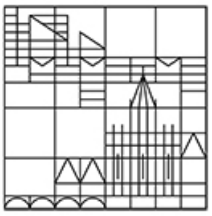
Numerik von Maximum Entropie Momentenproblemen in der Texturanalyse

Dissertation

zur Erlangung des akademischen Grades
eines **Doktors der Naturwissenschaften**
(Doctor rerum naturalium, Dr. rer. nat.)

vorgelegt von **Johannes Budday** an der

Universität Konstanz



Mathematisch-Naturwissenschaftliche Sektion
Fachbereich Mathematik und Statistik

Tag der mündlichen Prüfung: 24. Juli 2014

Referenten:
Prof. Dr. Michael Junk, Universität Konstanz
Prof. Dr. Markus Schweighofer, Universität Konstanz

Danksagung

Etwa zeitgleich zu meinem Diplomabschluss zum Thema *Numerische Behandlung der Rotation starrer Körper* an der Universität Konstanz initiierte Herr Prof. Dr. Michael Junk ein Projekt zum Thema *Maximum Entropie Momentenprobleme* in Kooperation mit Herrn Prof. Dr. Thomas Böhlke vom Karlsruher Institut für Technologie. Im Rahmen dieses Projekts entstand die Idee für die Themengebung dieser Dissertation.

Für das Angebot von Herrn Prof. Dr. Michael Junk mich im Rahmen einer Dissertation in dieses Projekt einzubinden, möchte ich mich sehr herzlich bedanken. Auch für die herzliche Aufnahme und Eingliederung in die Arbeitsgruppe, die sehr angenehme Arbeitsatmosphäre, die zahlreichen Diskussionen und unterhaltsamen Gespräche, und die immerwährende Bereitschaft mich bei meiner Arbeit zu unterstützen, möchte ich mich besonders bei Herrn Prof. Dr. Michael Junk aber auch bei allen weiteren Mitgliedern unserer Arbeitsgruppe sehr herzlich bedanken. Diese Zeit und die gemeinsame Arbeit in Forschung und Lehre wird mir immer in sehr angenehmer und freudiger Erinnerung bleiben.

Desweiteren möchte ich mich beim Fachbereich Mathematik & Statistik der Universität Konstanz bedanken, der es mir während der gesamten Zeit meiner Promotion durch die Finanzierung einer Stelle ermöglichte, diese Arbeit überhaupt aufnehmen zu können. Ein ganz besonderer Dank geht hier an Herrn Rainer Janßen, Frau Gisela Cassola und Frau Waltraud Pfeiffer, die bei allen Fragen und Bitten stets alles dafür taten, um mir für meine Arbeit im Fachbereich durch ihre Unterstützung und ihr Entgegenkommen die bestmöglichen Arbeitsbedingungen zu garantieren.

Ein sehr großer Dank geht an meine Eltern Erika und Jürgen, die mir es durch jahrelange Unterstützung erst ermöglichten, das Studium der Mathematik erfolgreich durchzuführen und mir somit verhalfen, die Grundlagen für diese Promotion zu schaffen. Ohne diese wertvolle Hilfe wäre das alles nicht möglich gewesen. Zuletzt geht ein ebenso großer Dank an meine engsten Freunde Johannes, Teresa und Matthias, meine Geschwister Mirjam und Marcus, meine liebe Freundin Christin und deren Eltern Christa und Herbert, welche mich alle immer wieder sehr lieb unterstützt haben, und mir auch in schwierigen Zeiten sehr wertvolle Stützen waren.

Vielen herzlichen Dank!

Konstanz, im Juni 2014

Inhaltsverzeichnis

1 Einleitung

Die vorliegende Dissertation beschäftigt sich mit einer Problemstellung, die unter anderem in der Materialforschung, spezieller im Fachgebiet der polykristallinen Metalle, häufig auftaucht. Polykristalline Metalle wie etwa Stahl oder Aluminiumlegierungen weisen ein sehr komplexes anisotropes (d.h. richtungsabhängiges) elastisches Verhalten auf, welches durch die kristalline Mikrostruktur des Metalls bestimmt wird. Möchte man zum Beispiel den Verformungsvorgang eines solchen polykristallinen Metalls optimieren bzw. numerisch simulieren (man denke dabei zum Beispiel an die Herstellung verschiedenster Fahrzeugteile in der Automobilindustrie), so wird es demnach von großer Bedeutung sein, diese Mikrostruktur genauestens zu kennen bzw. sie mathematisch beschreiben zu können. Betrachtet man diese innere Struktur etwas genauer, etwa mit Hilfe von Kristallschnittbildern des Metalls, so stellt man fest, dass das Metall aus lauter Einkristallen aufgebaut ist. Ein Einkristall zeichnet sich dadurch aus, dass alle Elementarzellen des Einkristalls identisch ausgerichtet sind. Die Ausrichtung eines Einkristalls kann also durch Angabe einer für den Einkristall charakteristischen Orientierung beschrieben werden, die bezüglich einer für alle Einkristalle fix gewählten Referenzelementarzelle angegeben wird. Mathematisch geschieht dies mit Hilfe der $SO(3)$-Rotationsmatrizen. Möchte man nun wissen, wieviele Kristalle einer zu untersuchenden Metallprobe die Orientierung $Q \in SO(3)$ haben, so liefert die Größe $f(Q)dQ$ in einer Umgebung von Q eine Antwort darauf, wobei mit f die sogenannte Kristallorientierungsverteilungsfunktion (kurz: **codf** - *crystalline orientation distribution function*) gemeint ist. Zahlreiche theoretische und numerische Studien belegen die Wichtigkeit der codf für eine präzise Voraussage des Verhaltens eines Materials. Deshalb gilt die codf in der Materialwissenschaft als Hauptrepräsentant der polykristallinen Mikrostruktur.[24] Eine Bestimmungsgleichung für die codf ist gegeben durch ein **allgemeines Momentenproblem**:

Finde eine positive Dichte $f \geq 0$ bezüglich einem Maß μ, die zu gegebenem $n \in \mathbb{N}$ bezüglich der Momentenfunktionen a_i und der Momente b_i folgende Bedingungen erfüllt:

$$\int_\Omega a_i f \, d\mu \;=\; b_i \quad \textit{für} \quad i = 1, \ldots, n \tag{1-1}$$

Dabei betrachten wir im Folgenden eine "natürliche" Klasse von Momentenproblemen:

$$\begin{aligned} \Omega \subset G: &\quad G \text{ lokalkompakte topologische Gruppe} \\ \mu: &\quad \text{Haar-Maß auf } G \\ a_i: &\quad \text{Darstellungsfunktionen auf } G \\ b_i: &\quad \text{Momente} \\ &\quad \text{(z.B. Messwerte, d.h. gegebene reelle Zahlen)} \end{aligned}$$

Warum diese Klasse als "natürlich" bezeichnet wird, wird im Folgenden anhand von Beispielen erläutert werden. Das Eingangsbeispiel mit den Kristallorientierungen legt nahe, warum man hierbei die Klasse von Momentenproblemen auf einer Gruppe betrachtet, denn durch diese Vorgehensweise wird man das Eingangsbeispiel für $G = SO(3)$ einfach in den Kontext einbinden können. Um jedoch über eine Gruppe integrieren zu können, wird ein entsprechendes Maß auf der Gruppe benötigt, welches das sogenannte Haar-Maß sein wird. Dabei handelt es sich um eine Verallgemeinerung des allseits bekannten Lebesgue-Maßes λ. Die Eigenschaft der Translationsinvarianz des Lebesgue-Maßes, d.h. dass für beliebige Borel-messbare Mengen $A \subset \mathbb{R}^m$ und für jedes $x \in \mathbb{R}^m$ die Bedingung $\lambda(A+x) = \lambda(A)$ gilt, wird auf Gruppenebene durch die sogenannte *Linksinvarianz* ersetzt. Ein Maß μ bezeichnet man dabei als *linksinvariant*, wenn für jede Borel-Menge B und jedes Gruppenelement g die Bedingung $\mu(gB) = \mu(B)$ gilt. Somit wird das Haar-Maß wie folgt definiert:[16]

1.1 Definition. Das linke **Haar-Maß** einer lokalkompakten topologischen Gruppe G ist das bis auf einen Faktor eindeutig bestimmte **linksinvariante** reguläre Borel-Maß, das auf nichtleeren offenen Teilmengen positiv ist.

Das rechte Haar-Maß erhält man analog und ist ebenso bis auf einen Faktor eindeutig bestimmt. Stimmen rechtes und linkes Haar-Maß überein, so nennt man die Gruppe *unimodular*. Im Falle von lokalkompakten abelschen Gruppen und im Falle von kompakten Gruppen ist dies gegeben. Desweiteren ist das Haar-Maß einer lokalkompakten Gruppe genau dann endlich, wenn die Gruppe kompakt ist.

1.2 Beispiel. Betrachten wir die additive Gruppe $(\mathbb{R}^m, +)$, so ist das Haar-Maß gerade durch das Lebesgue-Maß auf dem $\mathbb{R}^m$ gegeben.

