



TECHNISCHE UNIVERSITÄT CHEMNITZ

Fakultät für Mathematik
Professur für Angewandte Funktionalanalysis

Masterarbeit

**Schnelle Algorithmen für die Fouriertransformation auf der
Rotationsgruppe**

Erik Wünsche

Chemnitz, den 6. September 2021

Betreuer: Prof. Dr. Ralf Hielscher
Prüfer: Prof. Dr. Daniel Potts

Inhaltsverzeichnis

Abkürzungsverzeichnis	III
1 Einleitung	1
2 Die Rotationsgruppe	3
2.1 Rotationsmatrizen	3
2.2 Achse-Winkel-Parametrisierung	5
2.3 Die Euler-Winkel	6
2.4 Integration auf $\mathcal{SO}(3)$ und der Raum $L_2(\mathcal{SO}(3))$	8
3 Die kontinuierliche Fouriertransformation auf der Rotationsgruppe	11
3.1 Orthogonalpolynome	11
3.1.1 Die Chebyshev-Polynome erster Art	12
3.1.2 Die Jacobi-Polynome	12
3.2 Die Wigner-d Funktionen	13
3.3 Symmetrieeigenschaften der Wigner-d Matrizen	17
3.4 Effiziente Berechnung von Wigner-d Matrizen	18
3.5 Die Spherical Harmonics	22
3.6 Der Satz von Peter-Weyl	23
3.7 Die Wigner-D Funktionen	25
3.8 Harmonische Analysis	27
4 Die diskrete Fouriertransformation auf der Rotationsgruppe	32
4.1 Auswertung von N -bandbreitenbegrenzten Funktionen	33
4.2 Die Wignertransformation über Chebychev-Polynome	34
4.2.1 Die diskrete Polynomtransformation	34
4.2.2 Die schnelle Polynomtransformation	37
4.2.3 Transformation auf eine Fourierreihe	39
4.2.4 Die nichtäquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation	40
4.3 Die darstellungsbasierte Wignertransformation	43
4.3.1 Die darstellungsbasierte nichtäquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation	43
4.3.2 Symmetrieeigenschaften	46
4.4 Numerische Integration auf $\mathcal{SO}(3)$	49
4.4.1 Die Gauß-Quadratur	51
4.4.2 Die Clenshaw-Curtis-Quadratur	55
4.4.3 Quadratur auf der Rotationsgruppe	56
4.5 Berechnung von $\mathcal{SO}(3)$ -Fourierkoeffizienten	57

4.6	Die adjungierte Wignertransformation	60
4.6.1	Die adjungierte Wignertransformation über Chebyshev-Polynome . .	60
4.6.2	Die adjungierte darstellungsbasierte Wignertransformation	63
4.7	Numerische Ergebnisse	67
5	Fazit	75
	Literaturverzeichnis	76

Abkürzungsverzeichnis

DBWT	darstellungsbasierte Wignertransformation
DCT	diskrete Kosinustransformation
DSOFT	diskrete $\mathcal{SO}(3)$ -Fouriertransformation
DWT	diskrete Wignertransformation
FFT	schnelle Fouriertransformation
FPT	schnelle Polynomtransformation
FWT	schnelle Wignertransformation
NDFT	nichtäquidistante diskrete Fouriertransformation
NDSOFT	nichtäquidistante diskrete $\mathcal{SO}(3)$ -Fouriertransformation
NFFT	nichtäquidistante schnelle Fouriertransformation
NFSOFT	nichtäquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation
ODF	Orientierungsdichtefunktion
MDF	Missorientierungsdichtefunktion

1 Einleitung

Ein polykristallines Material besteht aus Körnern verschiedener Formen, Größen und Orientierungen. In der Kristallographie beschreibt eine Textur die Gesamtheit der Orientierungen der Körner eines polykristallinen Feststoffes. Manchmal können Gebiete mit verschiedener Orientierung nicht als einzelne Körner durch Grenzen voneinander getrennt werden. Stattdessen wechselt die Orientierung stetig. Materialwissenschaftler beschreiben die Textur eines solchen polykristallinen Materials mit der Orientierungsdichtefunktion (ODF). Diese gibt für jede Orientierung den Volumenanteil der Körner dieser Orientierung an einem infinitesimalen Probenvolumen an. Damit ist die ODF eine reellwertige Funktion auf der Rotationsgruppe $\mathcal{SO}(3)$. In der Praxis wird die ODF aus einer finiten Anzahl von Einzelmessungen oder aus Polfiguren geschätzt. Sie wird in den Materialwissenschaften genutzt um Aussagen über die Verformung von Materialien zu treffen. In den Geowissenschaften werden ODFs zur Untersuchung des Verformungszustandes und der Verformungsgeschichte von Gesteinen verwendet. Eine weitere Funktion auf $\mathcal{SO}(3)$ ist die Missorientierungsdichtefunktion (MDF). Sie wird in den Materialwissenschaften als Maß für anisotrope Korngrenzeigenschaften genutzt. Zur Darstellung und Auswertung von ODFs und MDFs können Fourierreihen auf $\mathcal{SO}(3)$ verwendet werden, weshalb die Berechnung von diskreten Fouriertransformationen auf $\mathcal{SO}(3)$ ein wichtiger Bestandteil der Texturanalyse ist, siehe [2, Kapitel 3 und 4].

Kostelec und Rockmore verwenden in [17] den Produktansatz der Wigner-D Funktionen und zerlegen die diskrete $\mathcal{SO}(3)$ -Fouriertransformation in eine bivariate schnelle Fouriertransformation (FFT) und eine direkte rekursive Auswertung von Wigner-d Funktionen. Dabei geben sie einen Algorithmus in $\mathcal{O}(N^4)$ an, der auf Kosten der Stabilität zu $\mathcal{O}(N^3 \log^2 N)$ verbessert werden kann. In [31] entwickelt Risbo die Reihe von Wigner-d Funktionen mit einer Komplexität von $\mathcal{O}(N^4)$ direkt in eine Fourierreihe, welche danach mit einer trivariaten FFT ausgewertet wird. In beiden Arbeiten wird die diskrete $\mathcal{SO}(3)$ -Fouriertransformation auf äquidistanten Rotationsgittern durchgeführt. Im Gegensatz dazu treten in praktischen Anwendungen wie beispielsweise in der Texturanalyse [2] bei der Auswertung von ODFs in Rotationen zufällig verteilte Punkte auf. Deshalb werden schnelle Algorithmen für die Berechnung der nichtäquidistanten diskreten $\mathcal{SO}(3)$ -Fouriertransformation gesucht.

Dazu verwenden Potts, Prestin und Vollrath in [24] die Idee von Kostelec und Rockmore aus [17] und transformieren die Reihe von Wigner-d Funktionen über Chebyshev-Polynome in eine Fourierreihe. Dabei benutzen sie die schnelle Polynomtransformation aus [25, 3] und erhalten einen Algorithmus mit Komplexität $\mathcal{O}(N^3 \log^2 N)$. Die Instabilität des Verfahrens umgehen die Autoren durch Einfügen des Stabilisierungsschrittes [26] in den Algorithmus. Da es vom Problem abhängig ist, wie häufig der Stabilisierungsschritt verwendet wird, ist es schwierig, die exakte Komplexität des Algorithmus anzugeben. Die erzeugte trivariate Fourierreihe wird zum Schluss mit einer trivariaten nichtäquidistanten schnellen

Fouriertransformation (NFFT), siehe [27] und [13], auf einem nichtäquidistanten Gitter ausgewertet.

Die Wigner-D Funktionen $D_n^{k,l}$ bilden eine Orthogonalbasis auf $L_2(\mathcal{SO}(3))$ und werden verwendet, um N -bandbreitenbegrenzte Funktionen f in $\mathcal{SO}(3)$ -Fourierreihen

$$f(\mathbf{R}_m) = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R}_m), \quad m = 1, \dots, M$$

mit $\mathcal{SO}(3)$ -Fourierkoeffizienten $\hat{f}_n^{k,l}$ zu entwickeln. Das Ziel dieser Arbeit ist es, einen effizienten und exakten Algorithmus für die Auswertung von N -bandbreitenbegrenzten Funktionen in Rotationen $\mathbf{R} \in \mathcal{SO}(3)$ anzugeben und die Vorteile gegenüber dem Verfahren von Potts aus [24] herauszustellen.

Analog zu Risbo [31] verwenden wir, dass die Wigner-D Funktionenelemente der Darstellungen von $\mathcal{SO}(3)$ und damit insbesondere Gruppenhomomorphismen sind. Damit kann die $\mathcal{SO}(3)$ -Fourierreihe in $\mathcal{O}(N^4)$ Schritten in eine trivariate Fourierreihe entwickelt werden. Im Gegensatz zu Risbo werten wir diese Fourierreihe mit einer trivariaten NFFT aus. Unter Verwendung einiger Symmetrieeigenschaften wird der Algorithmus, insbesondere für reellwertige Funktionen, weiter verbessert.

Die Arbeit ist folgendermaßen strukturiert. Zuerst wird in Kapitel 2 die Euler-Winkel-Parametrisierung der Rotationsgruppe $\mathcal{SO}(3)$ eingeführt. Weiterhin wird der Transformationsatz verwendet, um das Integral einer Funktion auf $\mathcal{SO}(3)$ durch drei Integrale über die Euler-Winkel darzustellen. Damit definieren wir den Raum der quadratintegrierbaren Funktionen $L_2(\mathcal{SO}(3))$.

Im dritten Kapitel definieren wir die Wigner-d Funktionen und geben Symmetrieeigenschaften sowie eine 3-Term-Rekursion zur effizienten Berechnung der Wigner-d Funktionen an. Danach definieren wir mithilfe der reellwertigen Wigner-d Funktionen die komplexwertigen Wigner-D Funktionen und zeigen mit der Darstellungstheorie, dass die Wigner-D Funktionen gerade die Matrixelemente der Darstellungen von $\mathcal{SO}(3)$ sind. Wegen des Satzes von Peter-Weyl bilden diese eine Orthogonalbasis auf $L_2(\mathcal{SO}(3))$. In Kapitel 3.8 verwenden wir die Orthogonalbasis der Wigner-D Funktionen um $\mathcal{SO}(3)$ -Fourierreihen aufzustellen und deren Konvergenz zu untersuchen.

Zur Auswertung von N -bandbreitenbegrenzten Funktionen in endlich vielen Rotationen wird im Kapitel 4.2 der Algorithmus von Potts aus [24] hergeleitet. Als Hauptresultat der Arbeit geben wir in Kapitel 4.3 den alternativen darstellungsbasierten Algorithmus an. Danach verwenden wir die $\mathcal{SO}(3)$ -Quadraturformel aus Kapitel 4.5, um die adjungierten Algorithmen in Kapitel 4.6 anzugeben. Abschließend werden die Laufzeiten der Algorithmen in Kapitel 4.7 verglichen.

2 Die Rotationsgruppe

Zur mathematischen Beschreibung von Rotationen werden Rotationsmatrizen verwendet. Um Rotationsmatrizen und insbesondere Rotationen leichter darzustellen, gibt es verschiedene Parametrisierungen. Das Ziel dieses Kapitels ist es die Euler-Winkel-Parametrisierung einzuführen und anhand dieser einen Integrationsbegriff auf $\mathcal{SO}(3)$ zu definieren.

2.1 Rotationsmatrizen

Rotationen beschreiben Drehungen des Koordinatensystems. Dabei lassen sie Längen, Winkel und die Orientierung invariant. Deshalb werden Rotationen im Euklidischen dreidimensionalen Raum durch lineare Operatoren $\mathbf{O}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ dargestellt, die ebenfalls längen- und winkeltreu sind. Damit erfüllt die Matrixdarstellung der linearen Operatoren im Euklidischen Skalarprodukt die Gleichung

$$\langle \mathbf{O}x, \mathbf{O}y \rangle_2 = x^T \mathbf{O}^T \mathbf{O} y = \langle x, y \rangle_2$$

für beliebige Vektoren $x, y \in \mathbb{R}^3$. Die Gleichung ist genau dann erfüllt, wenn $\mathbf{O}^T \mathbf{O}$ gleich der Identitätsmatrix \mathbf{I}_3 ist.

Definition 2.1. Eine reellwertige 3×3 Matrix \mathbf{O} heißt orthogonal, falls

$$\mathbf{O}^T \mathbf{O} = \mathbf{I}_3$$

gilt.

Lemma 2.2. Die orthogonalen Matrizen bilden unter Matrizenmultiplikation eine algebraische Gruppe. Diese wird orthogonale Gruppe $\mathcal{O}(3)$ genannt.

Beweis. Das Matrixprodukt zweier orthogonaler Matrizen \mathbf{O}_1 und \mathbf{O}_2 ist wegen

$$(\mathbf{O}_1 \mathbf{O}_2)^T (\mathbf{O}_1 \mathbf{O}_2) = \mathbf{O}_2^T \mathbf{O}_1^T \mathbf{O}_1 \mathbf{O}_2 = \mathbf{I}_3$$

wieder eine orthogonale Matrix. Die Matrizenmultiplikation ist assoziativ. Das neutrale Element ist die Identitätsmatrix \mathbf{I}_3 . Jede orthogonale Matrix ist nach Definition 2.1 regulär und die Inverse ist gerade die Transponierte der Matrix. \square

Zur Charakterisierung der orthogonalen Gruppe $\mathcal{O}(3)$ verwenden wir folgende Definition aus der Topologie, siehe [22, Kapitel 4.1].

Definition 2.3. Sei (X, T) ein topologischer Raum und die Gruppe G Teilmenge von X . Dann heißt G topologische Gruppe, wenn die Gruppenverknüpfung $G \times G \rightarrow G$ und die Inversionsabbildung $G \ni g \mapsto g^{-1} \in G$ stetig bezüglich der Topologie T sind.

Eine topologische Gruppe G heißt kompakt, wenn G als Teilmenge von X kompakt bezüglich der Topologie T ist.

Beispiel 2.4. Der Euklidische $\mathbb{R}^{3 \times 3}$ ist ein topologischer Raum mit einer durch eine Norm induzierten Topologie. Die Topologie ist unabhängig von der verwendeten Norm, da auf endlichdimensionalen Vektorräumen alle Normen äquivalent sind.

Matrixmultiplikation und Transposition sind stetige Abbildungen im Euklidischen $\mathbb{R}^{3 \times 3}$. Infolgedessen ist die orthogonale Gruppe $\mathcal{O}(3)$ eine topologische Gruppe.

Für die Determinante einer orthogonalen Matrix $\mathbf{O} \in \mathcal{O}(3)$ gilt

$$\det(\mathbf{O})^2 = \det(\mathbf{O}^T) \det(\mathbf{O}) = \det(\mathbf{O}^T \mathbf{O}) = \det(\mathbf{I}_3) = 1$$

und folglich $\det(\mathbf{O}) = \pm 1$.

Eine lineare Abbildung ist orientierungserhaltend, falls die Determinante der zugehörigen Matrix positiv ist. Wenn die Determinante negativ ist, wechselt die lineare Abbildung den Drehsinn des Koordinatensystems zwischen Rechts- und Linkssystem und wird Drehspiegelung genannt. Das motiviert die folgende Definition.

Definition 2.5. Die spezielle orthogonale Gruppe im \mathbb{R}^3 wird mit

$$\mathcal{SO}(3) := \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^T \mathbf{R} = \mathbf{I}_3; \det(\mathbf{R}) = 1\}$$

bezeichnet.

Bemerkung 2.6. Die spezielle orthogonale Gruppe $\mathcal{SO}(3)$ ist eine Untergruppe der orthogonalen Gruppe $\mathcal{O}(3)$. Sie wird Rotationsgruppe genannt und ist isomorph zur Menge der Rotationen, die unter Komposition ebenfalls eine Gruppe bilden.

Analog zu Beispiel 2.4 ist $\mathcal{SO}(3)$ eine topologische Gruppe im Euklidischen $\mathbb{R}^{3 \times 3}$.

Zur Veranschaulichung des Zusammenhanges von orthogonaler Gruppe $\mathcal{O}(3)$ und Rotationsgruppe $\mathcal{SO}(3)$ dient das folgende Lemma. Der Beweis wird später mit der Achse-Winkel-Parametrisierung von $\mathcal{SO}(3)$ durchgeführt.

Lemma 2.7. Die orthogonale Gruppe $\mathcal{O}(3)$ zerfällt in die zwei Zusammenhangskomponenten $\mathcal{SO}(3)$ und $\mathcal{O}(3) \setminus \mathcal{SO}(3)$.

Später verwenden wir den Satz von Peter-Weyl aus der Darstellungstheorie kompakter Gruppen. Dieser Satz verallgemeinert die Theorie der Fourieranalysis für Funktionen auf kompakten Gruppen. Deshalb benötigen wir das folgende Lemma.

Lemma 2.8. Die Rotationsgruppe $\mathcal{SO}(3)$ ist eine kompakte Gruppe.

Beweis. Es reicht zu zeigen, dass $\mathcal{SO}(3)$ abgeschlossen und beschränkt im $\mathbb{R}^{3 \times 3}$ ist.

Die Abbildung $f : \mathbb{R}^{3 \times 3} \ni \mathbf{O} \mapsto \mathbf{O}^T \mathbf{O} \in \mathbb{R}^{3 \times 3}$ ist die Komposition der stetigen Abbildungen Matrixmultiplikation und Transposition. Folglich ist f stetig und damit ist das Urbild $f^{-1}(\{\mathbf{I}_3\}) = \mathcal{O}(3)$ der abgeschlossenen Menge $\{\mathbf{I}_3\}$ abgeschlossen. Die Determinante \det ist eine stetige Abbildung. Deshalb ist das Urbild $\det^{-1}(\{1\})$ abgeschlossen im $\mathbb{R}^{3 \times 3}$. Es folgt, dass die Rotationsgruppe $\mathcal{SO}(3) = \mathcal{O}(3) \cap \det^{-1}(\{1\})$ abgeschlossen ist.

Die Frobeniusnorm einer Rotationsmatrix $\mathbf{R} \in \mathcal{SO}(3)$ erfüllt

$$\|\mathbf{R}\|_F = \text{Spur}(\mathbf{R}^T \mathbf{R}) = \text{Spur}(\mathbf{I}_3) = 3$$

und ist damit beschränkt. □

Bemerkung 2.9. Die Rotationsgruppe ist eine glatte Riemannsche Mannigfaltigkeit mit Dimension 3, siehe [20, Kapitel 3.1]. Das heißt, dass die Rotationsmatrizen als Elemente des $\mathbb{R}^{3 \times 3} \cong \mathbb{R}^9$ nur drei Freiheitsgrade haben.

Falls eine topologische Gruppe eine glatte Mannigfaltigkeit ist und die Abbildungen aus Definition 2.3 beliebig oft differenzierbar sind, dann spricht man von einer Lie-Gruppe, siehe [32, Kapitel 3.1]. Die Rotationsgruppe ist eine Lie-Gruppe, siehe [20, Kapitel 3.3].

2.2 Achse-Winkel-Parametrisierung

Morawiec beschreibt in [20, Kapitel 2] mehrere Parametrisierungen zur vereinfachten Darstellung von $\mathcal{SO}(3)$. Wir führen die Achse-Winkel-Parametrisierung ein, siehe [20, Kapitel 2.3]. Dazu verwenden wir den folgenden Satz aus [5, Kapitel 5.5].

Satz 2.10 (Satz vom Fußball). Jede Rotationsmatrix $\mathbf{R} \in \mathcal{SO}(3)$ hat den Eigenwert 1.

Beweis. Für die Eigenwerte λ_k mit $k = 1, 2, 3$ einer Rotationsmatrix $\mathbf{R} \in \mathcal{SO}(3)$ gilt wegen der Längeninvarianz von Rotationen

$$\|x\|_2 = \|\mathbf{R}x\|_2 = \|\lambda_k x\|_2 = |\lambda_k| \|x\|_2$$

und damit $|\lambda_k| = 1$.

Da die Determinante von \mathbf{R} gleich 1 ist, gilt $1 = \lambda_1 \cdot \lambda_2 \cdot \lambda_3$.

Falls alle Eigenwerte reell sind, gilt die Behauptung. Sei also o.B.d.A. λ_1 nicht reell. Da komplexe Eigenwerte von reellwertigen Matrizen immer als konjugiert komplexe Paare auftreten, gilt $1 = \lambda_1 \overline{\lambda_1} \lambda_3 = |\lambda_1| \lambda_3 = \lambda_3$. □

Angenommen es werden Punkte im \mathbb{R}^3 durch eine Rotationsmatrix $\mathbf{R} \in \mathcal{SO}(3)$ gedreht. Dann bleiben alle Punkte im Eigenraum des Eigenwertes 1 unverändert. Falls \mathbf{R} nicht die identische Rotation \mathbf{I}_3 ist, dann hat der Eigenraum Dimension 1 und ist eine Gerade im \mathbb{R}^3 . Wir wollen Rotationen als Drehungen um diese Gerade darstellen.

Bemerkung 2.11. Sei eine beliebige Rotation $\mathbf{R} \in \mathcal{SO}(3)$ gegeben. Weiterhin bezeichnen wir die 2-Sphäre mit \mathbb{S}^2 . Dann existiert eine Rotationsachse $\eta \in \mathbb{S}^2$ und ein Rotationswinkel $\omega \in [0, \pi]$, so dass wir die Rotation als $\mathbf{R} = \mathbf{R}_\eta(\omega)$ darstellen können.

Jede Rotationsmatrix \mathbf{R} hat den Eigenwert 1, siehe Satz 2.10. Der dazugehörige Eigenraum beschreibt die Rotationsachse η . Der Rotationswinkel ω wird durch

$$\omega = \arccos\left(\frac{\text{Spur}(\mathbf{R}) - 1}{2}\right)$$

berechnet.

Die angegebene Achse-Winkel-Parametrisierung der Rotationen ist für die Rotationswinkel 0 und π nicht bijektiv. So wird die identische Rotation durch $\mathbf{R}_\eta(0)$ für beliebige $\eta \in \mathbb{S}^2$ dargestellt. Weiterhin ist die Rotation $\mathbf{R}_\eta(\pi)$ um eine Rotationsachse η mit dem Winkel π gleich zur Rotation $\mathbf{R}_{-\eta}(\pi)$ um die entgegengesetzte Achse $-\eta$.

Nun können wir Lemma 2.7 beweisen.

Beweis von Lemma 2.7. Die Restriktion der Determinantenabbildung $\det|_{\mathcal{O}(3)}$ ist stetig und die Menge $\{1\}$ ist offen im Bild $\det|_{\mathcal{O}(3)}(\mathcal{O}(3)) = \{-1, 1\}$. Folglich ist das Urbild

$$\det|_{\mathcal{O}(3)}^{-1}(\{1\}) = \mathcal{SO}(3)$$

der offenen Menge $\{1\}$ offen. Analog ist $\mathcal{O}(3) \setminus \mathcal{SO}(3)$ offen in $\mathcal{O}(3)$.

Da $\mathbf{I}_3 \in \mathcal{SO}(3)$ und $-\mathbf{I}_3 \in \mathcal{O}(3) \setminus \mathcal{SO}(3)$ gilt, sind beide Mengen nicht leer und folglich ist die orthogonale Gruppe $\mathcal{O}(3)$ nicht zusammenhängend.

Die Rotationsgruppe $\mathcal{SO}(3)$ ist wegzusammenhängend, da es für jede Rotation $\mathbf{R}_\eta(\omega)$ einen Weg

$$\gamma(t) = \mathbf{R}_\eta(\omega \cdot t) \quad , \quad t \in [0, 1]$$

zur Identität gibt.

Multiplikation mit $-\mathbf{I}_3$ liefert im Raum der Drehspiegelungen $\mathcal{O}(3) \setminus \mathcal{SO}(3)$ Wege zur Spiegelung $-\mathbf{I}_3$. \square

Wir geben zwei wichtige Rotationen durch ihre Rotationsmatrizen an.

Beispiel 2.12. Seien $Y = (0, 1, 0)^T$ und $Z = (0, 0, 1)^T$ zwei Einheitsvektoren. Dann sind die Rotationsmatrizen für Rotationen um die Y - und Z -Achse gegeben durch

$$\mathbf{R}_Y(\omega) = \begin{pmatrix} \cos \omega & 0 & \sin \omega \\ 0 & 1 & 0 \\ -\sin \omega & 0 & \cos \omega \end{pmatrix} \quad \text{und} \quad \mathbf{R}_Z(\omega) = \begin{pmatrix} \cos \omega & -\sin \omega & 0 \\ \sin \omega & \cos \omega & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

2.3 Die Euler-Winkel

Die Rotationsgruppe wird am häufigsten mit Euler-Winkeln parametrisiert. Dabei wird eine Rotation als Komposition von drei Achsendrehungen dargestellt, siehe [20, Kapitel 2.4].

Definition 2.13. Sei der Torus $\mathbb{T} = \mathbb{R}/(2\pi\mathbb{Z})$ gegeben. Die Winkel $\alpha, \gamma \in \mathbb{T}$ und $\beta \in [0, \pi]$ werden Euler-Winkel der Rotation

$$\mathbf{R}(\alpha, \beta, \gamma) := \mathbf{R}_{ZYZ}(\alpha, \beta, \gamma) = \mathbf{R}_Z(\alpha) \mathbf{R}_Y(\beta) \mathbf{R}_Z(\gamma)$$

genannt.

Anstelle des Torus \mathbb{T} können wir auch das repräsentative Intervall $[0, 2\pi]$ mit identifizierten Randpunkten betrachten.

Bemerkung 2.14. Bei der Darstellung einer Rotation durch Euler-Winkel gibt es verschiedene Konventionen bei der Wahl der Achsen. Die hier verwendete ZYZ-Konvention ist in der Physik weit verbreitet. In der Texturanalyse wird häufig die ZXZ-Konvention verwendet. Dabei wird nicht um die Y-Achse, sondern um die X-Achse gedreht. Offensichtlich gilt

$$\mathbf{R}_X(\beta) = \mathbf{R}_{ZYZ}\left(-\frac{\pi}{2}, \beta, \frac{\pi}{2}\right) \quad (2.1)$$

für eine Drehung um die X-Achse. Dabei drehen wir die X-Achse in die Y-Achse, drehen mit dem Winkel β um die Y-Achse und drehen die X-Achse schließlich zurück. Damit lassen sich die beiden Achsenkonventionen durch

$$\mathbf{R}_{ZYZ}(\alpha, \beta, \gamma) = \mathbf{R}_{ZXZ}\left(\alpha + \frac{\pi}{2}, \beta, \gamma - \frac{\pi}{2}\right)$$

ineinander überführen, siehe [20, Kapitel 2.4.1].

Wir verwenden die ZYZ-Darstellung aus Definition 2.13. Die Rotation eines Koordinatensystems mit Euler-Winkeln wird in Abbildung 2.1 schrittweise dargestellt.

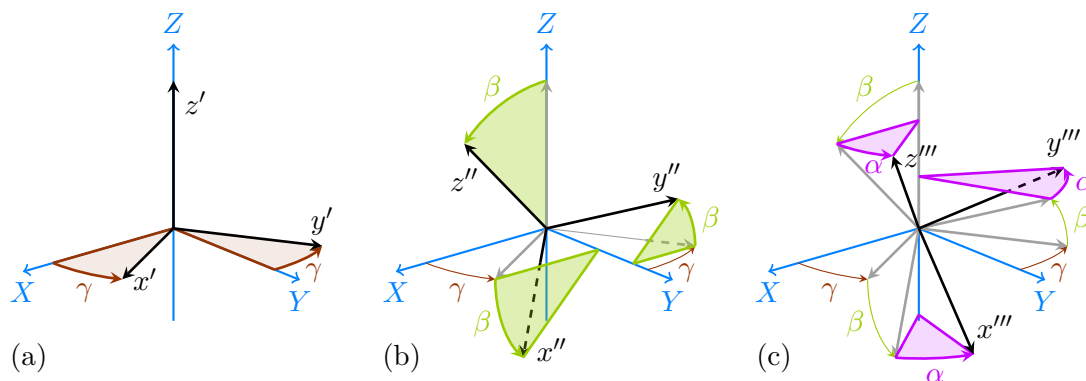


Abbildung 2.1: Rotation des Koordinatensystems (X, Y, Z) durch Euler-Winkel α, β, γ .

(a) Rotation um die Z-Achse mit Winkel γ liefert (x', y', z') .

(b) Danach erhalten wir (x'', y'', z'') durch Rotation mit β um die Y-Achse.

(c) Rotation mit α um die Z-Achse liefert das rotierte System (x''', y''', z''') .

Bemerkung 2.15. Für eine beliebige Rotation $\mathbf{R} \in \mathcal{SO}(3)$ existieren Euler-Winkel α, β, γ , so dass $\mathbf{R} = \mathbf{R}(\alpha, \beta, \gamma)$ gilt. Für $\beta \in (0, \pi)$ ist diese Parametrisierung bijektiv. In den Fällen $\beta = 0$ oder $\beta = \pi$ gelten für beliebige $\omega \in \mathbb{R}$ die Gleichungen

$$\mathbf{R}(\alpha, 0, \gamma) = \mathbf{R}(\alpha + \omega, 0, \gamma - \omega)$$

und

$$\mathbf{R}(\alpha, \pi, \gamma) = \mathbf{R}(\alpha + \omega, \pi, \gamma + \omega).$$

Wir können eine beliebige Rotation als Komposition von zwei Rotationen darstellen. Dabei finden nur variable Z -Achsen-Drehungen statt. Die Drehungen um die Y -Achse sind konstant. Dafür geben wir das folgende Lemma an.

Lemma 2.16. *Für eine Rotation in Euler-Winkel-Parametrisierung gilt*

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}(\alpha + \frac{\pi}{2}, \frac{\pi}{2}, \pi) \mathbf{R}(\beta, \frac{\pi}{2}, \gamma + \frac{\pi}{2}).$$

Beweis. Analog zu Gleichung (2.1) erfüllen die Achsendrehungen die Gleichung

$$\mathbf{R}_Y(\beta) = \mathbf{R}_X(-\frac{\pi}{2}) \mathbf{R}_Z(\beta) \mathbf{R}_X(\frac{\pi}{2})$$

für beliebige Winkel β . Durch Einsetzen von Gleichung (2.1) erhalten wir

$$\begin{aligned} \mathbf{R}_Y(\beta) &= \mathbf{R}_Z(-\frac{\pi}{2}) \mathbf{R}_Y(-\frac{\pi}{2}) \mathbf{R}_Z(\frac{\pi}{2}) \mathbf{R}_Z(\beta) \mathbf{R}_Z(-\frac{\pi}{2}) \mathbf{R}_Y(\frac{\pi}{2}) \mathbf{R}_Z(\frac{\pi}{2}) \\ &= \mathbf{R}_Z(-\frac{\pi}{2}) \mathbf{R}_Y(-\frac{\pi}{2}) \mathbf{R}_Z(\beta) \mathbf{R}_Y(\frac{\pi}{2}) \mathbf{R}_Z(\frac{\pi}{2}). \end{aligned}$$

Wir sehen leicht, dass

$$\mathbf{R}_Y(-\beta) = \mathbf{R}_Z(\pi) \mathbf{R}_Y(\beta) \mathbf{R}_Z(\pi) \tag{2.2}$$

gilt. Einsetzen liefert

$$\mathbf{R}_Y(\beta) = \mathbf{R}_Z(\frac{\pi}{2}) \mathbf{R}_Y(\frac{\pi}{2}) \mathbf{R}_Z(\pi) \mathbf{R}_Z(\beta) \mathbf{R}_Y(\frac{\pi}{2}) \mathbf{R}_Z(\frac{\pi}{2}).$$

Für die Euler-Winkel-Parametrisierung einer Rotation \mathbf{R} folgt

$$\mathbf{R}(\alpha, \beta, \gamma) = \mathbf{R}_Z(\alpha) \mathbf{R}_Y(\beta) \mathbf{R}_Z(\gamma) = \mathbf{R}_Z(\alpha + \frac{\pi}{2}) \mathbf{R}_Y(\frac{\pi}{2}) \mathbf{R}_Z(\pi) \mathbf{R}_Z(\beta) \mathbf{R}_Y(\frac{\pi}{2}) \mathbf{R}_Z(\gamma + \frac{\pi}{2})$$

und damit die Behauptung. \square

2.4 Integration auf $\mathcal{SO}(3)$ und der Raum $L_2(\mathcal{SO}(3))$

Wir wollen quadratintegrierbare Funktionen auf $\mathcal{SO}(3)$ durch Fourierreihen darstellen. Dazu benötigen wir das Skalarprodukt des Hilbertraums $L_2(\mathcal{SO}(3))$, das wiederum über einen Integrationsbegriff auf $\mathcal{SO}(3)$ definiert ist, siehe [37, Kapitel 3.1].

