag}[(2l+1)\tau B/2\pi]_{l=|m|,\dots,\tau B-1}$ ein. Aus (3.32) und Satz 3.3.3 folgt

$$(U_m D_m V_{\tau B})\,D_m^{\mathrm{T}} = E_{\tau B-|m|}.$$

Für $|m| < \tau B$ und beliebiges $b \in \mathbb{C}^{\tau B-|m|}$ finden wir demnach stets eine Lösung $x \in \mathbb{C}^{2\tau B}$ des Gleichungssystems $D_m x = b$, indem wir $x := V_{\tau B}D_m^{\mathrm{T}}U_m^{\mathrm{T}}$ setzen. Darüber hinaus gilt

$$(A_{\tau B}F_{2\tau B}I_{\tau B})\,(I_{\tau B}^{\mathrm{H}}F_{2\tau B}^{-1}A_{\tau B}^{\mathrm{T}}) = E_{2\tau B-1}.$$

Für beliebiges $b \in \mathbb{C}^{2\tau B-1}$ ist also eine Lösung des Gleichungssystems $A_{\tau B}F_{2\tau B}I_{\tau B}x = b$ durch $x := (2\tau B)^{-1}I_{\tau B}^{\mathrm{H}}F_{2\tau B}^{-1}A_{\tau B}^{\mathrm{T}}b \in \mathbb{C}^{2\tau B}$ gegeben. Für jede Permutationsmatrix $P \in \mathbb{R}^{n\times n}$ gilt selbstverständlich $P^{-1} = P^{\mathrm{T}}$ und daher $Px = b$ genau dann, wenn $x = P^{-1}b$ ($b \in \mathbb{C}^n$). Mit Blick auf (3.48) zeigen wir nun die Lösbarkeit der durch die Matrizen T_l beschriebenen Gleichungssysteme.

Lemma 3.3.12. *Für $l < \tau B$ besitzt das Gleichungssystem $T_l x = b$ für jede rechte Seite $b \in \mathbb{C}^{\tau B-l}$ mindestens eine Lösung $x \in \mathbb{C}^{2\tau B}$.*

Beweis. Es sei o. B. d. A. $\tau = 1$, andernfalls ersetze man im Folgenden B durch τB. Wir definieren die speziellen Polynome

$$p_{nl}(r) \;=\; p_{nl}(\gamma; r) \;\coloneqq\; r^l L_{n-l-1}^{(l+1/2)}\left(\frac{\gamma}{1+\gamma} r^2\right), \qquad r \in [0,\infty),\; l < n \leq B,$$

und konstruieren damit für festes $l < B$ die Matrix

$$O_l \;=\; O_l(\gamma, \varrho) \;\coloneqq\; \Big[p_{nl}(r_i)\Big]_{\substack{n=l+1,\ldots,B \\ i=0,\ldots,2B-1}} \;\in\; \mathbb{R}^{(B-l)\times 2B}.$$

Die Matrix T_l besitzt nach Folgerung 3.3.11 die Faktorisierung

$$T_l \;=\; \left(\frac{\pi}{1+\gamma}\right)^{3/2} M_{l,0} \cdot \operatorname{diag}\left[\left(\tfrac{\gamma}{1+\gamma}\right)^{n-1} N_{nl}\right]_{n=l+1,\ldots,B} \cdot O_l \cdot \operatorname{diag}\left[\mathrm{e}^{-\frac{\gamma}{1+\gamma} r_i^2}\right]_{i=0,\ldots,2B-1}.$$

Da stets $\operatorname{grad}(p_{nl}) = 2n - l - 2$ gilt, sind die Polynome p_{nl} für $n = l+1, \ldots, B$ linear unabhängig und vom Grad echt kleiner als $2B$. Die paarweise Verschiedenheit der $2B$ positiven Radien r_i liefert mit der obigen Faktorisierung der Matrix T_l somit $\operatorname{rang}(T_l) = \operatorname{rang}(O_l) = B - l$, was zu zeigen war. □

Die jeweils eindeutig bestimmte Lösung x des Gleichungssystems $T_l x = b$ mit minimaler Norm $\|x\|_2$ erhält man bekanntlich durch Multiplikation von b mit der *Pseudoinversen* $T_l^\dagger \in \mathbb{R}^{2\tau B \times (\tau B - l)}$ von T_l (s. [Kress, 1998, Abschn. 5.2]). Ersetzen wir also wie oben beschrieben in (3.48) den Fourier-Vektor $\hat{s}$ von s durch den Fourier-Vektor $\hat{f}$ von f, so besitzt das entstehende Gleichungssystem nach den obigen Überlegungen stets mindestens eine Lösung v. Wir erhalten eine solche Lösung und damit geeignete Skalierungskoeffizienten für s durch Auswerten des Matrix-Vektor-Produkts

(3.49)

$$\begin{aligned} v = {} & \left\{E_{2\tau B} \otimes P_{\tau B}^{\mathrm{T}}\right\} \cdot \overbrace{\left\{E_{(2\tau B)^2} \otimes \left(I_{\tau B}^{\mathrm{H}} F_{2\tau B}^{-1} A_{\tau B}^{\mathrm{T}}\right)\right\}}^{(2\tau B)^2 \text{ Blöcke der Größe } 2\tau B \times (2\tau B - 1)} \cdot \left\{E_{2\tau B} \otimes Q_{\tau B}^{\mathrm{T}}\right\} \\ & \times \left\{E_{2\tau B} \otimes \underbrace{\begin{bmatrix} V_{\tau B} D_{1-\tau B}^{\mathrm{T}} U_{1-\tau B}^{\mathrm{T}} & & \\ & \ddots & \\ & & V_{\tau B} D_{\tau B-1}^{\mathrm{T}} U_{\tau B-1}^{\mathrm{T}} \end{bmatrix}}_{2\tau B - 1 \text{ Blöcke } V_{\tau B} D_m^{\mathrm{T}} U_m^{\mathrm{T}} \text{ der Größe } 2\tau B \times (\tau B - |m|)}\right\} \cdot R_{\tau B}^{\mathrm{T}} \cdot \underbrace{\begin{bmatrix} \tilde{T}_0^\dagger & & \\ & \ddots & \\ & & \tilde{T}_{\tau B-1}^\dagger \end{bmatrix}}_{\tau B \text{ Blöcke (s. u.)}} \cdot S_{\tau B}^{\mathrm{T}} \cdot \hat{f}, \end{aligned}$$

wobei die Matrizen

$$\tilde{T}_l^\dagger \;=\; \underbrace{E_{2l+1} \otimes T_l^\dagger}_{2l+1 \text{ Blöcke der Größe } 2\tau B \times (\tau B - l)}, \qquad l < \tau B,$$

selbst wieder eine Blockstruktur besitzen. Von besonderer Bedeutung ist für uns die Tatsache, dass sich das Matrix-Vektor-Produkt (3.49) in $\mathcal{O}((\tau B)^4)$ Schritten auswerten lässt, wenn die von den Eingabedaten $\hat{f}$ unabhängigen Pseudoinversen $T_l^\dagger$ vorberechnet wurden. Dazu verwenden wir wie in Abschnitt 3.3.1 beschrieben den Clenshaw-Algorithmus für die Multiplikation mit den Matrizen D_m^{T}, während wir für die Multiplikationen mit der inversen Fourier-Matrix $F_{2\tau B}^{-1}$ die iFFT verwenden. Alternativ zum Clenshaw-Algorithmus

kann eine adjungierte FDPT verwendet werden, was hier jedoch keinen Einfluss auf die Komplexität hat.

Nachdem die Skalierungskoeffizienten v des Approximanten s auf oben beschriebene Weise berechnet wurden, kann s an M gestreuten Punkten in $\mathbb{R}^3$ mithilfe der in Abschnitt 3.3.2 beschriebenen FGT ausgewertet werden. Es ergibt sich eine Komplexität von $\mathcal{O}((\sigma n)^d \log(\sigma n) + (q\tau B)^3 + q^3 M)$ für diesen letzten Schritt, wobei n der Abschneideparameter der FGT ist, während σ und q den Überabtastungsfaktor bzw. Abschneideparameter der darin auftretenden iNFFT und ihrer Adjungierten bezeichnen. Dies vervollständigt die iNFSGLFT-B. Wir erhalten eine Rechenkomplexität von $\mathcal{O}((\tau B)^4 + (\sigma n)^d \log(\sigma n) + (q\tau B)^3 + q^3 M)$. Die Speicherkomplexität ist $\mathcal{O}((\tau B)^3 + (\sigma n)^3)$.