Die sogenannten **Momente** b_i sind zum Beispiel Messwerte. Beim eingangs erwähnten Beispiel der codf werden diese Messwerte als Informationen über Kristallorientierungen über einem zwei- oder dreidimensionalen räumlichen Gitter experimentell bestimmt, etwa über die Methode der Elektronenbeugung am Kristall.[17,24,32] Diese Vorgehensweise liefert folglich eine Punktmaß-Approximation der Orientierungsverteilung, da nur diskrete Orientierungen detektiert werden. Da für eine qualitative

Auswertung und Interpretation der Materialtextur jedoch eine glatte Darstellung der kompletten codf von großer Bedeutung ist, muss eine solche glatte Darstellung rekonstruiert werden. Dies gelingt mit Hilfe von Rekonstruktionsalgorithmen, wie etwa der Maximum Entropie Methode, auf welche später noch genauer eingegangen wird, indem man die Messwerte in Form von gewichteten Mittelwerten der detektierten Orientierungen verwendet.[24,35,36]

Eine sinnvolle Wahl der Mittelwerte, den sogenannten *tensoriellen Texturkoeffizienten*, wurde in der Texturanalysis durch Adams, Boehler, Guidi und Onat eingeführt[2,20] und wird im Laufe dieser Arbeit noch von Bedeutung sein. Sie ermöglichen eine koordinatenfreie Darstellung kristallographischer Informationen, eine Eigenschaft, welche in vorherigen (nicht-tensoriellen) Darstellungen nicht vorliegt.[13,14] In einer koordinatenfreien Formulierung können Resultate über anisotrope Tensorfunktionen[39,40] und homogenisierte Gleichungen[8] jedoch in einer sehr effizienten und kompakten Form formuliert werden.[24]

Es zeigt sich, dass das Problem der Rekonstruktion einer nicht-negativen codf aus gegebenen tensoriellen Texturkoeffizienten nicht nur im Bereich der Datenvisualisierung, sondern auch im Modellierungsprozess von Texturevolutionen auftaucht. Tatsächlich kann man unter Verwendung des codf-Rekonstruktionsproblems eine Evolutionsgleichung für die führenden Texturkoeffizienten herleiten. Die Ergebnisse liefern sinnvolle Voraussagen der Texturen und reproduzieren dabei die experimentell beobachteten stationären, verformten Texturen.[6,24]

Um den Begriff der in der Klasse der Momentenprobleme erwähnten **Darstellungsfunktion auf einer Gruppe** genauer zu erläutern, betrachten wir die folgende Definition[12]:

1.3 Definition. Sei G eine lokalkompakte Gruppe und V ein topologischer Vektorraum, dann heißt eine stetige Abbildung $\varrho : G \times V \longrightarrow V$ mit der Eigenschaft, dass $\varrho_g : v \longmapsto \varrho_g v$ linear ist und $\varrho_{gh} = \varrho_g \varrho_h$, eine **Darstellung von** G **auf** V. Bei Wahl eines $v \in V$ und einem linearen Funktional $\alpha \in V^*$ aus dem zugehörigen Dualraum V^* bezeichnen wir die Abbildung $g \longmapsto \alpha(\varrho_g v)$ als **Darstellungsfunktion auf** G.

Im Falle von $G = (\mathbb{R}, +)$ und $V = \mathbb{R}^m$ erhalten wir für ein beliebiges $A \in \mathbb{R}^{m\times m}$ mit $\varrho_g = \exp(gA)$ mögliche Beispiele solcher Darstellungen[1]. Wählt man nun in $\mathbb{R}^m$ die Standardbasis, so sind die Darstellungsfunktionen $g \longmapsto e_i^T(\varrho_g e_j)$ gerade die Matrixeinträge der Matrix ϱ_g.

Im Folgenden wollen wir anhand dreier Beispiele mögliche auftretende Darstellungsfunktionen und ihre Eigenschaften betrachten, um damit die Natürlichkeit der gewählten Klasse an Momentenprobleme zu unterstreichen:

[1] man beachte, dass aufgrund der Funktionalgleichung der matrixwertigen Exponentialfunktion für kommutative Matrizen die Bedingung $\varrho_{g+h} = \exp\big((g+h)A\big) = \exp(gA)\exp(hA) = \varrho_g\varrho_h$ erfüllt ist

1.4 Beispiel. $G = (\mathbb{R}, +)$, $V = \mathbb{R}^m$

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

Berechnen wir die zugehörige Matrixexponentialfunktion $\exp(tA)$ zu dieser nilpotenten Matrix A, so erhalten wir für die darstellende Matrix der Darstellung

$$\varrho_t = \exp(tA) = \begin{pmatrix} 1 & t & \frac{1}{2}t^2 & \cdots & \frac{1}{(m-1)!}t^{m-1} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \frac{1}{2}t^2 \\ \vdots & & \ddots & \ddots & t \\ 0 & \cdots & \cdots & 0 & 1 \end{pmatrix}.$$

Die auftretenden Darstellungsfunktionen sind demnach vom Typ $1, t, t^2, \ldots, t^{m-1}$, d.h. Monome bis zum Grad $m-1$. Eine wichtige Eigenschaft dieser Darstellungsfunktionen ist die **Translationsinvarianz** bezüglich der Gruppenoperation. Betrachtet man ein Polynom p, das durch diese Monome erzeugt wird, d.h. dass p sich mit gewissen Koeffizienten $\alpha_i \in \mathbb{R}$ in der Form $p(t) = \sum_{i=0}^{m-1} \alpha_i t^i$ darstellen lässt, so gilt für ein beliebiges $h \in \mathbb{R}$, dass sich auch das Polynom q mit $q(t) := p(t+h)$ mit gewissen Koeffizienten β_i in der Form $q(t) = \sum_{i=0}^{m-1} \beta_i t^i$ darstellen lässt, was sich durch simples Ausmultiplizieren nachrechnen lässt. Diese Translationsinvarianz wird später von Interesse sein.

Insgesamt erhalten wir mit (1-1) für $\Omega = \mathbb{R}$ die bekannten statistischen Momente

$$\int_{-\infty}^{\infty} t^k f_X(t)\, dt = E(X^k),$$

wobei E hier den Erwartungswert der k-ten Potenz einer stetigen Zufallsvariable X beschreibt und f_X die zu X zugehörige Dichtefunktion bezeichnet.

1.5 Beispiel. $G = (\mathbb{R}, +)$, $V = \mathbb{R}^2$

$$A = \begin{pmatrix} 0 & 1 \\ -(n\omega)^2 & 0 \end{pmatrix} \quad \omega > 0\ ,\ n \in \mathbb{N}$$

Für die darstellende Matrix der Darstellung erhalten wir in diesem Fall

$$\varrho_t = \exp(tA) = \begin{pmatrix} \cos(n\omega t) & \frac{1}{n\omega}\sin(n\omega t) \\ -n\omega \sin(n\omega t) & \cos(n\omega t) \end{pmatrix}.$$

Die auftretenden Darstellungsfunktionen sind hier vom Typ $\sin(n\omega t), \cos(n\omega t)$ und haben ebenso die Eigenschaft der Translationsinvarianz bezüglich der Gruppenoperation, wie man unter Verwendung bestimmter Additionstheoreme für trigonometrische Funktionen nachrechnen kann.

Für ein $T > 0$ liefern die Momente in diesem Fall für $\Omega = [0, T]$ die bis auf einen Faktor bestimmten FOURIER-Koeffizienten

$$\int_0^T \sin(n\omega t) f(t)\ dt \quad \text{und} \quad \int_0^T \cos(n\omega t) f(t)\ dt$$

der FOURIER-Entwicklung der Dichtefunktion f auf dem Intervall $[0, T]$.

1.6 Beispiel. $G = (\mathbb{R}, +)$, $V = \mathbb{R}^2$

$$A = \begin{pmatrix} -s_1 & 0 \\ 0 & -s_2 \end{pmatrix} \quad s_1, s_2 \in \mathbb{R} \setminus \{0\}$$

In diesem Fall erhalten wir für die darstellende Matrix der Darstellung trivialerweise

$$\varrho_t = \exp(tA) = \begin{pmatrix} e^{-s_1 t} & 0 \\ 0 & e^{-s_2 t} \end{pmatrix}.$$

Auch hier haben die auftretenden Darstellungsfunktionen vom Typ $e^{-s_1 t}, e^{-s_2 t}$ die Eigenschaft der Translationsinvarianz, wie man mit Hilfe der Funktionalgleichung der Exponentialfunktion nachrechnen kann.

Die Momente liefern in diesem Fall für $\Omega = \mathbb{R}_{\geq 0}$ Funktionswerte der LAPLACE-Transformierten von f an den Stellen s_1 und s_2, d.h.

$$\int_0^\infty e^{-s_1 t} f(t)\ dt \quad \text{und} \quad \int_0^\infty e^{-s_2 t} f(t)\ dt\ .$$

Zusammenfassend sollen diese drei Beispiele aufzeigen, weshalb die eingeführte Klasse von Momentenproblemen als natürlich bezeichnet wird. Aufgrund der auftretenden Darstellungsfunktionen wie Monome, trigonometrischer Funktionen und der Exponentialfunktion scheint diese Klassifizierung naheliegend.