Das Integral einer Funktion $f: \mathcal{SO}(3) \rightarrow \mathbb{R}$ über $\mathcal{SO}(3)$ ist gegeben durch

$$\int_{\mathcal{SO}(3)} f(\mathbf{R}) \, d\mathbf{R}$$

mit dem normierten Volumenelement $d\mathbf{R}$ auf $\mathcal{SO}(3)$.

Lemma 2.17. Sei eine Funktion $f: \mathcal{SO}(3) \rightarrow \mathbb{R}$ gegeben. Dann gilt für die Euler-Winkel-Parametrisierung der Rotationsgruppe

$$\int_{\mathcal{SO}(3)} f(\mathbf{R}) \, d\mathbf{R} = \frac{1}{8\pi^2} \int_0^{2\pi} \int_0^\pi \int_0^{2\pi} f(\mathbf{R}(\alpha, \beta, \gamma)) \sin \beta \, d\alpha \, d\beta \, d\gamma.$$

Dabei bezeichnet $d\mathbf{R}$ das normierte Volumenelement auf $\mathcal{SO}(3)$.

Beweis. Mit Definition 2.13 und Beispiel 2.12 stellen wir Rotationsmatrizen

$$\mathbf{R} = \mathbf{R}_Z(\alpha) \mathbf{R}_Y(\beta) \mathbf{R}_Z(\gamma)$$

durch Euler-Winkel dar.

Für die Koordinatentransformation zwischen einer Rotationsmatrix und ihrer Euler-Winkel-Parametrisierung berechnen wir die Jacobi-Matrix $\mathbf{J} \in \mathbb{R}^{9 \times 3}$ durch

$$\mathbf{J} = \frac{\partial \mathbf{R}}{\partial(\alpha, \beta, \gamma)} = \begin{pmatrix} -\cos \alpha \sin \gamma - \cos \beta \cos \gamma \sin \alpha & -\cos \alpha \cos \gamma \sin \beta & -\cos \gamma \sin \alpha - \cos \alpha \cos \beta \sin \gamma \\ \cos \alpha \cos \beta \cos \gamma - \sin \alpha \sin \gamma & -\cos \gamma \sin \alpha \sin \beta & \cos \alpha \cos \gamma - \cos \beta \sin \alpha \sin \gamma \\ 0 & -\cos \beta \cos \gamma & \sin \beta \sin \gamma \\ \cos \beta \sin \alpha \sin \gamma - \cos \alpha \cos \gamma & \cos \alpha \sin \beta \sin \gamma & \sin \alpha \sin \gamma - \cos \alpha \cos \beta \cos \gamma \\ -\cos \gamma \sin \alpha - \cos \alpha \cos \beta \sin \gamma & \sin \alpha \sin \beta \sin \gamma & -\cos \alpha \sin \gamma - \cos \beta \cos \gamma \sin \alpha \\ 0 & \cos \beta \sin \gamma & \cos \gamma \sin \beta \\ -\sin \alpha \sin \beta & \cos \alpha \cos \beta & 0 \\ \cos \alpha \sin \beta & \cos \beta \sin \alpha & 0 \\ 0 & -\sin \beta & 0 \end{pmatrix}.$$

Mithilfe des Transformationssatzes gilt

$$\int_{\mathcal{SO}(3)} d\mathbf{R} = \int_0^{2\pi} \int_0^\pi \int_0^{2\pi} \sqrt{\det(\mathbf{J}^T \mathbf{J})} \, d\alpha \, d\beta \, d\gamma.$$

Aus

$$\mathbf{J}^T \mathbf{J} = \begin{pmatrix} 2 & 0 & 2 \cos \beta \\ 0 & 2 & 0 \\ 2 \cos \beta & 0 & 2 \end{pmatrix}$$

erhalten wir $\det(\mathbf{J}^T \mathbf{J}) = 8 \sin^2 \beta$ und es gilt

$$\int_{\mathcal{SO}(3)} d\mathbf{R} = \sqrt{8} \int_0^{2\pi} \int_0^\pi \int_0^{2\pi} \sin \beta \, d\alpha \, d\beta \, d\gamma = 8\sqrt{8}\pi^2.$$

Normieren des Volumenelements $d\mathbf{R}$ liefert $d\mathbf{R} = \frac{1}{8\pi^2} \sin \beta \, d\alpha \, d\beta \, d\gamma$ und damit die Behauptung. \square

Analog zum Euklidischen definieren wir den Raum der quadratintegrierbaren Funktionen auf $\mathcal{SO}(3)$.

Definition 2.18. Sei $L_2(\mathcal{SO}(3))$ der Banachraum der messbaren Funktionen $f: \mathcal{SO}(3) \rightarrow \mathbb{C}$ mit endlicher Norm

$$\|f\|_{L_2(\mathcal{SO}(3))}^2 := \int_{\mathcal{SO}(3)} |f(\mathbf{R})|^2 d\mathbf{R}$$

in dem fast überall übereinstimmende Funktionen identifiziert werden. Seien $f, g \in L_2(\mathcal{SO}(3))$. Mit dem Skalarprodukt

$$\langle f, g \rangle_{L_2(\mathcal{SO}(3))} := \int_{\mathcal{SO}(3)} f(\mathbf{R}) \overline{g(\mathbf{R})} d\mathbf{R}$$

ist $L_2(\mathcal{SO}(3))$ ein Hilbertraum.

Definition 2.19. Sei $C(\mathcal{SO}(3))$ der Banachraum der stetigen Funktionen $f: \mathcal{SO}(3) \rightarrow \mathbb{C}$ mit der Supremumsnorm

$$\|f\|_{C(\mathcal{SO}(3))} := \max_{\mathbf{R} \in \mathcal{SO}(3)} |f(\mathbf{R})|.$$

Es ist offensichtlich, dass die stetigen Funktionen auf einer kompakten Menge quadratintegrierbar sind, d.h.

$$L_2(\mathcal{SO}(3)) \supset C(\mathcal{SO}(3)).$$

3 Die kontinuierliche Fouriertransformation auf der Rotationsgruppe

Wir wollen quadratintegrierbare Funktionen auf $\mathcal{SO}(3)$ durch Fourierreihen darstellen. Dazu definieren wir die komplexwertigen Wigner-D Funktionen auf $\mathcal{SO}(3)$ und zeigen mit dem Satz von Peter-Weyl, dass sie eine orthogonale Basis von $L_2(\mathcal{SO}(3))$ bilden.

Zur Definition der Wigner-D Funktionen verwenden wir einen Produktansatz bezüglich der Euler-Winkel von $\mathcal{SO}(3)$. Dabei multiplizieren wir komplexe Exponentialfunktionen im ersten und dritten Euler-Winkel mit einer reellwertigen Wigner-d Funktion im zweiten Euler-Winkel. Wigner-d Funktionen sind mit Jacobi-Polynomen konstruierbar und folgen ähnlich wie orthogonale Polynomsysteme einer 3-Term-Rekursion. Damit sind Wigner-d Funktionen in festen Funktionswerten effizient berechenbar.

Das Ziel dieses Kapitels besteht darin die normierten Wigner-D Funktionen als Orthonormalbasis des Hilbertraums $L_2(\mathcal{SO}(3))$ zur Definition von Fourierreihen auf $\mathcal{SO}(3)$ zu verwenden und die Konvergenz dieser $\mathcal{SO}(3)$ -Fourierreihen zu untersuchen.

3.1 Orthogonalpolynome

Sei Π_n der lineare Raum aller Polynome $p: \mathbb{R} \rightarrow \mathbb{C}$ mit Grad kleiner gleich $n \in \mathbb{N}$. Die folgende Definition erhalten wir von Szegő [33, Kapitel 2] und Plonka [23, Kapitel 6.5.1].

Definition 3.1. *Sei eine kompakte Menge Ω gegeben. Weiterhin haben wir eine zulässige Gewichtsfunktion $\omega: \Omega \rightarrow \mathbb{R}$, d.h. ω ist nichtnegativ, integrierbar auf Ω und hat höchstens endlich viele Nullstellen auf Ω .*

Für Funktionen $f, g \in L_2(\Omega)$ definieren wir das gewichtete Skalarprodukt

$$\langle f, g \rangle_\omega = \int_\Omega \omega(x) f(x) \overline{g(x)} dx.$$

Eine Folge $(p_j)_{j=0}^\infty$ von Polynomen $p_j \in \Pi_j$, $j \in \mathbb{N}_0$ heißt Folge von Orthogonalpolynomen in $L_2(\Omega)$ bezüglich ω , wenn die Gleichungen $\langle p_j, p_k \rangle_\omega = 0$ für alle $j \neq k$ erfüllt sind und jedes Polynom p_j Grad j hat.

Offensichtlich bilden die Orthogonalpolynome p_0, \dots, p_N eine Orthogonalbasis des Polynomraumes Π_N bezüglich des Skalarproduktes $\langle \cdot, \cdot \rangle_\omega$.

Jedes System von Orthogonalpolynomen erfüllt eine 3-Term-Rekursion, siehe [33, Kapitel 3.2]. Wir geben im Folgenden zwei Beispiele für orthogonale Polynomsysteme mit den zugehörigen Rekursionsformeln an.

3.1.1 Die Chebyshev-Polynome erster Art

In der Praxis werden die Nullstellen der Chebyshev-Polynome sehr häufig als Stützstellen bei der Polynominterpolation verwendet. Weitere Anwendungen sind die Approximation von Funktionen durch eine Reihe von Chebyshev-Polynomen und damit insbesondere die Clenshaw-Curtis-Quadratur.

Definition 3.2. Sei $n \in \mathbb{N}_0$. Die Chebyshev-Polynome erster Art $T_n: \mathbb{R} \rightarrow \mathbb{R}$ mit

$$T_n(x) = \cos(n \cdot \arccos x)$$

bilden ein System orthogonaler Polynome auf $[-1, 1]$ bezüglich der Gewichtsfunktion $\frac{1}{\sqrt{1-x^2}}$.

Die Chebyshev-Polynome erfüllen die folgende Rekursionsformel, siehe [12, Kapitel 1].

Bemerkung 3.3. Offensichtlich gilt $T_0(x) = 1$ und $T_1(x) = x$. Aus den Additionstheoremen erhalten wir

$$\cos((n+1)t) = 2 \cos t \cos(nt) - \cos((n-1)t)$$

und für $t = \arccos x$ folgt die 3-Term-Rekursion

$$T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x).$$

Damit ist $T_n(x)$ ein Polynom vom Grad n .

3.1.2 Die Jacobi-Polynome

Zur Konstruktion der Basisfunktionen von $L_2(\mathcal{SO}(3))$ verwenden wir die Wigner-d Funktionen. Diese lassen sich durch Jacobi-Polynome darstellen. Damit kommen wir zur folgenden Definition, siehe [12, Kapitel 2.4].

Definition 3.4. Seien $a, b \in \mathbb{N}_0$ fixiert. Weiterhin sei $s \in \mathbb{N}_0$. Die Jacobi-Polynome $P_s^{a,b}: \mathbb{R} \rightarrow \mathbb{R}$ mit

$$P_s^{a,b}(x) := \sum_{k=0}^s \binom{s+a}{s-k} \binom{s+b}{k} \left(\frac{x-1}{2}\right)^k \left(\frac{x+1}{2}\right)^{s-k}$$

bilden ein System orthogonaler Polynome auf dem Intervall $[-1, 1]$ bezüglich der Gewichtsfunktion $(1-x)^a (1+x)^b$.

Aus der Definition erhalten wir direkt das folgende Korollar.

Korollar 3.5. Es gilt

$$P_s^{a,b}(x) = (-1)^s P_s^{b,a}(-x). \quad (3.1)$$

Szegö [33, Kapitel 2.4] liefert die 3-Term Rekursion der Jacobi-Polynome für fixierte Indizes a, b .

Bemerkung 3.6. Sei $s \in \mathbb{N}_0$ und $P_{-1}^{a,b}(x) = 0$. Dann gilt die Rekursionsformel

$$\tilde{t}_{a,b,s} P_{s+1}^{a,b}(x) = (\tilde{u}_{a,b,s} x + \tilde{v}_{a,b,s}) P_s^{a,b}(x) - \tilde{w}_{a,b,s} P_{s-1}^{a,b}(x) \quad (3.2)$$

mit den Koeffizienten

$$\begin{aligned} \tilde{t}_{a,b,s} &= 2(s+1)(s+1+a+b)(2s+a+b), \\ \tilde{u}_{a,b,s} &= (2s+a+b)(2s+1+a+b)(2s+2+a+b), \\ \tilde{v}_{a,b,s} &= (2s+1+a+b)(a^2-b^2), \\ \tilde{w}_{a,b,s} &= 2(s+a)(s+b)(2s+2+a+b). \end{aligned}$$

Wir geben einige wichtige Jacobi-Polynome an und bestimmen die Funktionswerte von beliebigen Jacobi-Polynomen in speziellen Punkten.

Beispiel 3.7.

(a) Aus Definition 3.4 folgt

$$\begin{aligned} P_0^{a,b}(x) &= 1 \quad \text{und} \\ P_1^{a,b}(x) &= \frac{1}{2}(a+b+2)x + \frac{1}{2}(a-b). \end{aligned} \quad (3.3)$$

als Startwerte für die Rekursionsformel (3.2).

(b) Für die Funktionswerte der Jacobi-Polynome in den Randpunkten -1 und 1 gilt

$$\begin{aligned} P_s^{a,b}(1) &= \binom{s+a}{s} \quad \text{und} \\ P_s^{a,b}(-1) &= (-1)^s \binom{s+b}{s} \end{aligned}$$

(c) Die Funktionswerte der Jacobi-Polynome im Punkt 0 erfüllen die verkürzte Rekursionsgleichung

$$\tilde{t}_{a,b,s} P_{s+1}^{a,b}(0) = \tilde{v}_{a,b,s} P_s^{a,b}(0) - \tilde{w}_{a,b,s} P_{s-1}^{a,b}(0).$$

Die Startwerte sind $P_0^{a,b}(0) = 1$ und $P_1^{a,b}(0) = \frac{1}{2}(a-b)$.

3.2 Die Wigner-d Funktionen

Die Jacobi-Polynome $P_s^{a,b}$ bilden ein System orthogonaler Polynome, mit dem wir im Folgenden die Wigner-d Funktionen definieren, siehe [35, Kapitel 4.3.4].

Definition 3.8. Seien $n \in \mathbb{N}_0$ und $k, l \in \mathbb{Z}$ mit $|k|, |l| \leq n$. Weiterhin sind die Konstanten

$$\begin{aligned} a &= |k - l|, \\ b &= |k + l|, \\ s &= n - \frac{a + b}{2} = n - \max\{|k|, |l|\} \quad \text{und} \\ \nu &= \begin{cases} 0 & \text{falls } l \geq k, \\ k + l & \text{sonst} \end{cases} \end{aligned}$$

gegeben. Dann ist die Wigner-d Funktion $d_n^{k,l}: [-1, 1] \rightarrow \mathbb{R}$ vom Grad n definiert durch

$$d_n^{k,l}(x) := (-1)^\nu \binom{2n-s}{s+a}^{\frac{1}{2}} \binom{s+b}{b}^{-\frac{1}{2}} \left(\frac{1-x}{2}\right)^{\frac{a}{2}} \left(\frac{1+x}{2}\right)^{\frac{b}{2}} P_s^{a,b}(x).$$

Für festes $x \in [-1, 1]$ ist $d_n(x) := \left(d_n^{k,l}(x)\right)_{k,l=-n}^n$ die Wigner-d Matrix mit Zeilenindex k und Spaltenindex l .

Offensichtlich können wir $x \in [-1, 1]$ mit der Substitution $x = \cos \beta$ durch $\beta \in [0, \pi]$ ersetzen. Damit erhalten wir die Wigner-d Funktion $d_n^{k,l}(\cos \beta)$.

Varshalovich gibt in [35, Kapitel 4.3] weitere explizite Formeln zur Konstruktion der Wigner-d Funktionen an.

Bemerkung 3.9. Offensichtlich hat das Jacobi-Polynom aus Definition 3.4 den Polynomgrad $s = n - \frac{a+b}{2}$. Damit ist die Wigner-d Funktion $d_n^{k,l}(x)$ ein Polynom vom Grad n , wenn $k \pm l$ gerade ist. Falls $k \pm l$ ungerade ist, dann ist $(1-x^2)^{-\frac{1}{2}} \cdot d_n^{k,l}(x)$ ein Polynom vom Grad $n-1$.

Wir betrachten einige spezielle Wigner-d Matrizen.

Beispiel 3.10.

- (a) Wir berechnen die Wigner-d Matrix mit Grad 0. Aus Definition 3.8 und Gleichung (3.3) folgt

$$d_0(x) = P_0^{0,0}(x) = 1.$$

- (b) Die Wigner-d Matrix mit Grad 1 erhalten wir analog. Sei $\beta \in [0, \pi]$.

Für $k = l = 0$ gilt

$$d_1^{0,0}(\cos \beta) = P_1^{0,0}(\cos \beta) = \cos \beta$$

und für k oder l ungleich 0 erhalten wir

$$d_1^{k,l}(\cos \beta) = (-1)^\nu \binom{2}{a}^{\frac{1}{2}} \left(\sin \frac{\beta}{2}\right)^a \left(\cos \frac{\beta}{2}\right)^b.$$

Daraus folgt

$$d_1(\cos \beta) = \begin{pmatrix} \cos^2 \frac{\beta}{2} & \sqrt{2} \sin \frac{\beta}{2} \cos \frac{\beta}{2} & \sin^2 \frac{\beta}{2} \\ -\sqrt{2} \sin \frac{\beta}{2} \cos \frac{\beta}{2} & \cos \beta & \sqrt{2} \sin \frac{\beta}{2} \cos \frac{\beta}{2} \\ \sin^2 \frac{\beta}{2} & -\sqrt{2} \sin \frac{\beta}{2} \cos \frac{\beta}{2} & \cos^2 \frac{\beta}{2} \end{pmatrix}.$$

- (c) Wir verwenden Definition 3.8 zur Berechnung der Wigner-d Matrix $d_n(1)$ mit beliebigem Grad n . Falls $k \neq l$ ist, dann gilt $d_n^{k,l}(1) = 0$.

Falls $k = l$ ist, dann haben wir $a = 0$, $b = 2|k|$ und $s = n - |k|$ und es gilt

$$d_n^{k,k}(1) = \binom{2n - n + |k|}{n - |k|}^{\frac{1}{2}} \binom{n - |k| + 2|k|}{2|k|}^{-\frac{1}{2}} P_{n-|k|}^{0,2|k|}(1) = P_{n-|k|}^{0,2|k|}(1).$$

Aus Beispiel 3.7 folgt $P_{n-|k|}^{0,2|k|}(1) = 1$ und damit erhalten wir die Wigner-d Matrix

$$d_n(1) = I_n$$

- (d) Analog zum vorherigen Beispiel gilt $d_n^{k,l}(-1) = 0$, falls $k \neq -l$ ist. Für den Fall $k = -l$ haben wir $a = 2|k|$, $b = 0$ und $s = n - |k|$ und unter Verwendung von Beispiel 3.7 gilt

$$d_n^{k,-k}(-1) = \binom{2n - n + |k|}{n - |k| + 2|k|}^{\frac{1}{2}} \binom{n - |k|}{0}^{-\frac{1}{2}} P_{n-|k|}^{2|k|,0}(-1) = P_{n-|k|}^{2|k|,0}(-1) = (-1)^{n-|k|}.$$

Damit erhalten wir die vorzeichenbehaftete Flip-Matrix

$$d_n(-1) = \begin{pmatrix} & & & & 1 \\ & & & -1 & \\ & & & 1 & \\ & & \ddots & & \\ -1 & & & & \\ 1 & & & & \end{pmatrix}.$$

Analog zur Rekursionsformel (3.2) der Jacobi-Polynome suchen wir eine Rekursion der Wigner-d Funktionen über den Grad n . Damit wollen wir Wigner-d Matrizen effizient berechnen. Wir erhalten das folgende Lemma, siehe [30].

Lemma 3.11. *Seien $n \in \mathbb{N}$ und $k, l \in \mathbb{Z}$ mit $|k|, |l| \leq n$. Dann erfüllen die Wigner-d Funktionen die 3-Term-Rekursion*

$$t_n^{k,l} d_{n+1}^{k,l}(x) = (u_n^{k,l} x - v_n^{k,l}) d_n^{k,l}(x) - w_n^{k,l} d_{n-1}^{k,l}(x) \quad (3.4)$$

mit den Koeffizienten

$$\begin{aligned} t_n^{k,l} &= n \sqrt{\left((n+1)^2 - k^2\right)\left((n+1)^2 - l^2\right)}, \\ u_n^{k,l} &= n(n+1)(2n+1), \\ v_n^{k,l} &= (2n+1)kl, \\ w_n^{k,l} &= (n+1) \sqrt{(n^2 - k^2)(n^2 - l^2)}. \end{aligned}$$

Dabei setzen wir $d_n^{k,l}(x) = 0$, falls die Ungleichung $n < \max\{|k|, |l|\}$ erfüllt ist.

Beweis. Wir brauchen die Gleichung

$$\begin{aligned} \left(\frac{a+b}{2}\right) \left(\frac{a-b}{2}\right) &= \left(\frac{|k-l| + |k+l|}{2}\right) \left(\frac{|k-l| - |k+l|}{2}\right) \\ &= \frac{1}{4} \left(|k-l|^2 - |k-l||k+l| + |k+l||k-l| - |k+l|^2\right) = \frac{-4kl}{4} = -kl. \end{aligned} \quad (3.5)$$

Für $x \in \{-1, 1\}$ folgt die Behauptung aus Beispiel 3.10 (c) und (d). Sei also $x \in (-1, 1)$. Dann ist die Konstante

$$c = (-1)^\nu \left(\frac{2n-s}{s+a}\right)^{-\frac{1}{2}} \left(\frac{s+b}{b}\right)^{\frac{1}{2}} \left(\frac{1-x}{2}\right)^{-\frac{a}{2}} \left(\frac{1+x}{2}\right)^{-\frac{b}{2}}$$

wohldefiniert. Aus der Definition 3.8 der Wigner-d Funktionen folgt

$$\begin{aligned} P_s^{a,b}(x) &= c \cdot d_n^{k,l}(x), \\ P_{s-1}^{a,b}(x) &= c \sqrt{\frac{(2n-s)s}{(s+a)(s+b)}} \cdot d_{n-1}^{k,l}(x), \\ P_{s+1}^{a,b}(x) &= c \sqrt{\frac{(s+a+1)(s+b+1)}{(2n-s+1)(s+1)}} \cdot d_{n+1}^{k,l}(x). \end{aligned}$$

Einsetzen in die Rekursionsformel (3.2) der Jacobi-Polynome und Division der gesamten Gleichung durch 4 liefert

$$\begin{aligned} \frac{\tilde{t}_{a,b,s}}{4} c \sqrt{\frac{(s+a+1)(s+b+1)}{(2n-s+1)(s+1)}} \cdot d_{n+1}^{k,l}(x) &= \left(\frac{\tilde{u}_{a,b,s}}{4} x + \frac{\tilde{v}_{a,b,s}}{4}\right) c \cdot d_n^{k,l}(x) \\ &\quad - \frac{\tilde{w}_{a,b,s}}{4} c \sqrt{\frac{(2n-s)s}{(s+a)(s+b)}} \cdot d_{n-1}^{k,l}(x) \end{aligned}$$

mit den Koeffizienten $\tilde{t}_{a,b,s}$, $\tilde{u}_{a,b,s}$, $\tilde{v}_{a,b,s}$, $\tilde{w}_{a,b,s}$ aus Bemerkung 3.6.

Wir kürzen die Konstante c . Mit $s = n - \frac{a+b}{2}$ gilt für die Koeffizienten

$$\begin{aligned}\frac{\tilde{t}_{a,b,s}}{4} &= \sqrt{\frac{(s+a+1)(s+b+1)}{(2n-s+1)(s+1)}} = t_n^{k,l}, \\ \frac{\tilde{u}_{a,b,s}}{4} &= \frac{1}{4} (2n)(2n+1)(2n+2) = u_n^{k,l}, \\ \frac{\tilde{w}_{a,b,s}}{4} &= \sqrt{\frac{(2n-s)s}{(s+a)(s+b)}} = w_n^{k,l}\end{aligned}$$

und wegen Gleichung (3.5) gilt

$$\frac{\tilde{v}_{a,b,s}}{4} = (2n+1) \left(\frac{a+b}{2}\right) \left(\frac{a-b}{2}\right) = -v_n^{k,l}.$$

Damit folgt die Behauptung. □

3.3 Symmetrieeigenschaften der Wigner-d Matrizen

Zur effizienten Berechnung von Wigner-d Funktionen verwenden wir Symmetrieeigenschaften in den Wigner-d Matrizen. Varshalovich gibt in [35, Kapitel 4.4] eine Reihe von Symmetrieeigenschaften an.

Lemma 3.12. *Seien $n \in \mathbb{N}_0$, $k, l \in \mathbb{Z}$ mit $|k|, |l| \leq n$ und $x \in [-1, 1]$. Dann erfüllen die Wigner-d Matrizen die Symmetrieeigenschaften*

$$d_n^{k,l}(x) = (-1)^{k+l} d_n^{-k,-l}(x) = (-1)^{k+l} d_n^{l,k}(x) = d_n^{-l,-k}(x). \quad (3.6)$$

Beweis.

- i) Zuerst zeigen wir $d_n^{k,l}(x) = (-1)^{k+l} d_n^{-k,-l}(x)$. Negieren von k und l lässt die Konstanten a, b, s aus Definition 3.8 unverändert. Folglich gilt $d_n^{-k,-l}(x) = d_n^{k,l}(x) \cdot (-1)^{\nu+\tilde{\nu}}$ mit

$$\nu = \begin{cases} 0 & \text{falls } l \geq k, \\ k+l & \text{sonst} \end{cases} \quad \text{und} \quad \tilde{\nu} = \begin{cases} 0 & \text{falls } -l \geq -k, \\ -k-l & \text{sonst.} \end{cases}$$

Eine Fallunterscheidung liefert $(-1)^{\nu+\tilde{\nu}} = (-1)^{k+l}$.

- ii) Wir zeigen $d_n^{k,l}(x) = (-1)^{k+l} d_n^{l,k}(x)$ analog, da Vertauschen von k und l die Konstanten a, b, s aus Definition 3.8 unverändert lässt.

- iii) Aus i) und ii) folgt $d_n^{k,l}(x) = d_n^{-l,-k}(x)$. □

Lemma 3.13. Seien $n \in \mathbb{N}_0$, $k, l \in \mathbb{Z}$ mit $|k|, |l| \leq n$ und $x \in [-1, 1]$. Dann erfüllen die Wigner-d Matrizen die Symmetrieeigenschaften

$$d_n^{k,l}(-x) = (-1)^{n+k} d_n^{k,-l}(x) = (-1)^{n+l} d_n^{-k,l}(x). \quad (3.7)$$

Beweis.

i) Wir zeigen $d_n^{k,l}(-x) = (-1)^{n+k} d_n^{k,-l}(x)$. Definition 3.8 liefert

$$d_n^{k,l}(-x) = (-1)^\nu \binom{2n-s}{s+a}^{\frac{1}{2}} \binom{s+b}{b}^{-\frac{1}{2}} \left(\frac{1+x}{2}\right)^{\frac{a}{2}} \left(\frac{1-x}{2}\right)^{\frac{b}{2}} P_s^{a,b}(-x).$$

Offensichtlich bleibt

$$\binom{2n-s}{s+a}^{\frac{1}{2}} \binom{s+b}{b}^{-\frac{1}{2}} = \sqrt{\frac{(2n-s)! b! s!}{(s+a)! (2n-2s-a)! (s+b)!}} = \sqrt{\frac{(2n-s)! b! s!}{(s+a)! b! (s+b)!}}$$

beim Tausch von a und b konstant. Zusammen mit Gleichung (3.1) folgt

$$d_n^{k,l}(-x) = (-1)^{\nu+s} \binom{2n-s}{s+b}^{\frac{1}{2}} \binom{s+a}{a}^{-\frac{1}{2}} \left(\frac{1+x}{2}\right)^{\frac{a}{2}} \left(\frac{1-x}{2}\right)^{\frac{b}{2}} P_s^{b,a}(x).$$

Negieren von l verursacht den Tausch von a und b und lässt s unverändert. Damit folgt

$$d_n^{k,l}(-x) = (-1)^{\nu+s+\tilde{\nu}} d_n^{k,-l}(x)$$

mit den Konstanten

$$\nu = \begin{cases} 0 & \text{falls } l \geq k, \\ k+l & \text{sonst} \end{cases} \quad \text{und} \quad \tilde{\nu} = \begin{cases} 0 & \text{falls } -l \geq k, \\ k-l & \text{sonst.} \end{cases}$$

Eine Fallunterscheidung liefert $(-1)^{\nu+\tilde{\nu}+s} = (-1)^{n+k}$ und damit die Behauptung.

ii) Die Gleichung $(-1)^{n+k} d_n^{k,-l}(x) = (-1)^{n+l} d_n^{-k,l}(x)$ folgt direkt aus (3.6) i).

□

3.4 Effiziente Berechnung von Wigner-d Matrizen

Wir wollen Wigner-d Matrizen $d_n(x)$ für ein gegebenes $x \in [-1, 1]$ algorithmisch effizient bestimmen. Falls x die Randwerte -1 oder 1 annimmt, erhalten wir die zugehörigen Wigner-d Matrizen direkt aus Beispiel 3.10 (c) und (d). Sei also $x \in (-1, 1)$ fixiert.

Wir benötigen folgende Definition über die Symmetrie einer Matrix.

Definition 3.14. Eine quadratische Matrix $A = (a_{i,j})_{i,j=-n}^n$ heißt persymmetrisch, wenn sie symmetrisch bezüglich ihrer Gegendiagonale ist, d.h.

$$a_{i,j} = a_{-j,-i}$$

für alle $i, j = -n, \dots, n$.

Die Symmetrieeigenschaften aus Lemma 3.12 implizieren, dass Wigner-d Matrizen bis auf Vorzeichenwechsel symmetrisch und persymmetrisch sind. Dies wird in Abbildung 3.1a dargestellt. Es ist also hinreichend den Fundamentalsektor

$$\left(d_n^{k,l}(x) \right)_{(k,l) \in I} \quad \text{mit } I = \{(k,l) \mid k = -n, \dots, n; l \leq -|k|\}$$

der Wigner-d Matrix $d_n(x)$ zu berechnen und die restlichen Werte mithilfe der Symmetrieeigenschaften zu bestimmen.

Falls das Argument $x = 0$ ist, dann erhalten wir die zusätzlichen Symmetrieeigenschaften in Abbildung 3.1b aus Gleichung (3.7). Dadurch halbiert sich die Indexmenge des Fundamentalsektors der Wigner-d Matrix $d_n(x)$ zu

$$I = \{(k,l) \mid k = -n, \dots, 0; l \leq k\}.$$

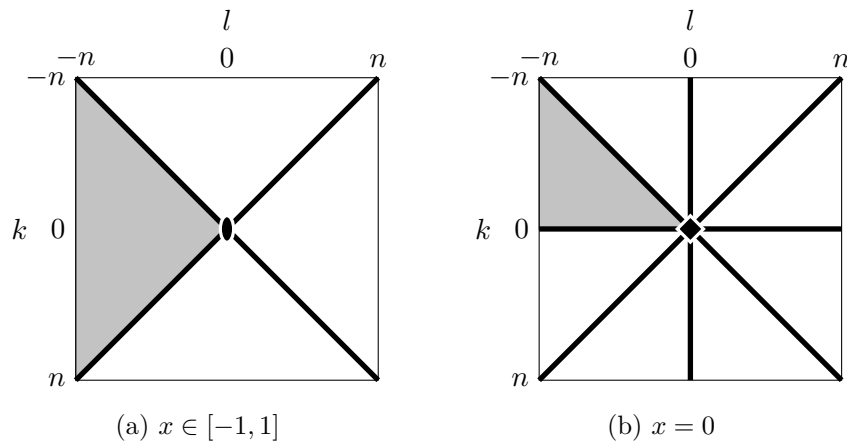


Abbildung 3.1: Darstellung der betragsmäßigen Symmetrie einer Wigner-d Matrix $d_n^{k,l}(x)$ mit Zeilenindex $k = -n, \dots, n$ und Spaltenindex $l = -n, \dots, n$. Die durchgezogenen Linien stellen Spiegelachsen dar und die graue Fläche markiert den Fundamentalsektor. Auf der linken Seite ist die Symmetrie für beliebige $x \in [-1, 1]$ abgebildet. Rechts haben wir zusätzliche Symmetrieeigenschaften, falls $x = 0$ gilt.