Wir wollen die iNFSGLFT-B abschließend als Matrix-Faktorisierung darstellen. Es seien dazu $x_0, \ldots, x_{M-1}$ die Punkte, an denen die Funktion f ausgewertet werden soll. Darüber hinaus seien y_j, $j = 0, \ldots, (2\tau B)^3 - 1$, die Zentren der Gaußschen Ansatzfunktion K_γ des Approximanten s. Wir definieren die spezielle DGT-Matrix

$$G \coloneqq \Big[K_\gamma(x_i - y_j)\Big]_{\substack{i=0,\ldots,M-1 \\ j=0,\ldots,(2\tau B)^3-1}} \in \mathbb{R}^{M \times (2\tau B)^3}.$$

Mit allen Überlegungen dieses Abschnitts erhalten wir insgesamt das folgende Resultat als Entsprechung zu Satz 3.2.2:

Satz 3.3.13. *Die Matrix $\tilde{H}$ in Definition 3.1 besitzt die näherungsweise Faktorisierung*

$$\tilde{H} \approx G \cdot X,$$

wobei X die bereits faktorisierte Matrix in (3.49) bezeichne.

Die Matrix G wiederum kann nun wie in Abschnitt 3.3.2 gezeigt näherungsweise faktorisiert werden. Bei den einzelnen Faktoren der Matrix X ergibt sich eine Faktorisierung der Matrizen D_m^{T} bei Verwendung des Clenshaw-Algorithmus aus Satz 2.2.21. Durch Umkehren der Produktreihenfolge und hermitesches Transponieren der einzelnen Faktoren erhalten wir hier wie im Fall der iNFSGLFT-A auch eine adjungierte iNFSGLFT-B mit der gleichen Rechen- und Speicherkomplexität wie die der iNFSGLFT-B. Für die Multiplikation mit den dabei auftretenden Matrizen D_m können wir den adjungierten Clenshaw-Algorithmus oder eine FDPT verwenden, während wir für die Multiplikation mit der Fourier-Matrix $F_{2\tau B}$ die FFT benutzen. Aus der iNFSGLFT-B und ihrer Adjungierten kann wieder eine iterative Vorwärtstransformation konstruiert werden, die wir als NFSGLFT-B bezeichnen.

3.3.4 Numerische Tests und Vergleich mit Variante A

Die iNFSGLFT-B wurde in der Programmiersprache C++ implementiert und auf dem in Abschnitt 2.3 beschriebenen System getestet. Die Berechnungen wurden mit doppelter Genauigkeit durchgeführt, die einzige Ausnahme hiervon bildet die Multiplikation mit den Pseudoinversen $T_l^\dagger$. Diese Multiplikationen wurden mit der *GNU Multiple Precision Arithmetic Library*[7] (GMP, Vers. 6.1.2) mit einer Genauigkeit von 1000 Nachkommastellen vorgenommen. Die Matrizen $T_l^\dagger$ selbst wurden in Mathematica mit hoher Genauigkeit

[7] Siehe `https://gmplib.org`

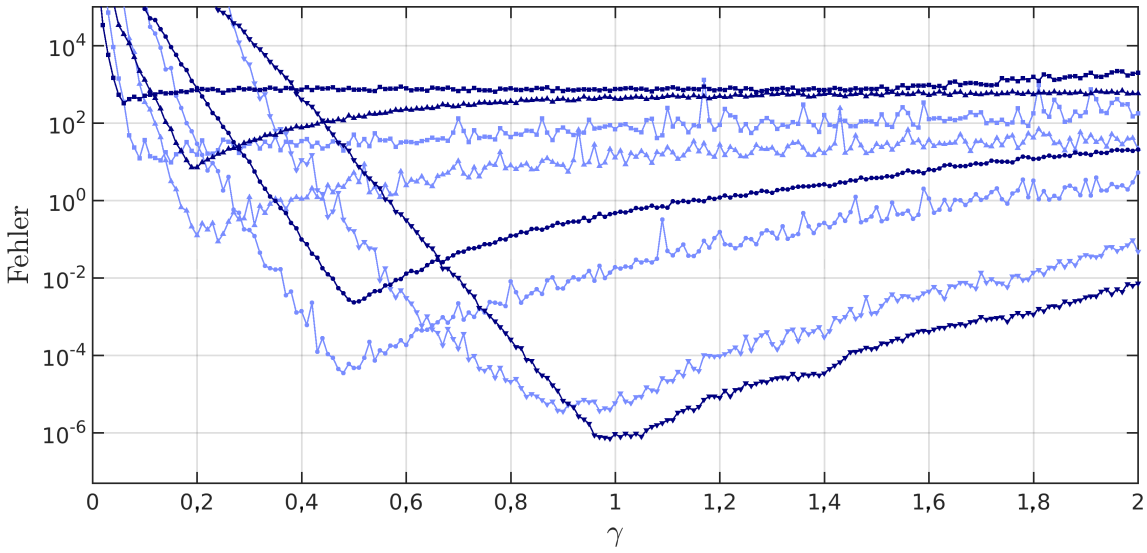

Abbildung 3.6. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-B für $B = 4$ bei $M = 10\,000$ gleichverteilten Punkten in $\mathbb{B}_5^3$ mit **(Viereck)** $\tau = 1$, **(Dreieck)** $\tau = 2$, **(Kreis)** $\tau = 4$ und **(gestürztes Dreieck)** $\tau = 8$ in Abhängigkeit von dem Spreizfaktor γ der Gaußschen Ansatzfunktion.

vorberechnet und mit 100 Nachkommastellen gespeichert. Für die Multiplikation mit den Matrizen D_m^{T} wurde der Clenshaw-Algorithmus verwendet. Es wurde die iFFT der FFTW benutzt (vgl. Abschn. 3.2.3). Bei den Gewichten der SO(3)-Quadraturformel in Satz 3.3.3 wurde auf die vorberechneten Werte aus dem Softwarepaket SpharmonicKit zurückgegriffen (vgl. Abschn. 2.3). Als FGT wurde die in Abschnitt 3.3.2 beschriebene Variante mit der von Keiner et al. [2009] implementierten iNFFT und ihrer Adjungierten zum Einsatz gebracht. Dabei wurde der Überabtastungsfaktor $\sigma = 2$ festgesetzt. Bei der Fehler- und Laufzeitmessung wurde wie in Abschnitt 3.2.3 stets die naive iNDSGLFT verwendet.

In einem ersten Test wurde der Fehler der iNFSGLFT-B in Abhängigkeit von dem Überabtastungsfaktor τ untersucht. Dazu wurde die Bandbreite $B = 4$ gewählt und $\varrho = 10$ festgesetzt. Bei der iNFFT und ihrer Adjungierten innerhalb der FGT wurden $n = 64$ und $q = 16$, bei der FGT selbst $\rho = 2\varrho = 20$ gewählt. Es wurden wie in Abschnitt 3.2.3 beschrieben zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ und $M = 10\,000$ zufällige Punkte $x_i \in \mathbb{B}_5^3$ generiert. Aus den SGL-Fourier-Koeffizienten wurden dann mit der iNFSGLFT-B die Funktionswerte $f(x_i)$ bei festem Spreizfaktor γ der Gaußschen Ansatzfunktion rekonstruiert, wobei für γ über die Werte $i/100$, $i = 1, \ldots, 200$, iteriert wurde. Dabei wurden für jedes γ neue Punkte und SGL-Fourier-Koeffizienten erzeugt. Mit zehn Testläufen wurde dann für $\tau = 1, 2, 4, 8$ jeweils der durchschnittliche absolute und relative Fehler (3.31) in Abhängigkeit von γ bestimmt. Abbildung 3.6 zeigt die Ergebnisse. Wählt man jeweils das bezüglich des absoluten Fehlers optimale γ, so liegt hier ein exponentielles Abfallen des absoluten Fehlers über τ nahe.