Die beobachtete Eigenschaften dieser Beispiele lässt sich in einem allgemeineren Kontext formulieren:

1.7 Bemerkung. Jede Darstellung einer lokalkompakten Gruppe auf einem endlichdimensionalen Vektorraum liefert einen translationsinvarianten Unterraum, welcher durch die Darstellungsfunktionen erzeugt wird.

Im Falle von kompakten Gruppen G liefert der Satz von PETER und WEYL sogar eine Zerlegung des Raumes $\mathbb{L}^2(G)$ von der Form

$$\mathbb{L}^2(G) = U_1 \oplus U_2 \oplus U_3 \oplus \ldots \tag{1-2}$$

in translationsinvariante Unterräume U_i von $\mathbb{L}^2(G)$, die durch Darstellungsfunktionen erzeugt werden. Dieses Resultat liefert eine Verallgemeinerung der FOURIER-Entwicklung einer Funktion $f \in \mathbb{L}^2(G)$ nach Darstellungsfunktionen auf kompakten Gruppen in der Form $f = \sum_{i=1}^{\infty} f_i$, wobei $f_i \in U_i$ gilt und jedes f_i somit eine Linearkombination aus Darstellungsfunktionen ist. Für die im Fall $G = SO(3)$ eingangs erwähnte codf existiert eine solche FOURIER-Entwicklung, die es im weiteren Verlauf dieser Arbeit aus dem Momentenproblem zu rekonstruieren gilt. In welcher Form dies gelingt, wird im Folgenden noch genauer betrachtet werden.

1.8 Beispiel. Im Falle der kompakten Gruppe $G = ([0,T], +_{\mathrm{mod}T})$ liefert diese Zerlegung für eine periodische Funktion $f \in \mathbb{L}^2(G)$ mit Periode $T > 0$ und der Grundfrequenz $\omega = \frac{2\pi}{T}$ die klassische FOURIER-Reihe

$$f(t) = \frac{c_0}{2} + \sum_{k=1}^{\infty} \big(c_k \cos(k\omega t) + d_k \sin(k\omega t)\big)$$

mit den FOURIER-Koeffizienten $c_k \in \mathbb{R}$ für $k \in \mathbb{N}_0$ und $d_k \in \mathbb{R}$ für $k \in \mathbb{N}$. Hieran kann man direkt ablesen, dass der Unterraum U_1 eindimensional und die Unterräume U_i für $i \in \mathbb{N}_{\geq 2}$ jeweils zweidimensional sind.

Im Folgenden behandeln wir die **Lösbarkeit** des Momentenproblems (1-1), wobei wir zunächst die Positivitätsbedingung $f \geq 0$ der Dichte weglassen und uns lediglich um die Erfüllung der Nebenbedingungen

$$\int_\Omega a_i f \; d\mu \;=\; b_i \quad \text{für} \quad i = 1, \ldots, n$$

integraler Form kümmern. Wenn dieses abgeänderte Problem lösbar ist, so werden im Allgemeinen unendlich viele Lösungen f existieren, wie man etwa für $n = 2$ am Beispiel $G = (\mathbb{R}, +)$, $\Omega = [-1, 1]$, $a_1 = b_1 = 1$, $a_2(x) = x$ und $b_2 = 0$ einfach nachvollziehen kann. Denn in diesem Beispiel erfüllen unter anderem alle Polynome der Form $f(x) = cx^2 + d$ die gegebenen Nebenbedingungen, solange die Koeffizienten die Bedingung $\frac{2}{3}c + 2d = 1$ erfüllen, für welche es unendlich viele Lösungen gibt. Selbst unter Berücksichtigung der Bedingung $f \geq 0$ an die gesuchte Dichte gibt es unendlich viele Kombinationen der Koeffizienten c und d, die zur Positivität von f führen. Um diese Problematik in den Griff zu bekommen, gibt es Methoden, welche aus diesen unendlich vielen Lösungen nach bestimmten Kriterien eine einzelne herausfiltern.

Die sogenannte GALERKIN-Methode[24] beruht auf der Wahl eines endlich dimensionalen Unterraumes von $\mathbb{L}^2_{loc}(G)$ wie zum Beispiel der linearen Hülle

$$V := \mathrm{span}(a_1, \ldots, a_n)$$

der Momentenfunktionen a_i, um über diesem Unterraum die Lösung des Problems

$$\min_{f \in V} \; \frac{1}{2}\|f\|^2 \quad , \quad \text{so dass} \quad \int_\Omega a_i f \; d\mu = b_i \;\; \text{für} \;\; i = 1, \ldots, n \tag{1-3}$$

zu berechnen. Durch Heranziehen des Formalismus von LAGRANGE erhält man die Lösung $f = \sum_{i=1}^{n} \lambda_i a_i$, wobei $\lambda \in \mathbb{R}^n$ die Lösung des linearen Gleichungssystems $M\lambda = b$ mit der symmetrischen Koeffizientenmatrix

$$M_{ij} = \int_\Omega a_i a_j \, d\mu$$

und der rechten Seite b, der vektoriellen Zusammenfassung der Momente b_i, ist. Man erhält also genau dann die eindeutige Lösbarkeit des Problems, wenn die Matrix M invertierbar ist, sprich die Momentenfunktionen a_i linear unabhängig sind.

Das folgende Beispiel, das auch unter dem Namen GIBBS'sches Phänomen bekannt ist, zeigt jedoch einen entscheidenden Nachteil dieser Methode für das Behandeln des ursprünglichen Momentenproblems auf. Dabei versuchen wir einen Ausschnitt eines positiven, periodischen Rechtecksignals mit Hilfe des folgenden Momentenproblems zu rekonstruieren:

1.9 Beispiel. $G = (\mathbb{R}, +)$, $\Omega = [-\pi, \pi]$, $n \in \mathbb{N}$,
$a_1(x) = 1$,
$a_{k+1}(x) = \sin(kx)$ für $k = 1, \ldots, n$,
$a_{k+1+n}(x) = \cos(kx)$ für $k = 1, \ldots, n$,
$f(x) = \mathbb{1}_R(x)$ mit $R = [-2.5, -1.5] \cup [-0.5, 0.5] \cup [1.5, 2.5]$

Berechnen wir nun ausgehend von diesem Rechtecksignal die Momente bzw. die FOURIER-Koeffizienten mit Hilfe der gegebenen Momentenfunktionen nach

$$b_i = \int_{-\pi}^{\pi} a_i(x) f(x) \, dx \quad \text{für} \quad i = 1, \ldots, 2n+1 \ ,$$

so können wir mit Hilfe der GALERKIN-Methode versuchen, die positive Dichte f durch das Lösen der daraus resultierenden Momentenbedingungen zu rekonstruieren. Dadurch versuchen wir also, das Rechtecksignal mit Hilfe von endlich vielen Schwingungen verschiedener Frequenzen zu approximieren. Berechnen wir die in diesem Falle diagonale Koeffizientenmatrix M und lösen sodann das Gleichungssystem $M\lambda = b$, so erhalten wir die rekonstruierte Dichte $\tilde{f}$ in der Form $\tilde{f} = \sum_{i=1}^{2n+1} \lambda_i a_i$. In den folgenden drei Grafiken sind für $n = 20, 50, 200$ (von oben nach unten) jeweils die positive Ausgangsdichte f und die rekonstruierte Dichte $\tilde{f}$ zu sehen. Obwohl wir zur Generierung der Daten eine positive Dichte herangezogen haben, wird es uns mit dieser Methode nicht gelingen, eine positive Dichte zu rekonstruieren. Unabhängig von der Wahl für $n \in \mathbb{N}$ wird die rekonstruierte Dichte für jeden endlichen Wert die in der Abbildung sichtbaren Unterschwingungen aufweisen. Dies ist im Allgemeinen immer zu erwarten, wenn man versucht, eine positive Dichte, gegeben in der Form einer FOURIER-Reihe, durch eine endliche Anzahl an Summanden der Reihe zu approximieren.

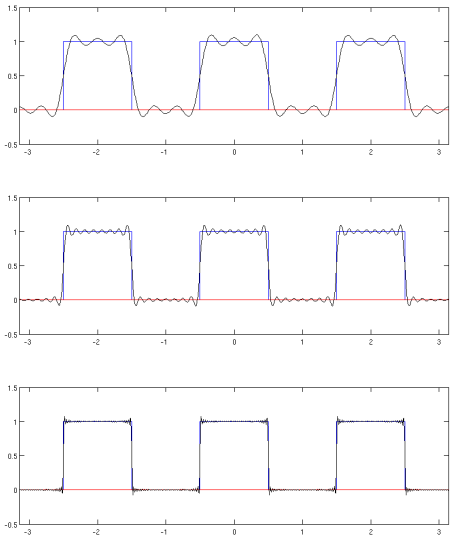


Abbildung 1.1: Positivitätsverletzung bei Galerkin-Methode

Da wir aber generell daran interessiert sind, eine Verteilungsfunktion zu rekonstruieren, ist die Positivität eine nicht zu vernachlässigende Eigenschaft. Daher wählen wir zur Rekonstruktion einen anderen Ansatz[24], die sogenannte **Maximum Entropie Methode**, welche von Jaynes im Bereich der statistischen Mechanik vorgestellt wurde.[22,23] Allgemeine Maximum Entropie Probleme wurden bereits in der Vergangenheit ausführlich bearbeitet.[10] Um einen Überblick über diese Methode zu gewinnen, sei auf das Buch von Wu[37] verwiesen, für Anwendungen im Kontext von Verteilungsstatistiken auf die Arbeiten von Mardia und Jupp[29] und Schaeben[31].