Unser Ziel ist es, alle Wigner-d Matrizen mit Grad kleiner gleich N zu berechnen. Dazu verwenden wir die 3-Term-Rekursion aus Lemma 3.11. Diese Rekursionsformel ist für steigenden Grad n numerisch stabil, siehe [30, Kapitel 2]. Diese Vorwärtsstabilität ist eine

Besonderheit der Wigner-d Funktionen. Im Vergleich dazu ist die 3-Term-Rekursion eines Systems orthogonaler Polynome üblicherweise rückwärtsstabil. Das wird bei der Auswertung einer Linearkombination von orthogonalen Polynomen mit dem Clenshaw-Algorithmus ausgenutzt, siehe [29, Kapitel 5.5].

Seien zwei Wigner-d Matrizen $d_n(x)$ und $d_{n-1}(x)$ gegeben. Dann liefert die 3-Term-Rekursion aus Lemma 3.11 die Wigner-d Matrixelemente $d_{n+1}^{k,l}(x)$ für ganzzahlige Indizes k, l mit $|k|, |l| \leq n$. Es fehlen die Matrixelemente mit $|k| = n+1$ oder $|l| = n+1$. Aus Definition 3.8 und Gleichung (3.3) folgt

$$d_{n+1}^{k, -(n+1)}(x) = (-1)^{n+1+k} \sqrt{\frac{(2n+2)!}{(n+1-k)!(n+1+k)!}} \left(\frac{1-x}{2}\right)^{\frac{n+1+k}{2}} \left(\frac{1+x}{2}\right)^{\frac{n+1-k}{2}} \quad (3.8)$$

für $k = -(n+1), \dots, n+1$. Mit den Symmetrieeigenschaften aus Lemma 3.12 erzeugen wir die restlichen Wigner-d Matrixelemente.

Falls der Grad n einer Wigner-d Matrix groß ist, dann werden die Funktionswerte einiger Wigner-d Funktionen betragsmäßig sehr klein. Das betrifft insbesondere die Wigner-d Funktionen mit betragsmäßig großen Indizes k und l . Falls das Argument $x = 0$ und der Grad $n \leq 512$ ist, dann sind die zugehörigen Wigner-d Matrizen im IEEE double Format ohne Probleme darstellbar. Um die Wigner-d Matrizen für beliebige $x \in [-1, 1]$ und größeren Grad n trotzdem darstellen zu können, wird in [30, Kapitel 3.2] ein spezieller Datentyp definiert.

In Kapitel 4 werden wir einen schnellen Algorithmus zur Auswertung von Funktionen auf der Rotationsgruppe angeben. Dabei benötigen wir die Wigner-d Matrizen im Argument $x = 0$. Zur rekursiven Berechnung dieser Wigner-d Matrizen $d_n(0)$ geben wir die Algorithmen 3.1 und 3.2 an.

Algorithmus 3.1 : Berechnung einer Wigner-d Matrix im Punkt $x = 0$

Input : $n \in \mathbb{N}$ Grad
 $d_{n-1}(0), d_n(0)$ Wigner-d Matrizen

for $k = -(n+1)$ *bis* 0 **do**
 | Berechne $d_{n+1}^{k, -(n+1)}(0)$ mit Gleichung (3.8)
end

for $l = -n$ *bis* 0 **do**
 | **for** $k = l$ *bis* 0 **do**
 | | Bestimme $d_{n+1}^{k,l}(0)$ mit der 3-Term-Rekursion (3.4) aus $d_n^{k,l}(0)$ und $d_{n-1}^{k,l}(0)$
 | **end**
end

Vervollständige $d_{n+1}(0)$ mit den Symmetrieeigenschaften (3.6) und (3.7)

Output : Wigner-d Matrix $d_{n+1}(0)$

Komplexität : $\mathcal{O}(n^2)$

Algorithmus 3.2 : Berechnung aller Wigner-d Matrizen $d_n(0)$ mit Grad $n \leq N$

Input : $N \in \mathbb{N}$ Maximaler Grad
 $d_0(0), d_1(0)$ Wigner-d Matrizen aus Beispiel 3.10 (a) und (b)

for $n = 1$ *bis* $N - 1$ **do**
 | Bestimme $d_{n+1}(0)$ mit Algorithmus 3.1 aus $d_n(0)$ und $d_{n-1}(0)$
end

Output : Wigner-d Matrizen $d_n(0)$ für $n = 0, \dots, N$

Komplexität : $\mathcal{O}(N^3)$

Angenommen wir suchen einen Algorithmus um eine einzelne Wigner-d Matrix vom Grad n zu bestimmen. Dann können wir die Idee von oben verbessern und speichern in jedem Rekursionsschritt nur die beiden letzten Wigner-d Matrizen ab.

Anstelle der Berechnung aller Wigner-d Matrizen bis zum Grad n wird alternativ in [30, Kapitel 2] ein effizienteres Verfahren beschrieben. Dazu geben wir die Rekursionsformel

$$\frac{1}{2} \sqrt{(n-l)(n+l+1)} d_n^{k,l+1}(x) = \frac{lx-k}{\sqrt{1-x^2}} d_n^{k,l}(x) - \frac{1}{2} \sqrt{(n+l)(n-l+1)} d_n^{k,l-1}(x) \quad (3.9)$$

mit Rekursion über den Spaltenindex l an. Diese 3-Term-Rekursion ist numerisch stabil, wenn $|l|$ fällt.

Für betragsmäßig große Indizes k und l werden einige Funktionswerte der Wigner-d Funktionen betragsmäßig sehr klein. Falls die Werte im IEEE double Format nicht mehr darstellbar sind und auf 0 gesetzt werden, schlägt beim Start $l = -n$ die gesamte Rekursion fehl. Deshalb wird in [30, Kapitel 3.1] eine Methode angegeben, deren Idee darin besteht, die Verhältnisse

$$\frac{d_n^{0,-n}(x)}{d_n^{0,-n+1}(x)}, \frac{d_n^{0,-n+1}(x)}{d_n^{0,-n+2}(x)}, \dots, \frac{d_n^{0,-2}(x)}{d_n^{0,-1}(x)}, \frac{d_n^{0,-1}(x)}{d_n^{0,0}(x)} \quad (3.10)$$

aus der Rekursionsformel (3.9) zu bestimmen. Als Startwert berechnen wir $d_n^{0,0}(x)$ effizient mit der Rekursion aus Lemma 3.11. Nun bestimmen wir die Wigner-d Matrixelemente $d_n^{0,l}(x)$ rekursiv für $l = -1, \dots, -n$ aus den Verhältnissen (3.10).

Als zweites berechnen wir analog die Verhältnisse für jede Zeile k und verwenden diese Verhältnisse zur rekursiven Bestimmung der Wigner-d Matrixelemente $d_n^{k,l}(x)$ für die Spaltenindizes $l = -1, \dots, -n$. Den Startwert $d_n^{k,0}(x)$ erhalten wir aus den anfangs berechneten Wigner-d Matrixelementen $\{d_n^{0,k}(x) \mid k = -n, \dots, 0\}$ mit der Symmetrieeigenschaft (3.6). Wir können diesen Schritt für alle Zeilen $k = -n, \dots, 0$ der Wigner-d Matrix parallelisieren. Als letztes liefern die Symmetrieeigenschaften die volle Wigner-d Matrix.

Im seltenen Fall, dass ein Verhältnis $d_n^{k,l-1}(x)/d_n^{k,l}(x)$ betragsmäßig unendlich groß wird, setzen wir $d_n^{k,l}(x) = 0$ und überspringen dieses Element der Wigner-d Matrix. Dazu bestimmen wir aus Gleichung (3.9) das Verhältnis $d_n^{k,l-1}(x)/d_n^{k,l+1}(x)$, siehe [30, Kapitel 3.1].

3.5 Die Spherical Harmonics

Wir werden den Satz von Peter-Weyl als zentrales Resultat der Darstellungstheorie verwenden, um eine orthogonale Basis auf $L_2(\mathcal{SO}(3))$ zu konstruieren. Dazu benötigen wir eine Orthonormalbasis auf $L_2(\mathbb{S}^2)$. Zur Definition der orthonormalen Basisfunktionen verwenden wir die folgende Parametrisierung der 2-Sphäre \mathbb{S}^2 , siehe [35, Kapitel 1.1.2].

Definition 3.15. Die Kugelkoordinaten r, ϑ, φ liefern eine Parametrisierung des \mathbb{R}^3 . Dabei bezeichnet der Radius r den Abstand zum Koordinatenursprung, $\vartheta \in [0, \pi]$ ist der Polarwinkel und $\varphi \in [0, 2\pi)$ der Azimutwinkel, siehe Abbildung 3.2. Wir können die kartesischen Koordinaten eines Punktes mit

$$\begin{aligned}x &= r \sin \vartheta \cos \varphi \\y &= r \sin \vartheta \sin \varphi \\z &= r \cos \vartheta\end{aligned}$$

aus den Kugelkoordinaten berechnen.

Die 2-Sphäre \mathbb{S}^2 ist definiert durch

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

Wir geben Punkte $\xi \in \mathbb{S}^2$ durch Kugelkoordinaten $(\vartheta, \varphi) \in [0, \pi] \times [0, 2\pi)$ mit Radius $r = 1$ an.

Mithilfe der sphärischen Koordinaten können wir Funktionen auf \mathbb{S}^2 definieren. Wir erhalten die folgende Definition aus [35, Kapitel 5.2.7].

Definition 3.16. Seien $n \in \mathbb{N}_0$ und $k \in \mathbb{Z}$ mit $|k| \leq n$ gegeben. Weiterhin haben wir das Jacobi-Polynom $P_{n-|k|}^{|k|, |k|}$ und die Konstante $\nu = \max\{k, 0\}$.

Die Spherical Harmonics $Y_n^k: \mathbb{S}^2 \rightarrow \mathbb{C}$ mit

$$Y_n^k(\vartheta, \varphi) = (-1)^\nu \frac{e^{i k \varphi}}{2^{|k|} n!} \sqrt{\frac{2n+1}{4\pi} (n-k)! (n+k)!} (\sin \vartheta)^{|k|} P_{n-|k|}^{|k|, |k|}(\cos \vartheta)$$

sind spezielle Funktionen auf der Sphäre.

Bemerkung 3.17. Die Spherical Harmonics sind die normierten Eigenfunktionen des Laplace-Operators mit separierbaren Koordinaten ϑ und φ .

Sie bilden eine Orthonormalbasis in $L_2(\mathbb{S}^2)$, siehe [7, Kapitel 7.3], und sind rotationsinvariant.

Varshalovich gibt in [35, Kapitel 5] weitere Definitionen der Spherical Harmonics an.

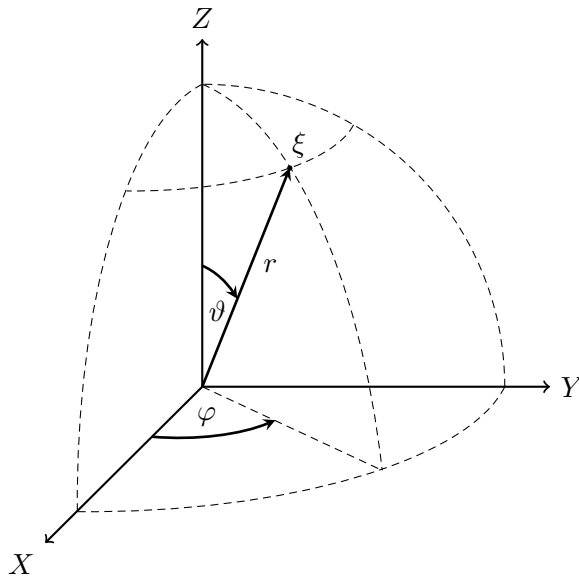


Abbildung 3.2: Darstellung des Punktes $\xi \in \mathbb{R}^3$ mit Kugelkoordinaten bestehend aus dem Polarwinkel $\vartheta \in [0, \pi]$, dem Azimutwinkel $\varphi \in [0, 2\pi)$ und dem Radius r .

3.6 Der Satz von Peter-Weyl

In der Darstellungstheorie werden die Eigenschaften einer topologischen Gruppe untersucht. Dazu werden die Gruppenelemente mit Homomorphismen auf Matrizen abgebildet. Der Satz von Peter-Weyl ist ein zentrales Resultat der Darstellungstheorie, das wir verwenden werden, um eine Basis auf $L_2(\mathcal{SO}(3))$ anzugeben. Zum Verständnis des Satzes benötigen wir einige grundlegende Begriffe aus der Darstellungstheorie, siehe bei Vilenkin [36, Kapitel 2.1 und 2.2].

Definition 3.18. Sei ein Vektorraum V über \mathbb{R} mit Dimension N gegeben. Die allgemeine lineare Gruppe $GL(V)$ ist die Gruppe aller bijektiven linearen Abbildungen $V \rightarrow V$. Eine Darstellung T von einer Gruppe G auf einen Vektorraum V ist ein Homomorphismus $T: G \rightarrow GL(V)$. Die Dimension der Darstellung T ist N .

Definition 3.19. Seien eine Gruppe G und zwei Vektorräume V und W gegeben. Die Darstellungen $T: G \rightarrow GL(V)$ und $\tilde{T}: G \rightarrow GL(W)$ heißen äquivalent, falls ein Vektorraumisomorphismus $\psi: V \rightarrow W$ existiert, so dass

$$\tilde{T}(g) = \psi \circ T(g) \circ \psi^{-1}$$

für alle $g \in G$ gilt.

Bemerkung 3.20. Offensichtlich ist die Äquivalenz von Darstellungen eine Äquivalenzrelation. Damit teilen wir die Darstellungen von einer Gruppe G in Äquivalenzklassen ein.

Definition 3.21. Sei eine Darstellung T von einer Gruppe G auf einen Vektorraum V gegeben. Ein Unterraum U von V heißt G -invariant bezüglich der Darstellung T , falls

$$T(g)U \subseteq U$$

für alle $g \in G$ gilt.

Eine Darstellung heißt irreduzibel, wenn die einzigen G -invarianten Unterräume von V die trivialen Unterräume V und $\{0\}$ sind. Falls ein nichttrivialer G -invarianter Unterraum von V existiert, dann heißt T reduzibel.

Die Restriktion von T auf einen nichttrivialen G -invarianten Unterraum liefert eine weitere Darstellung von G . Wir bezeichnen diese als Unterdarstellung von T .

Eine Darstellung T heißt vollständig reduzibel, wenn T eine direkte Summe von irreduziblen Unterdarstellungen $T_n: G \rightarrow GL(V_n)$, $n \in I$ ist, d.h.

$$V = \bigoplus_{n \in I} V_n.$$

Beispiel 3.22. Die bezüglich der Spherical Harmonics Y_n^k definierten Harmonischen Räume

$$\text{Harm}_n(\mathbb{S}^2) := \text{span}\{Y_n^k \mid k = -n, \dots, n\}$$

liefern ein vollständiges System $\mathcal{SO}(3)$ -invarianter Unterräume von $L_2(\mathbb{S}^2)$, d.h.

$$L_2(\mathbb{S}^2) = \text{clos}_{L_2} \bigoplus_{n=0}^{\infty} \text{Harm}_n(\mathbb{S}^2),$$

siehe [7, Kapitel 7.3]. Der Harmonische Raum $\text{Harm}_n(\mathbb{S}^2)$ hat als $\mathcal{SO}(3)$ -invariante Unterräume nur die trivialen Unterräume $\{0\}$ und $\text{Harm}_n(\mathbb{S}^2)$.

Sei eine Darstellung $D: \mathcal{SO}(3) \rightarrow GL(L_2(\mathbb{S}^2))$ von $\mathcal{SO}(3)$ auf $L_2(\mathbb{S}^2)$ gegeben. Dann ist D vollständig reduzibel und zerfällt für alle $n \in \mathbb{N}_0$ in irreduzible Unterdarstellungen $D_n: \mathcal{SO}(3) \rightarrow GL(\text{Harm}_n(\mathbb{S}^2))$.

Die Harmonischen Räume $\text{Harm}_n(\mathbb{S}^2)$ haben die Dimensionen $2n + 1$. Damit sind die Harmonischen Räume und insbesondere die Darstellungen D_n endlichdimensional.

Definition 3.23. Eine Darstellung T von einer Gruppe G auf einem Hilbertraum H heißt unitär, falls im Hilbertraumskalarprodukt $\langle \cdot, \cdot \rangle$ für alle $g \in G$ und $x, y \in H$ die Gleichung

$$\langle T(g)x, T(g)y \rangle = \langle x, y \rangle$$

gilt.

Sei zusätzlich eine Orthonormalbasis $\{e_i\}$ auf H gegeben. Die komplexwertige Funktion $t^{k,l}(g) = \langle T(g)e_k, e_l \rangle$ mit $g \in G$ heißt Matrixelement auf der Gruppe G .

Wir benötigen das folgende Lemma. Den Beweis gibt Vilenkin in [36, Kapitel 2.2.3] an.

Lemma 3.24. *Sei eine endlichdimensionale Darstellung T von einer kompakten Gruppe G auf einem Hilbertraum H gegeben, dann existiert ein G -invariantes Skalarprodukt auf H und T ist unitär bezüglich dieses Skalarproduktes.*

Damit können wir nun den zentralen Satz der Darstellungstheorie formulieren. Siehe bei Vilenkin [36, Kapitel 2.3] oder Weyl [39, Kapitel 3.11] für den Beweis.

Satz 3.25 (Satz von Peter-Weyl). *Sei $\Lambda = \{T_n\}$ die Menge aller Äquivalenzklassen der endlichdimensionalen, unitären, irreduziblen Darstellungen von einer kompakten Gruppe G . Weiterhin seien die Matrixelemente $t_n^{k,l}$ der Darstellungen T_n gegeben. Dann bildet die Menge der Funktionen*

$$\{t_n^{k,l}: G \rightarrow \mathbb{C} \mid T_n \in \Lambda; k, l = 1, \dots\}$$

eine Orthogonalbasis von $L_2(G)$.

3.7 Die Wigner-D Funktionen

Mithilfe des Satzes von Peter-Weyl aus der Darstellungstheorie bestimmen wir eine Orthogonalbasis von $L_2(\mathcal{SO}(3))$. Dazu folgen wir der Dissertation von Vollrath [37, Kapitel 3.2]. Zuerst definieren wir basierend auf den Wigner-d Funktionen das folgende Funktionensystem, siehe [35, Kapitel 4.3].

Definition 3.26. *Seien $n \in \mathbb{N}_0$ und $k, l \in \mathbb{Z}$ mit $|k|, |l| \leq n$. Weiterhin seien die Rotationen $\mathbf{R} \in \mathcal{SO}(3)$ durch Euler-Winkel parametrisiert. Dann ist die Wigner-D Funktion $\tilde{D}_n^{k,l}: \mathcal{SO}(3) \rightarrow \mathbb{C}$ vom Grad n durch*

$$\tilde{D}_n^{k,l}(\mathbf{R}(\alpha, \beta, \gamma)) := e^{-i k \alpha} d_n^{k,l}(\cos \beta) e^{-i l \gamma}$$

definiert.

Bemerkung 3.27. *Wir sehen leicht ein, dass für die Spherical Harmonics*

$$Y_n^k(\beta, \alpha) = (-1)^k \sqrt{\frac{2n+1}{4\pi}} \tilde{D}_n^{-k,0}(\mathbf{R}(\alpha, \beta, \gamma))$$

gilt. Deshalb betrachten wir die Wigner-D Funktionen als verallgemeinerte Spherical Harmonics, siehe [35, Kapitel 5.2.7].

Wir wollen nun zeigen, dass die Wigner-D Funktionen eine orthogonale Basis auf $L_2(\mathcal{SO}(3))$ bilden, siehe [37, Kapitel 3.2]. Dazu verwenden wir den Satz von Peter-Weyl.

Korollar 3.28. *Die Menge der Wigner-D Funktionen $\{\tilde{D}_n^{k,l} \mid n \in \mathbb{N}_0, \text{ und } k, l = -n, \dots, n\}$ bildet eine orthogonale Basis von $L_2(\mathcal{SO}(3))$.*

Beweis. Wir zeigen, dass die Wigner-D Funktionen zusammen mit $\mathcal{SO}(3)$ die Voraussetzungen des Satzes von Peter-Weyl 3.25 erfüllen.

In Lemma 2.8 haben wir bereits gezeigt, dass $\mathcal{SO}(3)$ eine kompakte Gruppe ist.

Beispiel 3.22 liefert für $n \in \mathbb{N}_0$ die endlichdimensionalen, irreduziblen Darstellungen D_n von $\mathcal{SO}(3)$ auf $\text{Harm}_n(\mathbb{S}^2)$.

Die irreduziblen Darstellungen D_n von $\mathcal{SO}(3)$ haben eine ungerade Dimension und sind eindeutig, siehe Woit [40, Kapitel 8.3]. Folglich existiert keine weitere endlichdimensionale, irreduzible Darstellung τ , die zu keiner der Darstellungen $\{D_n\}_{n=0}^\infty$ äquivalent ist.

Damit ist $\Lambda = \{D_n\}_{n=0}^\infty$ die Menge aller Äquivalenzklassen von endlichdimensionalen, irreduziblen Darstellungen von $\mathcal{SO}(3)$.

Wegen Lemma 3.24 sind die Darstellungen D_n unitär bezüglich einem $\mathcal{SO}(3)$ -invarianten Skalarprodukt $\langle \cdot, \cdot \rangle$. Damit erhalten wir für jede Darstellung D_n die Matrixelemente

$$D_n^{k,l}(\mathbf{R}) = \langle D_n(\mathbf{R})Y_n^k, Y_n^l \rangle_{\text{Harm}_n(\mathbb{S}^2)}$$

für $k, l = -n, \dots, n$ und für alle $\mathbf{R} \in \mathcal{SO}(3)$.

Es bleibt zu zeigen, dass die Wigner-D Funktionen ein analytischer Ausdruck der Matrixelemente $D_n^{k,l}(\mathbf{R})$ in Euler-Winkel-Parametrisierung sind. Dazu zeigt Vollrath in [37, Kapitel 3.3], dass die Matrixelemente $D_n^{k,l}$ die Eigenfunktionen des Laplace-Beltrami Operators sind, d.h.

$$\Delta_{\mathcal{SO}(3)} D_n^{k,l} = n(n+1) D_n^{k,l}.$$

Durch Lösen der partiellen Differentialgleichung mit einem Produktansatz bezüglich der Euler-Winkel erhalten wir die Wigner-D Funktionen als analytischen Ausdruck der Matrixelemente. \square

Die Wigner-D Funktionen bilden eine Orthogonalbasis, sind aber nicht normiert. Mit dem Lemma von Schur [36, Kapitel 2.2.8] erhalten wir

$$\|\tilde{D}_n^{k,l}\|_{L_2(\mathcal{SO}(3))}^2 = \frac{1}{2n+1},$$

siehe [36, Kapitel 2.3.5]. Damit kommen wir zu folgender Definition.

Definition 3.29. *Seien $n \in \mathbb{N}_0$ und $k, l \in \mathbb{Z}$ mit $|k|, |l| \leq n$. Weiterhin seien die Rotationen $\mathbf{R} \in \mathcal{SO}(3)$ durch Euler-Winkel parametrisiert. Dann ist die normierte Wigner-D Funktion $D_n^{k,l}: \mathcal{SO}(3) \rightarrow \mathbb{C}$ vom Grad n durch*

$$D_n^{k,l}(\mathbf{R}(\alpha, \beta, \gamma)) = \sqrt{2n+1} \tilde{D}_n^{k,l}(\mathbf{R}(\alpha, \beta, \gamma)) = \sqrt{2n+1} e^{-ik\alpha} d_n^{k,l}(\cos \beta) e^{-il\gamma}$$

definiert.

Aus Korollar 3.28 erhalten wir direkt das folgende Resultat.

Korollar 3.30. *Die Menge der normierten Wigner-D Funktionen*

$$\{D_n^{k,l}: \mathcal{SO}(3) \rightarrow \mathbb{C} \mid n \in \mathbb{N}_0, \text{ und } k, l = -n, \dots, n\}$$

bildet eine Orthonormalbasis von $L_2(\mathcal{SO}(3))$.

Analog zu den Harmonischen Räumen auf \mathbb{S}^2 geben wir die folgende Definition an.

Definition 3.31. *Die normierten Wigner-D Funktionen mit Grad $n \in \mathbb{N}_0$ spannen den Harmonischen Raum*

$$\text{Harm}_n(\mathcal{SO}(3)) := \text{span}\{D_n^{k,l} \mid k, l = -n, \dots, n\}$$

als Unterraum von $L_2(\mathcal{SO}(3))$ auf.

Aus Korollar 3.30 folgt

$$L_2(\mathcal{SO}(3)) = \text{clos}_{L_2} \bigoplus_{n=0}^{\infty} \text{Harm}_n(\mathcal{SO}(3)),$$

siehe [37, Kapitel 3.2].

3.8 Harmonische Analysis

Wir führen einige Begriffe und Resultate der Fourieranalysis speziell für die Rotationsgruppe ein, siehe [36, Kapitel 2.3.5]. Dazu definieren wir Fourierreihen auf $\mathcal{SO}(3)$ und untersuchen deren Konvergenz. Wir folgen dem Buch von Plonka [23, Kapitel 1.2].

Als erstes definieren wir ein Analogon zum Raum der trigonometrischen Polynome mit Grad $\leq N$.

Definition 3.32. *Sei $N \in \mathbb{N}$. Der Raum der N -bandbreitenbegrenzten Funktionen*

$$\mathbb{D}_N := \bigoplus_{n=0}^N \text{Harm}_n(\mathcal{SO}(3)),$$

ist ein Unterraum von $L_2(\mathcal{SO}(3))$. Er hat die Dimension

$$\dim(\mathbb{D}_N) = \sum_{n=0}^N (2n+1)^2 = \frac{1}{3}(N+1)(2N+1)(2N+3). \quad (3.11)$$

Weiterhin sei die geordnete Indexmenge

$$\mathcal{J}_N := \{(n, k, l) \mid n = 0, \dots, N; k, l = -n, \dots, n\}$$

mit $|\mathcal{J}_N| = \dim(\mathbb{D}_N)$ gegeben.

Wir wollen Funktionen aus $L_2(\mathcal{SO}(3))$ mit der Orthonormalbasis von $L_2(\mathcal{SO}(3))$ darstellen. Zuerst betrachten wir eine Projektion auf den endlichdimensionalen Unterraum \mathbb{D}_N . Dazu verwenden wir folgende Definition.

Definition 3.33. *Sei $f \in L_2(\mathcal{SO}(3))$ gegeben. Dann sind die $\mathcal{SO}(3)$ -Fourierkoeffizienten von f bezüglich der normierten Wigner-D Funktionen $D_n^{k,l}$ für alle 3-Tupel $(n, k, l) \in \mathcal{J}_N$*

definiert als

$$\hat{f}_n^{k,l} = \langle f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))}.$$

Weiterhin definieren wir die N -te $\mathcal{SO}(3)$ -Fourierpartialsomme von f durch

$$S_N f := \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l} \in \mathbb{D}_N.$$

Nach Definition ist $S_N: L_2(\mathcal{SO}(3)) \rightarrow L_2(\mathcal{SO}(3))$ ein linearer Operator, der die folgende Approximationseigenschaft besitzt.

Lemma 3.34. *Der $\mathcal{SO}(3)$ -Fourierpartialsommenoperator $S_N: L_2(\mathcal{SO}(3)) \rightarrow L_2(\mathcal{SO}(3))$ ist eine orthogonale Projektion auf \mathbb{D}_N , d.h.*

$$\|f - S_N f\|_{L_2(\mathcal{SO}(3))} = \min_{p \in \mathbb{D}_N} \|f - p\|_{L_2(\mathcal{SO}(3))}$$

für beliebige $f \in L_2(\mathcal{SO}(3))$. Insbesondere gilt

$$\|f - S_N f\|_{L_2(\mathcal{SO}(3))}^2 = \|f\|_{L_2(\mathcal{SO}(3))}^2 - \sum_{n=0}^N \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2.$$

Beweis. Für eine beliebige N -bandbreitenbegrenzte Funktion

$$p = \sum_{n=0}^N \sum_{k,l=-n}^n c_n^{k,l} D_n^{k,l}$$

mit beliebigen Koeffizienten $c_n^{k,l} \in \mathbb{C}$ und für alle $f \in L_2(\mathcal{SO}(3))$ gilt

$$\begin{aligned} \|f - p\|_{L_2(\mathcal{SO}(3))}^2 &= \|f\|_{L_2(\mathcal{SO}(3))}^2 - \langle p, f \rangle_{L_2(\mathcal{SO}(3))} - \langle f, p \rangle_{L_2(\mathcal{SO}(3))} + \|p\|_{L_2(\mathcal{SO}(3))}^2 \\ &= \|f\|_{L_2(\mathcal{SO}(3))}^2 + \sum_{n=0}^N \sum_{k,l=-n}^n \left(-c_n^{k,l} \overline{\hat{f}_n^{k,l}} - \overline{c_n^{k,l}} \hat{f}_n^{k,l} + |c_n^{k,l}|^2 \right) \\ &= \|f\|_{L_2(\mathcal{SO}(3))}^2 - \sum_{n=0}^N \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2 + \sum_{n=0}^N \sum_{k,l=-n}^n |c_n^{k,l} - \hat{f}_n^{k,l}|^2. \end{aligned}$$

Daraus folgt

$$\|f - p\|_{L_2(\mathcal{SO}(3))}^2 \geq \|f\|_{L_2(\mathcal{SO}(3))}^2 - \sum_{n=0}^N \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2,$$

wobei Gleichheit nur im Fall $c_n^{k,l} = \hat{f}_n^{k,l}$ für alle $(n, k, l) \in \mathcal{J}_N$ gilt, d.h. genau dann, wenn $p = S_N f$ ist. \square

Wir erweitern die Definition der N -ten $\mathcal{SO}(3)$ -Fourierpartialsomme und verwenden die Orthonormalbasis von $L_2(\mathcal{SO}(3))$ zur Approximation einer Funktion $f \in L_2(\mathcal{SO}(3))$. Damit erhalten wir einen zentralen Satz der Fourieranalysis.

Satz 3.35. *Jede Funktion $f \in L_2(\mathcal{SO}(3))$ hat eine eindeutige $\mathcal{SO}(3)$ -Fourierreihenentwicklung*

$$f = \sum_{n=0}^{\infty} \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}$$

mit den $\mathcal{SO}(3)$ -Fourierkoeffizienten $\hat{f}_n^{k,l}$. Dabei konvergiert die Folge der $\mathcal{SO}(3)$ -Fourierpartialsommen $(S_N f)_{N=0}^{\infty}$ in $L_2(\mathcal{SO}(3))$ gegen f , d.h.

$$\lim_{N \rightarrow \infty} \|S_N f - f\|_{L_2(\mathcal{SO}(3))} = 0.$$

Weiterhin gilt die Parsevalsche Gleichung

$$\|f\|_{L_2(\mathcal{SO}(3))}^2 = \sum_{n=0}^{\infty} \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2. \quad (3.12)$$

Beweis. Aus Lemma 3.34 erhalten wir für alle $N \in \mathbb{N}_0$ und beliebige f

$$\|S_N f\|_{L_2(\mathcal{SO}(3))}^2 = \sum_{n=0}^N \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2 \leq \|f\|_{L_2(\mathcal{SO}(3))}^2 < \infty.$$

Für $N \rightarrow \infty$ folgt die Besselsche Ungleichung

$$\sum_{n=0}^{\infty} \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2 \leq \|f\|_{L_2(\mathcal{SO}(3))}^2.$$

Sei $\varepsilon > 0$. Dann existiert ein Index $N(\varepsilon) \in \mathbb{N}$, so dass

$$\sum_{n=N(\varepsilon)}^{\infty} \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2 < \varepsilon$$

gilt. Für beliebige Indizes $M > N \geq N(\varepsilon)$ erhalten wir

$$\|S_M f - S_N f\|_{L_2(\mathcal{SO}(3))}^2 = \sum_{n=N+1}^M \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2 < \varepsilon.$$

Damit ist $(S_N f)_{N=0}^{\infty}$ eine Cauchyfolge und konvergiert im Hilbertraum $L_2(\mathcal{SO}(3))$. Angenommen $\lim_{N \rightarrow \infty} S_N f =: g$ mit $g \in L_2(\mathcal{SO}(3))$. Da Orthogonalprojektionen selbstadjung-

iert sind, erhalten wir mit Lemma 3.34 die Gleichung

$$\begin{aligned} \langle g, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} &= \lim_{N \rightarrow \infty} \langle S_N f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} = \lim_{N \rightarrow \infty} \langle f, S_N D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} \\ &= \langle f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} \end{aligned}$$

für alle $(n, k, l) \in \mathcal{J}_N$. Wegen Korollar 3.30 gilt $f = g$ fast überall.