In den folgenden Tests wurde innerhalb der FGT in Abhängigkeit von γ und ϱ stets

$$\rho = \rho(\varrho, \gamma) := \max\{2\varrho, \sqrt{50\ln(10)/\gamma}\}$$

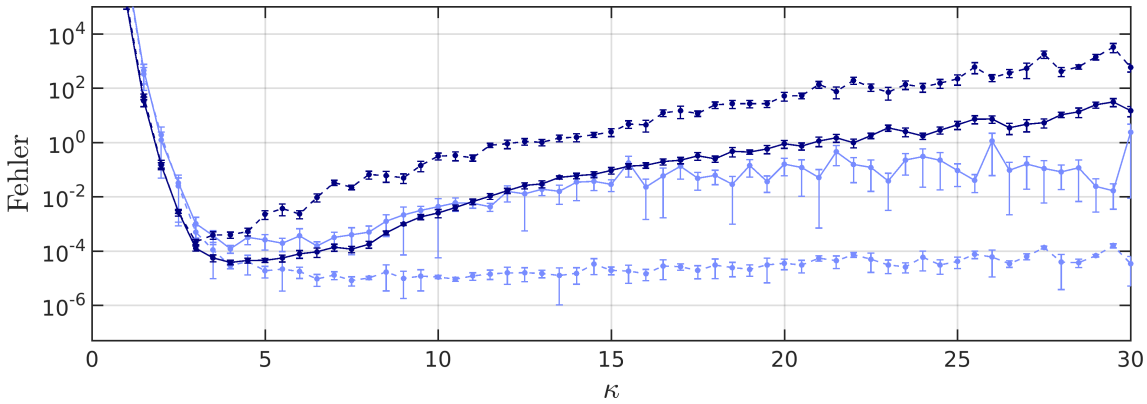

Abbildung 3.7. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-B für $B = \tau = 4$ bei $M = 1000$ gleichverteilten Punkten in $\mathbb{B}^3_\kappa$ mit $\varrho = 2\kappa$ und γ optimiert für den **(durchgezogen)** absoluten Fehler bzw. **(gestrichelt)** relativen Fehler, in Abhängigkeit von κ.

gewählt, da dann für den Gauß-Kern aus Definition 3.3.9 speziell $K_\gamma(\rho) \leq 10^{-50}$ gilt, was sich besonders bei kleinen Spreizfaktoren empirisch als vorteilhaft erwiesen hat.

Im zweiten Test wurde der Fehler der iNFSGLFT-B in Abhängigkeit von der Streuweite der Zielpunkte x_i untersucht. Dazu wurden für $B = 4$ wie in Abschnitt 3.2.3 beschrieben zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ und $M = 1000$ zufällige Punkte $x_i \in \mathbb{B}^3_\kappa$ erzeugt, wobei für κ über die Werte $i/2$, $i = 1, \ldots, 60$, iteriert wurde. Es wurde $\tau = 4$ und $\varrho = 2\kappa$ gewählt, während bei der FGT $n = 64$ und $q = 16$ gesetzt wurde. Es wurden dann mittels der iNFSGLFT-B jeweils die Funktionswerte $f(x_i)$ rekonstruiert, wobei für γ über die Werte $j/1000$, $j = 1, \ldots, 2000$, iteriert wurde. Für jedes κ und γ wurden dabei neue Punkte und SGL-Fourier-Koeffizienten erzeugt. Das obige Vorgehen wurde zehn mal wiederholt und so für jedes κ der durchschnittliche maximale absolute und relative Fehler in Abhängigkeit von γ bestimmt. Abbildung 3.7 zeigt die Ergebnisse für das jeweils optimale γ mit dem kleinsten absoluten bzw. relativen Fehler. Nach einer initialen Phase mit einem schnellen Abfallen des Fehlers beginnt dieser ab einem bestimmten Punkt wieder größer zu werden. Interessant ist hierbei, dass der relative Fehler im Test durch eine geeignete Wahl von γ ab $\kappa = 5$ sehr gut kontrolliert werden konnte. Darüber hinaus ist den Ergebnissen zu entnehmen, dass es im allgemeinen einen Unterschied macht, ob der Spreizfaktor γ der Gaußschen Ansatzfunktion für den absoluten oder den relativen Fehler optimiert wird. Für kleine Werte von κ ist die Wahl $\varrho = 2\kappa$ offensichtlich nicht geeignet. Hier bietet es sich an, ϱ größer zu wählen: Setzt man etwa $\varrho = 20$ und wählt die für $\kappa = 10$ zuvor bestimmten optimalen Spreizfaktoren $\gamma = 0{,}032$ für einen kleinen absoluten Fehler bzw. $\gamma = 0{,}051$ für einen kleinen relativen Fehler, so ergibt sich ein ganz anderes Bild (Abb. 3.8). Zum direkten Vergleich sind in Abbildung 3.9 die Ergebnisse des gleichen Versuchs mit der iNFSGLFT-A ($\sigma = 2$, $q = 1$ bzw. $q = 4$) gezeigt. Es ist wichtig zu bemerken, dass der Rechenaufwand in allen drei obigen Tests jeweils konstant war.

Im nächsten Test wurde der Fehler der iNFSGLFT-B in Abhängigkeit von der Bandbreite B untersucht. Es wurde $\tau = 4$ und $\varrho = 10$ gewählt, während bei der FGT $n = 64$

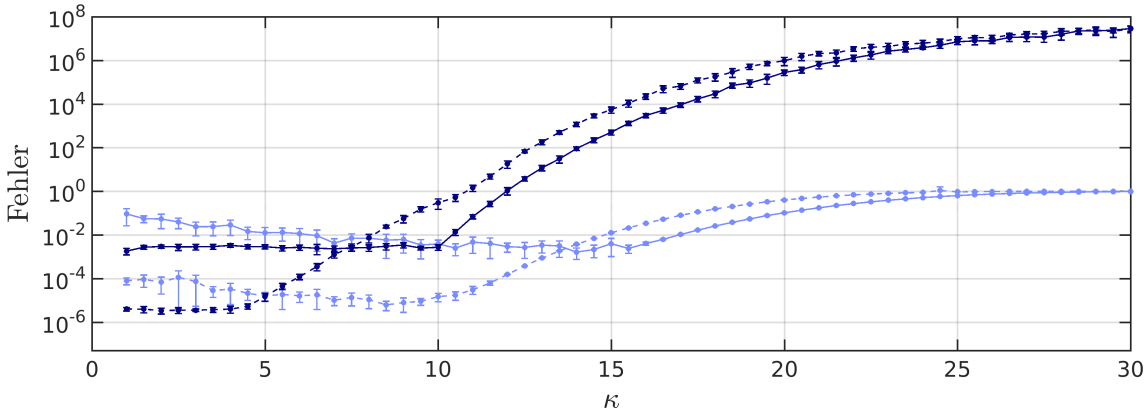

Abbildung 3.8. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-B für $B = \tau = 4$ bei $M = 1000$ gleichverteilten Punkten in $\mathbb{B}^3_\kappa$ mit $\varrho = 20$ und **(durchgezogen)** $\gamma = 0{,}032$ bzw. **(gestrichelt)** $\gamma = 0{,}051$ in Abhängigkeit von κ.

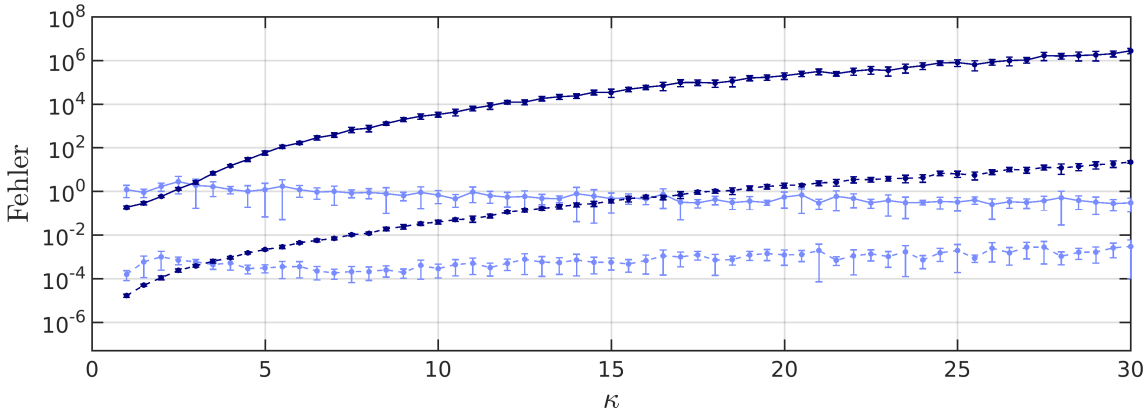

Abbildung 3.9. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-A für $B = 4$ bei $M = 1000$ gleichverteilten Punkten in $\mathbb{B}^3_\kappa$ mit $\sigma = 2$ und **(durchgezogen)** $q = 1$ bzw. **(gestrichelt)** $q = 4$ in Abhängigkeit von κ.

und $q = 16$ gesetzt wurde. Für die Bandbreiten $B = 2, 4, 8$ wurden wie in Abschnitt 3.2.3 beschrieben jeweils zufällige SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ und $M = 1000$ zufällige Punkte $x_i \in \mathbb{B}^3_5$ erzeugt. Es wurden dann die Funktionswerte $f(x_i)$ rekonstruiert, wobei für γ wieder über die Werte $i/1000$, $i = 1, \ldots, 2000$, iteriert wurde. Für jedes γ wurden dabei neue Punkte und SGL-Fourier-Koeffizienten erzeugt. Das obige Vorgehen wurde zehn mal wiederholt und so für jedes B der durchschnittliche maximale absolute und relative Fehler in Abhängigkeit von γ bestimmt. Abbildung 3.10 zeigt die Ergebnisse für das jeweils optimale γ mit dem minimalen absoluten bzw. relativen Fehler. Die Ergebnisse lassen ein subexponentielles Wachstum des absoluten Fehlers bezüglich B vermuten. Im Fall der iNFSGLFT-A wurde dieses Verhalten in Satz 3.2.4 bewiesen und in Abschnitt 3.2.3 numerisch validiert (vgl. Abb. 3.3).