Die grundlegende Idee dieser Methode basiert im Vergleich zur Galerkin-Methode darauf, die Zielfunktion $\frac{1}{2}\|f\|^2$ durch die statistische Entropie

$$E(f) := -\int_\Omega f \ln(f)\ d\mu$$

zu ersetzen und das Optimierungsproblem

$$E(f) \longrightarrow \max \quad \text{unter der Nebenbedingung} \quad \int_\Omega af\ d\mu = b \tag{1-4}$$

zu betrachten, wobei die Nebenbedingung hier nun vektoriell formuliert ist. In der Definition der Entropie wird durch $\ln(f)$ die Positivität von f gesichert, was die Aussage über die Lösbarkeit eines solchen Problems im Vergleich zur Galerkin-Methode jedoch stark beeinflusst. Betrachten wir zum Beispiel ein Momentenproblem mit durchweg positiven Momentenfunktionen a_i, einem Maß μ (d.h. $\mu \geq 0$) und Momenten b_i, von denen mindestens eines negativ ist, dann werden wir keine

positive Dichte f finden können, die das Momentenproblem löst. Selbst dann nicht, wenn die Momentenfunktionen a_i linear unabhängig sind.

Für diese Methode liefert der LAGRANGE-Formalismus eine positive Dichte

$$f_\lambda(g) = \exp\Big(-1 + \sum_{i=1}^{n} \lambda_i a_i(g) \Big) \ , \ g \in G \ , \tag{1-5}$$

in Abhängigkeit der LAGRANGE-Multiplikatoren $\lambda_i \in \mathbb{R}$. Die Frage, die sich hier nun aufdrängt, ist die nach der Existenz eines solchen $\lambda \in \mathbb{R}^n$, für das die Nebenbedingung

$$\int_\Omega a f_\lambda \ d\mu = b \tag{1-6}$$

erfüllt ist. Es ist ersichtlich, dass diese Frage nur dann positiv beantwortet werden kann, wenn die Momente b_i kompatibel mit den Momentenfunktionen a_i und der Positivitätsbedingung an die Dichte f sind. Im weiteren Verlauf werden wir bei der Behandlung solcher Probleme deshalb stets davon ausgehen, dass der Momentenvektor b für mindestens ein $f \in \mathbb{L}^2(G)$ mit $f \geq 0$ darstellbar ist als

$$b = \int_\Omega a f \ d\mu \ . \tag{1-7}$$

Die lineare Unabhängigkeit der Momentenfunktionen a_i ist auch hier für die Eindeutigkeit der LAGRANGE-Multiplikatoren notwendig, im Gegensatz zu (1-3) jedoch nicht hinreichend. Tatsächlich benötigt man hier zusätzlich eine schärfere Form der linearen Unabhängigkeit, die sogenannte *Pseudo*-HAAR Eigenschaft. Man sagt, dass die Momentenfunktionen a_i die Pseudo-HAAR Eigenschaft besitzen, wenn unter allen Linearkombinationen $\beta \cdot a$ nur diejenige mit $\beta = 0$ auf allen Mengen positiven Maßes verschwindet. Diese Eigenschaft ist schärfer als die lineare Unabhängigkeit der Momentenfunktionen, bei welcher $\beta \cdot a = 0$ mit $\beta \neq 0$ auf dem ganzen Raum unmöglich, auf Teilmengen des Raumes jedoch möglich ist. Diese Pseudo-HAAR Eigenschaft lässt sich auf die invarianten Unterräume U_i aus der Zerlegung (1-2) von PETER und WEYL übertragen und liefert folgendes Resultat, zu welchem detailliertere Ausführungen in der Arbeit von JUNK, BUDDAY und BÖHLKE nachzulesen sind:[24]

1.10 Satz. *Sei $\Omega \subset G$ kompakt, dann erhält man unter der Annahme* (1-7) *die Existenz einer eindeutigen Lösung von* (1-6)*, sofern die Momentenfunktionen a_i linear unabhängig und aus den Unterräumen U_i gewählt sind.*

Eine numerisch interessante Bestimmungsmethode dieser unter obigen Voraussetzungen eindeutig existierenden LAGRANGE-Multiplikatoren λ liefert der Übergang vom *primalen* Optimierungsproblem (1-4) zum zugehörigen *dualen* Optimierungsproblem:

$$\text{minimiere } \Phi : \mathbb{R}^n \longrightarrow \mathbb{R} \text{ mit}$$
$$\Phi(\lambda) := \int_\Omega \exp\Big(-1 + \sum_{i=1}^{n} \lambda_i a_i \Big)\, d\mu \; - \; \sum_{i=1}^{n} \lambda_i b_i \tag{1-8}$$

Man beachte, dass die Positivitätsbedingung sowie die Nebenbedingungen für die Momente durch die Definition der Zielfunktion Φ bereits direkt in das Optimierungsproblem eingearbeitet sind. Denn zur Bestimmung des Minimums von Φ suchen wir die Minimalstelle $\lambda \in \mathbb{R}^n$, die den Gradienten von Φ verschwinden lässt, d.h. jenes $\lambda \in \mathbb{R}^n$, für das für jedes $j \in \{1, 2, \ldots, n\}$ gilt:

$$\begin{aligned} \nabla\Phi(\lambda)_j \; &= \; \int_\Omega a_j \exp\Big(-1 + \sum_{i=1}^{n} \lambda_i a_i \Big)\, d\mu \; - \; b_j \; \stackrel{!}{=} \; 0 \\ &\iff \int_\Omega a_j f_\lambda \, d\mu \; = \; b_j \end{aligned}$$

Anhand der folgenden Tabelle wird ersichtlich, aufgrund welcher Eigenschaften das duale Problem (1-8) aus numerischen Gesichtspunkten interessanter ist als das primale Problem (1-4) und warum unter der Annahme (1-7) eine solche Minimalstelle λ überhaupt existiert:

primales Problem	duales Problem
unendlich-dimensional	**endlich-dimensional**
restringiert	**nicht restringiert**
strikt konvexe Zielfunktion	strikt konvexe Zielfunktion

Tabelle 1.1: Vergleich von primalem und dualem Optimierungsproblem

Im weiteren Verlauf werden wir wie schon zu Beginn dieser Einleitung das Maximum Entropie Momentenproblem überwiegend für den Fall $G = SO(3)$ betrachten, d.h. eine positive Dichte $f \in \mathbb{L}^2_{\geq 0}\big(SO(3)\big)$ suchen, die für ein gegebenes $n \in \mathbb{N}$ die Momentenbedingungen

$$\int_{SO(3)} a_i(Q) f(Q)\, dQ = b_i \quad \text{für} \quad i = 1, \ldots, n$$

erfüllt, für gegebene Momentenfunktionen $a_i \in \mathbb{L}^2\big(SO(3)\big)$ und gegebene Momente $b_i \in \mathbb{R}$. Da wir auf der Suche nach einer Orientierungsverteilungsfunktion sind, stellen wir zusätzlich die Normierungsbedingung

$$\int_{SO(3)} f(Q)\, dQ = 1 \; ,$$

welche jedoch problemlos als eine der Nebenbedingungen aufgenommen werden kann, etwa durch die Wahl von $a_1 = b_1 = 1$. Das Aufsuchen solch einer Verteilungsfunktion reduziert sich mit Hilfe der Maximum Entropie Methode auf das Lösen von (1-8), d.h. für diesen Fall das Lösen von

$$\min_{\lambda \in \mathbb{R}^n} \left(\int\limits_{SO(3)} \exp\Big(-1 + \sum_{i=1}^{n} \lambda_i a_i(Q) \Big) \, dQ \; - \; \sum_{i=1}^{n} \lambda_i b_i \right).$$

An dieser mathematischen Formulierung des Problems lässt sich die Idee der vorliegenden Arbeit sehr gut erklären:

Es sollen die verschiedenen Bestandteile dieses Problems numerisch effizient analysiert werden. So muss einerseits ein geeignetes Verfahren gefunden oder entwickelt werden, um das vorliegende *Optimierungsproblem* zu lösen (Kapitel 6). Um ein Optimierungsproblem zu lösen, muss man jedoch zuerst einmal in der Lage sein, die dem Problem zugrunde liegende Zielfunktion auswerten zu können. Um die hier vorliegende Zielfunktion jedoch auswerten zu können, muss zunächst einmal *über* $SO(3)$ *integriert* werden, d.h. die Frage nach dem Haar-Maß auf $SO(3)$ geklärt werden, sowie Möglichkeiten gefunden werden, diese Integration so effizient wie möglich durchführen zu können. Denn im Verlauf des Optimierungsvorganges muss sie für jede einzelne Zielfunktionsauswertung durchgeführt werden (Kapitel 5). Bei jeder durchzuführenden Integration muss wiederum der *Integrand* mehrmals *ausgewertet* werden, d.h. es muss geklärt werden, wie man die *Darstellungsfunktionen* $a_i \in \mathbb{L}^2\big(SO(3)\big)$ schnell und effizient *auswerten* kann. Dies wird besonders im Falle der codf von großer Bedeutung sein, da hierbei das zugehörige Momentenproblem, und somit die zu minimierende Zielfunktion, sehr hochdimensional ist. Jede einzelne Auswertung der Darstellungsfunktionen wird demnach extrem aufwändig sein. Deshalb wird diese Thematik in den Kapiteln 4, 3 und 2 sehr ausführlich behandelt werden. Im Folgenden werden diese numerischen Aspekte nun thematisch von innen nach außen bearbeitet, d.h. ausgehend von der Auswertung der Darstellungsfunktionen auf $SO(3)$ bis hin zur Optimierung. Abschließend werden in Kapitel 7 noch weitere Anwendungsbeispiele der Maximum Entropie Methode vorgestellt.