Für $n \rightarrow \infty$ erhalten wir mit Lemma 3.34 die Parsevalsche Gleichung (3.12). \square

Die $\mathcal{SO}(3)$ -Fourierkoeffizienten erfüllen die folgende Symmetrieeigenschaft.

Lemma 3.36. *Sei eine reellwertige Funktion $f \in L_2(\mathcal{SO}(3))$ gegeben. Dann erfüllen die $\mathcal{SO}(3)$ -Fourierkoeffizienten*

$$\hat{f}_n^{k,l} = (-1)^{k+l} \overline{\hat{f}_n^{-k,-l}}$$

für alle $(n, k, l) \in \mathcal{J}_N$.

Beweis. Sei $N \in \mathbb{N}$ und $(n, k, l) \in \mathcal{J}_N$ beliebig. Nach Definition der normierten Wigner-D Funktionen gilt

$$\overline{D_n^{-k,-l}(\mathbf{R})} = \sqrt{2n+1} e^{-i k \alpha} \overline{d_n^{-k,-l}(\cos \beta)} e^{-i l \gamma}.$$

Die Wigner-d Funktionen sind reellwertig. Aus Symmetrieeigenschaft (3.6) folgt

$$\overline{D_n^{-k,-l}(\mathbf{R})} = (-1)^{k+l} D_n^{k,l}(\mathbf{R}).$$

Einsetzen in die Definition der $\mathcal{SO}(3)$ -Fourierkoeffizienten liefert

$$\hat{f}_n^{k,l} = \langle f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} = (-1)^{k+l} \langle f, \overline{D_n^{-k,-l}} \rangle_{L_2(\mathcal{SO}(3))}.$$

Da f reellwertig ist, folgt die Behauptung. \square

Wir untersuchen die Konvergenz der $\mathcal{SO}(3)$ -Fourierkoeffizienten auf der Rotationsgruppe und geben ein Analogon zum Lemma von Riemann-Lebesgue an, siehe [23, Kapitel 1.4.1].

Lemma 3.37. *Für die $\mathcal{SO}(3)$ -Fourierkoeffizienten einer Funktion $f \in L_2(\mathcal{SO}(3))$ gilt*

$$\lim_{n \rightarrow \infty} \hat{f}_n^{k,l} = 0$$

für alle Indizes $k, l \in \mathbb{Z}$.

Beweis. Aus der Parsevalschen Gleichung (3.12) folgt

$$\lim_{n \rightarrow \infty} \sum_{k,l=-n}^n |\hat{f}_n^{k,l}|^2 = 0.$$

Insbesondere konvergiert $|\hat{f}_n^{k,l}|^2$ für feste $k, l \in \mathbb{Z}$ gegen 0. Daraus folgt die Behauptung. \square

Aus Satz 3.35 folgt, dass die $\mathcal{SO}(3)$ -Fourierreihe einer beliebigen Funktion $f \in L_2(\mathcal{SO}(3))$ in der Norm von $L_2(\mathcal{SO}(3))$ gegen f konvergiert. Dies wird Konvergenz im 2-ten Mittel genannt. Im Folgenden untersuchen wir gleichmäßige und punktweise Konvergenz von stetigen Funktionen, siehe [23, Kapitel 1.4.2]. Dazu müssen wir zusätzliche Bedingungen an die Funktionen stellen.

Lemma 3.38. *Wenn die $\mathcal{SO}(3)$ -Fourierkoeffizienten von $f \in C(\mathcal{SO}(3))$ die Eigenschaft*

$$\sum_{n=0}^{\infty} \sum_{k,l=-n}^n \sqrt{2n+1} |\hat{f}_n^{k,l}| < \infty \quad (3.13)$$

besitzen, dann konvergiert die $\mathcal{SO}(3)$ -Fourierreihe von f gleichmäßig gegen f auf $\mathcal{SO}(3)$, d.h.

$$\lim_{N \rightarrow \infty} \|f - S_N f\|_{C(\mathcal{SO}(3))} = 0.$$

Beweis. Wir schreiben die Unitäritätsbedingung der nicht-normalisierten Wigner-D Funktionen als

$$\sum_{k=-n}^n \tilde{D}_n^{k,l}(\mathbf{R}) \overline{\tilde{D}_n^{k,j}(\mathbf{R})} = \delta_{l,j},$$

siehe [35, Kapitel 4.1]. Für $j = l$ erhalten wir $\sum_{k=-n}^n |\tilde{D}_n^{k,l}(\mathbf{R})|^2 = 1$ und insbesondere gilt $|\tilde{D}_n^{k,l}(\mathbf{R})|^2 \leq 1$. Daraus folgt $|\tilde{D}_n^{k,l}(\mathbf{R})| \leq 1$ und mit der Eigenschaft (3.13) gilt

$$\left| \sum_{n=0}^{\infty} \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R}) \right| \leq \sum_{n=0}^{\infty} \sum_{k,l=-n}^n |\hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R})| \leq \sum_{n=0}^{\infty} \sum_{k,l=-n}^n \sqrt{2n+1} |\hat{f}_n^{k,l}| < \infty.$$

Aus dem Weierstraßschen Majorantenkriterium folgt die gleichmäßige Konvergenz der Fourierreihe gegen eine stetige Funktion

$$g(\mathbf{R}) := \sum_{n=0}^{\infty} \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R}).$$

Gleichmäßige Konvergenz impliziert die Konvergenz im 2-ten Mittel. Mit Satz 3.35 erhalten wir $f = g$. \square

Aus der gleichmäßigen Konvergenz folgt die punktweise Konvergenz der Fourierreihen von stetigen Funktionen, falls die Eigenschaft (3.13) erfüllt ist.

4 Die diskrete Fouriertransformation auf der Rotationsgruppe

Wir verwenden die diskrete Fouriertransformation auf $\mathcal{SO}(3)$, um N -bandbreitenbegrenzte Funktionen in endlich vielen Rotationen auszuwerten. Dabei berechnen wir die Funktionswerte von $\mathcal{SO}(3)$ -Fourierpartialsommen in nichtäquidistanten Rotationen. Das Ziel dieses Kapitels ist es, zwei effiziente Algorithmen für die Berechnung der diskreten Fouriertransformation auf $\mathcal{SO}(3)$ anzugeben und diese in Bezug auf Laufzeit und Stabilität zu vergleichen.

Weiterhin geben wir die adjungierten Algorithmen für die Berechnung der adjungierten diskreten Fouriertransformation auf $\mathcal{SO}(3)$ an. Wir verwenden diese zusammen mit einer exakten Quadraturformel auf $\mathcal{SO}(3)$ zur Berechnung des Integrals $\langle f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))}$, d.h. zur Berechnung der $\mathcal{SO}(3)$ -Fourierkoeffizienten $\hat{f}_n^{k,l}(f)$ einer N -bandbreitenbegrenzten Funktion.

Zuerst definieren wir den nichtäquidistanten Fourieroperator auf $\mathcal{SO}(3)$, siehe [37, Kapitel 3.2].

Definition 4.1. Seien $N \in \mathbb{N}$, eine endliche Menge von Rotationen

$$\mathcal{R}_M = \{\mathbf{R}_m \in \mathcal{SO}(3) : m = 1, \dots, M\}$$

und ein Vektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N} \in \mathbb{C}^{|\mathcal{J}_N|}$ gegeben. Dann bezeichnen wir den Vektor $\mathbf{f} = (f_m)_{m=1}^M \in \mathbb{C}^M$ als nichtäquidistante diskrete $\mathcal{SO}(3)$ -Fouriertransformation (NDSOFT) von $\hat{\mathbf{f}}$ bezüglich \mathcal{R}_M , wenn

$$f_m = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R}_m), \quad m = 1, \dots, M \quad (4.1)$$

gilt. Der zugehörige Operator $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} : \mathbb{C}^{|\mathcal{J}_N|} \rightarrow \mathbb{C}^M$ hat die Matrixform

$$\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} = \left(D_n^{k,l}(\mathbf{R}_m) \right)_{(m=1, \dots, M); ((n,k,l) \in \mathcal{J}_N)} \in \mathbb{C}^{M \times |\mathcal{J}_N|}$$

und es gilt

$$\mathbf{f} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} \hat{\mathbf{f}}.$$

4.1 Auswertung von N -bandbreitenbegrenzten Funktionen

Wir verwenden die NDSOFT zur Auswertung einer Funktion $f \in \mathbb{D}_N$ in endlich vielen Rotationen, siehe [37, Kapitel 3.2].

Korollar 4.2. *Seien $N \in \mathbb{N}$ und eine endliche Menge von Rotationen*

$$\mathcal{R}_M = \{\mathbf{R}_m \in \mathcal{SO}(3) : m = 1, \dots, M\}$$

gegeben. Dann wird eine N -bandbreitenbegrenzte Funktion $f \in \mathbb{D}_N$ mit gegebenem $\mathcal{SO}(3)$ -Fourierkoeffizientenvektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ mit der NDSOFT $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}$ durch

$$(f(\mathbf{R}_m))_{m=1, \dots, M} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} \hat{\mathbf{f}}$$

in den Knoten $\mathbf{R}_1, \dots, \mathbf{R}_M$ ausgewertet.

Beweis. Einsetzen der Rotationen $\mathbf{R}_1, \dots, \mathbf{R}_M$ in die $\mathcal{SO}(3)$ -Fourierreihe von f liefert die Behauptung. \square

Bemerkung 4.3. *Zur Auswertung einer Funktion aus $L_2(\mathcal{SO}(3))$ verwenden wir deren N -te $\mathcal{SO}(3)$ -Fourierpartialsumme als Bestapproximation, siehe Satz 3.35 und Lemma 3.38. Folglich ist es hinreichend N -bandbreitenbegrenzte Funktionen zu untersuchen.*

Wegen Korollar 4.2 können wir die NDSOFT eines Vektors bestimmen, indem wir die entsprechende N -bandbreitenbegrenzte Funktion auswerten.

Wir suchen schnelle Algorithmen für die NDSOFT. Dazu betrachten wir die $\mathcal{SO}(3)$ -Fourierpartialsumme

$$f_m = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R}(\alpha_m, \beta_m, \gamma_m)) \quad (4.2)$$

in den mit Euler-Winkeln parametrisierten Rotationen $\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \in \mathcal{R}_M \subset \mathcal{SO}(3)$. Wir folgen den Überlegungen von Vollrath [37, Kapitel 4].

Das Ziel ist es, die $\mathcal{SO}(3)$ -Fourierpartialsumme (4.2) effizient in eine echte Fourierpartialsumme mit komplexen Exponentialen als Orthonormalbasis umzuwandeln, welche danach mit einer NFFT ausgewertet wird. Falls die Rotationen aus \mathcal{R}_M auf einem äquidistanten Gitter liegen, dann reicht eine äquidistante FFT.

Wir setzen die Definition der normierten Wigner-D Funktionen in Gleichung (4.2) ein und erhalten

$$f_m = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} \sqrt{2n+1} d_n^{k,l}(\cos \beta_m) e^{-i k \alpha_m} e^{-i l \gamma_m}.$$

Vertauschen der Summationsreihenfolge liefert

$$f_m = \sum_{k,l=-N}^N e^{-i k \alpha_m} e^{-i l \gamma_m} \sum_{n=\max\{|k|,|l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{k,l}(\cos \beta_m). \quad (4.3)$$

Damit haben wir komplexe Exponentiale bezüglich des ersten und dritten Euler-Winkels. Das Ziel ist es, für fixierte Indizes $k, l = -N, \dots, N$ die Linearkombination der Wigner-d Funktionen in eine Fourierpartialsumme zu transformieren, d.h. wir suchen Fourierkoeffizienten $(h_n^{k,l})_{n=-N, \dots, N}$ mit

$$\sum_{n=\max\{|k|,|l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{k,l}(\cos \beta) = \sum_{n=-N}^N h_n^{k,l} e^{-i n \beta} \quad (4.4)$$

für $\beta \in [0, \pi]$. Im Folgenden bezeichnen wir eine Koeffiziententransformation, welche $\mathcal{SO}(3)$ -Fourierkoeffizienten in echte Fourierkoeffizienten transformiert, als Wignertransformation. Dazu vergleichen wir zwei Algorithmen.

4.2 Die Wignertransformation über Chebychev-Polynome

Wir verwenden den Algorithmus von Potts [24] um eine Wignertransformation für Gleichung (4.4) anzugeben. Dazu transformieren wir in einem ersten Schritt die Linearkombination der Wigner-d Funktionen in eine Reihe von Chebyshev-Polynomen, siehe [24, Kapitel 3.1, 3.2], um sie in einem zweiten Schritt in eine echte Fourierreihe, siehe [24, Kapitel 3.3] umzuwandeln. Wir fassen in diesem Kapitel das Paper [24] zusammen.

4.2.1 Die diskrete Polynomtransformation

Wir führen eine Polynomtransformation von der Linearkombination der Wigner-d Funktionen auf Chebyshev-Polynome ein, siehe [24, Kapitel 3.1]

Dazu seien $N \in \mathbb{N}$ und die Indizes $k, l = -N, \dots, N$ fixiert. Wegen Bemerkung 3.9 ist die Wigner-d Funktion $d_n^{k,l}(\cos \beta)$ ein Polynom vom Grad n bezüglich dem Argument $\cos \beta$, falls $k + l$ gerade ist. Wenn $k + l$ ungerade ist, dann ist $(\sin \beta)^{-1} d_n^{k,l}(\cos \beta)$ ein Polynom vom Grad $n - 1$ bezüglich $\cos \beta$. Insbesondere sind die Linearkombinationen

$$\sum_{n=\max\{|k|,|l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{k,l}(\cos \beta)$$

für $k + l$ gerade und

$$(\sin \beta)^{-1} \sum_{n=\max\{|k|,|l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{k,l}(\cos \beta)$$

für $k + l$ ungerade Polynome vom Grad N bzw. $N - 1$. Diese Polynome sind durch Chebyshevreihen darstellbar, da die Chebyshev-Polynome ein System von Orthogonalpolynomen bilden, siehe Kapitel 3.1.1.

Damit geben wir eine Polynomtransformation auf Chebyshev-Polynome an, d.h. es existieren Chebyshevkoeffizienten $(t_n^{k,l})_{n=0,\dots,N}$, so dass

$$\sum_{n=\max\{|k|,|l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{k,l}(\cos \beta) = \begin{cases} \sum_{n=0}^N t_n^{k,l} T_n(\cos \beta) & \text{falls } k+l \text{ gerade,} \\ \sin \beta \sum_{n=0}^{N-1} t_n^{k,l} T_n(\cos \beta) & \text{falls } k+l \text{ ungerade} \end{cases} \quad (4.5)$$

gilt. Falls $k + l$ ungerade ist oder $\hat{f}_N^{k,l} = 0$ gilt, dann ist der Polynomgrad kleiner als N und wir erhalten $t_N^{k,l} = 0$ bei der Polynominterpolation durch ein Polynom vom Grad N . Deshalb führen wir im Folgenden, unabhängig davon ob $k + l$ gerade ist, eine Polynomtransformation auf Chebyshev-Polynome mit Grad $\leq N$ durch.

Ein Polynom vom Grad höchstens N kann durch seine Funktionswerte in den Chebyshev-knoten $\left\{ \cos \frac{(2m+1)\pi}{2(N+1)} \mid m = 0, \dots, N \right\}$ eindeutig dargestellt werden. Sei

$$\Omega_N = \left\{ \frac{(2m+1)\pi}{2(N+1)} \mid m = 0, \dots, N \right\}.$$

Dann berechnen wir die Chebyshevkoeffizienten $(t_n^{k,l})_{n=0,\dots,N}$ aus den $\mathcal{SO}(3)$ -Fourierkoeffizienten $(\hat{f}_n^{k,l})_{n=\max\{|k|,|l|\},\dots,N}$, indem wir die Knoten $\beta_m \in \Omega_N$ in Gleichung (4.5) einsetzen und das entstehende lineare Gleichungssystem lösen. Wir benötigen das folgende Lemma über die Operatoren der Chebyshevinterpolation, siehe [23, Kapitel 3.5].

Lemma 4.4. *Sei $N \in \mathbb{N}$. Die trigonometrischen Matrizen*

$$\mathbf{T} = \left(\cos \frac{(2m+1)n\pi}{2(N+1)} \right)_{m,n=0,\dots,N} \quad (4.6)$$

und

$$\mathbf{T}^{-1} = \left(\frac{2 - \delta_{0,m}}{N+1} \cos \frac{(2n+1)m\pi}{2(N+1)} \right)_{m,n=0,\dots,N} \quad (4.7)$$

sind zueinander invers.

Beweis. Für $\eta \in \mathbb{Z}$ gilt

$$-2i \sin \left(\frac{\pi \eta}{2N+2} \right) \sum_{n=0}^N e^{\frac{i\pi(2n+1)\eta}{2N+2}} = \left(e^{\frac{-i\pi\eta}{2N+2}} - e^{\frac{i\pi\eta}{2N+2}} \right) \sum_{n=0}^N e^{\frac{i\pi(2n+1)\eta}{2N+2}} = 1 - e^{i\pi\eta}.$$

Sei $0 < |\eta| < 2N + 2$. Dann ist $\sin \frac{\pi \eta}{2N+2} \neq 0$ und es folgt

$$\sum_{n=0}^N e^{\frac{i \pi (2n+1) \eta}{2N+2}} = \frac{1 - (-1)^\eta}{-2i \sin \frac{\pi \eta}{2N+2}} = \begin{cases} 0 & \text{falls } \eta \text{ gerade,} \\ \frac{i}{\sin \frac{\pi \eta}{2N+2}} & \text{falls } \eta \text{ ungerade.} \end{cases}$$

Damit erhalten wir die Gleichung

$$\sum_{n=0}^N \cos \frac{\pi (2n+1) \eta}{2N+2} = \Re \left(\sum_{n=0}^N e^{\frac{i \pi (2n+1) \eta}{2N+2}} \right) = 0, \quad (4.8)$$

für $0 < |\eta| < 2N + 2$.

Für die Komponenten der Matrix $\mathbf{T}^{-1} \mathbf{T}$ gilt

$$(\mathbf{T}^{-1} \mathbf{T})_{m,n} = \sum_{t=0}^N \mathbf{T}_{m,t}^{-1} \mathbf{T}_{t,n} = \frac{2 - \delta_{0,m}}{N+1} \sum_{t=0}^N \cos \frac{(2t+1)m\pi}{2(N+1)} \cos \frac{(2t+1)n\pi}{2(N+1)}.$$

Mit dem Additionstheorem $2 \cos x \cos y = \cos(x+y) + \cos(x-y)$ erhalten wir

$$(\mathbf{T}^{-1} \mathbf{T})_{m,n} = \frac{2 - \delta_{0,m}}{2(N+1)} \left(\sum_{t=0}^N \cos \frac{(2t+1)(m+n)\pi}{2(N+1)} + \sum_{t=0}^N \cos \frac{(2t+1)(m-n)\pi}{2(N+1)} \right).$$

Da $-N \leq m-n \leq m+n \leq 2N$ gilt, folgt mit Gleichung (4.8)

$$(\mathbf{T}^{-1} \mathbf{T})_{m,n} = \frac{2 - \delta_{0,m}}{2(N+1)} \left((N+1) \delta_{m+n,0} + (N+1) \delta_{m-n,0} \right) = \frac{(2 - \delta_{0,m})(\delta_{m,-n} + \delta_{m,n})}{2}.$$

Eine Fallunterscheidung liefert $(\mathbf{T}^{-1} \mathbf{T})_{m,n} = \delta_{m,n}$. □

Wir geben die Transformation von $\mathcal{SO}(3)$ -Fourierkoeffizienten auf Chebyshevkoeffizienten durch lineare Operatoren an.

Lemma 4.5. *Seien $N \in \mathbb{N}$, fixierte Indizes $k, l = -N, \dots, N$ und die reellwertigen Matrizen*

$$\begin{aligned} \tilde{\mathbf{D}}^{k,l} &= \left(d_n^{k,l} \left(\cos \frac{(2m+1)\pi}{2(N+1)} \right) \right)_{(m=0,\dots,N); (n=\max\{|k|,|l|\},\dots,N)}, \\ \mathbf{C}^{k,l} &= \text{diag} \left(\sqrt{2n+1} \right)_{n=\max\{|k|,|l|\},\dots,N}, \\ \mathbf{S} &= \text{diag} \left(\sin \frac{(2m+1)\pi}{2(N+1)} \right)_{m=0,\dots,N} \end{aligned} \quad (4.9)$$

sowie die Chebyshevmatrix \mathbf{T}^{-1} aus Gleichung (4.7) gegeben. Weiterhin wird der Basiswechsel (4.5) zwischen dem $\mathcal{SO}(3)$ -Fourierkoeffizientenvektor $\hat{\mathbf{f}}^{k,l} = (\hat{f}_{\max\{|k|,|l|\}}^{k,l}, \dots, \hat{f}_N^{k,l})^T$ und dem Chebyshevkoeffizientenvektor $\mathbf{t}^{k,l} = (t_0^{k,l}, \dots, t_N^{k,l})^T$ durch die Polynomtransformation

$\mathbf{W}^{k,l} \in \mathbb{R}^{(N+1) \times (N+1 - \max\{|k|, |l|\})}$ mit

$$\mathbf{t}^{k,l} = \mathbf{W}^{k,l} \hat{\mathbf{f}}^{k,l}$$

beschrieben. Dann gilt

$$\mathbf{W}^{k,l} = \begin{cases} \mathbf{T}^{-1} \tilde{\mathbf{D}}^{k,l} \mathbf{C}^{k,l} & \text{falls } k+l \text{ gerade,} \\ \mathbf{T}^{-1} \mathbf{S}^{-1} \tilde{\mathbf{D}}^{k,l} \mathbf{C}^{k,l} & \text{falls } k+l \text{ ungerade.} \end{cases}$$

Beweis. Wir setzen die Knoten des äquidistanten Gitters Ω_N in Gleichung (4.5) ein und schreiben das entstehende Gleichungssystem als Matrix-Vektor-Multiplikation

$$\tilde{\mathbf{D}}^{k,l} \mathbf{C}^{k,l} \hat{\mathbf{f}}^{k,l} = \begin{cases} \mathbf{T} \mathbf{t}^{k,l} & \text{falls } k+l \text{ gerade,} \\ \mathbf{S} \mathbf{T} \mathbf{t}^{k,l} & \text{falls } k+l \text{ ungerade.} \end{cases}$$

mit der trigonometrischen Matrix \mathbf{T} aus Gleichung (4.6). Wegen Lemma 4.4 ist \mathbf{T}^{-1} die inverse Matrix von \mathbf{T} . Da die Hauptdiagonaleinträge der Diagonalmatrix \mathbf{S} ungleich 0 sind, ist \mathbf{S} invertierbar und es folgt die Behauptung. \square

Für fixierte Indizes $k, l = -N, \dots, N$ benutzen wir die 3-Term-Rekursion (3.4) der Wigner-d Funktionen und berechnen die Matrix-Vektor-Multiplikation mit $\tilde{\mathbf{D}}^{k,l}$ durch den Clenshaw Algorithmus [29, Kapitel 5.5] in $\mathcal{O}(N^2)$ Schritten. Für die Multiplikation mit \mathbf{T}^{-1} verwenden wir die diskrete Kosinustransformation (DCT), siehe [1], mit einem Aufwand von $\mathcal{O}(N \log N)$. Die Multiplikationen mit den Diagonalmatrizen \mathbf{S}^{-1} und $\mathbf{C}^{k,l}$ sind sehr schnell in $\mathcal{O}(N)$ Schritten machbar. Damit erhalten wir eine Komplexität von $\mathcal{O}(N^2)$ für die Polynomtransformation $\mathbf{t}^{k,l} = \mathbf{W}^{k,l} \hat{\mathbf{f}}^{k,l}$. Da es $(2N+1)^2$ verschiedene Chebyshevkoeffizientenvektoren $\mathbf{t}^{k,l}$ gibt, erhalten wir eine approximierte Gesamtkomplexität von $\mathcal{O}(N^4)$ für die Transformation von $\mathcal{SO}(3)$ -Fourierkoeffizienten auf Chebyshevkoeffizienten.

4.2.2 Die schnelle Polynomtransformation

Um den numerischen Aufwand der Polynomtransformation $\mathbf{t}^{k,l} = \mathbf{W}^{k,l} \hat{\mathbf{f}}^{k,l}$ zu verringern, suchen wir einen schnelleren Algorithmus für die Transformation mit dem Operator $\tilde{\mathbf{D}}^{k,l}$. Dazu verwenden wir die Idee von Potts, siehe [24, Kapitel 3.2].

Bisher waren die Wigner-d Funktionen $d_n^{k,l}$ nur für Indizes $|k|, |l| \leq n$ definiert. Wir verändern die Rekursionsformel (3.4), so dass die Wigner-d Funktionen für beliebige ganzzahlige Indizes $k, l = -N, \dots, N$ rekursiv berechenbar sind. Zusätzlich definieren wir verallgemeinerte Wigner-d Funktionen, die eine verallgemeinerte Rekursionsformel erfüllen, in der wir den Grad der Wigner-d Funktion anstelle eines einzelnen Schrittes um $c \in \mathbb{N}$ Schritte erhöhen können.

Bemerkung 4.6. Sei $N \in \mathbb{N}$. Für $k, l = -N, \dots, N$ und $n = 0, \dots, N-1$ haben wir die erweiterte 3-Term-Rekursion

$$d_{n+1}^{k,l}(\cos \beta) = (\varphi_n^{k,l} \cos \beta + \psi_n^{k,l}) d_n^{k,l}(\cos \beta) + \varrho_n^{k,l} d_{n-1}^{k,l}(\cos \beta).$$

Mit $a = \min\{|k|, |l|\}$ und $b = \max\{|k|, |l|\}$ sowie den Koeffizienten $t_n^{k,l}, u_n^{k,l}, v_n^{k,l}, w_n^{k,l}$ aus Lemma 3.11 erhalten wir die Koeffizienten

$$\varphi_0^{k,l} = \begin{cases} 1 & \text{falls } k = l, \\ -1 & \text{falls } k + l \text{ gerade, } k \neq l, \\ 0 & \text{sonst,} \end{cases} \quad \varphi_n^{k,l} = \begin{cases} (-1)^{k+l+1} & \text{falls } 0 < n < b - a, \\ \frac{kl}{|kl|} & \text{falls } b - a < n < b, \\ \frac{u_n^{k,l}}{t_n^{k,l}} & \text{falls } b \leq n, \end{cases}$$

$$\psi_n^{k,l} = \begin{cases} 1 & \text{falls } 0 \leq n < b, \\ 0 & \text{falls } k = l = 0, \\ -\frac{v_n^{k,l}}{t_n^{k,l}} & \text{sonst,} \end{cases} \quad \varrho_n^{k,l} = \begin{cases} 0 & \text{falls } n \leq b, \\ -\frac{w_n^{k,l}}{t_n^{k,l}} & \text{sonst.} \end{cases}$$

Für den Start der Rekursion setzen wir $d_{-1}^{k,l} = 0$ für alle $k, l = -N, \dots, N$ und

$$d_0^{k,l}(\cos \beta) = \begin{cases} \frac{\sqrt{(2a)!}}{2^a a!} & \text{falls } k + l \text{ gerade,} \\ \frac{\sqrt{(2a)!}}{2^a a!} \sin \beta & \text{falls } k + l \text{ ungerade.} \end{cases}$$

Für die zweite Modifikation der Rekursionsformel verwenden wir die folgende Definition.

Definition 4.7. Sei ein Verschiebungsparameter $c \in \mathbb{N}$ gegeben. Dann sind die verallgemeinerten Wigner-d Funktionen

$$\begin{aligned} d_{-1}^{k,l}(\cos \beta, c) &= 0, \\ d_0^{k,l}(\cos \beta, c) &= 1, \\ d_{n+1}^{k,l}(\cos \beta, c) &= (\varphi_{n+c}^{k,l} \cos \beta + \psi_{n+c}^{k,l}) d_n^{k,l}(\cos \beta, c) + \varrho_{n+c}^{k,l} d_{n-1}^{k,l}(\cos \beta, c) \end{aligned}$$

gegeben.

Wir kombinieren Bemerkung 4.6 mit Definition 4.7 und erhalten das folgende Resultat.

Lemma 4.8. Seien $N \in \mathbb{N}_0$, $k, l = -N, \dots, N$ und $c, n = 0, \dots, N$. Dann gilt

$$d_{n+c}^{k,l}(\cos \beta) = d_c^{k,l}(\cos \beta, n) d_n^{k,l}(\cos \beta) + \varrho_n^{k,l} d_{c-1}^{k,l}(\cos \beta, n+1) d_{n-1}^{k,l}(\cos \beta).$$

Beweis. Induktion über c liefert die Behauptung. □

Wir verwenden die schnelle Polynomtransformation (FPT) von Potts, siehe [25], auf den verallgemeinerten Wigner-d Funktionen. Ein Algorithmus auf der Sphäre wird in [18] und [3] beschrieben.

Dadurch verbessern wir die approximative Komplexität des linearen Operators $\tilde{\mathbf{D}}^{k,l}$ von $\mathcal{O}(N^2)$ zu $\mathcal{O}(N \log^2 N)$ und erhalten eine Gesamtkomplexität von $\mathcal{O}(N^3 \log^2 N)$ für die Transformation von $\mathcal{SO}(3)$ -Fourierkoeffizienten auf Chebyshevkoeffizienten.

Kleine Ungenauigkeiten in der DCT implizieren numerische Instabilitäten, wenn sie mit großen Funktionswerten von den verallgemeinerten Wigner-d Funktionen multipliziert werden, vergleiche [28]. Deshalb beschreibt Potts in [26] einen Stabilisierungsschritt, der im Algorithmus eingefügt wird. Die Idee besteht darin, kritische Polynome aus der Summe zu entfernen und separat zu verarbeiten. Im schlimmsten Fall werden alle Polynome separat verarbeitet. Dadurch wird die FPT zurückgeführt auf die diskrete Polynomtransformation aus Kapitel 4.2.1.

4.2.3 Transformation auf eine Fourierreihe

Ein weiterer Basiswechsel ist notwendig, um die Chebyshevreihe aus Gleichung (4.5) in eine Fourierreihe zu transformieren, siehe [24, Kapitel 3.3], d.h. wir suchen Koeffizienten $(h_n^{k,l})_{n=-N,\dots,N}$, so dass

$$\sum_{n=-N}^N h_n^{k,l} e^{-i n \beta} = \begin{cases} \sum_{n=0}^N t_n^{k,l} T_n(\cos \beta) & \text{falls } k+l \text{ gerade,} \\ \sin \beta \sum_{n=0}^{N-1} t_n^{k,l} T_n(\cos \beta) & \text{falls } k+l \text{ ungerade} \end{cases} \quad (4.10)$$

gilt.