Im abschließenden vierten Test wurde die Laufzeit der iNFSGLFT-B untersucht und mit der der Variante A verglichen. Dazu wurden für die Bandbreiten $B = 4$ und $B = 8$ zufällige

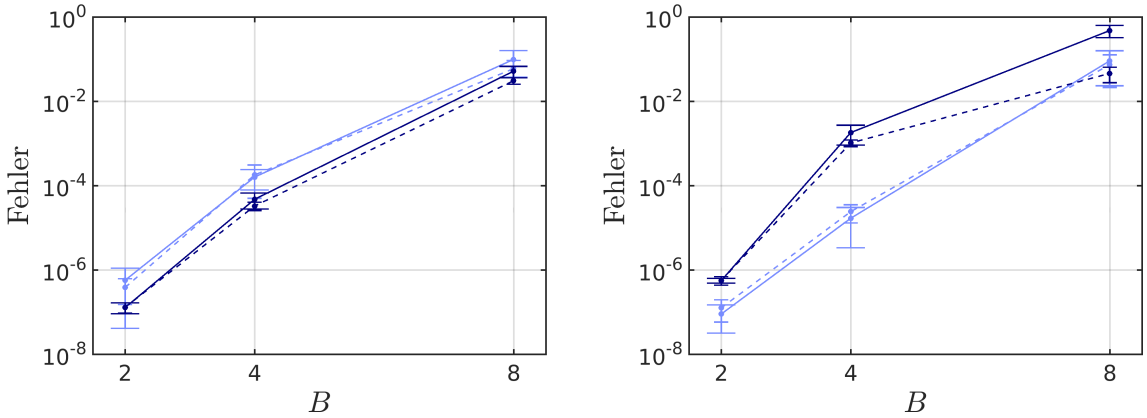

Abbildung 3.10. Durchschnittlicher maximaler **(dunkelblau)** absoluter und **(hellblau)** relativer Fehler der iNFSGLFT-B bei $M = 1000$ gleichverteilten Punkten in $\mathbb{B}_5^3$ mit γ optimiert für **(links)** den absoluten Fehler und **(rechts)** den relativen Fehler, in Abhängigkeit von der Bandbreite B. Die gestrichelten Linien zeigen das Ergebnis unter Verwendung der DGT anstelle der FGT. Hierbei wurden die Querstriche bei der Standardabweichung halb so breit gezeichnet.

SGL-Fourier-Koeffizienten $\hat{f}_{nlm}$ generiert. Für $B = 4$ wurde dies wie oben getan, während für $B = 8$ der Real- und Imaginärteil von $\hat{f}_{nlm}$ jeweils gleichverteilt in $[-\alpha_{nlm}, \alpha_{nlm}]$ mit $\alpha_{nlm} := \exp(-(2n - l - 2)/2)$ erzeugt wurden (man beachte, dass $2n - l - 2$ der Grad von H_{nlm} ist, s. Abschn. 1.5). Es wurde $\tau = 4$ und $\varrho = 10$ gewählt, während bei der FGT $n = 64$ gesetzt wurde. Für $B = 4$ wurde $\gamma = 0{,}485$ und bei der FGT $q = 6$ gewählt, für die Bandbreite $B = 8$ wurde $\gamma = 0{,}907$ und $q = 12$ gesetzt. Es handelt sich hierbei um die im vorigen dritten Test bezüglich des relativen Fehlers optimierten Werte des Spreizfaktors γ. Bei der iNFSGLFT-A wurde für $B = 4$ die Einstellung $q = 5$ vorgenommen, für $B = 8$ wurde $q = 4$ gesetzt. Diese Werte von q wurden so gewählt, dass der relative Fehler der iNFSGLFT-A und der der iNFSGLFT-B möglichst gut übereinstimmten. Im weiteren Verlauf des Tests wurden wie in Abschnitt 3.2.3 beschrieben M zufällige Punkte $x_i \in \mathbb{B}_5^3$ generiert, wobei für M über die Werte 10^k, $k = 4, \ldots, 7$, iteriert wurde. Es wurden dann mittels der iNFSGLFT-A und -B die Funktionswerte $f(x_i)$ rekonstruiert. Für jede der beiden Varianten, jedes B und jedes M wurden dabei neue Punkte und SGL-Fourier-Koeffizienten erzeugt. Das obige Vorgehen wurde zehn mal wiederholt und so die Laufzeit, sowie der durchschnittliche maximale absolute und relative Fehler der iNFSGLFT-A und -B in Abhängigkeit von B und M bestimmt. Um den vorhandenen physikalischen Arbeitsspeicher von 8 Gigabyte voll auszunutzen, wurde der Test für die Bandbreite $B = 4$ noch einmal mit $M = 40\,000\,000$ Punkten wiederholt. Tabelle 3.2 zeigt die gesammelten Ergebnisse. Es ist zunächst festzuhalten, dass sowohl die iNFSGLFT-A, als auch die iNFSGLFT-B ab einer gewissen Anzahl an Zielpunkten in diesem Test schneller sind, als der naive Algorithmus. Bei der Variante B wird dieser Punkt hier allerdings erst wesentlich später erreicht, als bei der iNFSGLFT-A. Mit steigender Anzahl M an Zielpunkten vergrößert sich der Anteil der FGT an der Laufzeit der iNFSGLFT-B immer weiter, bis sich die Laufzeit fast vollständig aus der der FGT ergibt; dies lässt sich

zwanglos mit der Komplexität der iNFSGLFT-B begründen (vgl. S. 60). Die Ergebnisse legen nahe, dass sich das Laufzeitverhältnis mit wachsendem M bei fester Bandbreite B und fester Parameterwahl jeweils einem konstanten Wert annähert. In diesem Test verbessert sich das Laufzeitverhältnis mit wachsender Anzahl M an Punkten zu Gunsten der iNFSGLFT-B, die Variante A ist bei den betrachteten Werten von M jedoch noch jeweils schneller. Von der Bandbreite $B = 4$ hin zu $B = 8$ kann die iNFSGLFT-A in diesem Test ihren Laufzeitvorteil weiter ausbauen.

3.3.5 Fehlerabschätzung

Im letzten Abschnitt dieser Arbeit untersuchen wir den maximalen absoluten Fehler der iNSGLFT-B. Vermöge der Dreiecksungleichung können wir hierbei den Fehler, der von der approximativen FGT verursacht wird, und denjenigen, der bei der Annäherung der Funktion f durch den Approximanten s verbleibt, getrennt voneinander behandeln.

Der Fehler der in Abschnitt 3.3.2 besprochenen FGT wurde im eindimensionalen Fall von Kunis [2006, Abschn. 4.1.2] untersucht. Mit einer ähnlichen Strategie wie im Beweis von Satz 3.1.9 verallgemeinern wir [Kunis, 2006, Lem. 4.6] hier auf den d-dimensionalen Fall. Der Fehler der in der FGT zum Einsatz kommenden iNFFT und ihrer Adjungierten wurde in Abschnitt 3.1 abgeschätzt und wird im Folgenden nicht mehr berücksichtigt.