2 Darstellungstheorie kompakter Gruppen

Wie bereits in der Einleitung erwähnt wurde, steht die FOURIER-Entwicklung der codf in engem Zusammenhang mit der Darstellungstheorie *kompakter Gruppen*. Deshalb soll dieses Kapitel einen Überblick über jene Grundlagen dieser Theorie liefern, welche in unserem Fall von Interesse sein werden.[12,16,33,34]

2.1 Kompakte Gruppen & Lie-Gruppen

Zum besseren Verständnis des Kontextes der folgenden Abschnitte zur Darstellungstheorie kompakter Gruppen soll dieser Abschnitt einen kurzen Überblick der wichtigsten Definitionen rund um kompakte Gruppen geben.

2.1 Definition. Ein topologischer Raum (X, τ) heißt **Hausdorff-Raum** oder **separierter Raum**, wenn für alle $x, y \in X$ mit $x \neq y$ disjunkte, offene Umgebungen $U(x), U(y) \subset X$ existieren.

2.2 Definition. Ein HAUSDORFF-Raum (X, τ) heißt **kompakt**, wenn jede offene Überdeckung $X = \bigcup_{j \in J} U_j$ mit $U_j \in \tau$ für alle Indizes $j \in J$ einer gegebenen Indexmenge J eine endliche Teilüberdeckung $X = U_{j_1} \cup U_{j_2} \cup \cdots \cup U_{j_n}$ mit $j_1, j_2, \ldots, j_n \in J$ besitzt. (X, τ) heißt **lokalkompakt**, wenn jede Umgebung eines beliebigen Punktes $x \in X$ eine kompakte Umgebung enthält.

Mit Hilfe dieser Definitionen können wir nun die Begriffe *topologische* und *kompakte Gruppe* definieren:[16]

2.3 Definition. Ein Paar (G, τ) aus einer Gruppe G und einer Topologie τ heißt **topologische Gruppe**, wenn die Gruppenverknüpfung $G \times G \longrightarrow G$ und die zur Gruppe gehörende Inversenabbildung stetige Abbildungen sind. Dabei versieht man $G \times G$ mit der zugehörigen Produkttopologie. Eine topologische Gruppe heißt **(lokalkompakt) kompakt**, wenn sie als HAUSDORFF-Raum (G, τ) (lokalkompakt) kompakt ist.

Im späteren Verlauf wird unser Fokus, wie bereits eingangs erwähnt wurde, auf einer für uns wichtigen Gruppe liegen, der sogenannten *speziellen orthogonalen Gruppe*

$$G = SO(d) := \left\{ Y \in \mathbb{R}^{d \times d} \;\middle|\; YY^T - \mathbb{1} = 0 \;\text{ und }\; \det Y = 1 \right\}$$

im Fall $\boldsymbol{d=3}$. Diese Gruppe ist ein Repräsentant des speziellen Gruppentyps der sogenannten LIE-Gruppen, welche in der folgenden Definition genauer beschrieben werden:[12]

2.4 Definition. Eine topologische Gruppe G heißt **Lie-Gruppe**, wenn G eine differenzierbare Mannigfaltigkeit ist, so dass die Gruppenverknüpfung $G \times G \longrightarrow G$ und die zur Gruppe gehörende Inversenabbildung beliebig oft differenzierbar, d.h. C^∞-Abbildungen sind.

2.5 Bemerkung. Nicht jede LIE-Gruppe ist kompakt, wie man an der Gruppe $G = (\mathbb{R}^d, +)$ sieht. Die Gruppe $SO(d)$ hingegen ist ein Beispiel für eine kompakte LIE-Gruppe.[33]

Die weiteren Resultate in diesem Kapitel sind im Allgemeinen für kompakte topologische Gruppen (kurz: **kompakte Gruppen**) hergeleitet.

2.2 Zu Darstellungen kompakter Gruppen

Dieser Abschnitt ist in Anlehnung an die Arbeit von JUNK, BUDDAY und BÖHLKE verfasst.[24] Im Folgenden erinnern wir an die Definition einer Darstellung einer kompakten Gruppe auf einem topologischen Vektorraum und an die Definition der sogenannten Darstellungsfunktionen auf einer kompakten Gruppe, welche bereits in der Einleitung in Definition 1.3 auf Seite 3 erwähnt wurden:

2.6 Definition. Sei G eine kompakte Gruppe und V ein topologischer Vektorraum, dann heißt eine stetige Abbildung $\varrho : G \times V \longrightarrow V$ mit der Eigenschaft, dass $\varrho_g : v \longmapsto \varrho_g v$ linear ist und $\varrho_{gh} = \varrho_g \varrho_h$, eine **Darstellung von G auf V**. Bei Wahl eines $v \in V$ und einem linearen Funktional $\alpha \in V^*$ aus dem zugehörigen Dualraum V^* bezeichnen wir die Abbildung $g \longmapsto \alpha(\varrho_g v)$ als **Darstellungsfunktion auf G**.

Ein wichtiges Beispiel für eine solche Darstellung ist die sogenannte *reguläre Darstellung* $\mathcal{D} : G \times \mathbb{L}^2(G) \longrightarrow \mathbb{L}^2(G)$ einer kompakten Gruppe G auf dem Vektorraum $\mathbb{L}^2(G)$, welche durch $(\mathcal{D}_h f)(g) := f(h^{-1}g)$ definiert ist. Dass es sich hierbei tatsächlich um eine Darstellung gemäß Definition 2.6 handelt, sehen wir daran, dass $\mathcal{D}_{h_1 h_2} = \mathcal{D}_{h_1}\mathcal{D}_{h_2}$ gilt:

$$\begin{aligned}(\mathcal{D}_{h_1h_2}f)(g) &= f\big((h_1h_2)^{-1}g\big) = f(h_2^{-1}h_1^{-1}g) = (\mathcal{D}_{h_2}f)(h_1^{-1}g)\\ &= \mathcal{D}_{h_1}(\mathcal{D}_{h_2}f)(g) = (\mathcal{D}_{h_1}\mathcal{D}_{h_2}f)(g)\end{aligned}$$

Wie bereits in Bemerkung 1.7 auf Seite 5 erwähnt wurde, liefert jede Darstellung einer kompakten Gruppe G auf einem endlich-dimensionalen Vektorraum V einen bezüglich der regulären Darstellung invarianten Unterraum von $\mathbb{L}^2(G)$, der durch

Darstellungsfunktionen erzeugt wird (kurz: $\mathcal{D}$-invarianter Unterraum). Um dies zu sehen, nehmen wir zunächst eine beliebige Darstellung $\varrho : G \times V \longrightarrow V$ von G auf einem Vektorraum V mit $\dim V = d < \infty$, wählen eine Basis $v_1, \ldots, v_d$ von V und berechnen die entsprechende Matrix $B(g)$ der linearen Abbildung ϱ_g für ein beliebiges $g \in G$. Es lässt sich zeigen, dass die Matrixeinträge $b_{ij}(g)$ stetige Funktionen auf G und somit Funktionen aus $\mathbb{L}^2(G)$ sind. Genauer gesagt erzeugen sie einen Unterraum U von $\mathbb{L}^2(G)$, welcher invariant unter der regulären Darstellung ist, d.h. es gilt $\mathcal{D}_g U \subset U$ für alle $g \in G$. Um dies zu sehen, betrachten wir die zugehörige Dualbasis $v_1^*, \ldots, v_d^*$ des Dualraums V^* und verwenden die Definition der transponierten linearen Abbildung, welche besagt, dass die transponierte Abbildung $L^T : V^* \longrightarrow V^*$ einer linearen Abbildung $L : V \longrightarrow V$ durch $L^T(w) = w \circ L$ gegeben ist. Mit Hilfe der Dualbasis erhalten wir die Matrixeinträge von $B(g)$ durch $b_{ij}(g) = v_i^*(\varrho_g v_j)$. Bei den Matrixeinträgen handelt es sich demnach um Darstellungsfunktionen auf G. Für eine beliebige Linearkombination $f(g) := \sum_{i,j=1}^{d} c_{ij} b_{ij}(g)$ der Matrixeinträge von $B(g)$, d.h. es gilt $f \in U$, erhalten wir sodann