Lemma 4.9. Seien $N \in \mathbb{N}$, fixierte Indizes $k, l = -N, \dots, N$ und zwei Matrizen $\mathbf{H} \in \mathbb{R}^{(2N+1) \times (N+1)}$ und $\mathbf{O} \in \mathbb{C}^{(2N+1) \times (2N+1)}$ mit

$$\mathbf{H} = \begin{pmatrix} & & & & \frac{1}{2} \\ & & \ddots & & \\ & \frac{1}{2} & & & \\ 1 & & \frac{1}{2} & & \\ & & & \ddots & \\ & & & & \frac{1}{2} \end{pmatrix} \quad \text{und} \quad \mathbf{O} = \frac{i}{2} \begin{pmatrix} 0 & 1 & & & \\ -1 & \ddots & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & -1 & 0 \end{pmatrix}.$$

gegeben. Weiterhin wird der Basiswechsel (4.10) zwischen dem Chebyshevkoeffizientenvektor $\mathbf{t}^{k,l} = (t_0^{k,l}, \dots, t_N^{k,l})^T$ und dem Fourierkoeffizientenvektor $\mathbf{h}^{k,l} = (h_{-N}^{k,l}, \dots, h_N^{k,l})^T$ durch die Matrix $\mathbf{A}^{k,l} \in \mathbb{C}^{(2N+1) \times (N+1)}$ mit

$$\mathbf{h}^{k,l} = \mathbf{A}^{k,l} \mathbf{t}^{k,l}$$

beschrieben. Dann gilt

$$\mathbf{A}^{k,l} = \begin{cases} \mathbf{H} & \text{falls } k+l \text{ gerade,} \\ \mathbf{O} \mathbf{H} & \text{falls } k+l \text{ ungerade.} \end{cases}$$

Beweis. Wegen $T_n(\cos \beta) = \cos(n\beta) = \frac{1}{2}(e^{in\beta} + e^{-in\beta})$ gilt

$$\sum_{n=0}^N t_n^{k,l} T_n(\cos \beta) = \sum_{n=0}^N \frac{1}{2} t_n^{k,l} (e^{in\beta} + e^{-in\beta}) = \frac{1}{2} t_0^{k,l} + \sum_{n=-N}^N \frac{1}{2} t_{|n|}^{k,l} e^{-in\beta}$$

Damit erhalten wir

$$h_n^{k,l} = \begin{cases} t_0^{k,l} & \text{falls } n=0, \\ \frac{1}{2} t_{|n|}^{k,l} & \text{sonst} \end{cases}$$

und es folgt die Behauptung für den Fall $k+l$ gerade.

Wenn $k+l$ ungerade ist, dann gilt mit $\sin \beta = \frac{i}{2}(e^{-i\beta} - e^{i\beta})$ analog

$$\sin \beta \sum_{n=0}^{N-1} t_n^{k,l} T_n(\cos \beta) = \frac{i}{2} (e^{-i\beta} - e^{i\beta}) \sum_{n=-(N-1)}^{N-1} h_n^{k,l} e^{-in\beta}.$$

Wir setzen $h_N^{k,l} = h_{-N}^{k,l} = 0$ und erhalten

$$\sin \beta \sum_{n=0}^{N-1} t_n^{k,l} T_n(\cos \beta) = \frac{i}{2} \sum_{n=-N}^N (h_{n-1}^{k,l} - h_{n+1}^{k,l}) e^{-in\beta}.$$

Damit folgt die Behauptung für $k+l$ ungerade. □

Die Matrix $\mathbf{A}^{k,l}$ ist dünn besetzt. Dadurch ist die Matrix-Vektor-Multiplikation in $\mathcal{O}(N)$ Schritten machbar. Da es $(2N+1)^2$ Variationen der Indizes k und l gibt, erhalten wir eine Gesamtkomplexität von $\mathcal{O}(N^3)$ für die Transformation von Chebyshev- auf Fourierkoeffizienten.

4.2.4 Die nichtäquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation

Durch Hintereinanderausführung der Operatoren aus den Lemmata 4.5 und 4.9 erhalten wir eine Wignertransformation. Wir verwenden diese im folgenden zentralen Resultat über die NDSOFT, siehe [24, Kapitel 3.4].

Satz 4.10. *Seien $N \in \mathbb{N}$ und eine durch Euler-Winkel parametrisierte Rotationsmenge*

$$\mathcal{R}_M = \{\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \mid m = 1, \dots, M\} \subset \mathcal{SO}(3)$$

mit Kardinalität $M \in \mathbb{N}$ gegeben. Weiterhin haben wir die Blockdiagonalmatrix

$$\mathbf{W} = \text{diag} \left(\mathbf{W}^{k,l} \right)_{k,l=-N,\dots,N} \in \mathbb{R}^{(2N+1)^2(N+1) \times |\mathcal{J}_N|}$$

bestehend aus den Matrizen $\mathbf{W}^{k,l}$ aus Lemma 4.5. Wir haben die Blockdiagonalmatrix

$$\mathbf{A} = \text{diag} \left(\mathbf{A}^{k,l} \right)_{k,l=-N,\dots,N} \in \mathbb{C}^{(2N+1)^3 \times (2N+1)^2(N+1)},$$

welche aus den Matrizen $\mathbf{A}^{k,l}$ aus Lemma 4.9 zusammengesetzt ist und die trivariate nichtäquidistante Fouriermatrix

$$\mathbf{F}_{\mathcal{R}_M, N} = \left(e^{-i(k,n,l) \cdot (\alpha_m, \beta_m, \gamma_m)^T} \right)_{(m=1,\dots,M); (k,n,l=-N,\dots,N)} \in \mathbb{C}^{M \times (2N+1)^3}. \quad (4.11)$$

Dann gilt für die NDSOFT $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}$ aus Definition 4.1 der Zusammenhang

$$\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} = \mathbf{F}_{\mathcal{R}_M, N} \mathbf{A} \mathbf{W}.$$

Beweis. Durch Einsetzen der mit Euler-Winkeln parametrisierten Rotationen $\mathbf{R}_m \in \mathcal{R}_M$ in die NDSOFT erhalten wir

$$f_m = \sum_{k,l=-N}^N e^{-i k \alpha_m} e^{-i l \gamma_m} \sum_{n=\max\{|k|, |l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{k,l}(\cos \beta_m) \quad (4.3)$$

für $m = 1, \dots, M$. Mit dem Basiswechsel aus Lemma 4.5 tauschen wir die Linearkombination der Wigner-d Funktionen gegen eine Linearkombination von Chebyshev-Polynomen und es folgt

$$f_m = \sum_{k,l=-N}^N e^{-i k \alpha_m} e^{-i l \gamma_m} \sum_{n=0}^N t_n^{k,l} T_n(\cos \beta_m) (\sin \beta_m)^{k+l \bmod 2}.$$

Lemma 4.9 liefert einen Wechsel von Chebyshevkoeffizienten auf Fourierkoeffizienten, d.h.

$$f_m = \sum_{k,l=-N}^N e^{-i k \alpha_m} e^{-i l \gamma_m} \sum_{n=-N}^N h_n^{k,l} e^{-i n \beta_m} = \sum_{k,n,l=-N}^N h_n^{k,l} e^{-i(k,n,l) \cdot (\alpha_m, \beta_m, \gamma_m)^T}.$$

Damit erhalten wir eine trivariate diskrete Fouriertransformation und schreiben die Fouriermatrix wie in Gleichung (4.11). \square

Für das Matrix-Vektor-Produkt mit der trivariaten Fouriermatrix $\mathbf{F}_{\mathcal{R}_M, N}$ verwenden wir die trivariate NFFT mit einer approximativen Komplexität von $\mathcal{O}(M + N^3 \log N)$. Damit haben wir schnelle Algorithmen für die linearen Operatoren der NDSOFT aus Satz 4.10 und erhalten die nichtäquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation (NFSOFT) in Algorithmus 4.1.

Bei der Transformation von $\mathcal{SO}(3)$ -Fourierkoeffizienten über Chebyshevkoeffizienten auf echte Fourierkoeffizienten wird die Mächtigkeit der Koeffizientenmengen größer. Das wird in Abbildung 4.1 verdeutlicht, siehe [37, Abbildung 4.1].

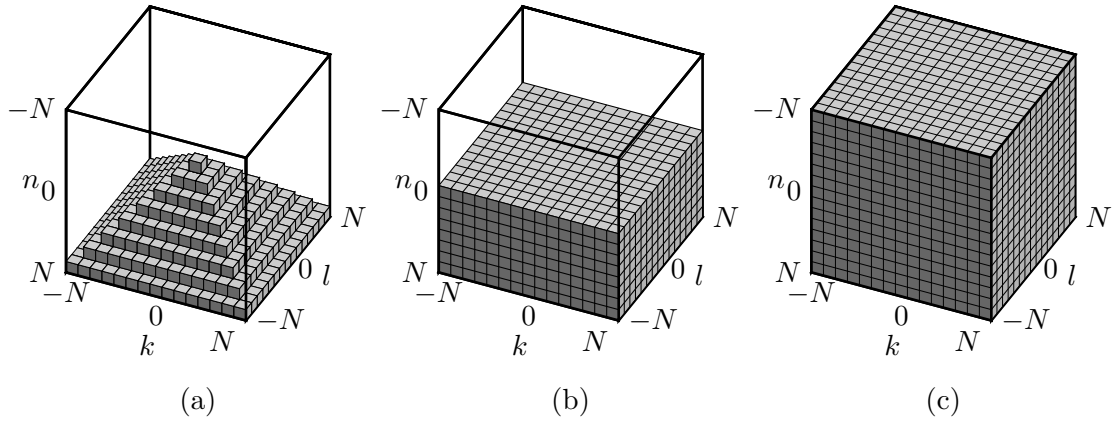


Abbildung 4.1: Darstellung der Koeffizientenmengen, die bei der Wignertransformation aus Satz 4.10 auftreten. Die Koordinaten (n, k, l) geben den Grad n und die Indizes k, l des zugehörigen Koeffizienten an.

(a) Darstellung der $\mathcal{SO}(3)$ -Fourierkoeffizienten $\hat{f}_n^{k,l}$ mit $(n, k, l) \in \mathcal{J}_N$.

(b) Chebyshevkoeffizienten $t_n^{k,l}$ aus Lemma 4.5, die beim Übergang von Wigner-d Funktionen auf Chebyshev-Polynome berechnet werden.

(c) Fourierkoeffizienten $h_n^{k,l}$ aus Lemma 4.9, mit denen wir schließlich eine echte Fourierreihe erhalten.

Algorithmus 4.1 : NFSOFT

Input : $N \in \mathbb{N}$ Bandbreite
 $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ Koeffizientenvektor
 \mathcal{R}_M Menge von Rotationen

1. Berechne die Chebyshevkoeffizienten $\mathbf{t} = \mathbf{W} \hat{\mathbf{f}}$ wie in Lemma 4.5 mit der FPT aus [25] in $\mathcal{O}(N^3 \log^2 N)$ Schritten.
2. Berechne die Fourierkoeffizienten $\mathbf{h} = \mathbf{A} \mathbf{t}$ aus den Chebyshevkoeffizienten wie in Lemma 4.9 in $\mathcal{O}(N^3)$ Schritten.
3. Berechne $\mathbf{f} = \mathbf{F}_{\mathcal{R}_M, N} \mathbf{h}$ mit der Fouriermatrix $\mathbf{F}_{\mathcal{R}_M, N}$ aus Gleichung (4.11) in den Rotationen aus \mathcal{R}_M mit trivariater NFFT in $\mathcal{O}(M + N^3 \log N)$ Schritten.

Output : NDSOFT $\mathbf{f} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} \hat{\mathbf{f}}$

Komplexität : $\mathcal{O}(M + N^3 \log^2 N)$

4.3 Die darstellungsbasierte Wignertransformation

Wir wollen einen alternativen Algorithmus für die NDSOFT angeben. Die Wignertransformation aus Kapitel 4.2 verwendet unter anderem den Wigner-d Operator $\tilde{\mathbf{D}}^{k,l}$ aus Gleichung (4.9). Dieser berechnet in jedem Knoten $\beta_m \in \Omega_N$ eine Linearkombination der in β_m ausgewerteten Wigner-d Funktionen bis zum Grad $N \in \mathbb{N}$. In der NFSOFT verwenden wir dafür die FPT von Potts [25]. Wegen Instabilität benötigen wir zusätzlich den Stabilisierungsschritt [26].

Das Ziel dieses Kapitels ist es den Wigner-d Operator $\tilde{\mathbf{D}}^{k,l}$, d.h. insbesondere die FPT [25], zu vermeiden. Stattdessen nutzen wir aus, dass die Wigner-D Funktionen die Matricelemente von Darstellungen sind. Darstellungen sind Homomorphismen und folglich strukturerhaltend. Damit geben wir eine Wignertransformation von $\mathcal{SO}(3)$ -Fourierkoeffizienten auf echte Fourierkoeffizienten an, in der das Argument der zur Berechnung benötigten Wigner-d Funktionen konstant ist, d.h. wir werten die Wigner-d Funktionen im gesamten Algorithmus nur in einem Punkt aus. Die Idee für dieses Verfahren stammt von Risbo, siehe [31]. Wir werden Symmetrieeigenschaften in den Fourierkoeffizienten ausnutzen, um den Algorithmus zu verbessern.

4.3.1 Die darstellungsbasierte nichtäquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation

Wir geben die Wignertransformation für den Basiswechsel von Wigner-d Funktionen auf komplexe Exponentiale an.

Definition 4.11. Seien $N \in \mathbb{N}$ und ein Vektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N} \in \mathbb{C}^{|\mathcal{J}_N|}$ gegeben. Wir definieren den linearen Operator $\mathbf{G} \in \mathbb{R}^{(2N+1)^3 \times |\mathcal{J}_N|}$ durch das Matrix-Vektor-Produkt

$$(\mathbf{G} \hat{\mathbf{f}})_{k,j,l} := \sum_{n=\max\{|k|,|j|,|l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{j,k}(0) d_n^{j,l}(0) \quad (4.12)$$

für alle $k, j, l = -N, \dots, N$ und bezeichnen \mathbf{G} als darstellungsbasierte Wignertransformation (DBWT).

Wir zeigen, dass die Koeffiziententransformation \mathbf{G} wirklich eine Wignertransformation ist. Dabei geben wir eine Alternative für die Berechnung der NDSOFT an und erhalten das folgende Hauptresultat.

Satz 4.12. [Darstellungsbasierte NDSOFT] Seien $N \in \mathbb{N}$ und eine durch Euler-Winkel parametrisierte Knotenmenge

$$\mathcal{R}_M = \{\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \mid m = 1, \dots, M\} \subset \mathcal{SO}(3)$$

mit Kardinalität $M \in \mathbb{N}$ sowie die verschobene Knotenmenge

$$\tilde{\mathcal{R}}_M = \{\mathbf{R}(\alpha_m - \frac{\pi}{2}, \beta_m, \gamma_m + \frac{\pi}{2}) \mid m = 1, \dots, M\} \subset \mathcal{SO}(3)$$

gegeben. Weiterhin haben wir die DBWT \mathbf{G} aus Definition 4.11 und die trivariate nichtäquidistante Fouriermatrix

$$\mathbf{F}_{\tilde{\mathcal{R}}_M, N} = \left(e^{-i(k,j,l) \cdot (\alpha_m - \frac{\pi}{2}, \beta_m, \gamma_m + \frac{\pi}{2})^T} \right)_{(m=1, \dots, M); (k,j,l=-N, \dots, N)} \in \mathbb{C}^{M \times (2N+1)^3}. \quad (4.13)$$

Dann gilt für die NDSOFT $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}$ aus Definition 4.1 der Zusammenhang

$$\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} = \mathbf{F}_{\tilde{\mathcal{R}}_M, N} \mathbf{G}.$$

Beweis. Als erstes nutzen wir aus, dass die Wigner-D Funktionen gerade die Matrixelemente der Darstellungen $\tilde{D}^n: \mathcal{SO}(3) \rightarrow \mathbb{C}^{(2n+1) \times (2n+1)}$ sind. Da Darstellungen Gruppenhomomorphismen sind, gilt

$$\tilde{D}^n(\mathbf{R}\mathbf{Q}) = \tilde{D}^n(\mathbf{R}) \tilde{D}^n(\mathbf{Q})$$

für beliebige Rotationen $\mathbf{R}, \mathbf{Q} \in \mathcal{SO}(3)$. Daraus folgt

$$D_n^{k,l}(\mathbf{R}\mathbf{Q}) = \frac{1}{\sqrt{2n+1}} \sum_{j=-n}^n D_n^{k,j}(\mathbf{R}) D_n^{j,l}(\mathbf{Q})$$

für die normierten Wigner-D Funktionen als normierte Matrixelemente der Darstellungen. Wir verwenden Lemma 2.16 und erhalten

$$\begin{aligned} D_n^{k,l}(\mathbf{R}(\alpha, \beta, \gamma)) &= \frac{1}{\sqrt{2n+1}} \sum_{j=-n}^n D_n^{k,j}(\mathbf{R}(\alpha + \frac{\pi}{2}, \frac{\pi}{2}, \pi)) D_n^{j,l}(\mathbf{R}(\beta, \frac{\pi}{2}, \gamma + \frac{\pi}{2})) \\ &= \sqrt{2n+1} \sum_{j=-n}^n e^{-ik(\alpha - \frac{\pi}{2})} e^{-ij\beta} e^{-il(\gamma + \frac{\pi}{2})} (-1)^{k+j} d_n^{k,j}(0) d_n^{j,l}(0) \end{aligned} \quad (4.14)$$

für beliebige $(n, k, l) \in \mathcal{J}_N$. Die Symmetrieeigenschaft (3.6) der Wigner-d Funktionen liefert

$$(-1)^{k+j} d_n^{k,j}(0) = d_n^{j,k}(0).$$

Durch Einsetzen der normierten Wigner-D Funktionen aus Gleichung (4.14) in die $\mathcal{SO}(3)$ -Fourierpartialsomme (4.1) folgt

$$f_m = \sum_{n=0}^N \sum_{k,j,l=-n}^n \hat{f}_n^{k,l} \sqrt{2n+1} e^{-ik(\alpha_m - \frac{\pi}{2})} e^{-ij\beta_m} e^{-il(\gamma_m + \frac{\pi}{2})} d_n^{j,k}(0) d_n^{j,l}(0)$$

für $m = 1, \dots, M$. Vertauschen der Summationsreihenfolge liefert

$$f_m = \sum_{k,j,l=-N}^N \left(\sum_{n=\max\{|k|, |j|, |l|\}}^N \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{j,k}(0) d_n^{j,l}(0) \right) e^{-i(k,j,l) \cdot (\alpha_m - \frac{\pi}{2}, \beta_m, \gamma_m + \frac{\pi}{2})^T}$$

und es folgt die Behauptung. \square

Damit schreiben wir die $\mathcal{SO}(3)$ -Fourierpartialsomme aus Gleichung (4.2) als echte Fourierpartialsomme

$$f_m = \sum_{j,k,l=-N}^N \hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\alpha_m - \frac{\pi}{2}, \beta_m, \gamma_m + \frac{\pi}{2})^T}, \quad m = 1, \dots, M$$

mit den Fourierkoeffizienten $\hat{g}_{k,j,l} = (\mathbf{G} \hat{\mathbf{f}})_{k,j,l}$ für Indizes $k, j, l = -N \dots N$.
Unter Verwendung der Substitution

$$(\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}) := (\alpha - \frac{\pi}{2}, \beta, \gamma + \frac{\pi}{2})$$

erhalten wir für $m = 1, \dots, M$ die Fourierpartialsomme

$$f_m = \sum_{k,j,l=-N}^N \hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\tilde{\alpha}_m, \tilde{\beta}_m, \tilde{\gamma}_m)^T}.$$

Zur Berechnung des geordneten Fourierkoeffizientenvektors $(\hat{g}_{k,j,l})_{k,j,l=-N}^N$ mit der DBWT benötigen wir alle Wigner-d Matrizen mit Grad $\leq N$ im Punkt 0. Dafür verwenden wir Algorithmus 3.2. Mit Satz 4.12 erhalten wir einen schnellen Algorithmus für die NDSOFT.

Algorithmus 4.2 : Darstellungsbasierte NFSOFT

Input : $N \in \mathbb{N}$ Bandbreite
 $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ Koeffizientenvektor
 \mathcal{R}_M Menge von Rotationen mit Kardinalität M

1. **Wignertransformation:** Berechne $\hat{\mathbf{g}} = \mathbf{G} \hat{\mathbf{f}}$ mit Definition 4.11 in $\mathcal{O}(N^4)$ Schritten.
2. **Knoten bestimmen:** Seien die Rotationen $\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \in \mathcal{R}_M$ für $m = 1, \dots, M$ mit Euler-Winkeln parametrisiert. Berechne

$$(\tilde{\alpha}_m, \tilde{\gamma}_m) = (\alpha_m - \frac{\pi}{2}, \gamma_m + \frac{\pi}{2})$$

für $m = 1, \dots, M$.

3. **NFFT:** Berechne $\mathbf{f} = \mathbf{F} \hat{\mathbf{g}}$ mit der Fouriermatrix

$$\mathbf{F} = \left(e^{-i(k,j,l) \cdot (\tilde{\alpha}_m, \beta_m, \tilde{\gamma}_m)^T} \right)_{(m=1, \dots, M); (k,j,l=-N, \dots, N)}$$

in den Punkten $\{(\tilde{\alpha}_m, \beta_m, \tilde{\gamma}_m) \mid m = 1, \dots, M\}$. Verwende die trivariate NFFT in $\mathcal{O}(M + N^3 \log N)$ Schritten.

Output : NDSOFT $\mathbf{f} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} \hat{\mathbf{f}}$

Komplexität : $\mathcal{O}(M + N^4)$

Die Schritte 1 und 2 sind parallelisierbar.

4.3.2 Symmetrieeigenschaften

Wir wollen die darstellungsbasierte NFSOFT, siehe Algorithmus 4.2, verbessern. Dazu verwenden wir Symmetrieeigenschaften der Fourierkoeffizienten $(\hat{g}_{k,j,l})_{k,j,l=-N}^N$, die mit der DBWT im ersten Schritt von Algorithmus 4.2 berechnet werden.

Lemma 4.13. *Seien $N \in \mathbb{N}$ und ein Koeffizientenvektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ gegeben, der für alle $(n,k,l) \in \mathcal{J}_N$ die Gleichung*

$$\hat{f}_n^{k,l} = (-1)^{k+l} \overline{\hat{f}_n^{-k,-l}} \quad (4.15)$$

erfüllt. Weiterhin verwenden wir die DBWT \mathbf{G} aus Definition 4.11 zur Berechnung der Fourierkoeffizienten $\hat{\mathbf{g}} = (\hat{g}_{k,j,l})_{k,j,l=-N}^N = \mathbf{G} \hat{\mathbf{f}}$. Dann gilt die Symmetrieeigenschaft

$$\hat{g}_{k,j,l} = \overline{\hat{g}_{-k,-j,-l}} \quad (4.16)$$

für alle Indizes $k, j, l = -N, \dots, N$.

Beweis. Da die Wigner-d Funktionen reellwertig sind, gilt

$$\overline{\hat{g}_{-k,-j,-l}} = \sum_{n=\max\{|k|,|j|,|l|\}}^N \sqrt{2n+1} \overline{\hat{f}_n^{-k,-l}} d_n^{-j,-k}(0) d_n^{-j,-l}(0).$$

Aus Voraussetzung (4.15) und der Symmetrieeigenschaft (3.7) der Wigner-d Funktionen folgt die Behauptung. \square

Falls wir die darstellungsbasierte NFSOFT verwenden, um eine N -bandbreitenbegrenzte Funktion in endlich vielen Rotationen auszuwerten, dann nutzen wir das folgende Resultat.

Korollar 4.14. *Seien $N \in \mathbb{N}$ und eine reellwertige Funktion $f \in \mathbb{D}_N$ gegeben. Weiterhin werden die Fourierkoeffizienten $\hat{\mathbf{g}} = (\hat{g}_{k,j,l})_{k,j,l=-N}^N = \mathbf{G} \hat{\mathbf{f}}$ mit der DBWT \mathbf{G} aus Definition 4.11 aus den $\mathcal{SO}(3)$ -Fourierkoeffizienten von f berechnet. Dann gilt die Symmetrieeigenschaft (4.16) für alle Indizes $k, j, l = -N, \dots, N$.*

Beweis. Da f reellwertig ist, folgt aus Lemma 3.36, dass die Gleichung (4.15) für alle $\mathcal{SO}(3)$ -Fourierkoeffizienten von f gilt. Aus Lemma 4.13 folgt die Behauptung. \square

Lemma 4.15. *Seien $N \in \mathbb{N}$ und ein Koeffizientenvektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ gegeben. Weiterhin werden die Fourierkoeffizienten $\hat{\mathbf{g}} = (\hat{g}_{k,j,l})_{k,j,l=-N}^N = \mathbf{G} \hat{\mathbf{f}}$ mit der DBWT \mathbf{G} aus Definition 4.11 berechnet. Dann gilt die Symmetrieeigenschaft*

$$\hat{g}_{k,j,l} = (-1)^{k+l} \hat{g}_{k,-j,l} \quad (4.17)$$

für alle Indizes $k, j, l = -N, \dots, N$.

Beweis. Für $\beta = \frac{\pi}{2}$ folgt aus der Symmetrieeigenschaft (3.7) der Wigner-d Funktionen

$$d_n^{j,k}(0) d_n^{j,l}(0) = (-1)^{2n+k+l} d_n^{-j,k}(0) d_n^{-j,l}(0).$$

Wegen $(-1)^{2n} = 1$ folgt die Behauptung direkt durch Einsetzen in Gleichung (4.12). \square

Wir nutzen diese beiden Symmetrieeigenschaften bei der Berechnung der Fourierkoeffizienten aus. Dadurch verkürzt sich der Rechenaufwand der DBWT. Zuerst verwenden wir Korollar 4.14 und erhalten das folgende Resultat.

Satz 4.16. *Seien $N \in \mathbb{N}$ und ein Koeffizientenvektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ gegeben, der die Gleichung*

$$\hat{f}_n^{k,l} = (-1)^{k+l} \overline{\hat{f}_n^{-k,-l}} \quad (4.15)$$

für alle $(n,k,l) \in \mathcal{J}_N$ erfüllt. Weiterhin verwenden wir die DBWT \mathbf{G} aus Definition 4.11 zur Berechnung der Fourierkoeffizienten $\hat{\mathbf{g}} = (\hat{g}_{k,j,l})_{k,j,l=-N}^N = \mathbf{G} \hat{\mathbf{f}}$. Dann gilt

$$\sum_{k,j,l=-N}^N \hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} = \Re \left(\sum_{j,k=-N}^N \sum_{l=0}^N (2 - \delta_{l,0}) \hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} \right).$$

Beweis. Wir zerlegen die Fourierpartialsumme auf der linken Seite und schreiben

$$\begin{aligned} & \sum_{k,j=-N}^N \hat{g}_{k,j,0} e^{-i(k,j,0) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} + \sum_{k,j=-N}^N \sum_{l=1}^N \hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} \\ & + \sum_{k,j=-N}^N \sum_{l=1}^N \hat{g}_{k,j,-l} e^{-i(k,j,-l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T}. \end{aligned} \quad (4.18)$$

Umsortieren von Summanden liefert für $l = 0, \dots, N$ die Gleichung

$$\sum_{k,j=-N}^N \hat{g}_{k,j,-l} e^{-i(k,j,-l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} = \sum_{k,j=-N}^N \hat{g}_{-k,-j,-l} e^{-i(-k,-j,-l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T}.$$

Da die Voraussetzung (4.15) für alle Koeffizienten $\hat{f}_n^{k,l}$ erfüllt ist, haben wir Symmetrieeigenschaft (4.16) der Fourierkoeffizienten $\hat{\mathbf{g}}$ und es folgt

$$\sum_{k,j=-N}^N \hat{g}_{k,j,-l} e^{-i(k,j,-l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} = \sum_{k,j=-N}^N \overline{\hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T}}. \quad (4.19)$$

Einsetzen in die aufgeteilte Fourierpartialsumme aus Formel (4.18) liefert

$$\sum_{k,j=-N}^N \hat{g}_{k,j,0} e^{-i(k,j,0) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} + \sum_{k,j=-N}^N \sum_{l=1}^N 2 \Re \left(\hat{g}_{k,j,l} e^{-i(k,j,l) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T} \right).$$

Für $l = 0$ folgt aus Gleichung (4.19), dass $\sum_{k,j=-N}^N \hat{g}_{k,j,0} e^{-i(k,j,0) \cdot (\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})^T}$ reellwertig ist und damit erhalten wir die Behauptung. \square

Die Fourierpartialsomme mit den Fourierkoeffizienten $\hat{\mathbf{g}} = \mathbf{G} \hat{\mathbf{f}}$ hat nun eine verkürzte Darstellung. Dadurch kann die trivariate NFFT im dritten Schritt der darstellungsbasierten NFSOFT mit halben Aufwand berechnet werden. Weiterhin müssen wir im Schritt 1 nur die Hälfte der Fourierkoeffizienten $\hat{\mathbf{g}}$ bestimmen. Mit der Symmetrieeigenschaft (4.17) vereinfachen wir die Berechnung von $\hat{\mathbf{g}}$ noch weiter und erhalten Algorithmus 4.3.

Algorithmus 4.3 : Darstellungsbasierte NFSOFT unter Verwendung der Symmetrie

Input : $N \in \mathbb{N}$ Bandbreite
 $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ Koeffizientenvektor (definiert als $\mathcal{SO}(3)$ -Fourierkoeffizientenvektor eine eindeutige Funktion $f \in \mathbb{D}_N$)
 \mathcal{R}_M Menge von Rotationen mit Kardinalität M

1. **Wignertransformation:** Berechne $\hat{\mathbf{g}} = (\hat{g}_{k,j,l})_{k,j,l=-N}^N = \mathbf{G} \hat{\mathbf{f}}$ mit Definition 4.11 in $\mathcal{O}(N^4)$ Schritten. Gehe dazu wie folgt vor:
 - 1: **if** f reellwertig **then** $L \leftarrow 0$ **else** $L \leftarrow -N$
 - 2: Initialisiere $\hat{g}_{k,j,l} \leftarrow 0$ für alle $k, j = -N \dots N$ und $l = L, \dots, N$
 - 3: **for** $n = 0$ bis N **do**
 - 4: Bestimme $d_n(0)$ mit dem Rekursionsschritt aus Algorithmus 3.1. Verwende $d_0(0)$ und $d_1(0)$ aus Beispiel 3.10 (a) und (b) als Startwerte.
 - 5: **for** $l = \max\{L, -n\}$ bis n **do**
 - 6: **for** $k = -n$ bis n **do**
 - 7: **for** $j = 0$ bis n **do**
 - 8: $\hat{g}_{k,j,l} \leftarrow \hat{g}_{k,j,l} + \sqrt{2n+1} \hat{f}_n^{k,l} d_n^{j,k}(0) d_n^{j,l}(0)$
 - 9: **if** $n = N$ **then**
 - 10: $\hat{g}_{k,-j,l} \leftarrow (-1)^{k+l} \hat{g}_{k,j,l}$
 - 11: **if** f reellwertig **and** $l = 0$ **then**
 - 12: $\hat{g}_{k,j,0} \leftarrow \frac{1}{2} \hat{g}_{k,j,0}$
 - 13: **end**
 - 14: **end**
 - 15: **end**
 - 16: **end**
 - 17: **end**
 - 18: **end**
-

-
2. **Knoten bestimmen:** Seien die Rotationen $\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \in \mathcal{R}_M$ für $m = 1, \dots, M$ mit Euler-Winkeln parametrisiert. Berechne

$$(\tilde{\alpha}_m, \tilde{\gamma}_m) = (\alpha_m - \frac{\pi}{2}, \gamma_m + \frac{\pi}{2})$$

für $m = 1, \dots, M$.

3. **NFFT:** Berechne $\mathbf{f} = \mathbf{F} \hat{\mathbf{g}}$ mit der Fouriermatrix

$$\mathbf{F} = \left(e^{-i(k,j,l) \cdot (\tilde{\alpha}_m, \beta_m, \tilde{\gamma}_m)^T} \right)_{(m=1, \dots, M); (k,j=-N, \dots, N \text{ und } l=L, \dots, N)}$$

in den Punkten $\{(\tilde{\alpha}_m, \beta_m, \tilde{\gamma}_m) \mid m = 1, \dots, M\}$. Verwende die trivariate NFFT in $\mathcal{O}(M + N^3 \log N)$ Schritten. Setze \mathbf{f} auf $2 \Re(\mathbf{f})$, wenn f reellwertig ist.