Lemma 3.3.14. *Es gilt in Abschnitt 3.3.2 mit* $z_{ij} := x_i - y_j$ *und* $\beta := \max_{ij} \|z_{ij}\|_2$

$$\begin{aligned} |K_\gamma(z_{ij}) - \tilde{K}_{\gamma,n}(z_{ij})| \;\le\; (2^d - 1)\Bigg(&\left(2 + \frac{1}{\gamma\rho(\rho-\beta)}\right)^d \mathrm{e}^{-\gamma(\rho-\beta)^2} \\ &+ \left(1 + \frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right)^{d-1}\left(\frac{\sqrt{\pi}}{\sqrt{\gamma}\rho} + \frac{2}{\pi n}\frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right)^d \mathrm{e}^{-(\pi n)^2/4\gamma\rho^2}\Bigg). \end{aligned}$$

Beweis. Es sei $z_{ij} = [\xi_0, \ldots, \xi_{d-1}]$. Wir setzen abkürzend

$$M \;=\; M(\gamma, \rho, z_{ij}) \;:=\; \max_{j\in\{0,\ldots,d-1\}} \sum\nolimits_{\zeta\in\mathbb{Z}\setminus\{0\}} \mathrm{e}^{-\gamma(\xi_j-\rho\zeta)^2}.$$

Aus der Definition von K_γ und $\tilde{K}_\gamma$ ergibt sich mit $\exp(-\gamma\xi^2) \le 1$ zunächst

$$|K_\gamma(z_{ij}) - \tilde{K}_\gamma(z_{ij})| \;=\; \sum\nolimits_{z\in\mathbb{Z}^d\setminus\{0\}} \mathrm{e}^{-\gamma\|z_{ij}-\rho z\|_2^2} \;\le\; \sum_{l=1}^{d} \binom{d}{l} M^l. \tag{3.50}$$

Die Summanden auf der rechten Seite ergeben sich aus dem Fall, dass genau eine der d Komponenten von z ungleich Null ist, dass genau zwei der d Komponenten von z ungleich Null sind, usw. Der Beweis von [Kunis, 2006, Lem. 4.6] zeigt die Abschätzung

$$M \;\le\; \left(2 + \frac{1}{\gamma\rho(\rho-\beta)}\right) \mathrm{e}^{-\gamma(\rho-\beta)^2}.$$

Damit ergibt sich in (3.50)

$$|K_\gamma(z_{ij}) - \tilde{K}_\gamma(z_{ij})| \;\le\; (2^d - 1)\left(2 + \frac{1}{\gamma\rho(\rho-\beta)}\right)^d \mathrm{e}^{-\gamma(\rho-\beta)^2}. \tag{3.51}$$

M	absoluter/relativer Fehler iNFSGLFT-A	absoluter/relativer Fehler iNFSGLFT-B	Laufzeit iNDSGLFT	Laufzeit iNFSGLFT-A	Laufzeit iNFSGLFT-B	Laufzeitverhältnis iNFSGLFT-B/-A
1 E +4	$(1{,}42 \pm 0{,}25)$E −4	$(3{,}24 \pm 0{,}55)$E −3	1,06E −1 s	4,62E −2 s	1,32E +0 s	28,57
	$(1{,}27 \pm 0{,}61)$E −4 -----	$(1{,}19 \pm 0{,}78)$E −4			(4,53E −1 s)	
1 E +5	$(1{,}59 \pm 0{,}19)$E −4	$(2{,}94 \pm 1{,}15)$E −3	9,94E −1 s	4,32E −1 s	1,99E +0 s	4,61
	$(3{,}74 \pm 3{,}31)$E −4 -----	$(3{,}41 \pm 1{,}52)$E −4			(1,11E +0 s)	
1 E +6	$(1{,}68 \pm 0{,}23)$E −4	$(3{,}76 \pm 1{,}09)$E −3	9,84E +0 s	4,30E +0 s	8,59E +0 s	2,00
	$(1{,}13 \pm 0{,}83)$E −3 -----	$(1{,}08 \pm 0{,}98)$E −3			(7,72E +0 s)	
1 E +7	$(1{,}68 \pm 0{,}26)$E −4	$(3{,}96 \pm 1{,}10)$E −3	9,85E +1 s	4,32E +1 s	7,34E +1 s	1,70
	$(2{,}34 \pm 0{,}76)$E −3 -----	$(2{,}81 \pm 1{,}23)$E −3			(7,25E +1 s)	
4 E +7	$(1{,}81 \pm 0{,}19)$E −4	$(3{,}60 \pm 1{,}18)$E −3	3,99E +2 s	1,94E +2 s	3,13E +2 s	1,61
	$(3{,}63 \pm 1{,}58)$E −3 -----	$(4{,}61 \pm 1{,}91)$E −3			(3,12E +2 s)	

(a)

M	absoluter/relativer Fehler iNFSGLFT-A	absoluter/relativer Fehler iNFSGLFT-B	Laufzeit iNDSGLFT	Laufzeit iNFSGLFT-A	Laufzeit iNFSGLFT-B	Laufzeitverhältnis iNFSGLFT-B/-A
1 E +4	$(4{,}87 \pm 0{,}44)$E −3	$(6{,}90 \pm 1{,}71)$E −4	9,16E −1 s	4,20E −2 s	3,14E +1 s	747,62
	$(1{,}90 \pm 0{,}76)$E −3 -----	$(1{,}84 \pm 1{,}26)$E −3			(8,39E +0 s)	
1 E +5	$(5{,}51 \pm 0{,}54)$E −3	$(8{,}30 \pm 1{,}58)$E −4	9,00E +0 s	3,06E −1 s	3,47E +1 s	113,40
	$(4{,}20 \pm 1{,}82)$E −3 -----	$(3{,}85 \pm 2{,}09)$E −3			(1,20E +1 s)	
1 E +6	$(5{,}61 \pm 0{,}61)$E −3	$(9{,}36 \pm 1{,}86)$E −4	9,05E +1 s	2,99E +0 s	7,09E +1 s	23,71
	$(1{,}69 \pm 0{,}70)$E −2 -----	$(1{,}12 \pm 0{,}81)$E −2			(4,88E +1 s)	
1 E +7	$(6{,}20 \pm 0{,}69)$E −3	$(1{,}05 \pm 0{,}16)$E −3	9,37E +2 s	3,19E +1 s	4,65E +2 s	14,58
	$(6{,}21 \pm 3{,}64)$E −2 -----	$(3{,}42 \pm 4{,}02)$E −2			(4,40E +2 s)	

(b)

Tabelle 3.2. Laufzeitvergleich der iNFSGLFT-A und -B für **(a)** $B = 4$ und **(b)** $B = 8$. Bei der iNFSGLFT-B ist in Klammern die Laufzeit der FGT mit angegeben.

Nun setzen wir für gerades $n \in \mathbb{N}$

$$\begin{aligned} N_1 &= N_1(\gamma,\rho,n) := \sum\nolimits_{\kappa\in\mathbb{Z}\setminus I_n} |\psi_\kappa|, \\ N_2 &= N_2(\gamma,\rho,n) := \sum\nolimits_{\kappa\in I_n} |\psi_\kappa|, \end{aligned}$$

mit den Fourier-Koeffizienten ψ_κ in (3.41). Ähnlich wie oben ergibt sich mit ω_k aus (3.40)

$$|\tilde{K}_\gamma(x) - \tilde{K}_{\gamma,n}(x)| \le \sum\nolimits_{k\in\mathbb{Z}^d\setminus I_n^d} |\omega_k| \le \sum_{l=1}^{d} \binom{d}{l} N_1^l N_2^{d-l}. \tag{3.52}$$

Der Beweis von [Kunis, 2006, Lem. 4.6] zeigt hier die Abschätzung

$$N_1 \le \frac{\sqrt{\pi}}{\sqrt{\gamma}\rho}\left(1 + \frac{2\gamma\rho^2}{n\pi^2}\right) \mathrm{e}^{-(\pi n)^2/4\gamma\rho^2}.$$

Darüber hinaus gilt

$$N_2 \le \frac{\sqrt{\pi}}{\sqrt{\gamma}\rho} \sum_{\kappa=-n/2}^{n/2} \mathrm{e}^{-(\pi\kappa)^2/\gamma\rho^2} \le \frac{\sqrt{\pi}}{\sqrt{\gamma}\rho}\left(1 + 2\int_0^\infty \mathrm{e}^{-(\pi\xi)^2/\gamma\rho^2}\mathrm{d}\xi\right) = \frac{\sqrt{\pi}}{\sqrt{\gamma}\rho}\left(1 + \frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right).$$

Damit ergibt sich in (3.52)

$$|\tilde{K}_\gamma(x) - \tilde{K}_{\gamma,n}(x)| \le (2^d - 1)\left(1 + \frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right)^{d-1}\left(\frac{\sqrt{\pi}}{\sqrt{\gamma}\rho} + \frac{2}{\pi n}\frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right)^d \mathrm{e}^{-(\pi n)^2/4\gamma\rho^2}. \tag{3.53}$$

Die Abschätzung im Lemma ergibt sich nun aus (3.51) und (3.53) mit der Dreiecksungleichung. □

Folgerung 3.3.15. *Der Fehler, der in Abschnitt 3.3.2 bei der Approximation der Summen S_i durch die Summen $\tilde{S}_i$ verbleibt, kann abgeschätzt werden durch*

$$\begin{aligned} \max_{i\in\{0,\dots,l-1\}} |S_i - \tilde{S}_i| \le (2^d - 1)\Bigg(&\left(2 + \frac{1}{\gamma\rho(\rho-\beta)}\right)^d \mathrm{e}^{-\gamma(\rho-\beta)^2} \\ &+ \left(1 + \frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right)^{d-1}\left(\frac{\sqrt{\pi}}{\sqrt{\gamma}\rho} + \frac{2}{\pi n}\frac{\sqrt{\gamma}\rho}{\sqrt{\pi}}\right)^d \mathrm{e}^{-(\pi n)^2/4\gamma\rho^2}\Bigg)\|a\|_1, \end{aligned}$$

wobei $a := [\alpha_0, \dots, \alpha_{m-1}]$.