$$\begin{aligned}
(\mathcal{D}_h f)(g) = f(h^{-1}g) &= \sum_{i,j=1}^{d} c_{ij} b_{ij}(h^{-1}g) = \sum_{i,j=1}^{d} c_{ij} v_i^*(\varrho_{h^{-1}g} v_j) = \sum_{i,j=1}^{d} c_{ij} v_i^*(\varrho_{h^{-1}} \varrho_g v_j) \\
&= \sum_{i,j=1}^{d} c_{ij} (v_i^* \circ \varrho_{h^{-1}})(\varrho_g v_j) = \sum_{i,j=1}^{d} c_{ij} \underbrace{\varrho_{h^{-1}}^T(v_i^*)}_{\in V^*}(\varrho_g v_j) \\
&= \sum_{i,j=1}^{d} c_{ij} \Big(\sum_{k=1}^{d} d_{ik} v_k^*(\varrho_g v_j) \Big) =: \sum_{k,j=1}^{d} q_{kj} v_k^*(\varrho_g v_j) = \sum_{k,j=1}^{d} q_{kj} b_{kj}(g) \ .
\end{aligned}$$

Es gilt also $\mathcal{D}_h f \in U$, was die Invarianz des Unterraumes U beweist. Mit einer analogen Argumentation lässt sich zeigen, dass eine Darstellung $\overline{\varrho} : G \times W \longrightarrow W$ derselben Gruppe G, jedoch auf einem anderen endlich-dimensionalen Vektorraum W, welche sich von ϱ lediglich durch eine feste unitäre Koordinatentransformation unterscheidet, denselben Unterraum $U \subset \mathbb{L}^2(G)$ erzeugt. Aus diesem Grund führen wir mit folgender Definition den Begriff von *zueinander äquivalenten* Darstellungen ein:[33]

2.7 Definition. Sei G eine kompakte Gruppe, dann heißen zwei Darstellungen $\varrho : G \times V \longrightarrow V$ und $\overline{\varrho} : G \times W \longrightarrow W$ von G auf den endlich-dimensionalen Vektorräumen V und W **zueinander äquivalent** genau dann, wenn es eine unitäre[1] Abbildung $A : V \longrightarrow W$ gibt, so dass für alle $g \in G$ gilt:

$$\overline{\varrho}_g = A \varrho_g A^{-1} \tag{2-1}$$

[1] zu jeder Darstellung D einer kompakten Gruppe G auf einem endlich-dimensionalen Vektorraum U gibt es ein auf U inneres Produkt $\langle \cdot, \cdot \rangle$ mit der Eigenschaft $\langle D_g u, D_g v \rangle = \langle u, v \rangle$ für alle $u, v \in U$ und alle $g \in G$[33]

Definieren wir mit Hilfe von (2-1) eine *Äquivalenzrelation* auf der Menge aller Darstellungen der kompakten Gruppe G, so können wir nun festhalten, dass alle Repräsentanten einer Äquivalenzklasse denselben Unterraum von $\mathbb{L}^2(G)$ erzeugen. Im Folgenden bezeichnen wir eine Äquivalenzklasse mit α, den zugehörigen Unterraum entsprechend mit U_α.

Wie bereits in der Einleitung angedeutet wurde, benötigen wir den Satz von Peter und Weyl, um ein Resultat zur Entwicklung einer Funktion aus $\mathbb{L}^2(G)$ nach Darstellungsfunktionen auf G zu erhalten. Um dieses Resultat allgemein formulieren zu können, bedarf es jedoch noch ein wenig Vorarbeit.

Zunächst benötigen wir den Begriff der *irreduziblen* Darstellung:

2.8 Definition. Eine Darstellung $\varrho : G \times V \longrightarrow V$ einer kompakten Gruppe G auf einem endlich-dimensionalen Vektorraum V heißt genau dann **irreduzibel**, wenn $\{0\}$ und V die einzigen ϱ-invarianten Unterräume von V sind.

Zur Herleitung eines Irreduzibilitäts-Kriteriums einer Darstellung benötigen wir das folgende Lemma, welches auf den Kommutator $[A_1, A_2] := A_1A_2 - A_2A_1$ zweier Automorphismen $A_1, A_2 \in \mathrm{Aut}(V)$ und die Identitätsabbildung $I \in \mathrm{Aut}(V)$ zurückgreift:

2.9 Lemma. *Sei $\varrho : G \times V \longrightarrow V$ eine Darstellung der kompakten Gruppe G auf einem endlich-dimensionalen, komplexen Vektorraum V, und sei $U \subsetneq V$ mit $U \neq \{0\}$ ein ϱ-invarianter Unterraum von V, dann existiert ein Automorphismus $A \in \mathrm{Aut}(V)$ mit $[A, \varrho_g] = 0$ für alle $g \in G$ und $A \neq \lambda I$ für alle $\lambda \in \mathbb{C}$.*

Beweis. Nach Voraussetzung ist U ein (ϱ-invarianter) Unterraum von V, weshalb wir V orthogonal zerlegen können in $V = U \oplus U^\perp$. Da $U \neq \{0\}$ und $U \subsetneq V$ gilt, gilt ebenso $U^\perp \neq \{0\}$ und $U^\perp \subsetneq V$. Mit Hilfe von Theorem 1.7 und Proposition 1.9 auf Seite 68 in dem Buch von Bröcker und Dieck[12] erhalten wir somit, dass auch $U^\perp$ ein ϱ-invarianter Unterraum von V ist. Definieren wir den Automorphismus $A : V \longrightarrow V$ durch $A|_U = I$ und $A|_{U^\perp} = 2I$, so gilt zum einen für alle $\lambda \in \mathbb{C}$ bereits $A \neq \lambda I$, und zum anderen für ein beliebiges $g \in G$ und alle $v \in V$

$$\begin{aligned} A\varrho_g v - \varrho_g A v &= A\varrho_g(u + u^\perp) - \varrho_g A(u + u^\perp) \\ &= A\varrho_g u + A\varrho_g u^\perp - \varrho_g A u - \varrho_g A u^\perp \\ &= A\underbrace{\varrho_g u}_{\in U} + A\underbrace{\varrho_g u^\perp}_{\in U^\perp} - \varrho_g u - 2\varrho_g u^\perp \\ &= \varrho_g u + 2\varrho_g u^\perp - \varrho_g u - 2\varrho_g u^\perp = 0 \ . \end{aligned}$$

Es gilt also $[A, \varrho_g] = 0$ für alle $g \in G$, was das Lemma schließlich beweist. □

Mit Hilfe dieses Lemmas können wir nun folgendes Irreduzibilitäts-Kriterium für eine Darstellung beweisen:

2.10 Satz. *Sei $\varrho : G \times V \longrightarrow V$ eine Darstellung der kompakten Gruppe G auf einem endlich-dimensionalen, komplexen Vektorraum V, dann sind folgende zwei Aussagen äquivalent:*

(i) *ϱ ist irreduzibel*

(ii) $\{A \in \mathrm{Aut}(V) \mid \forall g \in G : \ [A, \varrho_g] = 0\} = \{A \in \mathrm{Aut}(V) \mid A = \lambda I$ *für ein* $\lambda \in \mathbb{C}\}$

Beweis. $(i) \Rightarrow (ii)$ Wir zeigen die Mengengleichheit. Sei dazu ϱ irreduzibel:
"$\subseteq$": sei $A \in \mathrm{Aut}(V)$ so gewählt, dass für alle $g \in G$ die Bedingung $[A, \varrho_g] = 0$ gilt, dann liefert das Lemma von Schur (siehe Theorem 1.10 auf Seite 69 in dem Buch von Bröcker und Dieck[12]) bereits $A = \lambda I$ für ein $\lambda \in \mathbb{C}$.
"$\supseteq$": sei $A \in \mathrm{Aut}(V)$ gegeben durch $A = \lambda I$ für ein $\lambda \in \mathbb{C}$, dann gilt für alle $g \in G$ und alle $v \in V$:

$$[A, \varrho_g]v \ = \ A\varrho_g v - \varrho_g A v \ = \ \lambda\varrho_g v - \lambda\varrho_g v \ = \ 0$$

$(ii) \Rightarrow (i)$ Die Kontraposition von Lemma 2.9 besagt, dass es unter Voraussetzung der Mengengleichheit

$$\{A \in \mathrm{Aut}(V) \mid \forall g \in G : \ [A, \varrho_g] = 0\} = \{A \in \mathrm{Aut}(V) \mid A = \lambda I \text{ für ein } \lambda \in \mathbb{C}\}$$

keine nichttrivialen ϱ-invarianten Unterräume von V geben kann. Somit ist die Darstellung ϱ irreduzibel. □