Output : NDSOFT $\mathbf{f} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N} \hat{\mathbf{f}}$

Komplexität : $\mathcal{O}(M + N^4)$

Die Schritte 1 und 2 sind parallelisierbar.

Bemerkung 4.17. In Schritt 2 verschieben wir die Knoten für die NFFT, welche in Schritt 3 durchgeführt wird. Wegen

$$e^{-i(k,j,l) \cdot (\alpha_m - \frac{\pi}{2}, \beta_m, \gamma_m + \frac{\pi}{2})^T} = i^{k-l} e^{-i(k,j,l) \cdot (\alpha_m, \beta_m, \gamma_m)^T}$$

können wir alternativ alle Koeffizienten $\hat{g}_{k,j,l}$ mit i^{k-l} multiplizieren.

- 2'. **Fourierkoeffizienten vorberechnen:** Berechne

$$\hat{g}_{k,j,l} = i^{k-l} \hat{g}_{k,j,l}$$

für alle $k, j = -N, \dots, N$ und $l = L, \dots, N$.

Die Berechnungen aus Schritt 2' können wir auch im ersten Schritt direkt vor Zeile 10 durchführen. Damit haben wir einen alternativen Algorithmus. Falls die Euler-Winkel auf einem äquidistanten Gitter liegen, ist Schritt 3 mit einer trivariaten FFT berechenbar.

4.4 Numerische Integration auf $\mathcal{SO}(3)$

Wir wollen die $\mathcal{SO}(3)$ -Fourierkoeffizienten einer N -bandbreitenbegrenzten Funktion berechnen, welche durch Funktionswerte in diskreten Punkten gegeben ist. Deshalb benötigen wir eine Quadraturformel auf $\mathcal{SO}(3)$ für das Integral $\langle f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))}$.

Sei eine Funktion $f: \mathcal{SO}(3) \rightarrow \mathbb{C}$ in Euler-Winkel-Parametrisierung mit

$$f(\mathbf{R}(\alpha, \beta, \gamma)) = f_1(\alpha) f_2(\cos \beta) f_3(\gamma)$$

durch Funktionswerte in diskreten Punkten gegeben. Weiterhin ist f_2 ein Polynom und f_1, f_3 sind trigonometrische Polynome. Dann wollen wir das Integral

$$\int_{\mathcal{SO}(3)} f(\mathbf{R}) \, d\mathbf{R}$$

aus den Funktionswerten von f in diskreten Punkten berechnen.

Dazu verwenden wir das normierte Volumenelement $d\mathbf{R}$ aus Lemma 2.17 und erhalten das Integral

$$\frac{1}{8\pi^2} \int_0^{2\pi} \int_0^\pi \int_0^{2\pi} f(\mathbf{R}(\alpha, \beta, \gamma)) \sin \beta \, d\alpha \, d\beta \, d\gamma.$$

Wir berechnen die drei univariaten Integrale getrennt.

Zunächst geben wir eine Definition zur numerischen Integration an, siehe [38, Kapitel 4].

Definition 4.18. Seien $n \in \mathbb{N}$, ein Körper $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, eine kompakte Menge Ω und der Vektorraum $V(\Omega)$ der auf Ω integrierbaren \mathbb{K} -wertigen Funktionen gegeben. Weiterhin haben wir eine zulässige Gewichtsfunktion $\omega: \Omega \rightarrow \mathbb{R}$, d.h. ω ist nichtnegativ, integrierbar auf Ω und hat höchstens endlich viele Nullstellen auf Ω . Sei zusätzlich das lineare Funktional $I: V(\Omega) \rightarrow \mathbb{K}$ mit

$$I(f) = \int_{\Omega} \omega(x) f(x) \, dx$$

gegeben. Weiterhin haben wir Stützstellen $x_0, \dots, x_n \in \Omega$ und Gewichte $\omega_0, \dots, \omega_n \geq 0$. Dann heißt das lineare Funktional $Q_n: V(\Omega) \rightarrow \mathbb{K}$ mit

$$Q_n(f) = \sum_{k=0}^n \omega_k f(x_k)$$

Quadraturformel der Ordnung n .

Sei zusätzlich eine Folge $(p_j)_{j=0}^\infty$ von Orthogonalpolynomen $p_j \in \Pi_j$, $j \in \mathbb{N}_0$ in $L_2(\Omega)$ bezüglich ω gegeben. Falls für die ersten $L+1$ Orthogonalpolynome

$$Q_n(p_j) = I(p_j), \quad j = 0, \dots, L$$

gilt, dann nennen wir L den Exaktheitsgrad von Q_n .

Da die Orthogonalpolynome p_0, \dots, p_L eine Basis des Polynomraumes Π_L bilden, ist eine Quadraturformel mit Exaktheitsgrad L für alle Polynome aus Π_L exakt.

Wir können die Qualität von Quadraturformeln durch ihre Effizienz bewerten, siehe [10] und [19].

Definition 4.19. Auf der Integrationsdomain Ω sei eine Quadraturformel Q_n mit $M \in \mathbb{N}$ Stützstellen und Exaktheitsgrad $L \in \mathbb{N}$ gegeben. Die Effizienz

$$E = \frac{L}{(1 + \dim(\Omega)) M}$$

ist der Quotient aus L Gleichungen, die für Exaktheit der Quadraturformel erfüllt sein müssen und $(1 + \dim(\Omega)) M$ freien Parametern. Dabei haben wir M Quadraturgewichte und $\dim(\Omega) \cdot M$ Koordinaten in M Quadraturknoten.

Quadraturformeln werden dazu verwendet das Integral einer Funktion näherungsweise zu bestimmen. Wir integrieren im weiteren Verlauf über Polynome bzw. trigonometrische Polynome. Dazu bestimmen wir Quadraturgewichte und Quadraturknoten, so dass die resultierenden Quadraturformeln die Polynome bzw. trigonometrischen Polynome bis zu einem gewissen Grad exakt integrieren.

Im Folgenden werden zwei verschiedene Quadraturformeln, die Gauß-Quadratur und die Clenshaw-Curtis-Quadratur, angegeben. Danach kombinieren wir diese zu einer Quadraturformel für Integrale auf $\mathcal{SO}(3)$.

4.4.1 Die Gauß-Quadratur

Die Gauß-Quadratur liefert eine Strategie zur Wahl von Quadraturgewichten und Quadraturknoten, so dass der Exaktheitsgrad maximal ist. Für $n \rightarrow \infty$ konvergieren Gauß-Quadraturformeln Q_n für alle stetigen Funktionen f und haben kein Problem mit Rundungsfehlern, siehe [34].

Wir geben die Gauß-Quadratur für Funktionen auf abgeschlossenen Intervallen an. Für den Beweis siehe [38, Kapitel 4.3].

Satz 4.20. Seien $n \in \mathbb{N}$, ein Intervall $[a, b]$ und eine zulässige Gewichtsfunktion ω gegeben. Weiterhin haben wir ein gewichtetes Skalarprodukt $\langle \cdot, \cdot \rangle$ auf $C([a, b])$ und ein System von Orthogonalpolynomen $p_j \in \Pi_j$, $j = 0, \dots, n + 1$ bezüglich $\langle \cdot, \cdot \rangle$.

Dann existiert genau eine Quadraturformel der Form

$$Q_n(f) = \sum_{k=0}^n \omega_k f(x_k)$$

zur Berechnung des Integrals $I(f) = \int_a^b \omega(x) f(x) dx$, die für alle Polynome aus Π_{2n+1} exakt ist.

Dabei sind die Quadraturknoten x_0, \dots, x_n die Nullstellen des $(n + 1)$ -ten Orthogonalpolynoms p_{n+1} .

Für $j = 0, \dots, n$ seien die Lagrange-Polynome $\ell_j \in \Pi_n$ diejenigen Polynome vom Grad n , für die $\ell_j(x_k) = \delta_{k,j}$ in allen Stützstellen $k = 0, \dots, n$ gilt. Damit erhalten wir die Quadraturgewichte

$$\omega_k = \int_a^b \omega(x) \ell_j(x) dx$$

für $k = 0, \dots, n$.

Wir untersuchen periodische Funktionen. Dazu wählen wir anstelle eines kompakten Intervalls den Torus als Integrationsdomain. Als Orthogonalbasis sind die algebraischen Polynome ungeeignet. Deshalb definieren wir das folgende Orthogonalsystem, siehe [23, Kapitel 1.2].

Definition 4.21. *Das Funktionensystem*

$$\{1, \sin(kx), \cos(kx) \mid k \in \mathbb{N}\}$$

bildet eine Orthogonalbasis des $\mathbb{L}_2(\mathbb{T})$.

Sei $n \in \mathbb{N}$. Dann ist

$$\mathcal{T}_n = \text{span}\{1, \sin(kx), \cos(kx) \mid k = 1, \dots, n\}$$

der Raum der trigonometrischen Polynome mit Grad $\leq n$.

Wir wollen eine Quadraturformel für trigonometrische Polynome \mathcal{T}_n auf dem Torus angeben. Für den Beweis der Exaktheit brauchen wir das folgende Lemma, siehe [23, Kapitel 3.1].

Lemma 4.22. *Sei $n \in \mathbb{N}$ gegeben. Für alle $j \in \mathbb{Z}$ gilt*

$$\sum_{k=0}^{n-1} e^{2\pi i j k/n} = n \cdot \delta_{0, j \bmod n}.$$

Beweis. Wir verwenden die Darstellung über die Einheitswurzel $w_n^k := e^{2\pi i k/n}$.

Falls $j \bmod n = 0$ gilt, dann existiert $l \in \mathbb{Z}$ mit $j = ln$ und es folgt $w_n^j = (w_n^n)^l = 1$.

Sei also $j = ln + m$ mit $l \in \mathbb{Z}$ und $m \in \{1, \dots, n-1\}$. Dann ist $w_n^j = (w_n^n)^l \cdot w_n^m = w_n^m \neq 1$.

Für $x \neq 1$ gilt die Teleskopsumme

$$\sum_{k=0}^{n-1} x^k = \frac{1 - x^n}{1 - x},$$

woraus die Behauptung folgt. □

Wir geben eine Quadraturformel für periodische Funktionen an, siehe [23, Kapitel 9.3].

Satz 4.23. *Sei $n \in \mathbb{N}$ ungerade. Für $k = 0, \dots, n$ seien die äquidistanten Stützstellen $x_k = \frac{2\pi k}{n+1}$ und die konstanten Quadraturgewichte $\omega_k = \frac{2\pi}{n+1}$ gegeben.*

Für alle trigonometrischen Polynome $f \in \mathcal{T}_n$ mit Höchstgrad n ist die Quadraturformel

$$Q_n^G(f) := \sum_{k=0}^n \omega_k f(x_k)$$

gleich dem Integral $I(f) = \int_{\mathbb{T}} f(x) dx$, d.h. die Quadraturformel Q_n^G ist exakt für alle $f \in \mathcal{T}_n$. Die Quadraturformel Q_n^G hat maximalen Exaktheitsgrad.

Beweis. Wir zeigen zuerst, dass Q_n^G exakt in \mathcal{T}_n ist. Sei $f \in \mathcal{T}_n$ beliebig. Dann ist f eindeutig als Linearkombination

$$f(x) = a_0 + \sum_{k=0}^n a_k \sin(kx) + b_k \cos(kx)$$

mit Konstanten a_k, b_k darstellbar. Folglich reicht es zu zeigen, dass die Orthogonalbasisfunktionen $1, \sin(kx)$ und $\cos(kx)$ für $k = 1, \dots, n$ exakt integriert werden.

Offensichtlich gilt $I(1) = 2\pi = Q_n^G(1)$.

Sei $k \in \{1, \dots, n\}$ beliebig. Dann gilt $I(\sin(kx)) = I(\cos(kx)) = 0$. Unter Verwendung der Exponentialdarstellung von Sinus und Kosinus sowie von Lemma 4.22 erhalten wir $Q_n^G(\sin(kx)) = Q_n^G(\cos(kx)) = 0$.

Es bleibt zu zeigen, dass keine Quadraturformel Q_n existiert, die alle trigonometrischen Polynome vom Grad $\leq (n+1)$ exakt integriert. Angenommen es existiert eine solche Quadraturformel Q_n . Sei das Knotenpolynom

$$\ell(x) = \sin\left(\frac{n+1}{2}x\right) \in \mathcal{T}_{\frac{n+1}{2}}$$

gegeben. Offensichtlich ist $\ell(x_k) = 0$ für alle Stützstellen $k = 0, \dots, n$. Damit folgt für das trigonometrische Polynom

$$q(x) = \sin^2\left(\frac{n+1}{2}x\right) = \frac{1}{2}(1 - \cos((n+1)x)) \in \mathcal{T}_{n+1}$$

der Widerspruch

$$0 < \int_0^{2\pi} \sin^2\left(\frac{n+1}{2}x\right) dx = \sum_{k=0}^n \omega_k \sin^2\left(\frac{n+1}{2}x_k\right) = 0.$$

□

Die angegebene Quadratur Q^G hat den maximalen Exaktheitsgrad. Deshalb nennen wir sie Gauß-Quadratur.

Bei der üblichen Gauß-Quadratur aus Satz 4.20 hat der Raum der exakt integrierten Polynome $2n+2$ Freiheitsgrade. Im Raum der trigonometrischen Polynome haben wir nur $2n+1$ Freiheitsgrade. Die zusätzliche Annahme der Periodizität kostet einen Freiheitsgrad. Wir berechnen im Folgenden die Quadraturknoten und -gewichte über den Ansatz der Gauß-Quadratur aus Satz 4.20.

Bemerkung 4.24. Sei $n \in \mathbb{N}$ ungerade. Wir vollziehen den Beweis der Gauß-Quadratur aus Satz 4.20 für trigonometrische Polynome auf dem Torus nach. Dabei erhalten wir Exaktheit bis zum Grad n , falls die Quadraturknoten gerade die Nullstellen von einer Linearkombination des n -ten und $(n+1)$ -ten Orthogonalpolynoms sind, d.h. von

$$a \sin\left(\frac{n+1}{2}x\right) + b \cos\left(\frac{n+1}{2}x\right)$$

für $a, b \in \mathbb{R}$ mit $a \cdot b \neq 0$.

Die Nullstellen sind für alle Koeffizienten a, b äquidistant. Durch Verschieben der Integrationsgrenzen auf dem Torus ist die Stützstellenmenge, welche wir für beliebige a, b erhalten, auf $\{\frac{2\pi k}{n+1} \mid k = 0, \dots, n\}$ transformierbar.

Wir definieren die trigonometrischen Lagrange-Polynome für $j = 0, \dots, n$ durch

$$\ell_j(x) = c_j \cdot \cos\left(\frac{1}{2}(x - x_j)\right) \cdot \prod_{\substack{k=0 \\ k \neq j}}^n \sin\left(\frac{1}{2}(x - x_k)\right) \in \mathcal{T}_{\frac{n+1}{2}}$$

mit einer Konstante $c_j \in \mathbb{R} \setminus \{0\}$.

Da die Stützstellen auf dem Torus äquidistant sind, ist jedes trigonometrische Lagrange-Polynom durch Verschiebung des 0-ten trigonometrischen Lagrange-Polynoms konstruierbar, d.h. $\ell_j(x) = \ell_0(x - x_j)$. Damit sind alle Konstanten c_j gleich. Wir setzen $x = x_0$ und erhalten die Konstante

$$\frac{1}{c_0} = \prod_{k=1}^n \sin\left(\frac{1}{2}(x_0 - x_k)\right) = \prod_{k=1}^n \sin\left(-\frac{\pi k}{n+1}\right) = -\frac{n+1}{2^n}.$$

Wegen $\ell_j(x) = \ell_0(x - x_j)$ sind alle Quadraturgewichte gleich und es gilt

$$\begin{aligned} \omega_j &= \int_0^{2\pi} \ell_j(x) dx = -\frac{2^n}{n+1} \int_0^{2\pi} \cos\frac{x}{2} \cdot \prod_{k=1}^n \sin\left(\frac{x}{2} - \frac{k\pi}{n+1}\right) dx \\ &= \frac{1}{n+1} \int_0^{2\pi} \cot\frac{x}{2} \sin\left(\frac{n+1}{2}x\right) dx = \frac{2\pi}{n+1}. \end{aligned}$$

Wir wollen mit der Quadratur aus Satz 4.23 komplexe Exponentiale exakt integrieren. Die Stützstellen und Quadraturgewichte sind sehr einfach berechenbar.

Bemerkung 4.25. Aus der Eulerschen Formel $e^{i k x} = \cos(k x) + i \sin(k x)$ folgt

$$\mathcal{T}_n \cong \{e^{i k x} \mid k = -n, \dots, n\}.$$

Sei $n = 2N + 1$ ungerade. Wegen Satz 4.23 ist die Quadraturformel

$$Q_{2N+1}^G(f) = \frac{\pi}{N+1} \sum_{j=0}^{2N+1} f\left(\frac{\pi j}{N+1}\right) = \int_{\mathbb{T}} f(x) dx \quad (4.20)$$

exakt für alle trigonometrischen Polynome mit Höchstgrad $2N + 1$, d.h. für alle $f \in \mathcal{T}_{2N+1}$. Die Quadraturformel (4.20) wird Trapezregel genannt, siehe [38, Kapitel 4.2.2].

Wir untersuchen die Effizienz der Gauß-Quadratur, siehe [10].

Bemerkung 4.26. Für die Gauß-Quadraturformel Q_n^G aus Satz 4.23 erhalten wir die

Effizienz

$$E = \frac{2n}{2(n+1)},$$

und damit $E \rightarrow 1$ für $n \rightarrow \infty$.

4.4.2 Die Clenshaw-Curtis-Quadratur

Wir wollen reellwertige Polynome mit Gewichtsfunktion $\omega \equiv 1$ auf dem Intervall $[-1, 1]$ exakt integrieren. Dazu suchen wir eine Quadraturformel der Ordnung $n \in \mathbb{N}$. Da die zu $\omega \equiv 1$ gehörenden Orthogonalpolynome die Legendre-Polynome sind, wird häufig die Gauß-Legendre-Quadratur verwendet, siehe [38, Kapitel 4.3.3]. Diese ist exakt für Polynome mit Grad $\leq 2n + 1$, siehe Satz 4.20. Die Quadraturknoten und Gewichte sind durch Lösen eines tridiagonalen Eigenwertproblems, siehe [8], in $\mathcal{O}(n^2)$ Operationen berechenbar.

Alternativ existieren Clenshaw-Curtis-Quadraturformeln, deren Idee darin besteht, den Integranden f durch das Interpolationspolynom in den Chebyshevextrempunkten darzustellen. Die Quadraturgewichte sind mit der FFT in $\mathcal{O}(n \log n)$ Operationen berechenbar, siehe [23, Kapitel 6.4.2].

Zum Vergleich von der Clenshaw-Curtis-Quadratur mit der Gauß-Legendre-Quadratur verwenden wir [34].

Bemerkung 4.27. *Die Clenshaw-Curtis-Quadratur integriert Polynome bis zum Grad n exakt. Damit benötigen wir im Vergleich zur Gauß-Quadratur die doppelte Anzahl an Stützstellen. In der Praxis ist die Clenshaw-Curtis-Quadratur für die meisten Integranden f genauso akkurat wie die Gauß-Quadratur, siehe [21] und [34]. Analog zur Gauß-Quadratur konvergiert die Clenshaw-Curtis-Quadratur für alle stetigen Funktionen f und hat keine Probleme mit Rundungsfehlern. Damit ist sie so robust wie die Gauß-Quadratur und lässt sich für gegebenes n schneller berechnen.*

Wir geben eine Clenshaw-Curtis-Quadraturformel für gerades n an. Der Beweis kann bei Plonka oder Filippi nachvollzogen werden, siehe [23, Kapitel 6.4.2] und [4, Kapitel 3].

Satz 4.28. *Sei $n \in \mathbb{N}$ mit $n = 2N$ gerade und*

$$\varepsilon_N(j) = \begin{cases} 1/2 & \text{falls } j \in \{0, N\}, \\ 1 & \text{sonst.} \end{cases}$$

Weiterhin haben wir die Quadraturknoten $x_k = \cos \frac{k\pi}{2N}$ für $k = 0, \dots, 2N$ und die symmetrischen Clenshaw-Curtis Quadraturgewichte

$$\omega_k^N = \omega_{2N-k}^N := \frac{\varepsilon_{2N}(k)}{N} \sum_{j=0}^N \varepsilon_N(j) \frac{2}{1-4j^2} \cos \frac{kj\pi}{N} \quad (4.21)$$

für $k = 0, \dots, N$ gegeben. Für alle Polynome mit Höchstgrad $2N$ ist die Quadraturformel

$$Q_{2N}^{CC}(f) := \sum_{k=0}^{2N} \omega_k^N f(x_k)$$

gleich dem Integral $\int_{-1}^1 f(x) dx$, d.h. sie ist exakt für alle Polynome $f \in \Pi_{2N}$.

Bemerkung 4.29. Polynome mit Grad $\leq 2N$ sind durch eine Linearkombination der Chebyshev-Polynome T_0, \dots, T_{2N} darstellbar. Zur Auswertung dieser Chebyshevreihe in den Clenshaw-Curtis-Quadraturknoten wird die DCT mit Komplexität $\mathcal{O}(N \log N)$ verwendet. Die Quadraturgewichte werden mit einer FFT in $\mathcal{O}(N \log N)$ Operationen berechnet, siehe [23, Kapitel 6.4.2].

Wir untersuchen die Effizienz der Clenshaw-Curtis-Quadratur, siehe [10].

Bemerkung 4.30. Für die Clenshaw-Curtis-Quadraturformel Q_{2N}^{CC} aus Satz 4.28 erhalten wir die Effizienz

$$E = \frac{2N}{2(2N+1)}$$

und damit $E \rightarrow \frac{1}{2}$ für $N \rightarrow \infty$.

4.4.3 Quadratur auf der Rotationsgruppe

Gräf vergleicht in [10, 9] verschiedene Quadraturformeln auf $\mathcal{SO}(3)$, die sich in der Wahl der Stützstellen unterscheiden.

Wir kombinieren die Gauß- und Clenshaw-Curtis-Quadraturformeln zur Berechnung des Integrals

$$\int_{\mathcal{SO}(3)} f(\mathbf{R}) d\mathbf{R}.$$

Dazu verwenden wir das Volumenelement aus Lemma 2.17 und erhalten den folgenden Satz, siehe [24, Kapitel 3.5].

Satz 4.31. Sei $N \in \mathbb{N}$. Weiterhin haben wir für $a, c = 0, \dots, 2N+1$ und $b = 0, \dots, 2N$ die Quadraturknoten

$$x_{a,b,c} = \mathbf{R} \left(\frac{a\pi}{N+1}, \frac{b\pi}{2N}, \frac{c\pi}{N+1} \right)$$

auf der mit Euler-Winkeln parametrisierten Rotationsgruppe $\mathcal{SO}(3)$. Seien die Quadraturgewichte

$$\omega_{a,b,c} = \frac{\omega_b^N}{8(N+1)^2} \tag{4.22}$$

mit den Clenshaw-Curtis-Quadraturgewichten ω_b^N aus Gleichung (4.21) gegeben.
 Sei $f: \mathcal{SO}(3) \rightarrow \mathbb{C}$ bezüglich der Euler-Winkel separierbar, so dass der Produktansatz

$$f(\mathbf{R}(\alpha, \beta, \gamma)) = f_1(\alpha) \cdot f_2(\cos \beta) \cdot f_3(\gamma)$$

gilt. Dabei sind $f_1, f_3 \in \mathcal{T}_{2N}$ trigonometrische Polynome mit Grad $\leq 2N$ und $f_2 \in \Pi_{2N}$ ein Polynom mit Grad $\leq 2N$.

Dann ist die Quadraturformel

$$Q_N^{\mathcal{SO}}(f) = \sum_{a,c=0}^{2N+1} \sum_{b=0}^{2N} \omega_{a,b,c} f(x_{a,b,c})$$

gleich dem Integral $I(f) = \int_{\mathcal{SO}(3)} f(\mathbf{R}) d\mathbf{R}$, d.h. die Quadraturformel ist exakt.

Beweis. Wir verwenden die Euler-Winkel-Parametrisierung und schreiben das Integral $I(f)$ mit Lemma 2.17 als

$$I(f) = \frac{1}{8\pi^2} \int_0^{2\pi} f_1(\alpha) d\alpha \int_0^\pi \sin \beta f_2(\cos \beta) d\beta \int_0^{2\pi} f_3(\gamma) d\gamma.$$

Substitution im zweiten Integral liefert

$$\int_0^\pi f_2(\cos \beta) \sin \beta d\beta = \int_{-1}^1 f_2(x) dx.$$

Aus $f_2 \in \Pi_{2N}$ folgt, dass die Clenshaw-Curtis-Quadratur Q_{2N}^{CC} aus Satz 4.28 im zweiten Integral exakt ist. Wegen $f_1, f_3 \in \mathcal{T}_{2N} \subset \mathcal{T}_{2N+1}$ ist die Gauß-Quadratur Q_{2N+1}^G aus Satz 4.23 im ersten und dritten Integral exakt. Folglich gilt

$$I(f) = Q_{2N+1}^G(f_1) \cdot Q_{2N}^{CC}(f_2) \cdot Q_{2N+1}^G(f_3)$$

und damit die Behauptung. □

Wir werden später zeigen, dass die Effizienz der $\mathcal{SO}(3)$ -Quadraturformel $Q_N^{\mathcal{SO}}$ für $N \rightarrow \infty$ gegen $\frac{1}{24}$ konvergiert, siehe [10].

4.5 Berechnung von $\mathcal{SO}(3)$ -Fourierkoeffizienten

Seien komplexe Werte auf dem mit Euler-Winkeln parametrisierten äquidistanten Gitter

$$\mathcal{I}_N = \left\{ \mathbf{R} \left(\frac{a\pi}{N+1}, \frac{b\pi}{2N}, \frac{c\pi}{N+1} \right) \mid a, c = 0, \dots, 2N+1; b = 0, \dots, 2N \right\} \quad (4.23)$$

gegeben. Dann wollen wir die eindeutige N -bandbreitenbegrenzte Funktion $f \in \mathbb{D}_N$ interpolieren. Dazu berechnen wir die $\mathcal{SO}(3)$ -Fourierkoeffizienten von f , d.h. wir bestimmen das Integral $\langle f, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))}$ für beliebige 3-Tupel $(n, k, l) \in \mathcal{J}_N$. Hierfür verwenden wir die $\mathcal{SO}(3)$ -Quadraturformel aus Satz 4.31.

Wir geben das folgende Lemma für die Berechnung der $\mathcal{SO}(3)$ -Fourierkoeffizienten einer N -bandbreitenbegrenzten Funktion an, siehe [24, Kapitel 3.5]. Alternativ wird in [15] ein Gauß-Legendre-Quadratur basiertes Verfahren beschrieben.

Lemma 4.32. *Seien $N \in \mathbb{N}$ und $f \in \mathbb{D}_N$ eine N -bandbreitenbegrenzte Funktion. Weiterhin haben wir die Quadraturgewichte $(\omega_{a,b,c})_{a,c=0; b=0}^{2N+1, 2N}$ aus Gleichung (4.22). Dann erfüllen die $\mathcal{SO}(3)$ -Fourierkoeffizienten $\hat{f}_n^{k,l}$ für beliebige $(n, k, l) \in \mathcal{J}_N$ die Gleichung*

$$\hat{f}_n^{k,l} = \sum_{a,c=0}^{2N+1} \sum_{b=0}^{2N} \omega_{a,b,c} f(\mathbf{R}(\alpha_a, \beta_b, \gamma_c)) \overline{D_n^{k,l}(\mathbf{R}(\alpha_a, \beta_b, \gamma_c))} \quad (4.24)$$

mit $\mathbf{R}(\alpha_a, \beta_b, \gamma_c) = \mathbf{R}\left(\frac{a\pi}{N+1}, \frac{b\pi}{2N}, \frac{c\pi}{N+1}\right)$.

Beweis. Sei $(m, r, s) \in \mathcal{J}_N$ und $f = D_m^{r,s}$. Dann gilt

$$\begin{aligned} \left(D_m^{r,s} \cdot \overline{D_n^{k,l}} \right) (\mathbf{R}(\alpha, \beta, \gamma)) &= \sqrt{(2n+1)(2m+1)} e^{-i\alpha(r-k)} d_m^{r,s}(\cos \beta) d_n^{k,l}(\cos \beta) e^{-i\gamma(s-l)} \\ &= f_1(\alpha) \cdot f_2(\cos \beta) \cdot f_3(\gamma) \end{aligned}$$

mit $f_1, f_3 \in \mathcal{T}_{2N}$.

Sei $r \neq k$ oder $s \neq l$. Wegen Lemma 4.22 ist die rechte Seite von Gleichung (4.24) gleich 0. Da die normierten Wigner-D Funktionen orthogonal sind, erhalten wir die $\mathcal{SO}(3)$ -Fourierkoeffizienten

$$\hat{f}_n^{k,l} = \langle D_m^{r,s}, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} = 0$$

und damit die Behauptung.

Sei also $r = k$ und $s = l$. Dann ist $f = D_m^{k,l}$ und wegen Bemerkung 3.9 ist $d_m^{k,l}(x) d_n^{k,l}(x)$ ein Polynom vom Grad $n + m \leq 2N$.

Damit ist die Quadraturformel Q_N^{SO} aus Satz 4.31 für alle Funktionen $\left(D_m^{r,s} \overline{D_n^{k,l}} \right) (\mathbf{R})$ mit Indizes $(m, r, s) \in \mathcal{J}_N$ exakt, d.h. es gilt

$$\hat{f}_n^{k,l} = \langle D_m^{r,s}, D_n^{k,l} \rangle_{L_2(\mathcal{SO}(3))} = Q_N^{SO} \left(D_m^{r,s} \cdot \overline{D_n^{k,l}} \right).$$

Weiterhin ist die Quadraturformel Q_N^{SO} auch für jede Linearkombination exakt, d.h. Q_N^{SO} ist für alle Funktionen der Form

$$\sum_{(m,r,s) \in \mathcal{J}_N} c_m^{r,s} \left(D_m^{r,s} \overline{D_n^{k,l}} \right) (\mathbf{R}) = \left(\sum_{(m,r,s) \in \mathcal{J}_N} c_m^{r,s} D_m^{r,s}(\mathbf{R}) \right) \overline{D_n^{k,l}(\mathbf{R})}$$

mit beliebigen Koeffizienten $c_m^{r,s} \in \mathbb{C}$ exakt. Dementsprechend ist die Quadraturformel Q_N^{SO} für das Produkt einer beliebigen N -bandbreitenbegrenzten Funktion mit $\overline{D_n^{k,l}(\mathbf{R})}$ exakt. \square

Offensichtlich ist die auf dem Gitter \mathcal{I}_N interpolierte N -bandbreitenbegrenzte Funktion eindeutig.

Wir schreiben Formel (4.24) zur Berechnung der $\mathcal{SO}(3)$ -Fourierkoeffizienten als Hintereinanderausführung von linearen Operatoren, siehe [24, Kapitel 3.5].

Korollar 4.33. *Seien eine N -bandbreitenbegrenzte Funktion $f \in \mathbb{D}_N$, der Vektor der Funktionswerte*

$$\mathbf{f} = (f(\mathbf{R}(\alpha_a, \beta_b, \gamma_c)))_{\mathbf{R}(\alpha_a, \beta_b, \gamma_c) \in \mathcal{I}_N}$$

und der Vektor der $\mathcal{SO}(3)$ -Fourierkoeffizienten

$$\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$$

gegeben. Weiterhin haben wir den linearen Operator

$$\mathbf{D}_{\mathcal{I}_N, \mathcal{J}_N} = \left(D_n^{k,l}(\mathbf{R}(\alpha_a, \beta_b, \gamma_c)) \right)_{(\mathbf{R}(\alpha_a, \beta_b, \gamma_c) \in \mathcal{I}_N); ((n,k,l) \in \mathcal{J}_N)} \in \mathbb{C}^{(2N+2)^2(2N+1) \times |\mathcal{J}_N|},$$

der NDSOFT aus Definition 4.1 sowie die Diagonalmatrix

$$\mathbf{Q}_N = \text{diag}(\omega_{a,b,c}^N)_{a=0,\dots,2N+1; b=0,\dots,2N; c=0,\dots,2N+1} \in \mathbb{R}^{(2N+2)^2(2N+1) \times (2N+2)^2(2N+1)}$$

der Quadraturgewichte aus Gleichung (4.22).