Bemerkung 3.3.16. Im Fall $d = 1$ und $\beta = \rho/2$ erhalten wir [Kunis, 2006, Satz 4.7]. Man beachte, dass in den numerischen Tests im vorigen Abschnitt stets $\beta/\rho \le 3/4$ gilt.

Wir wenden uns nun der Untersuchung des Fehlers, der innerhalb der iNFSGLFT-B durch die Annäherung der Funktion f durch den Approximanten s entsteht, zu. Wir beweisen hier das folgende wichtige Resultat.

Lemma 3.3.17. *Die SGL-Fourier-Partialsummen*

$$S_N := \sum_{n=1}^{N} \sum\nolimits_{|m|\le l<n} \langle s, H_{nlm}\rangle_H H_{nlm} \tag{3.54}$$

des in (3.45) definierten Approximanten s konvergieren für $N \to \infty$ punktweise gegen s.

Beweis. Eine geeignete Teilfolge $(S_{N_k})_{k=0}^{\infty}$ von $(S_N)_{N=1}^{\infty}$ konvergiert fast überall gegen s [Bauer, 1992, Satz 15.7]. In einem ersten Schritt zeigen wir, dass die Folge der Partialsummen S_N auf kompakten Teilmengen von $\mathbb{R}^3\setminus\{0\}$ gleichmäßig gegen eine Funktion $\tilde{s}$ konvergiert. Die Funktionen $\tilde{s}$ und s stimmen fast überall überein, da die Teilfolge (S_{N_k}) den gleichen Grenzwert besitzen muss wie (S_N). Da die SGL-Fourier-Partialsummen S_N stetig sind, muss $\tilde{s}$ auf $\mathbb{R}^3\setminus\{0\}$ auch stetig sein und somit dort überall mit s übereinstimmen. Die Konvergenz im Nullpunkt untersuchen wir anschließend gesondert.

Mit dem in (3.47) definierten Translationsoperator T gilt (vgl. Gl. 3.45)

$$R(\alpha_j,\beta_k,0)^{-1}T_2(r_i)\,K_\gamma \;=\; T(R(\alpha_j,\beta_k,0)^{-1}r_i\,e_2)\,K_\gamma,$$

wobei e_2 den kanonischen Einheitsvektor entlang der ξ_2-Achse bezeichnet (vgl. Abb. 1.1). Daher können wir den Approximanten s in (3.45) auch in der Form

$$s \;=\; \sum_{i=0}^{(2\tau B)^3-1} \varepsilon_i\,T(y_i)\,K_\gamma$$

schreiben, wobei die y_i die Zentren der Gaußschen Ansatzfunktion K_γ und die ε_i die zugehörigen Skalierungskoeffizienten seien. Es sei nun $\Omega\subset\mathbb{R}^3\setminus\{0\}$ eine kompakte Menge, sowie $x=[r,\vartheta,\varphi]\in\Omega$. Aus der Definition 1.5.1 der SGL-Basisfunktionen, Lemma 3.3.10 und dem sphärischen Additionstheorem (Lem. 1.1.9) folgt mit $y_i=[r_i,\vartheta_i,\varphi_i]$ in (3.54)

$$\begin{aligned}(3.55)\quad \zeta_n(x) &:= \sum\nolimits_{|m|\le l<n}\langle s,H_{nlm}\rangle_H H_{nlm}\\ &= \frac{\sqrt{\pi}}{2}(1+\gamma)^{-3/2}\left(\frac{\gamma}{1+\gamma}\right)^{n-1}\sum_{i=0}^{(2\tau B)^3-1}\varepsilon_i\,\mathrm{e}^{-\frac{\gamma}{1+\gamma}r_i^2}\\ &\quad\times\sum_{l=0}^{n-1}\frac{\Gamma(n-l)}{\Gamma(n+1/2)}(rr_i)^l L_{n-l-1}^{(l+1/2)}(r^2)L_{n-l-1}^{(l+1/2)}\left(\frac{\gamma}{1+\gamma}r_i^2\right)(2l+1)\,P_l(\cos\phi_i),\end{aligned}$$

wobei ϕ_i den Winkel zwischen x und y_i bezeichne. Wir setzen in Abhängigkeit von den Daten $\hat{f}$ (vgl. Gl. 3.49)

$$\mu(\hat{f}) \;=\; \mu(\hat{f};B,\tau;\varrho,\gamma) \;:=\; \max_{i\in\{0,\dots,(2\tau B)^3-1\}}|\varepsilon_i|\,\mathrm{e}^{-\frac{1}{2}\frac{\gamma}{1+\gamma}r_i^2}.$$

Aus der Abschätzung $|P_l|\le 1$ (vgl. Lem. 1.1.4) und Lemma 1.4.2 folgt damit

$$(3.56)\quad |\zeta_n(x)|\le 4\sqrt{\pi}\mu(\hat{f})\,(2n-1)\left(\frac{\tau B}{\sqrt{1+\gamma}}\right)^3\left(\frac{\gamma}{1+\gamma}\right)^{n-1}\mathrm{e}^{r^2/2}\sum_{l=0}^{n-1}\binom{n-1}{l}\frac{(r\varrho)^l}{\Gamma(l+3/2)}.$$

Mit Blick auf die rechte Seite zeigen wir unter Berücksichtigung von Definition 1.2.1 mithilfe von Lemma 1.4.1

$$\begin{aligned}(3.57)\qquad \sum_{l=0}^{n-1}\binom{n-1}{l}\frac{(r\varrho)^l}{\Gamma(l+3/2)} &= \frac{2}{\sqrt{\pi}}\,{}_1F_1(1-n;3/2;-r\varrho)\\ &= \frac{2}{\sqrt{\pi}}\binom{n-1/2}{n-1}^{-1}L_{n-1}^{(1/2)}(-r\varrho).\end{aligned}$$

Es folgt aus Lemma 1.4.5, der Stirlingschen[8] Formel, sowie der Tatsache, dass r auf Ω von Null weg beschränkt ist, die Existenz eines $N = N(\Omega, \varrho)$ und eines $C_1 = C_1(\Omega, \varrho)$ mit

$$\frac{2}{\sqrt{\pi}} \binom{n-1/2}{n-1}^{-1} L_{n-1}^{(1/2)}(-r\varrho) \;\leq\; C_1 \mathrm{e}^{2\sqrt{(n-1)r\varrho}} \qquad (n \geq N).$$

Wir setzen $\eta = \eta(\Omega) := \max_{x\in\Omega} \|x\|_2$. Die Abschätzung (3.56), sowie die Tatsache, dass r auf Ω auch nach oben beschränkt ist ($\eta < \infty$), zeigen die Existenz eines $C_2 = C_2(\Omega, B, \tau, \varrho, \gamma, \hat{f})$ mit

$$|\zeta_n(x)| \leq C_2 (2n-1) \mathrm{e}^{2\sqrt{(n-1)\eta\varrho}} \left(\frac{\gamma}{1+\gamma}\right)^{n-1} \qquad (n \geq N). \tag{3.58}$$

Es ist damit für jedes $\lambda \in (0,1)$ auf Ω die asymptotische Abschätzung

$$\zeta_n(x) \;=\; \mathcal{O}\left(\left(\frac{\gamma}{1+\gamma}\right)^{\lambda n}\right) \qquad (n \to \infty)$$

gleichmäßig erfüllt.