Mit Hilfe dieses Satzes gelingt es nun die Äquivalenzklassen von Darstellungen einer kompakten Gruppe G etwas zu charakterisieren:

2.11 Satz. *Sei G eine kompakte Gruppe und $\varrho : G \times V \longrightarrow V$ und $\mu : G \times W \longrightarrow W$ zwei zueinander äquivalente Darstellungen von G auf den endlich-dimensionalen, komplexen Vektorräumen V und W, dann gilt: ϱ ist genau dann irreduzibel, wenn μ irreduzibel ist.*

Beweis. Nach Voraussetzung sind ϱ und μ zueinander äquivalent, d.h. es gibt eine unitäre Abbildung $A : V \longrightarrow W$ mit $\mu_g = A\varrho_g A^{-1}$ für alle $g \in G$. Da beide Beweisrichtungen exakt analog geführt werden, beschränken wir uns darauf, eine Richtung zu zeigen. Sei dazu ohne Einschränkung der Allgemeinheit ϱ irreduzibel. Um die daraus resultierende Irreduzibilität von μ zu zeigen, genügt es aufgrund von Satz 2.10 die Gleichheit

$$\{B \in \mathrm{Aut}(W) \mid \forall g \in G : \ [B, \mu_g] = 0\} = \{B \in \mathrm{Aut}(W) \mid B = \lambda I \text{ für ein } \lambda \in \mathbb{C}\}$$

zu zeigen. Für die Inklusion "$\subseteq$" wählen wir also ein beliebiges $B \in \mathrm{Aut}(W)$ mit $[B, \mu_g] = 0$ für alle $g \in G$, d.h. $B\mu_g = \mu_g B$ für alle $g \in G$. Daraus können wir die Irreduzibilität von μ nun wie folgt schließen:

$$\begin{aligned} B\mu_g = \mu_g B &\Rightarrow BA\varrho_g A^{-1} = A\varrho_g A^{-1} B \\ &\Rightarrow A^{-1}BA\varrho_g = \varrho_g A^{-1}BA \\ &\Rightarrow [A^{-1}BA, \varrho_g] = 0 \\ &\Rightarrow A^{-1}BA = \lambda I \text{ für ein } \lambda \in \mathbb{C}, \text{ da } \varrho \text{ irreduzibel} \\ &\Rightarrow B = \lambda A I A^{-1} = \lambda A A^{-1} = \lambda I \end{aligned}$$

Die Inklusion "$\supseteq$" ist trivial und verläuft analog zu jener im Beweis zu Satz 2.10. □

Vor diesem Hintergrund kann man nun zeigen, dass im Falle von kompakten Gruppen G der Raum $\mathbb{L}^2(G)$ in eine orthogonale Summe endlich-dimensionaler Unterräume U_α zerfällt, wobei α dabei auf die Äquivalenzklassen der endlich-dimensionalen, irreduziblen Darstellungen von G beschränkt ist. Dies wird im folgenden Satz, welcher als Satz von Peter und Weyl bekannt ist, festgehalten. Der Beweis hierzu ist dem Buch von Sternberg[34] zu entnehmen.

2.12 Satz. *Sei G eine kompakte Gruppe und $\hat{G}$ die Menge der Äquivalenzklassen der endlich-dimensionalen, irreduziblen Darstellungen von G, dann gilt*

$$\mathbb{L}^2(G) = \bigoplus_{\alpha \in \hat{G}} U_\alpha \ .$$

Dieser Satz liefert schließlich, dass jedes $f \in \mathbb{L}^2(G)$ in der Form $f = \sum_{\alpha \in \hat{G}} \hat{f}_\alpha$, der sogenannten Fourier-Entwicklung von f nach Darstellungsfunktionen auf G, mit eindeutig bestimmten $\hat{f}_\alpha \in U_\alpha$ geschrieben werden kann. Die Unterräume U_α sind jene, die bereits in der Einleitung erwähnt wurden. Um die Fourier-Entwicklung präzisieren zu können, benötigt man Basen dieser Unterräume. Eine Basis des Unterraumes U_α kann man mit jeder beliebigen Darstellung $\varrho : G \times V \longrightarrow V$ aus der Äquivalenzklasse α und einer beliebigen Basis $v_1, \ldots, v_d$ von V berechnen. Mit der zugehörigen Dualbasis erhalten wir dann, wie bereits beschrieben, durch $b_{ij}(g) := v_i^*(\varrho_g v_j)$ eine spezielle Basis von U_α. Im folgenden Abschnitt wird für den Fall $G = SO(3)$ eine spezielle Wahl einer solchen Basis vorgestellt.

2.3 Darstellungen von $SU(2)$ und $SO(3)$

Das Ziel in diesem Abschnitt wird es sein, im Falle der kompakten Gruppe $SO(3)$ jeweils einen Repräsentanten der Äquivalenzklassen der irreduziblen Darstellungen zu finden, um damit den Raum $\mathbb{L}^2\big(SO(3)\big)$ wie beschrieben zerlegen und Basen der entsprechenden Unterräume angeben zu können. Dazu machen wir zunächst einen kleinen Umweg und betrachten Darstellungen der kompakten Gruppe

$$SU(2) := \left\{ Y \in \mathbb{C}^{2\times 2} \ \mid \ \overline{Y}^T Y - \mathbb{1} = 0 \ \text{ und } \ \det Y = 1 \right\} ,$$

der sogenannten *speziellen unitären Gruppe*, auf dem Vektorraum $\mathcal{V}_r(\mathbb{C}^2)$ der **homogenen Polynome vom Grad r in zwei Variablen über** $\mathbb{C}^2$. Für die Dimension dieses Vektorraumes gilt $\dim \mathcal{V}_r(\mathbb{C}^2) = r+1$. Betrachten wir für ein festes $r \in \mathbb{N}_0$ die Darstellung

$$\begin{aligned}\varrho^r : SU(2) \times \mathcal{V}_r(\mathbb{C}^2) &\longrightarrow \mathcal{V}_r(\mathbb{C}^2) \\ (Y,p) &\longmapsto \varrho^r_Y p, \text{ mit } (\varrho^r_Y p)(z) := p(Y^T z),\ z=(z_1,z_2)\in\mathbb{C}^2\ ,\end{aligned}$$

so erhalten wir mit Proposition 5.1 auf Seite 85 in dem Buch von Bröcker und Dieck[12] für alle $r \in \mathbb{N}_0$ die Irreduzibilität dieser Darstellungen. Nach Proposition 5.3 auf Seite 86 erhalten wir sogar, dass jede irreduzible Darstellung μ von $SU(2)$ auf einem beliebigen endlich-dimensionalen Vektorraum X isomorph ist zu einer der Darstellungen ϱ^r, d.h. dass es ein $r_0 \in \mathbb{N}_0$ gibt, sodass X isomorph ist zu $\mathcal{V}_{r_0}(\mathbb{C}^2)$ bzw. die Darstellungen μ und ϱ^{r_0} zueinander äquivalent sind. Da für $r_1, r_2 \in \mathbb{N}_0$ mit $r_1 \neq r_2$ aus Dimensionsgründen die Darstellungen ϱ^{r_1} und ϱ^{r_2} nicht zueinander äquivalent sein können, erhalten wir demnach durch die Darstellungen ϱ^r aus jeder Äquivalenzklasse der irreduziblen Darstellungen von $SU(2)$ genau einen Repräsentanten und können das Resultat von Peter und Weyl in diesem Fall durch

$$\mathbb{L}^2\big(SU(2)\big) = \bigoplus_{r\in\mathbb{N}_0} U_r$$

notieren, wobei wir mit U_r den zu der irreduziblen Darstellung ϱ^r gehörenden $\mathcal{D}$-invarianten Unterraum von $\mathbb{L}^2\big(SU(2)\big)$ bezeichnen.