Dann ist $\mathbf{D}_{\mathcal{I}_N, \mathcal{J}_N}^H$ die adjungierte diskrete $\mathcal{SO}(3)$ -Fouriertransformation (DSOFT) und es gilt

$$\hat{\mathbf{f}} = \mathbf{D}_{\mathcal{I}_N, \mathcal{J}_N}^H \mathbf{Q}_N \mathbf{f}.$$

Beweis. Folgt direkt aus Lemma 4.32. □

Wir berechnen die Effizienz der $\mathcal{SO}(3)$ -Quadraturformel Q_N^{SO} aus Satz 4.31, siehe [10, Kapitel 3].

Bemerkung 4.34. *Aus Bemerkung 2.9 wissen wir, dass die Rotationsgruppe $\mathcal{SO}(3)$ dreidimensional ist. Weiterhin ist die Quadraturformel Q_N^{SO} für das Produkt einer beliebigen N -bandbreitenbegrenzten Funktion mit $D_n^{k,l}$ für $(n,k,l) \in \mathcal{J}_N$ exakt. Die Anzahl der Stützstellen beträgt $(2N+2)^2(2N+1)$. Damit erhalten wir die Effizienz*

$$E = \frac{\dim(\mathbb{D}_N)}{4(2N+2)^2(2N+1)}.$$

Aus Gleichung (3.11) folgt

$$E = \frac{2N+3}{24(2N+2)}$$

und damit konvergiert die Effizienz gegen $\frac{1}{24}$ für $N \rightarrow \infty$.

Bei der Hintereinanderausführung der Quadraturformel Q_N^{SO} und der adjungierten DSOFIT werden die Quadraturgewichte vorberechnet und die Koordinaten der Quadraturknoten liegen auf einem äquidistanten Gitter. Vergleichen wir die Anzahl der Abtastpunkte mit der Anzahl an berechneten $\mathcal{SO}(3)$ -Fourierkoeffizienten, dann erhalten wir eine Oversamplingrate von 6 für $N \rightarrow \infty$.

Wir suchen einen Algorithmus für die adjungierte DSOFIT.

4.6 Die adjungierte Wignertransformation

Sei das äquidistante Gitter

$$\mathcal{I}_N = \left\{ \mathbf{R} \left(\frac{a\pi}{N+1}, \frac{b\pi}{2N}, \frac{c\pi}{N+1} \right) \mid a, c = 0, \dots, 2N+1; b = 0, \dots, 2N \right\} \quad (4.23)$$

auf $\mathcal{SO}(3)$ gegeben. Wegen Korollar 4.33 verwenden wir die $\mathcal{SO}(3)$ -Quadratur \mathbf{Q}_N aus Satz 4.31 und die adjungierte DSOFIT $\mathbf{D}_{\mathcal{I}_N, \mathcal{J}_N}^H$ zur Berechnung der $\mathcal{SO}(3)$ -Fourierkoeffizienten einer N -bandbreitenbegrenzten Funktion $f \in \mathbb{D}_N$.

Wir geben im Folgenden zwei Algorithmen für die Berechnung der adjungierten NDSOFIT an. Zuerst adjungieren wir die NDSOFIT aus Kapitel 4.2 und danach die NDSOFIT aus Kapitel 4.3. In beiden Fällen ist die adjungierte NDSOFIT gerade die Hintereinanderausführung einer inversen NFFT und einer adjungierten Wignertransformation. Kircheis gibt in [16] einen Algorithmus für die inverse NFFT an.

Aus der adjungierten NDSOFIT erhalten wir die adjungierte DSOFIT. Bei dieser ist das Gitter \mathcal{I}_N äquidistant. Folglich verwenden wir eine inverse FFT anstelle der inversen NFFT. Ein Algorithmus für die inverse FFT folgt direkt aus der FFT, siehe [23, Kapitel 3.2.2].

4.6.1 Die adjungierte Wignertransformation über Chebyshev-Polynome

Wir folgen der Idee von Potts aus Kapitel 4.2. Dazu adjungieren wir die linearen Operatoren der Wignertransformation, welche die $\mathcal{SO}(3)$ -Fourierkoeffizienten über Chebyshevkoeffizienten in echte Fourierkoeffizienten transformiert, siehe [24, Kapitel 3.4].

Korollar 4.35. *Sei die Blockdiagonalmatrix*

$$\mathbf{W} = \text{diag} \left(\mathbf{W}^{k,l} \right)_{k,l=-N,\dots,N} \in \mathbb{R}^{(2N+1)^2(N+1) \times |\mathcal{J}_N|},$$

bestehend aus den Matrizen $\mathbf{W}^{k,l}$ für $k, l = -N, \dots, N$ aus Lemma 4.5 gegeben. Weiterhin haben wir die Matrizen $\mathbf{C}^{k,l}$, $\tilde{\mathbf{D}}^{k,l}$, \mathbf{S} , \mathbf{T} aus Lemma 4.5 sowie die Diagonalmatrix

$$\mathbf{K} = \text{diag} \left(\frac{2 - \delta_{0,n}}{N+1} \right)_{n=0,\dots,N}.$$

Dann gilt für den adjungierten Operator

$$\mathbf{W}^H = \text{diag} \left((\mathbf{W}^{k,l})^H \right)_{k,l=-N,\dots,N}.$$

Dabei ist

$$(\mathbf{W}^{k,l})^H = \begin{cases} \mathbf{C}^{k,l} (\tilde{\mathbf{D}}^{k,l})^H \mathbf{T} \mathbf{K} & \text{falls } k+l \text{ gerade,} \\ \mathbf{C}^{k,l} (\tilde{\mathbf{D}}^{k,l})^H \mathbf{S}^{-1} \mathbf{T} \mathbf{K} & \text{falls } k+l \text{ ungerade} \end{cases}$$

für alle Indizes $k, l = -N, \dots, N$.

Beweis. Beim Adjungieren einer Blockdiagonalmatrix erhalten wir die Blockdiagonalmatrix der adjungierten Blöcke.

Adjungieren von $\mathbf{W}^{k,l}$ aus Lemma 4.5 liefert

$$(\mathbf{W}^{k,l})^H = \begin{cases} (\mathbf{C}^{k,l})^H (\tilde{\mathbf{D}}^{k,l})^H \mathbf{T}^{-H} & \text{falls } k+l \text{ gerade,} \\ (\mathbf{C}^{k,l})^H (\tilde{\mathbf{D}}^{k,l})^H \mathbf{S}^{-H} \mathbf{T}^{-H} & \text{falls } k+l \text{ ungerade.} \end{cases}$$

Die Operatoren \mathbf{S}^{-1} und $\mathbf{C}^{k,l}$ sind reellwertige Diagonalmatrizen. Aus Lemma 4.4 folgt

$$\mathbf{T}^{-H} = \mathbf{T}^{-T} = \mathbf{T} \cdot \text{diag} \left(\frac{2 - \delta_{0,n}}{N+1} \right)_{n=0,\dots,N}.$$

Einsetzen liefert die Behauptung. \square

Wir analysieren die Komplexität der adjungierten Operatoren von $\mathbf{W}^{k,l}$ für beliebige Indizes $k, l = -N, \dots, N$. Die Diagonalmatrizen haben einen approximativen Aufwand von $\mathcal{O}(N)$. Für die Matrix-Vektor-Multiplikation der trigonometrischen Matrix \mathbf{T} verwenden wir die DCT mit Komplexität $\mathcal{O}(N \log N)$. Der adjungierte Wigner-d Operator $(\tilde{\mathbf{D}}^{k,l})^H$ wird mit der adjungierten FPT von Potts [25] mit approximativer Komplexität $\mathcal{O}(N \log^2 N)$ berechnet. Da es $(2N+1)^2$ Variationen der Indizes k, l gibt, hat der Operator \mathbf{W}^H eine Gesamtkomplexität von $\mathcal{O}(N^3 \log^2 N)$.

Korollar 4.36. *Sei die Blockdiagonalmatrix*

$$\mathbf{A} = \text{diag} \left(\mathbf{A}^{k,l} \right)_{k,l=-N,\dots,N} \in \mathbb{C}^{(2N+1)^3 \times (2N+1)^2(N+1)},$$

bestehend aus den Matrizen $\mathbf{A}^{k,l}$ für $k, l = -N, \dots, N$ aus Lemma 4.9 gegeben. Weiterhin haben wir die dünn besetzten Matrizen \mathbf{H} und \mathbf{O} aus Lemma 4.9. Dann gilt für den adjungierten Operator

$$\mathbf{A}^H = \text{diag} \left((\mathbf{A}^{k,l})^H \right)_{k,l=-N,\dots,N}.$$

Dabei ist

$$(\mathbf{A}^{k,l})^H = \begin{cases} \mathbf{H}^T & \text{falls } k+l \text{ gerade,} \\ -\mathbf{H}^T \mathbf{O} & \text{falls } k+l \text{ ungerade.} \end{cases}$$

für alle Indizes $k, l = -N, \dots, N$.

Beweis. Beim Adjungieren einer Blockdiagonalmatrix erhalten wir die Blockdiagonalmatrix der adjungierten Blöcke.

Seien $k, l = -N, \dots, N$ beliebig. Adjungieren von $\mathbf{A}^{k,l}$ aus Lemma 4.9 liefert

$$(\mathbf{A}^{k,l})^H = \begin{cases} \mathbf{H}^H & \text{falls } k+l \text{ gerade,} \\ \mathbf{H}^H \mathbf{O}^H & \text{falls } k+l \text{ ungerade.} \end{cases}$$

Die Definition der Matrizen aus Lemma 4.9 impliziert, dass \mathbf{H} reellwertig ist und $\mathbf{O}^H = -\mathbf{O}$ gilt. \square

Der Operator \mathbf{A}^H ist dünn besetzt und hat eine Komplexität von $\mathcal{O}(N^3)$.

Durch Hintereinanderausführung der adjungierten Operatoren aus den Korollaren 4.35 und 4.36 erhalten wir die adjungierte Wignertransformation. Damit liefert Satz 4.10 über die NDSOFT die adjungierte NDSOFT, siehe [24, Kapitel 3.4].

Korollar 4.37. *Sei eine mit Euler-Winkeln parametrisierte Menge $\mathcal{R}_M \subset \mathcal{SO}(3)$ mit Kardinalität M gegeben. Weiterhin sind die Blockdiagonalmatrizen $\mathbf{W} \in \mathbb{R}^{(2N+1)^2(N+1) \times |\mathcal{J}_N|}$ und $\mathbf{A} \in \mathbb{C}^{(2N+1)^3 \times (2N+1)^2(N+1)}$ sowie die trivariate Fouriermatrix $\mathbf{F}_{\mathcal{R}_M, N} \in \mathbb{C}^{M \times (2N+1)^3}$ aus Satz 4.10 gegeben. Dann gilt für den adjungierten Operator der NDSOFT $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}$ aus Definition 4.1 der Zusammenhang*

$$\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}^H = \mathbf{W}^H \mathbf{A}^H \mathbf{F}_{\mathcal{R}_M, N}^H.$$

Beweis. Adjungieren der NDSOFT aus Satz 4.10 liefert die Behauptung. \square

Für das Matrix-Vektor-Produkt der adjungierten trivariaten Fouriermatrix $\mathbf{F}_{\mathcal{R}_M, N}^H$ verwenden wir eine inverse trivariate NFFT, siehe [16], mit einer approximativen Komplexität von $\mathcal{O}(M + N^3 \log N)$.

Algorithmus 4.4 : Adjungierte NDSOFT

Input : $N \in \mathbb{N}$ Bandbreite
 \mathcal{R}_M Menge von Rotationen
 $\boldsymbol{\xi}_1 \in \mathbb{C}^M$ geordneter Vektor der Länge M

1. Berechne $\boldsymbol{\xi}_2 = \mathbf{F}_{\mathcal{R}_M, N}^H \boldsymbol{\xi}_1$ mit der adjungierten Fouriermatrix

$$\mathbf{F}_{\mathcal{R}_M, N}^H = \left(e^{i(k,n,l) \cdot (\alpha_m, \beta_m, \gamma_m)^T} \right)_{(k,n,l=-N,\dots,N); (m=1,\dots,M)}$$

in den Euler-Winkeln der Rotationen $\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \in \mathcal{R}_M$ mit inverser trivariater NFFT, siehe [16], in $\mathcal{O}(M + N^3 \log N)$ Schritten.

2. Berechne $\boldsymbol{\xi}_3 = \mathbf{A}^H \boldsymbol{\xi}_2$ mit Korollar 4.36 in $\mathcal{O}(N^3)$ Schritten.
3. Berechne $\mathbf{f} = \mathbf{W}^H \boldsymbol{\xi}_3$ mit Korollar 4.35 in $\mathcal{O}(N^3 \log^2 N)$ Schritten.

Output : adjungierte NDSOFT $\hat{\mathbf{f}} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}^H \boldsymbol{\xi}_1$

Komplexität : $\mathcal{O}(M + N^3 \log^2 N)$

Bemerkung 4.38. Wir wollen die $\mathcal{SO}(3)$ -Fourierkoeffizienten einer N -bandbreitenbegrenzten Funktion $f \in \mathbb{D}_N$ mit Hilfe von Korollar 4.33 berechnen. Dazu seien Werte auf dem mit Euler-Winkeln parametrisierten äquidistanten Gitter $\mathcal{I}_N \subset \mathcal{SO}(3)$ aus Gleichung (4.23) gegeben. Zuerst gewichten wir die gegebenen Funktionswerte mit der Diagonalmatrix der Quadraturgewichte \mathbf{Q}_N . Danach berechnen wir die adjungierte DSOF

$$\mathbf{D}_{\mathcal{I}_N, \mathcal{J}_N}^H = \mathbf{W}^H \mathbf{A}^H \mathbf{F}_{\mathcal{I}_N, N}^H.$$

mit Algorithmus 4.4. Da die Rotationsmenge \mathcal{I}_N äquidistant ist, verwenden wir im ersten Schritt des Algorithmus die adjungierte trivariate äquidistante Fouriermatrix

$$\mathbf{F}_{\mathcal{I}_N, N}^H = \left(e^{i(k,n,l) \cdot (\alpha_a, \beta_b, \gamma_c)^T} \right)_{(k,n,l) \in \mathcal{I}_N; (\mathbf{R}(\alpha_a, \beta_b, \gamma_c) \in \mathcal{I}_N)} \in \mathbb{C}^{(2N+1)^3 \times (2N+2)^2 (2N+1)}$$

mit einer inversen trivariaten FFT in $\mathcal{O}(N^3 \log N)$ Schritten. Dadurch hat der Algorithmus eine Gesamtkomplexität von $\mathcal{O}(N^3 \log^2 N)$.

4.6.2 Die adjungierte darstellungsbasierte Wignertransformation

Wir verwenden die Idee aus Kapitel 4.3 und adjungieren die DBWT, welche $\mathcal{SO}(3)$ -Fourierkoeffizienten in echte Fourierkoeffizienten transformiert. Die Idee stammt von Risbo, siehe [31].

Korollar 4.39. Seien ein geordneter Vektor $\mathbf{a} = (a_{k,j,l})_{k,j,l=-N}^N \in \mathbb{C}^{(2N+1)^3}$ und die DBWT \mathbf{G} aus Definition 4.11 gegeben. Dann ist der adjungierte Operator von \mathbf{G} für alle 3-Tupel $(n, k, l) \in \mathcal{J}_N$ durch das Matrix-Vektor-Produkt

$$(\mathbf{G}^H \mathbf{a})_n^{k,l} := \sqrt{2n+1} \sum_{j=-n}^n d_n^{j,k}(0) d_n^{j,l}(0) a_{k,j,l} \quad (4.25)$$

definiert.

Beweis. Für einen beliebigen Vektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N} \in \mathbb{C}^{|\mathcal{J}_N|}$ gilt im Euklidischen Skalarprodukt

$$\langle \mathbf{G} \hat{\mathbf{f}}, \mathbf{a} \rangle_2 = \sum_{k,j,l=-N}^N \left(\sum_{n=\max\{|k|, |j|, |l|\}}^N \sqrt{2n+1} d_n^{j,k}(0) d_n^{j,l}(0) \hat{f}_n^{k,l} \right) \overline{a_{k,j,l}}.$$

Die Wigner-d Funktionen sind reellwertig. Vertauschen der Summationsreihenfolge liefert

$$\langle \mathbf{G} \hat{\mathbf{f}}, \mathbf{a} \rangle_2 = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} \left(\sqrt{2n+1} \sum_{j=-n}^n d_n^{j,k}(0) d_n^{j,l}(0) a_{k,j,l} \right)$$

und es folgt

$$\langle \mathbf{G} \hat{\mathbf{f}}, \mathbf{a} \rangle_2 = \langle \hat{\mathbf{f}}, \mathbf{G}^H \mathbf{a} \rangle_2.$$

□

Die Matrix-Vektor-Multiplikation mit dem adjungierten Operator \mathbf{G}^H hat eine approximative Komplexität von $\mathcal{O}(N^4)$.

Aus Satz 4.12 über die darstellungsbasierte NDSOFT erhalten wir eine weitere Formel der adjungierten NDSOFT.

Korollar 4.40. *Seien ein mit Euler-Winkeln parametrisiertes Gitter $\mathcal{R}_M \subset \mathcal{SO}(3)$ mit Kardinalität M sowie das verschobene Gitter*

$$\tilde{\mathcal{R}}_M = \left\{ \mathbf{R} \left(\alpha - \frac{\pi}{2}, \beta, \gamma + \frac{\pi}{2} \right) \mid \mathbf{R}(\alpha, \beta, \gamma) \in \mathcal{R}_M \right\} \subset \mathcal{SO}(3)$$

gegeben. Weiterhin haben wir die DBWT $\mathbf{G} \in \mathbb{R}^{(2N+1)^3 \times |\mathcal{J}_N|}$ aus Definition 4.11 und die trivariate Fouriermatrix $\mathbf{F}_{\tilde{\mathcal{R}}_M, N} \in \mathbb{C}^{M \times (2N+1)^3}$ aus Gleichung (4.13). Dann gilt für den adjungierten Operator der NDSOFT $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}$ aus Definition 4.1 der Zusammenhang

$$\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}^H = \mathbf{G}^H \mathbf{F}_{\tilde{\mathcal{R}}_M, N}^H.$$

Beweis. Adjungieren der NDSOFT aus Satz 4.12 liefert die Behauptung. \square

Für das Matrix-Vektor-Produkt der adjungierten trivariaten Fouriermatrix $\mathbf{F}_{\tilde{\mathcal{R}}_M, N}^H$ verwenden wir die inverse trivariate NFFT, siehe [16], mit einer approximativen Komplexität von $\mathcal{O}(M + N^3 \log N)$.

Wir geben einen Algorithmus für die adjungierte darstellungsbasierte NFSOFT an. Dabei verwenden wir die Strategie aus Bemerkung 4.17, d.h. anstatt im zweiten Schritt alle Rotationen zu verschieben, multiplizieren wir die Koeffizienten mit dem Faktor i^{l-k} .

Algorithmus 4.5 : Adjungierte darstellungsbasierte NFSOFT

Input : $N \in \mathbb{N}$ Bandbreite
 \mathcal{R}_M Menge von Rotationen
 $\boldsymbol{\xi}_1 \in \mathbb{C}^M$ geordneter Vektor der Länge M

1. **inverse NFFT:** Berechne $\boldsymbol{\xi}_2 = \mathbf{F}^H \boldsymbol{\xi}_1$ mit der adjungierten Fouriermatrix

$$\mathbf{F}^H = \left(e^{i(k,j,l) \cdot (\alpha_m, \beta_m, \gamma_m)^T} \right)_{(k,j,l=-N, \dots, N); (m=1, \dots, M)}$$

in den Euler-Winkeln der Rotationen $\mathbf{R}(\alpha_m, \beta_m, \gamma_m) \in \mathcal{R}_M$. Verwende die inverse trivariate NFFT aus [16] in $\mathcal{O}(M + N^3 \log N)$ Schritten.

2. **Koeffizienten anpassen:** Berechne $(\boldsymbol{\xi}_3)_{k,j,l} = i^{l-k} (\boldsymbol{\xi}_2)_{k,j,l}$ für $k, j, l = -N, \dots, N$.
3. **adjungierte Wignertransformation:** Berechne $\hat{\mathbf{f}} = \mathbf{G}^H \boldsymbol{\xi}_3$ mit Korollar 4.39 in $\mathcal{O}(N^4)$ Schritten.

Output : adjungierte NDSOFT $\hat{\mathbf{f}} = \mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}^H \boldsymbol{\xi}_1$

Komplexität : $\mathcal{O}(M + N^4)$

Wegen Bemerkung 4.38 ist zur Berechnung der adjungierten DSOFD eine inverse trivariate FFT im ersten Schritt des Algorithmus ausreichend. Damit hat der Algorithmus eine Gesamtkomplexität von $\mathcal{O}(N^4)$.

Wir wollen ein Verfahren zur Berechnung der adjungierten DBWT \mathbf{G}^H angeben und Symmetrieeigenschaften ausnutzen, um den numerischen Aufwand zu verringern.

Bemerkung 4.41. *Seien ein geordneter Vektor $\mathbf{a} = (a_{k,j,l})_{k,j,l=-N}^N \in \mathbb{C}^{(2N+1)^3}$ und die adjungierte DBWT \mathbf{G}^H aus Gleichung (4.25) gegeben. Aus der Symmetrieeigenschaft (3.6) der Wigner-d Funktionen folgt*

$$(\mathbf{G}^H \mathbf{a})_n^{k,l} = \sqrt{2n+1} \sum_{j=0}^n d_n^{j,k}(0) d_n^{j,l}(0) \left(a_{k,j,l} + (-1)^{k+l} \delta_{j,0} a_{k,-j,l} \right).$$

Durch Vorberechnung der Terme $a_{k,j,l} + (-1)^{k+l} \delta_{j,0} a_{k,-j,l}$ wird der Aufwand für große Bandbreiten N nahezu halbiert.

Falls wir die adjungierte Wignertransformation verwenden, um die $\mathcal{SO}(3)$ -Fourierkoeffizienten einer reellwertigen N -bandbreitenbegrenzten Funktion $f \in \mathbb{D}_N$ zu berechnen, dann liefert die Symmetrieeigenschaft aus Lemma 3.36 die Gleichung

$$(\mathbf{G}^H \mathbf{a})_n^{k,l} = (-1)^{k+l} \overline{(\mathbf{G}^H \mathbf{a})_n^{-k,-l}}.$$

Damit sind in jeder Iteration $n = 0, \dots, N$ nur die Hälfte der Werte zu ermitteln.

Wir verwenden diese Symmetrieeigenschaften und geben Algorithmus 4.6 zur Berechnung der $\mathcal{SO}(3)$ -Fourierkoeffizienten einer N -bandbreitenbegrenzten Funktion an.

Algorithmus 4.6 : Berechnung von $\mathcal{SO}(3)$ -Fourierkoeffizienten mit der adjungierten darstellungsbasierten NFSOFT unter Verwendung der Symmetrie

Input : $N \in \mathbb{N}$ Bandbreite
 \mathcal{I}_N äquidistantes Rotationsgitter aus Gleichung (4.23)
 \mathbf{f} Vektor der Funktionswerte einer N -bandbreitenbegrenzten Funktion $f \in \mathbb{D}_N$ ausgewertet auf \mathcal{I}_N

1. **Quadratur:** Gewichte die Funktionswerte $\mathbf{f}^Q = \mathbf{Q}_N \mathbf{f}$ mit der Diagonalmatrix der $\mathcal{SO}(3)$ -Quadraturgewichte aus Korollar 4.33.

2. **inverse FFT:** Berechne $\boldsymbol{\xi} = \mathbf{F}^H \mathbf{f}^Q$ mit der adjungierten Fouriermatrix

$$\mathbf{F}^H = \left(e^{i(k,j,l) \cdot (\alpha_a, \beta_b, \gamma_c)^T} \right)_{(k,j,l=-N, \dots, N); (\mathbf{R}(\alpha_a, \beta_b, \gamma_c) \in \mathcal{I}_N)}.$$

Verwende die inverse trivariate FFT in $\mathcal{O}(N^3 \log N)$ Schritten.

3. **adjungierte Wignertransformation:** Berechne $\hat{\mathbf{f}} = \mathbf{G}^H \left((i^{l-k})_{k,j,l=-N, \dots, N} \right) \boldsymbol{\xi}$ mit Korollar 4.39 in $\mathcal{O}(N^4)$ Schritten. Gehe dazu wie folgt vor:

- 1: Initialisiere $(\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N} \leftarrow 0$
- 2: **if** f reellwertig **then** $L \leftarrow 0$ **else** $L \leftarrow -N$
- 3: Setze $a_{k,j,l} \leftarrow i^{l-k} \left((\boldsymbol{\xi})_{k,j,l} + (-1)^{k+l} \delta_{j,0} (\boldsymbol{\xi})_{k,-j,l} \right)$ für alle Indizes $k = -N, \dots, N$, $l = L, \dots, N$ und $j = 0, \dots, N$
- 4: **for** $n = 0$ bis N **do**
- 5: Bestimme $d_n(0)$ mit dem Rekursionsschritt aus Algorithmus 3.1. Verwende $d_0(0)$ und $d_1(0)$ aus Beispiel 3.10 (a) und (b) als Startwerte.
- 6: **for** $k = -n$ bis n **do**
- 7: **for** $l = \max\{L, -n\}$ bis n **do**
- 8: **for** $j = 0$ bis n **do**
- 9: $\hat{f}_n^{k,l} \leftarrow \hat{f}_n^{k,l} + \sqrt{2n+1} a_{k,j,l} d_n^{j,k}(0) d_n^{j,l}(0)$
- 10: **end**
- 11: **if** f reellwertig **then**
- 12: $\hat{f}_n^{-k,-l} \leftarrow (-1)^{k+l} \overline{\hat{f}_n^{k,l}}$
- 13: **end**
- 14: **end**
- 15: **end**
- 16: **end**

Output : $\mathcal{SO}(3)$ -Fourierkoeffizientenvektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N}$ von $f \in \mathbb{D}_N$

Komplexität : $\mathcal{O}(N^4)$

4.7 Numerische Ergebnisse

Wir vergleichen die Laufzeiten der verschiedenen Wignertransformationen von $\mathcal{SO}(3)$ -Fourierkoeffizienten auf echte Fourierkoeffizienten und die daraus resultierenden NFSOFT-Algorithmen. Dazu verwenden wir die numerischen Ergebnisse von Potts, aus [24, Kapitel 4]. Die angegebenen Algorithmen werden mit Matlab R2020b auf einem 3.6GHz AMD FX™ 8150 Prozessor mit 8GB RAM und Windows (64 bit) unter Verwendung von doppelter Rechengenauigkeit getestet. Weiterhin werden die FFTW 3.0.1 [6] und die NFFT 3.5.2 [13] Bibliotheken sowie die MTEX-Toolbox [11] benötigt.

Im Folgenden generieren wir zufällige $\mathcal{SO}(3)$ -Fourierkoeffizienten aus dem komplexen Quadrat $[-\frac{1}{2}, \frac{1}{2}] + [-\frac{1}{2}, \frac{1}{2}]$ und zufällige Knoten $(\alpha, \beta, \gamma) \in [0, 2\pi) \times [0, \pi) \times [0, 2\pi)$.

Wir untersuchen vier algorithmische Vorgehensweisen für die Wignertransformation.

1. Die diskrete Wignertransformation (DWT) kombiniert die Transformation von $\mathcal{SO}(3)$ -Fourierkoeffizienten auf Chebyshevkoeffizienten aus Kapitel 4.2.1 mit dem Schritt aus Kapitel 4.2.3.
2. Die schnelle Wignertransformation (FWT) verwendet im Gegensatz zur DWT die schnelle Polynomtransformation aus Kapitel 4.2.2 zur Berechnung des Wigner-d Operators $\tilde{D}^{k,l}$. Da die verwendete DCT instabil ist, wird zusätzlich der Stabilisierungsschritt von Potts durchgeführt, siehe [26].
3. Die DBWT aus Kapitel 4.3 wird in Schritt 1 von Algorithmus 4.3 angegeben.
4. Die reellwertige DBWT ist eine Verbesserung der DBWT für den Fall, dass die auszuwertende Funktion reellwertig ist.

Die Hintereinanderausführung der Wignertransformationen mit einer NFFT liefert Algorithmen für die NFSOFT. Diese NFSOFT-Algorithmen unterscheiden sich grundlegend in der verwendeten Wignertransformation. Die anschließend durchgeführte NFFT ist annähernd gleich.

Siehe [24, Kapitel 4] für einen Vergleich von DWT und FWT.

In den folgenden Laufzeittests werden einige Parameter der implementierten Algorithmen voreingestellt. In der NFFT verwenden wir den Oversamplingfaktor $\sigma = 1.5$ und die Kaiser-Bessel-Fensterfunktionen mit Cut-off Parameter $m = 4$. In der FWT aus Kapitel 4.2 wird das Matrix-Vektor-Produkt des Wigner-d Operators $\tilde{D}^{k,l}$ aus Gleichung (4.9) mit einem gegebenen Vektor durch die FPT aus [25] durchgeführt. Dabei verwenden wir den Stabilisierungsschritt von Potts, siehe [26], und wählen einen internen Schwellwert $\kappa = 10^3$. Die Parameter m und κ regeln die Genauigkeit der NFFT und FPT, siehe [18], und [14]. Potts untersucht in [24, Kapitel 4] den Fehler der NFSOFT in Bezug auf die Wahl von m und κ .

Bemerkung 4.42. *DWT und FWT verwenden die Transformation über Chebyshev-Polynome, siehe [24], und wurden von Vollrath in der NFFT-Bibliothek [13] implementiert. Die DBWT aus Algorithmus 4.3 wird in der MTEX-Toolbox [11] zur Auswertung von $\mathcal{SO}(3)$ -Fourierreihen verwendet.*

Wir vergleichen die mit der DWT und DBWT erzeugten Fourierkoeffizienten bei exakter Rechnung, siehe [24, Kapitel 3.4].

Lemma 4.43. Seien $N \in \mathbb{N}$ und ein Vektor $\hat{\mathbf{f}} = (\hat{f}_n^{k,l})_{(n,k,l) \in \mathcal{J}_N} \in \mathbb{C}^{|\mathcal{J}_N|}$ gegeben. Dann erfüllen die Koeffizienten $(h_j^{k,l})_{k,j,l=-N}^N = \mathbf{A} \mathbf{W} \hat{\mathbf{f}}$ aus Satz 4.10 und $(\hat{g}_{k,j,l})_{k,j,l=-N}^N = \mathbf{G} \hat{\mathbf{f}}$ aus Gleichung (4.12) die Eigenschaft

$$h_j^{k,l} = i^{k-l} \hat{g}_{k,j,l}$$

für alle Indizes $k, j, l = -N, \dots, N$.

Beweis. Aus den Sätzen 4.10 und 4.12 über die NDSOFT $\mathbf{D}_{\mathcal{R}_M, \mathcal{J}_N}$ folgt

$$\mathbf{F}_{\mathcal{R}_M, N} \mathbf{A} \mathbf{W} = \mathbf{F}_{\tilde{\mathcal{R}}_M, N} \mathbf{G}.$$

Durch verschieben der Fouriermatrix $\mathbf{F}_{\tilde{\mathcal{R}}_M, N}$ erhalten wir

$$\mathbf{F}_{\tilde{\mathcal{R}}_M, N} = \mathbf{F}_{\mathcal{R}_M, N} \cdot \text{diag} \left(e^{i \pi (k-l)/2} \right)_{k,j,l=-N, \dots, N}.$$

Da die Fouriermatrix invertierbar ist, folgt die Behauptung. \square

Damit sind die Symmetrieeigenschaften aus den Lemmata 4.13 und 4.15 auf den Koeffizientenvektor $\mathbf{h} = \mathbf{A} \mathbf{W} \hat{\mathbf{f}}$ aus Kapitel 4.2 übertragbar.

Beispiel 4.44. Wir vergleichen die Laufzeiten der Wignertransformationen in Abhängigkeit von der Bandbreite.