Die Konvergenz von (S_N) gegen s im Nullpunkt ergibt sich aus der Formel (3.55) zusammen mit Lemma 1.4.4, indem wir dort $\psi := \gamma/(1+\gamma)$ und jeweils $\xi := \gamma r_i^2/(1+\gamma)$ setzen. □

Satz 3.3.18. *Der Fehler, der in Abschnitt 3.3 bei der Annäherung der Funktion f durch den Approximanten s verbleibt, kann unter Verwendung der obigen Bezeichnungen für $x \in \mathbb{R}^3$ abgeschätzt werden durch*

$$|f(x) - s(x)| \leq 8\mu(\hat{f}) \mathrm{e}^{r^2/2} \left(\frac{\tau B}{\sqrt{1+\gamma}}\right)^3 \sum_{n=\tau B+1}^{\infty} (2n-1) \left(\frac{\gamma}{1+\gamma}\right)^{n-1} \binom{n-1/2}{n-1}^{-1} L_{n-1}^{(1/2)}(-r\varrho).$$

Beweis. Mit der punktweisen Konvergenz der SGL-Fourier-Partialsummen in (3.54) und der Dreiecksungleichung erhalten wir für beliebiges $x \in \mathbb{R}^3$

$$\begin{aligned} |f(x) - s(x)| &= \left| \sum_{n=1}^{B} \sum\nolimits_{|m|\leq l<n} \hat{f}_{nlm} H_{nlm}(x) - \sum_{n=1}^{\infty} \sum\nolimits_{|m|\leq l<n} \hat{s}_{nlm} H_{nlm}(x) \right| \\ &\leq \sum_{n=1}^{\tau B} \sum\nolimits_{|m|\leq l<n} |\hat{f}_{nlm} - \hat{s}_{nlm}| |H_{nlm}(x)| + \sum_{n=\tau B+1}^{\infty} \underbrace{\left| \sum\nolimits_{|m|\leq l<n} \hat{s}_{nlm} H_{nlm}(x) \right|}_{=|\zeta_n(x)|}. \end{aligned}$$

Die Herleitung der iNFSGLFT-B zeigt $\hat{s}_{nlm} = \hat{f}_{nlm}$ für $n \leq \tau B$. Die Aussage folgt damit aus der auch für $r = 0$ gültigen Abschätzung (3.56) mit der daran anschließenden Umformung (3.57). □

[8] James Stirling (⋆1692 in Garden bei Stirling, Schottland, †1770 in Edinburgh, ebd.)

Abkürzungsverzeichnis

CG ..... *conjugate gradient*
CGNE ..... *conjugate-gradient normal-equation error*
CGNR ..... *conjugate-gradient normal-equation residual*
CPU ..... *central processing unit*

DCT ..... *discrete cosine transform*
DFT ..... *discrete Fourier transform*
DGT ..... *discrete Gauss transform*
DLT ..... *discrete Legendre transform*
DSFT ..... *discrete spherical Fourier transform*
DSGLFT ..... *discrete SGL Fourier transform*
DSOFT ..... *discrete* SO(3) *Fourier transform*

FCT ..... *fast cosine transform*
FDPT ..... *fast discrete polynomial transform*
FFT ..... *fast Fourier transform*
FFTW ..... *Fastest Fourier Transform in the West*
FGT ..... *fast Gauss transform*
FLT ..... *fast Legendre transform*
FSFT ..... *fast spherical Fourier transform*
FSGLFT ..... *fast spherical Gauss-Laguerre Fourier transform*
FSOFT ..... *fast* SO(3) *Fourier transform*

GMP ..... *GNU Multiple Precision Arithmetic Library*
GNU ..... „*GNU's not Unix*“ (rekursives Akronym)
GTO ..... *Gaussian-type orbital*

NDFT ..... *nonequispaced discrete Fourier transform*
NFFT ..... *nonequispaced fast Fourier transform*
NFSFT ..... *nonequiangular fast spherical Fourier transform*
NDSGLFT ..... *nonlattice discrete SGL Fourier transform*
NFSGLFT ..... *nonlattice fast SGL Fourier transform*

SGL ..... *sphärisch*[*e*] *Gauß-Laguerresch*[*e*] [Basisfunktionen etc.] bzw. *spherical Gauss-Laguerre* [*basis functions* etc.]
SO ..... *speziell*[*e*] *orthogonal*[*e*] [Gruppe] bzw. *special orthogonal* [*group*]

Abbildungsverzeichnis

E.1 Übersicht über die in dieser Arbeit behandelten Klassen von schnellen Algorithmen mit ihren Beziehungen zueinander 7

1.1 Kugelkoordinaten im $\mathbb{R}^3$ 10

2.1 Abtastwinkel des sphärischen Abtasttheorems von Driscoll und Healy 30
2.2 Gewichte des sphärischen Abtasttheorems von Driscoll und Healy 31
2.3 Abtastpunkte und Gewichte der Gauß-Quadratur für das Hermite-Gewicht auf der positiven Halbachse 34

3.1 Fehler der iNFSGLFT-A in Abhängigkeit von dem Abschneideparameter der NFFT 77
3.2 Fehler der iNFSGLFT-A in Abhängigkeit von der Streuweite der Zielpunkte . . 79
3.3 Fehler der iNFSGLFT-A in Abhängigkeit von der Bandbreite 80
3.4 Fehler der NFSGLFT-A in Abhängigkeit von der Dichte der Zielpunkte 81
3.5 Fehler der NFSGLFT-A in Abhängigkeit von der Streuweite der Zielpunkte . . . 81
3.6 Fehler der iNFSGLFT-B in Abhängigkeit von dem Überabtastungsfaktor 96
3.7 Fehler der iNFSGLFT-B in Abhängigkeit von der Streuweite der Zielpunkte . . 97
3.8 Fehler der iNFSGLFT-B in Abhängigkeit von der Streuweite der Zielpunkte . . 98
3.9 Fehler der iNFSGLFT-A in Abhängigkeit von der Streuweite der Zielpunkte . . 98
3.10 Fehler der iNFSGLFT-B in Abhängigkeit von der Bandbreite 99

Tabellenverzeichnis

2.1 Fehler einer naiven iDSGLFT bzw. iFSGLFT und anschließenden naiven DSGLFT bzw. FSGLFT in Abhängigkeit von der Bandbreite 57
2.2 Laufzeit einer naiven iDSGLFT bzw. iFSGLFT und anschließenden naiven DSGLFT bzw. FSGLFT in Abhängigkeit von der Bandbreite 57

3.1 Laufzeit der naiven iNDSGLFT und der iNFSGLFT-A in Abhängigkeit von der Anzahl an Zielpunkten 78
3.2 Laufzeitvergleich der iNFSGLFT-A und -B 101

Literaturverzeichnis

Abramowitz, M., und Stegun, I. A., Herausgeber (1972). *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables.* US Department of Commerce, National Bureau of Standards, Gaithersburg, MD, USA, 10. Auflage.

Ahmed, N., Natarajan, T., und Rao, K. R. (1974). Discrete cosine transform. *IEEE Trans. Comput.*, C-23(1):90–93.

Andrews, G. E., Askey, R., und Roy, R. (1999). *Special Functions.* Cambridge University Press, Cambridge, UK.

Arfken, G. B., und Weber, H. J. (2005). *Mathematical Methods for Physicists.* Elsevier Academic Press, Burlington, MA, USA, 6. Auflage.

Baszenski, G., und Tasche, M. (1997). Fast polynomial multiplication and convolutions related to the discrete cosine transform. *Linear Algebra Appl.*, 252(1–3):1–25.

Batir, N. (2008). Inequalities for the gamma function. *Arch. Math.*, 91(6):554–563.

Bauer, H. (1992). *Maß- und Integrationstheorie.* De Gruyter, Berlin, 2. Auflage.

Berg, L. (1968). *Asymptotische Darstellungen und Entwicklungen.* Deutscher Verlag der Wissenschaften, Berlin.

Biedenharn, L. C., und Louck, J. D. (1981). *Angular Momentum in Quantum Physics: Theory and Application.* Addison-Wesley, Boston, MA, USA.

Chirikjian, G. S., und Kyatkin, A. B. (2016). *Harmonic Analysis for Engineers and Applied Scientists.* Dover Publications, Mineola, NY, USA, 2. Auflage.

Clenshaw, C. W. (1955). A note on the summation of Chebyshev series. *Math. Comp.*, 9(51):118–120.

Cooley, J. W., Lewis, P. A. W., und Welch, P. D. (1970). The fast Fourier transform algorithm: programming considerations in the calculation of sine, cosine and Laplace transforms. *J. Sound Vib.*, 12(3):315–337.

Cooley, J. W., und Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19(90):297–301.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., und Stein, C. (2001). *Introduction to Algorithms.* MIT Press, Cambridge, MA, USA, 2. Auflage.

Dai, F., und Xu, Y. (2013). *Approximation Theory and Harmonic Analysis on Spheres and Balls.* Springer, New York, NY, USA.

Driscoll, J. R., und Healy, D. M. (1994). Computing Fourier transforms and convolutions on the 2-sphere. *Adv. Appl. Math.*, 15(2):202–250.

Driscoll, J. R., Healy, D. M., und Rockmore, D. N. (1997). Fast discrete polynomial transforms with applications to data analysis for distance transitive graphs. *SIAM J.*

Comput., 26(4):1066–1099.

Dudgeon, D. E., und Mersereau, R. M. (1984). *Multidimensional Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, NJ, USA.

Dunkl, C. F., und Xu, Y. (2001). *Orthogonal Polynomials of Several Variables.* Cambridge University Press, Cambridge, UK.

Edmonds, A. R. (1996). *Angular Momentum in Quantum Mechanics.* Princeton University Press, Princeton, NJ, USA, 4. Auflage.

Erdélyi, A., Magnus, W., Oberhettinger, F., und Tricomi, F. G. (1954). *Tables of Integral Transforms, Bd. 2.* McGraw-Hill, New York, NY, USA.

Freeden, W., Gervens, T., und Schreiner, M. (1998). *Constructive Approximation on the Sphere.* Oxford University Press, Oxford, UK.

Gautschi, W. (1997). *Numerical Analysis: An Introduction.* Birkhäuser Boston, Cambridge, MA, USA.

Golub, G. H., und van Loan, C. F. (1996). *Matrix Computations.* Johns Hopkins University Press, Baltimore, MD, USA, 3. Auflage.

Healy, D. M., Rockmore, D. N., Kostelec, P. J., und Moore, S. (2003). FFTs for the 2-sphere – improvements and variations. *J. Fourier Anal. Appl.*, 9(4):341–385.

Hildebrand, F. B. (1956). *Introduction to Numerical Analysis.* McGraw-Hill, New York, NY, USA.

Keiner, J., Kunis, S., und Potts, D. (2009). Using NFFT 3 – a software library for various nonequispaced fast Fourier transforms. *ACM Trans. Math. Softw.*, 36(4): Artikel 19.

Kostelec, P. J., und Rockmore, D. N. (2008). FFTs on the rotation group. *J. Fourier Anal. Appl.*, 14(2):145–179.

Kress, R. (1998). *Numerical Analysis.* Springer, New York, NY, USA.

Kunis, S. (2006). *Nonequispaced FFT: Generalisation and Inversion.* Dissertation, Universität zu Lübeck, Institut für Mathematik.

Kunis, S., und Potts, D. (2003). Fast spherical Fourier algorithms. *J. Comput. Appl. Math.*, 161(1):75–98.

Kunis, S., Potts, D., und Steidl, G. (2006). Fast Gauss transforms with complex parameters using NFFTs. *J. Numer. Math.*, 14(4):295–303.

Leistedt, B., und McEwen, J. D. (2012). Exact wavelets on the ball. *IEEE Trans. Signal Process.*, 60(12):6257–6269.

Levy, H., und Lessman, F. (1992). *Finite Difference Equations.* Dover Publications, Mineola, NY, USA.

Lohöfer, G. (1998). Inequalities for the associated Legendre functions. *J. Approx. Theory*, 95(2):178–193.

Maizlish, O., und Prymak, A. (2015). Convex polynomial approximation in $\mathbb{R}^d$ with Freud weights. *J. Approx. Theory*, 192:60–68.

Makhoul, J. (1980). A fast cosine transform in one and two dimensions. *IEEE Trans. Acoust., Speech, Signal Process.*, 28(1):27–34.

McEwen, J. D., Büttner, M., Leistedt, B., Peiris, H. V., und Wiaux, Y. (2015). A novel sampling theorem on the rotation group. *IEEE Signal Process. Lett.*, 22(12):2425–2429.

McEwen, J. D., und Wiaux, Y. (2011). A novel sampling theorem on the sphere. *IEEE Trans. Signal Process.*, 59(12):5876–5887.

Mhaskar, H. N. (1996). *Introduction to the Theory of Weighted Polynomial Approximation.* World Scientific, Singapore.

Plonka, G., und Tasche, M. (2005). Fast and numerically stable algorithms for discrete cosine transforms. *Linear Algebra Appl.*, 394:309–345.

Potts, D. (2003). *Schnelle Fourier-Transformationen für nichtäquidistante Daten und Andwendungen.* Habilitationsschrift, Universität zu Lübeck, Institut für Mathematik.

Potts, D., Prestin, J., und Vollrath, A. (2009). A fast algorithm for nonequispaced Fourier transforms on the rotation group. *Numer. Algorithms*, 52(3):355–384.

Potts, D., Steidl, G., und Tasche, M. (1998). Fast algorithms for discrete polynomial transforms. *Math. Comp.*, 67(224):1577–1590.

Potts, D., Steidl, G., und Tasche, M. (2001). Fast Fourier transforms for nonequispaced data: a tutorial. In Benedetto, J. J., und Ferreira, P. J. S. G., Herausgeber, *Modern Sampling Theory*, Applied and Numerical Harmonic Analysis, Seiten 247–270. Birkhäuser, Boston, MA, USA.

Prestin, J., und Wülker, C. (2017). Fast Fourier transforms for spherical Gauss-Laguerre basis functions. In Pesenson, I., Le Gia, Q. T., Mayeli, A., Mhaskar, H. N., und Zhou, D.-X., Herausgeber, *Novel Methods in Harmonic Analysis, Bd. 1: Frames and Other Bases in Abstract and Function Spaces*, Applied and Numerical Harmonic Analysis, Seiten 237–263. Birkhäuser, Basel, Schweiz.

Ritchie, D. W. (2005). High-order analytic translation matrix elements for real-space six-dimensional polar Fourier correlations. *J. Appl. Cryst.*, 38(5):808–818.

Ritchie, D. W. (2011). *High Performance Algorithms for Molecular Shape Recognition.* Habilitationsschrift, Université Henri Poincaré, Nancy, Frankreich.

Ritchie, D. W., und Kemp, G. J. L. (2000). Protein docking using spherical polar Fourier correlations. *Proteins*, 39(2):178–194.

Steen, N. M., Byrne, G. D., und Gelbard, E. M. (1969). Gaussian quadratures for the integrals $\int_0^\infty \exp(-x^2) f(x) \mathrm{d}x$ and $\int_0^b \exp(-x^2) f(x) \mathrm{d}x$. *Math. Comp.*, 23(107):661–671.

Strang, G. (1999). The discrete cosine transform. *SIAM Rev.*, 41(1):135–147.

Szegő, G. (1975). *Orthogonal Polynomials.* American Mathematical Society, Providence, RI, USA, 4. Auflage.

van Loan, C. F. (1992). *Computational Frameworks for the Fast Fourier Transform.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Watson, G. N. (1995). *A Treatise on the Theory of Bessel Functions.* Cambridge University Press, Cambridge, UK, 2. Auflage.

Werner, D. (2011). *Funktionalanalysis.* Springer, Berlin / Heidelberg, 7. Auflage.

Lebenslauf

Name	Christian Wülker
Geburtstag und -ort	11. März 1989, Hannover
Staatsangehörigkeit	deutsch
Familienstand	ledig

Schulischer Werdegang

1995 – 1999	Grundschule Bonner Straße, Hannover
1999 – 2001	Orientierungsstufe Birkenstraße, Hannover
2001 – 2008	Sophienschule, Gymnasium in Hannover
2008	Abitur

Zivildienst

2008 – 2009	Ambulanter Pflegedienst der Gustav-Brandt'sche-Stiftung, Hannover

Studium

2009 – 2012	Bachelorstudium *Medizinische Ingenieurwissenschaft*, Universität zu Lübeck
2012	Bachelor of Science, Titel der Abschlussarbeit: *Collocation methods for partial differential equations based on radial basis functions*, Verleihung des Philips-Best-Bachelor-Preises
2012 – 2014	Masterstudium *Biomedical Engineering*, Ruprecht-Karls-Universität Heidelberg
2014	Master of Science, Titel der Abschlussarbeit (in Kooperation mit Philips Research Hamburg): *Time-of-flight PET image reconstruction using origin ensembles*

Weiterer Werdegang

seit September 2014	Stipendiat am Institut für Mathematik, Universität zu Lübeck