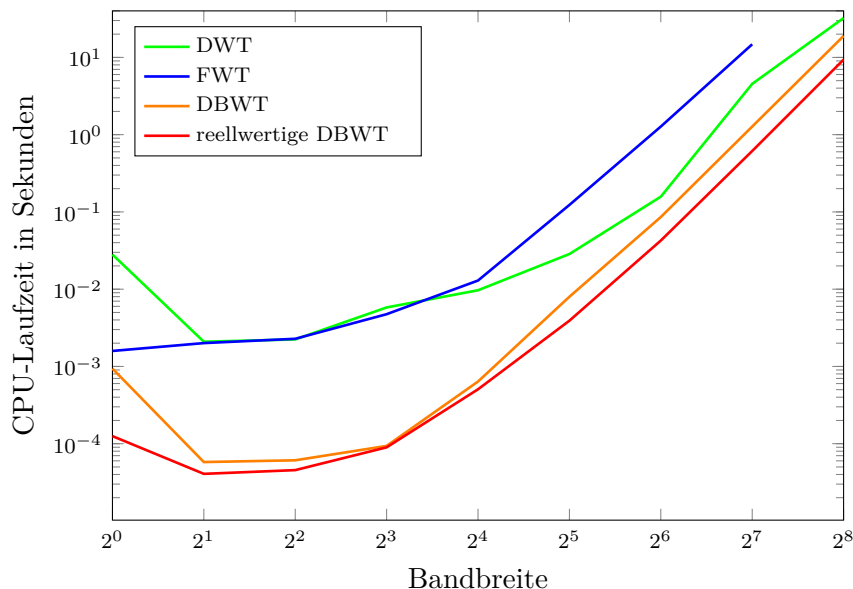


Abbildung 4.2: Die benötigte Laufzeit in Abhängigkeit von der Bandbreite, um eine Wignertransformation mit Vorberechnungen durchzuführen.

In der FWT ist die Vorberechnung der FPT sehr aufwändig. Beispielsweise kann die FWT für Bandbreiten größer als 128 nicht mehr vorberechnet werden, da der Arbeitsspeicher zu klein ist.

Wie in Abbildung 4.2 erkennbar ist, ist die DBWT eine schnellere Alternative zu den Algorithmen von Potts [24], d.h. zur FWT bzw. zur DWT. Die Laufzeit der DBWT wird noch kleiner, wenn die auszuwertende Funktion reellwertig ist.

Beispiel 4.45. In der FWT sind die Vorberechnungen der FPT sehr teuer. Bei mehrmaliger Verwendung der FWT reicht es aus, diese Vorberechnungen für eine Bandbreite N einmal durchzuführen. Bei erneuter Anwendung der FWT können die Vorberechnungen wiederverwendet werden. Deshalb vergleichen wir im Folgenden die Laufzeiten der Wignertransformationen ohne Vorberechnung.

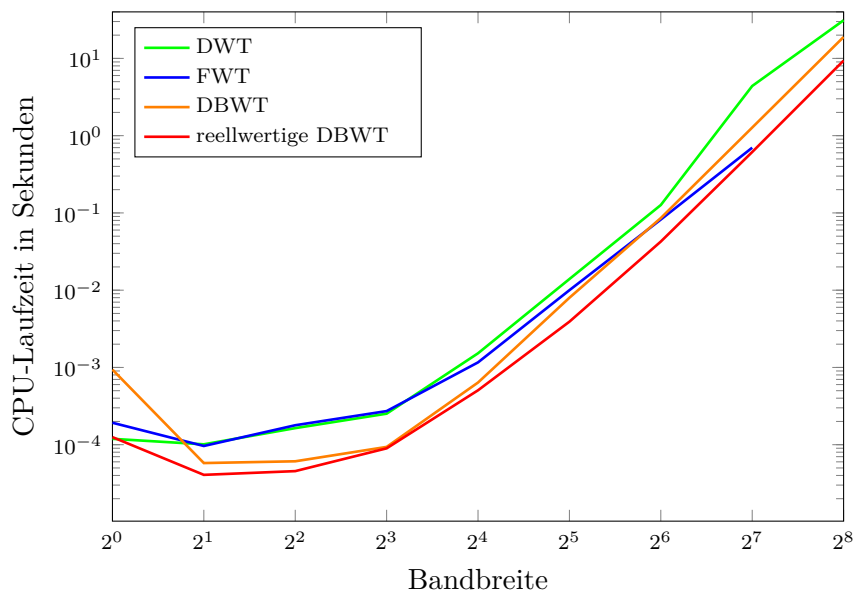


Abbildung 4.3: Die benötigte Laufzeit in Abhängigkeit von der Bandbreite, um eine Wignertransformation ohne Vorberechnungen durchzuführen.

Aufgrund der vorliegenden technischen Gegebenheiten ist die Vorberechnung der FWT für Bandbreiten größer als 128 nicht mehr durchführbar. Deshalb können wir die FWT für große Bandbreiten nicht verwenden.

Aus Abbildung 4.3 folgern wir, dass die FWT ohne Vorberechnung für Bandbreiten ab 64 schneller ist als die DBWT.

Sei die Bandbreite 128. Wir geben in Abbildung 4.4 die Gesamtlaufzeit für den mehrfachen Aufruf einer Wignertransformation an, wenn die Vorberechnung nur beim ersten Aufruf durchgeführt wird.

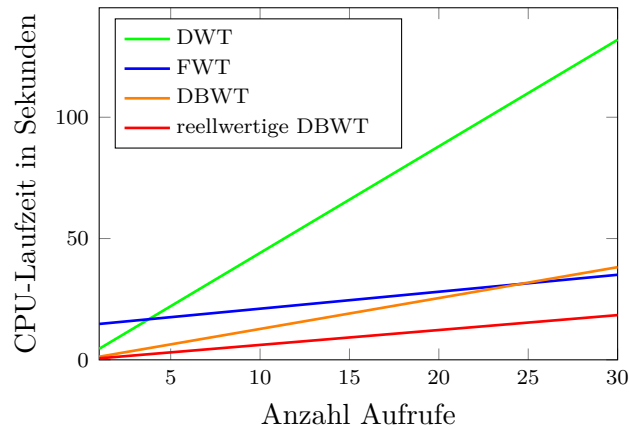


Abbildung 4.4: Die Gesamtlauzeit für mehrere Aufrufe der Wignertransformationen mit einmaliger Vorberechnung bei einer festen Bandbreite von 128.

Bei einer Bandbreite von 128 ist die FWT mit einmaliger Vorberechnung ab dem 25-ten Aufruf schneller als die DBWT. Dabei werden nur diejenigen Aufrufe gezählt, in denen die auszuwertende Funktion nicht reellwertig ist, da die reellwertige DBWT bei Bandbreite 128 genauso schnell ist wie die FWT, siehe Abbildung 4.3.

Beispiel 4.46. Als Nächstes vergleichen wir die Laufzeiten der NFSOFT-Algorithmen für verschiedene Bandbreiten N . Dazu verwenden wir zusätzlich zur Wignertransformation mit Vorberechnung eine NFFT in N^3 Knoten.

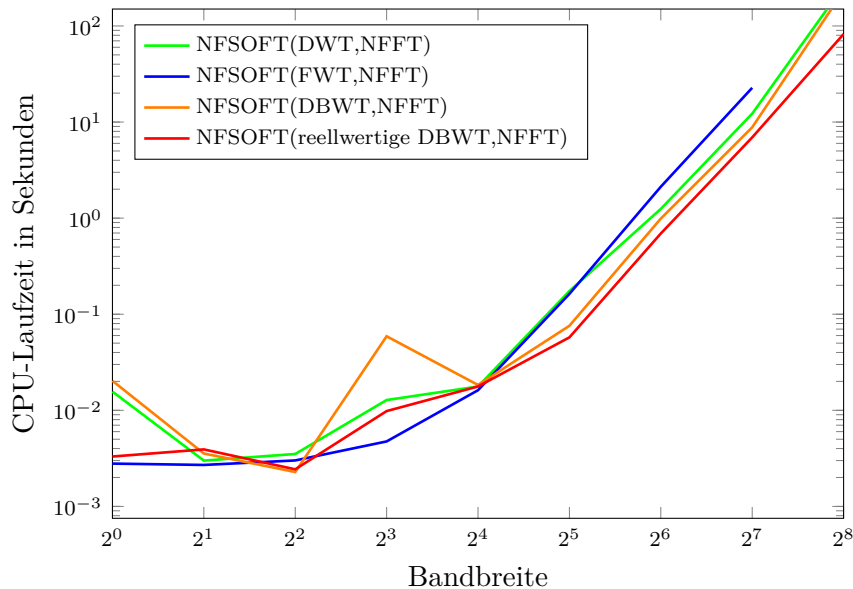


Abbildung 4.5: Die benötigte Laufzeit in Abhängigkeit von der Bandbreite N zur Berechnung der NFSOFT in N^3 Rotationen.

Die Komplexität der NFFT ist in den verschiedenen NFSOFT-Algorithmen gleich. Deshalb verursachen nur die Wignertransformationen Unterschiede in den Laufzeiten der NFSOFT-Algorithmen. Nachfolgend werden einige Laufzeiten der NFSOFT-Algorithmen mit Vorberechnung angegeben.

Algorithmus	Bandbreite 64	Bandbreite 128
NFSOFT(DWT,NFFT)	1.247	12.234
NFSOFT(FWT,NFFT)	2.121	22.719
NFSOFT(DBWT,NFFT)	0.989	8.814
NFSOFT(reellwertige DBWT,NFFT)	0.692	6.922

Tabelle 4.1: Laufzeiten in Sekunden der NFSOFT-Algorithmen bei Bandbreite 64 bzw. 128, bei der Auswertung in 64^3 bzw. 128^3 Rotationen.

Wir verwenden die NFSOFT, um eine N -bandbreitenbegrenzte Funktion in $M \in \mathbb{N}$ Rotationen auszuwerten. Dabei sind die Wignertransformationen unabhängig von der Wahl und Anzahl der Rotationen, weshalb wir die Fourierkoeffizienten immer mit gleicher Laufzeit berechnen können.

Beispiel 4.47. Wir vergleichen im Folgenden die Laufzeiten der NFSOFT-Algorithmen in Abhängigkeit von der Anzahl an Auswertungspunkten.

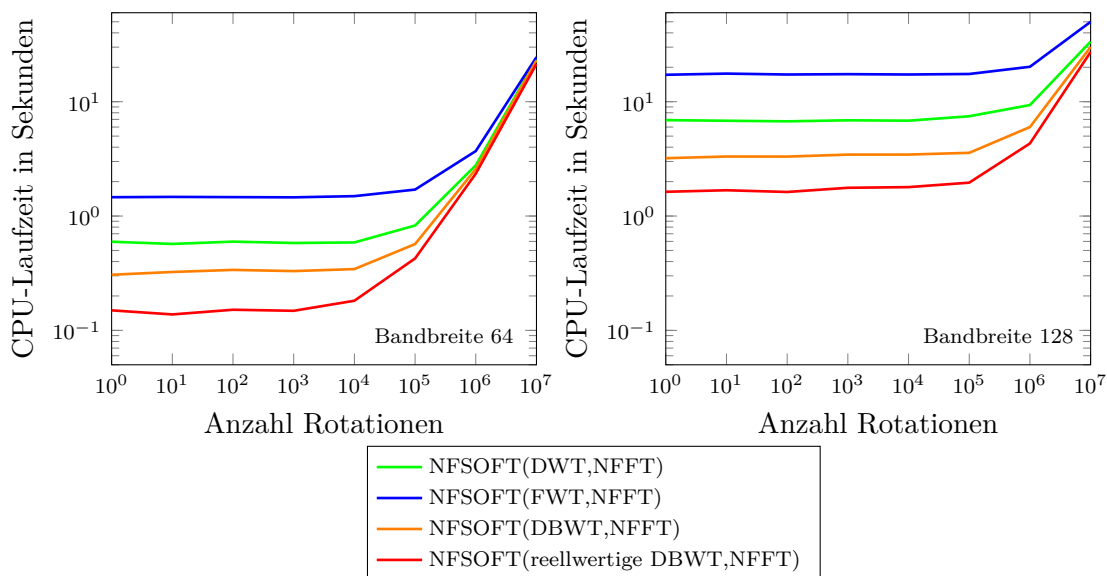


Abbildung 4.6: Die für die Berechnung der NFSOFT benötigte Laufzeit in Abhängigkeit von der Anzahl an Rotationen für die Bandbreiten $N = 64$ (links) und $N = 128$ (rechts).

Falls eine N -bandbreitenbegrenzte Funktion mit der NFSOFT in sehr vielen Rotationen ausgewertet wird, dann wird die Laufzeit der NFFT sehr groß. Dementsprechend ist die

Laufzeit nur geringfügig von der Wahl der Wignertransformation abhängig.

Wir geben einige Vorteile der DBWT aus Kapitel 4.3 gegenüber der FWT aus Kapitel 4.2 an.

Bemerkung 4.48. *Bei der FWT benötigen wir die Werte der Wigner-d Funktionen in den Chebyshevknoten. Dazu wird eine FPT mit einem Stabilisierungsschritt eingeführt, siehe [25] und [26]. Die Vorberechnung der FWT ist sehr aufwändig, siehe Beispiel 4.44. Im Gegensatz dazu werden bei der DBWT nur Wigner-d Funktionen im Punkt 0 verwendet. Dadurch können wir die 3-Term-Rekursion der Wigner-d Funktionen effizient ausnutzen und es wird keine FPT benötigt.*

In der Theorie ist die FWT mit einer approximativen Komplexität von $\mathcal{O}(N^3 \log^2 N)$ schneller als die DBWT mit $\mathcal{O}(N^4)$.

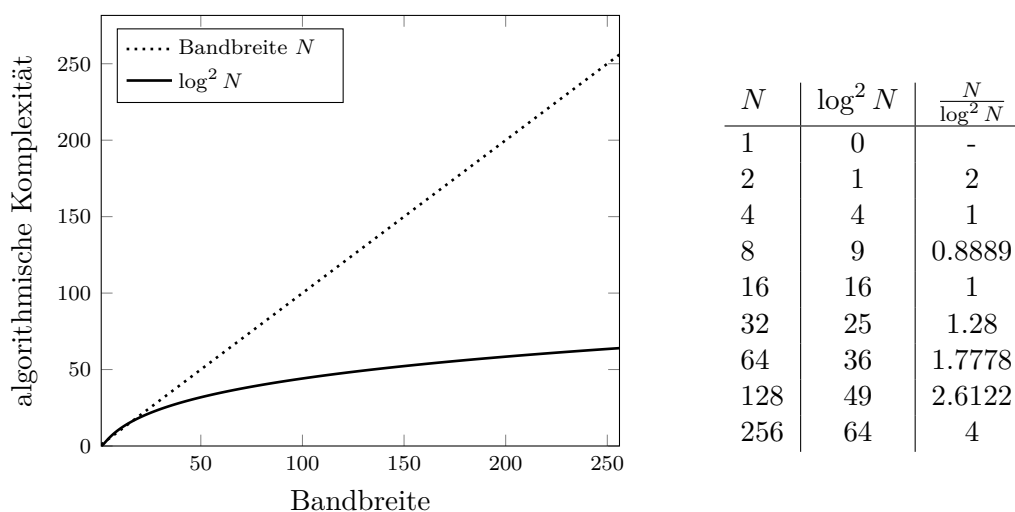


Abbildung 4.7: Vergleich der Graphen und Funktionswerte von den Funktionen $N \mapsto N$ und $N \mapsto \log^2 N$.

Für Bandbreiten kleiner gleich 128 unterscheiden sich die approximativen Komplexitäten der FWT und der DBWT um weniger als den Faktor 3. Wir können diesen Unterschied vernachlässigen.

Beispiel 4.49. In [24, Kapitel 4] wird eine Fehlerabschätzung der NFSOFT(FWT,NFFT) durchgeführt. Als Referenz werden die exakten Werte mit der NDSOFT bestimmt, welche aus der DWT und einer nichtäquidistanten diskreten Fouriertransformation (NDFT) zusammengesetzt ist. Die NDSOFT(DWT,NDFT) ist sehr langsam, aber exakt.

Wir bestimmen im Folgenden den Fehler der NFSOFT(DBWT,NFFT) aus Kapitel 4.3.

Seien eine Bandbreite $N \in \mathbb{N}$, ein zufälliger $\mathcal{SO}(3)$ -Fourierkoeffizientenvektor der Länge $|\mathcal{J}_N|$ und N^3 zufällige Rotationen gegeben. Weiterhin berechnen wir mit den Algorithmen NFSOFT(FWT,NFFT) bzw. NFSOFT(DBWT,NFFT) einen Vektor $\mathbf{f}_{NFSOFT} \in \mathbb{C}^{N^3}$. Die

NDSOFT(DWT,NDFT) liefert $\mathbf{f} \in \mathbb{C}^{N^3}$. In Abbildung 4.8 wird der Fehler

$$E_\infty = \frac{\|\mathbf{f} - \mathbf{f}_{NFSOFT}\|_\infty}{\|\mathbf{f}\|_1}$$

in Abhängigkeit von der Bandbreite dargestellt.

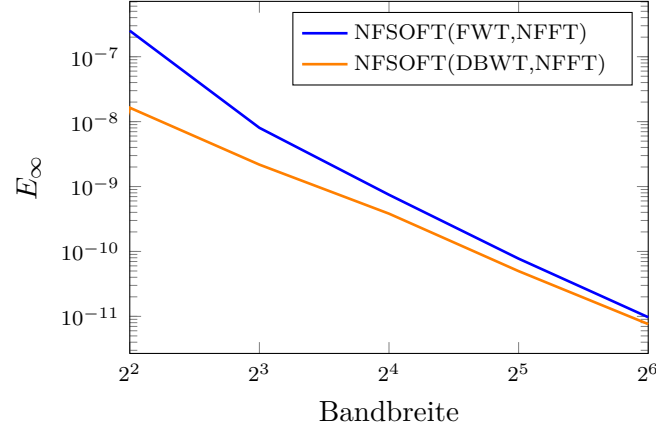


Abbildung 4.8: Genauigkeit der NFSOFT-Algorithmen in Abhängigkeit von der Bandbreite bei internen Parametern $m = 4$ und $\kappa = 1000$.

Der Algorithmus der DBWT wird um den Faktor 2 schneller, wenn die auszuwertende N -bandbreitenbegrenzte Funktion f reellwertig ist. Deshalb untersuchen wir eine weitere Eigenschaft von f , welche zusätzliche Symmetrieeigenschaften in den $\mathcal{SO}(3)$ -Fourierkoeffizienten liefert.

Definition 4.50. Eine Funktion $f: \mathcal{SO}(3) \rightarrow \mathbb{C}$ heißt antipodal, falls $f(\mathbf{R}) = f(\mathbf{R}^{-1})$ für alle Rotationen $\mathbf{R} \in \mathcal{SO}(3)$ gilt.

Lemma 4.51. Eine N -bandbreitenbegrenzte Funktion $f \in \mathbb{D}_N$ ist genau dann antipodal, wenn für die $\mathcal{SO}(3)$ -Fourierkoeffizienten

$$\hat{f}_n^{k,l} = (-1)^{k+l} \hat{f}_n^{-l,-k}$$

für alle $(n, k, l) \in \mathcal{J}_N$ gilt.

Beweis. Für Achsendrehungen gilt

$$\mathbf{R}_Z(\alpha) \mathbf{R}_Y(\beta) \mathbf{R}_Z(\gamma) \mathbf{R}_Z(-\gamma) \mathbf{R}_Y(-\beta) \mathbf{R}_Z(-\alpha) = \mathbf{I}_3.$$

Wir sehen leicht ein, dass

$$\mathbf{R}_Y(-\beta) = \mathbf{R}_Z(\pi) \mathbf{R}_Y(\beta) \mathbf{R}_Z(\pi) \tag{2.2}$$

gilt. Damit erhalten wir die Euler-Winkel-Parametrisierung einer inversen Rotation

$$\mathbf{R}(\alpha, \beta, \gamma)^{-1} = \mathbf{R}(\pi - \gamma, \beta, \pi - \alpha).$$

Für die Fourierreihendarstellung von f gilt

$$f(\mathbf{R}(\pi - \gamma, \beta, \pi - \alpha)) = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} e^{i k \gamma} e^{i l \alpha} (-1)^{k+l} d_n^{k,l}(\cos \beta).$$

Wir verwenden die Symmetrieeigenschaft (3.6) der Wigner-d Funktionen und vertauschen die Summationsreihenfolge. Daraus erhalten wir

$$f(\mathbf{R}(\pi - \gamma, \beta, \pi - \alpha)) = \sum_{n=0}^N \sum_{l,k=-n}^{-n} \hat{f}_n^{-k,-l} (-1)^{k+l} e^{-i l \alpha} e^{-i k \gamma} d_n^{l,k}(\cos \beta)$$

und es folgt die Behauptung. □

Damit können wir die algorithmische Komplexität der Wignertransformationen weiter verringern.

Bemerkung 4.52. *Die adjungierten Wignertransformationen haben die selben Laufzeiten wie die zugehörigen Wignertransformationen. Fehlerabschätzungen der adjungierten Algorithmen können bei Potts in [24, Kapitel 4] nachvollzogen werden.*

5 Fazit

In Kapitel 4.3 wird die darstellungsbasierte NFSOFT zur effizienten Auswertung der $\mathcal{SO}(3)$ -Fourierreihe

$$f(\mathbf{R}_m) = \sum_{n=0}^N \sum_{k,l=-n}^n \hat{f}_n^{k,l} D_n^{k,l}(\mathbf{R}_m), \quad m = 1, \dots, M$$

einer N -bandbreitenbegrenzten Funktion $f \in \mathbb{D}_N$ in endlich vielen Rotationen $\mathbf{R}_m \in \mathcal{SO}(3)$ angegeben. Die Idee besteht darin, die nichtäquidistante diskrete $\mathcal{SO}(3)$ -Fouriertransformation

$$\mathbf{f} = \mathbf{D}_{\mathcal{R}_m, \mathcal{J}_N} \hat{\mathbf{f}} = \mathbf{F}_{\tilde{\mathcal{R}}_M, N} \mathbf{G} \hat{\mathbf{f}}$$

in eine Fouriermatrix $\mathbf{F}_{\tilde{\mathcal{R}}_M, N}$ und die darstellungsbasierte Wignertransformation (DBWT) \mathbf{G} zu zerlegen. Der entsprechende NFSOFT-Algorithmus 4.3 ist in der MTEX-Toolbox [11] als C-Skript implementiert. Der zugehörige adjungierte Algorithmus 4.6 wird in Kapitel 4.6.2 beschrieben.

Wir sehen in den numerischen Untersuchungen aus Kapitel 4.7, dass die darstellungsbasierte NFSOFT bei gleicher Genauigkeit effizienter ist als die NFSOFT von Potts, siehe [24]. Der Algorithmus von Potts hat erst einen Vorteil, wenn viele Funktionen mit der gleichen Bandbreite ausgewertet werden und diese Bandbreite größer als 64 ist. Falls die auszuwertenden Funktionen reellwertig sind, dann ist die darstellungsbasierte NFSOFT bei Bandbreiten unter 128 schneller als die NFSOFT von Potts.

Wir geben einen Ausblick über mögliche Verbesserungen der DBWT.

Im implementierten Algorithmus werden die Wigner-d Matrizen mit Grad kleiner gleich N rekursiv bestimmt. Alternativ ist eine Vorberechnung der Wigner-d Matrizen sinnvoll. Falls die auszuwertende Funktion antipodal ist, erfüllen die $\mathcal{SO}(3)$ -Fourierkoeffizienten zusätzliche Symmetrieeigenschaften. Damit können wir analog zu Kapitel 4.3.2 den numerischen Aufwand des Algorithmus verringern. Falls zusätzliche Symmetrien auf der Rotationsgruppe gegeben sind, lässt sich der Algorithmus weiter verbessern.

Die äquidistante schnelle $\mathcal{SO}(3)$ -Fouriertransformation und die adjungierte darstellungsbasierte NFSOFT aus Algorithmus 4.6 wurden in MTEX noch nicht implementiert. Damit wollen wir zukünftig die $\mathcal{SO}(3)$ -Fourierkoeffizienten einer N -bandbreitenbegrenzten Funktion aus den Funktionswerten in diskreten Punkten berechnen.

Literaturverzeichnis

- [1] G. Baszenski und M. Tasche. „Fast Polynomial Multiplication and Convolutions Related to the Discrete Cosine Transform“. In: *Linear Algebra and its Applications* 252.1-3 (1997), S. 1–25. DOI: 10.1016/0024-3795(95)00696-6.
- [2] H. J. Bunge. *Texture Analysis in Materials Science*. Butterworth-Heinemann, 1982.
- [3] J. R. Driscoll und D. M. Healy. „Computing Fourier Transforms and Convolutions on the 2-Sphere“. In: *Advances in Applied Mathematics* 15.2 (1994), S. 202–250. ISSN: 0196-8858. DOI: 10.1006/aama.1994.1008.
- [4] S. Filippi. *Untersuchungen über die Fourier-Tschebyscheff-Approximation von Stammfunktionen*. Forschungsberichte des Landes Nordrhein-Westfalen. Springer-Verlag, 2013. DOI: 10.1007/978-3-663-07463-2.
- [5] G. Fischer. *Lineare Algebra: Eine Einführung für Studienanfänger*. 18. Aufl. Springer Spektrum, Wiesbaden, 2014. DOI: 10.1007/978-3-658-03945-5.
- [6] Matteo Frigo und Steven G. Johnson. *FFTW, C subroutine library*. 2020. URL: <http://www.fftw.org>.
- [7] J. Gallier und J. Quaintance. *Differential Geometry and Lie Groups: A Second Course*. Springer International Publishing, 2020. ISBN: 978-3-030-46047-1. DOI: 10.1007/978-3-030-46047-1_7.
- [8] G. H. Golub und J. H. Welsch. „Calculation of Gauss quadrature rules“. In: *Mathematics of computation* 23.106 (1969), S. 221–230.
- [9] M. Gräf. „Efficient algorithms for the computation of optimal quadrature points on Riemannian manifolds“. Diss. Universitätsverlag Chemnitz, 2013.
- [10] M. Gräf und D. Potts. „Sampling Sets and Quadrature Formulae on the Rotation Group“. In: *Numerical Functional Analysis and Optimization* 30.7-8 (2009), S. 665–688.
- [11] R. Hielscher u. a. *MTEX 5.7.0, Matlab Toolbox*. 2021. URL: <https://mtex-toolbox.github.io/index>.
- [12] P. Junghanns. *Orthogonale Polynome*. Leipzig: EAGLE, 2009. ISBN: 978-3-937219-28-8.
- [13] J. Keiner, S. Kunis und D. Potts. *NFFT 3.5.2, Softwarepackage, C subroutine library*. 2020. URL: <http://www.tu-chemnitz.de/~potts/nfft>.
- [14] J. Keiner und D. Potts. „Fast evaluation of quadrature formulae on the sphere“. In: *Mathematics of computation* 77.261 (2008), S. 397–419. DOI: 10.1090/S0025-5718-07-02029-7.

- [15] Z. Khalid u. a. „Gauss-Legendre Sampling on the Rotation Group“. In: *IEEE Signal Processing Letters* 23.2 (2015), S. 207–211. DOI: 10.1109/LSP.2015.2503295.
- [16] M. Kircheis und D. Potts. „Direct inversion of the nonequispaced fast Fourier transform“. In: *Linear Algebra and its Applications* 575 (2019), S. 106–140. DOI: 10.1016/j.laa.2019.03.028.
- [17] P. J. Kostelec und D. N. Rockmore. „FFTs on the Rotation Group“. In: *Journal of Fourier analysis and applications* 14.2 (2008), S. 145–179. DOI: 10.1007/s00041-008-9013-5.
- [18] S. Kunis und D. Potts. „Fast spherical Fourier algorithms“. In: *Journal of Computational and Applied Mathematics* 161.1 (2003), S. 75–98. DOI: 10.1016/S0377-0427(03)00546-6.
- [19] A. McLaren. „Optimal Numerical Integration on a Sphere“. In: *Mathematics of Computation* 17 (1963), S. 361–383. DOI: 10.1090/S0025-5718-1963-0159418-2.
- [20] A. Morawiec. *Orientations and rotations: Computations in crystallographic textures*. Springer, 2004. DOI: 10.1007/978-3-662-09156-2.
- [21] H. O’hara und F. J. Smith. „Error estimation in the Clenshaw–Curtis quadrature formula“. In: *The Computer Journal* 11.2 (1968), S. 213–219.
- [22] E. Ossa. *Topologie*. Wiesbaden: Vieweg+Teubner Verlag, 1992. ISBN: 978-3-322-96891-3. DOI: 10.1007/978-3-322-96891-3_4.
- [23] G. Plonka u. a. *Numerical Fourier Analysis*. Springer International Publishing, 2018. ISBN: 978-3-030-04306-3. DOI: 10.1007/978-3-030-04306-3_1.
- [24] D. Potts, J. Prestin und A. Vollrath. „A fast algorithm for nonequispaced Fourier transforms on the rotation group“. In: *Numerical Algorithms* 52.3 (2009), S. 355. DOI: 10.1007/s11075-009-9277-0.
- [25] D. Potts, G. Steidl und M. Tasche. „Fast algorithms for discrete polynomial transforms“. In: *Mathematics of Computation* 67.224 (1998), S. 1577–1590.
- [26] D. Potts, G. Steidl und M. Tasche. „Fast and stable algorithms for discrete spherical Fourier transforms“. In: *Linear Algebra and its Applications* 275 (1998), S. 433–450. DOI: 10.1016/S0024-3795(97)10013-1.
- [27] D. Potts, G. Steidl und M. Tasche. „Fast Fourier transforms for nonequispaced data: A tutorial“. In: *J.J. Benedetto and P.J.S.G. Ferreira, editors, Modern Sampling Theory. Applied and Numerical Harmonic Analysis*. (2001). DOI: 10.1007/978-1-4612-0143-4_12.
- [28] D. Potts, G. Steidl und M. Tasche. „Numerical Stability of Fast Trigonometric Transforms - A Worst Case Study“. In: *J. Concrete Appl. Math* 1 (2003), S. 1–36.
- [29] W. H. Press u. a. *Numerical Recipes in C*. 2. Aufl. Cambridge: Cambridge University Press, 1992.

- [30] G. Prezeau und M. Reinecke. „Algorithm for the evaluation of reduced Wigner matrices“. In: *The Astrophysical Journal Supplement Series* 190.2 (2010), S. 267–274. DOI: 10.1088/0067-0049/190/2/267.
- [31] T. Risbo. „Fourier transform summation of Legendre series and D-functions“. In: *Journal of Geodesy* 70.7 (1996), S. 383–396. DOI: 10.1007/BF01090814.
- [32] S. Scherer. *Symmetrien und Gruppen in der Teilchenphysik*. Springer Berlin Heidelberg, 2016. ISBN: 978-3-662-47734-2. DOI: 10.1007/978-3-662-47734-2_3.
- [33] G. Szegő. *Orthogonal Polynomials*. American Math. Soc: Colloquium publ 23. American Mathematical Society, 1939. ISBN: 9780821810231.
- [34] L. N. Trefethen. „Is Gauss quadrature better than Clenshaw–Curtis?“. In: *SIAM review* 50.1 (2008), S. 67–87. DOI: 10.1137/060659831.
- [35] D. A. Varshalovich, A. N. Moskalev und V. K. Khersonskii. *Quantum Theory of Angular Momentum*. WORLD SCIENTIFIC, 1988. DOI: 10.1142/0270. eprint: <https://www.worldscientific.com/doi/pdf/10.1142/0270>.
- [36] N. Ja. Vilenkin und A. U. Klimyk. *Representation of Lie Groups and Special Functions: Volume 1: Simplest Lie Groups, Special Functions and Integral Transforms*. Springer Netherlands, 1991. ISBN: 978-94-011-3538-2. DOI: 10.1007/978-94-011-3538-2_3.
- [37] A. Vollrath. „The Nonequispaced Fast SO(3) Fourier Transform, Generalisations and Applications“. Diss. Zentrale Hochschulbibliothek Lübeck, 2010.
- [38] J. Werner. *Numerische Mathematik 1: Lineare und nichtlineare Gleichungssysteme, Interpolation, numerische Integration*. 1. Aufl. 1992. DOI: 10.1007/978-3-663-07747-3.
- [39] H. Weyl. *The theory of groups and quantum mechanics*. Dover Publications, 1931.
- [40] P. Woit. *Quantum Theory, Groups and Representations*. Springer, 2017. DOI: 10.1007/978-3-319-64612-1.