Betrachten wir nun irreduzible Darstellungen der kompakten Gruppe $SO(3)$, so können wir mit Hilfe der Bemerkung auf Seite 86 in dem Buch von Bröcker und Dieck[12] einen Zusammenhang zu jenen der kompakten Gruppe $SU(2)$ herleiten. Mit Hilfe eines Epimorphismus $\pi : SU(2) \longrightarrow SO(3)$ lässt sich zeigen, dass die irreduziblen Darstellungen von $SO(3)$ in bijektivem Zusammenhang zu den irreduziblen Darstellungen ϱ^{2r} von $SU(2)$ auf $\mathcal{V}_{2r}(\mathbb{C}^2)$ für $r \in \mathbb{N}_0$ stehen. Über diese Bijektion erhalten wir somit zu jeder Äquivalenzklasse der irreduziblen Darstellungen von $SO(3)$ einen Repräsentanten, d.h. zu jedem $r \in \mathbb{N}_0$ eine irreduzible Darstellung von $SO(3)$ auf einem Vektorraum W_r mit der Dimension $\dim W_r = \dim \mathcal{V}_{2r}(\mathbb{C}^2) = 2r+1$. Mit Hilfe der Äquivalenzklassen dieser irreduziblen Darstellungen können wir folglich den kompletten Raum $\mathbb{L}^2\big(SO(3)\big)$ zerlegen. Schließlich erhalten wir mit Proposition 5.10 auf Seite 89 in dem Buch von Bröcker und Dieck[12] folgendes für uns wichtiges Resultat:

2.13 Satz. *Für jedes $r \in \mathbb{N}_0$ ist der Vektorraum W_r isomorph zu dem Vektorraum $\mathcal{H}_r(\mathbb{R}^3)$ der* ***homogenen, harmonischen Polynome vom Grad r in drei Variablen über*** *$\mathbb{R}^3$, folglich gilt $\dim \mathcal{H}_r(\mathbb{R}^3) = \dim W_r = 2r+1$. Ein Repräsentant der zu $r \in \mathbb{N}_0$ gehörenden Äquivalenzklasse der irreduziblen Darstellungen von $SO(3)$ ist gegeben durch die irreduzible Darstellung*

$$\begin{aligned} \varrho : SO(3) \times \mathcal{H}_r(\mathbb{R}^3) &\longrightarrow \mathcal{H}_r(\mathbb{R}^3) \\ (R, f) &\longmapsto \varrho_R f \ \textit{mit}\ (\varrho_R f)(x) := f(R^T x),\ x \in \mathbb{R}^3 \ . \end{aligned}$$

Dies liefert uns also die Zerlegung des Raumes $\mathbb{L}^2\big(SO(3)\big)$ ebenso in der Form

$$\mathbb{L}^2\big(SO(3)\big) = \bigoplus_{r \in \mathbb{N}_0} U_r \ ,$$

wobei wir hier mit U_r nun den zur irreduziblen Darstellung ϱ aus Satz 2.13 gehörenden $\mathcal{D}$-invarianten Unterraum von $\mathbb{L}^2\big(SO(3)\big)$ bezeichnen. Um nun wie bereits angekündigt eine Basis des Unterraumes U_r zu bestimmen, verwenden wir die irreduzible Darstellung ϱ aus Satz 2.13 als Repräsentant der entsprechenden Äquivalenzklasse, und benötigen deshalb zunächst eine Basis des Vektorraumes $\mathcal{H}_r(\mathbb{R}^3)$. Da homogene Funktionen über $\mathbb{R}^3$ durch die Einschränkung auf die Einheitssphäre S^2 bereits eindeutig bestimmt sind, können wir uns auf den Raum $\mathcal{H}_r(S^2)$ beschränken. Das hat den Vorteil, dass mit den Kugelflächenfunktionen bereits eine bekannte Orthonormalbasis (ONB) von $\mathcal{H}_r(S^2)$ zur Verfügung steht. Im nächsten Abschnitt berechnen wir damit nun eine Basis des Unterraumes U_r.

Aufgrund der Wichtigkeit der erwähnten Polynome für weitere Aussagen in dieser Arbeit, gibt die folgende Tabelle einen Überblick der Dimensionen des Vektorraumes $\mathcal{V}_r(\mathbb{R}^3)$ der homogenen Polynome vom Grad r über $\mathbb{R}^3$ im Vergleich zu den Dimensionen des Vektorraumes $\mathcal{H}_r(\mathbb{R}^3)$ der homogenen, harmonischen Polynome vom Grad r über $\mathbb{R}^3$ in Abhängigkeit des Grades $r \in \mathbb{N}_0$:[12]

Grad r	homogen	homogen, harmonisch
	$\dim \mathcal{V}_r(\mathbb{R}^3) = \frac{1}{2}(r+1)(r+2)$	$\dim \mathcal{H}_r(\mathbb{R}^3) = 2r+1$
0	1	1
1	3	3
2	6	5
3	10	7
4	15	9
5	21	11
6	28	13
7	36	15

Tabelle 2.1: Dimensionen der Polynomräume $\mathcal{V}_r(\mathbb{R}^3)$ und $\mathcal{H}_r(\mathbb{R}^3)$

2.3.1 Basis der Kugelflächenfunktionen

Wir betrachten den Vektorraum $\mathcal{H}_r(S^2)$ der homogenen, harmonischen Polynome vom Grad r über der Einheitssphäre S^2. Nach Satz 2.13 gilt $\dim \mathcal{H}_r(S^2) = 2r+1$ und man kann zeigen, dass die sogenannten Kugelflächenfunktionen Y_r^m vom Grad r mit $m \in \{-r, \ldots, r\}$ eine ONB von $\mathcal{H}_r(S^2)$ bilden. Deshalb können wir für ein $r \in \mathbb{N}_0$ den Vektorraum $\mathcal{H}_r(S^2)$ in der Form

$$\mathcal{H}_r(S^2) = \operatorname{span}\{Y_r^m \mid m = -r, \ldots, r\} \subset \mathbb{L}^2(S^2)$$

schreiben. Nach Satz 2.13 haben wir mit

$$\begin{aligned} \varrho : SO(3) \times \mathcal{H}_r(S^2) &\longrightarrow \mathcal{H}_r(S^2) \\ (R, f) &\longmapsto \varrho_R f \text{ mit } (\varrho_R f)(x) := f(R^T x),\ x \in S^2 \end{aligned}$$

eine irreduzible Darstellung von $SO(3)$ auf $\mathcal{H}_r(S^2)$. Wählen wir als Basis von $\mathcal{H}_r(S^2)$ also $v_{-r}, \ldots, v_r$ mit $v_m = Y_r^m$, so erhalten wir durch $v_m^*(f) := \langle f, Y_r^m \rangle_{\mathbb{L}^2(S^2)}$ die zugehörige Dualbasis. Die Basisfunktionen, die den $\mathcal{D}$-invarianten Unterraum U_r von $\mathbb{L}^2\big(SO(3)\big)$ erzeugen, erhalten wir demnach durch die Matrixeinträge b_{mn} der linearen Abbildung ϱ_R, d.h. es gilt

$$b_{mn}(R) = v_m^*(\varrho_R v_n) = \langle \varrho_R Y_r^n, Y_r^m \rangle_{\mathbb{L}^2(S^2)} =: T_r^{mn}(R)\ .$$

Die Orthonormalität der Kugelflächenfunktionen Y_r^m liefert die Orthogonalität der sogenannten Wigner-*Funktionen* T_r^{mn} in $\mathbb{L}^2\big(SO(3)\big)$, d.h.

$$\langle T_r^{mn}, T_{r'}^{m'n'} \rangle_{\mathbb{L}^2(SO(3))} \ = \ \frac{1}{2r+1}\ \delta_{rr'}\ \delta_{mm'}\ \delta_{nn'}\ .$$

Weitere Informationen zu den Wigner-Funktionen T_r^{mn} sind zum Beispiel dem Buch von Bunge[14] zu entnehmen. Die linearen Hüllen der Wigner-Funktionen zu einem jeweils festen $r \in \mathbb{N}_0$ liefern uns somit alle Unterräume $U_r \subset \mathbb{L}^2\big(SO(3)\big)$ der Zerlegung von $\mathbb{L}^2\big(SO(3)\big)$ nach Peter und Weyl. Mit den Wigner-Funktionen T_r^{mn} als natürliche Basis gilt folglich

$$\mathbb{L}^2\big(SO(3)\big) = \bigoplus_{r \in \mathbb{N}_0} U_r\ .$$

Es ist also möglich, sich ausgehend von homogenen, harmonischen Polynomen Basen der invarianten Unterräume U_r zu konstruieren. Im folgenden Kapitel werden wir einen Zusammenhang zwischen homogenen, harmonischen Polynomen und Tensoren herleiten, welcher es uns ermöglichen wird, ausgehend von einer Tensorbasis Basen der invarianten Unterräume U_r zu konstruieren. Für die Lösbarkeit des später zu lösenden Maximum Entropie Problems ist die Wahl der Ausgangsbasis jedoch irrelevant, da es nach Satz 1.10 auf Seite 9 der Einleitung lediglich darauf ankommt, dass

die Momentenfunktionen aus den Räumen U_r gewählt werden. Die Tatsache, dass wir, wie bereits in der Einleitung erwähnt wurde, im weiteren Verlauf mit Tensoren arbeiten werden, liegt eher an praktischen Vorteilen. Zum einen ermöglicht uns diese Wahl auf eine sehr einfache Art und Weise, weitere Unterräume mit zusätzlichen Symmetrien (wie etwa der Kristallsymmetrie) konstruieren, und somit diese Symmetrieeigenschaften unmittelbar auf die codf übertragen zu können. Zum anderen beeinflusst sie die numerische Effizienz und Robustheit beim Lösen des Maximum Entropie Problems.

3 Tensoren

Da wir die codf mit Hilfe von Darstellungen auf Tensorräumen approximieren wollen, werden in diesem Kapitel zunächst alle notwendigen Grundlagen über Tensoren zusammengetragen, die für ein tieferes Verständnis der weiteren Tensorkalkulationen benötigt werden.

3.1 Grundlagen

Es gibt verschiedene Möglichkeiten einen Tensor zu definieren. Die wohl einfachste Definition betrachtet einen Tensor schlicht als mehrdimensional indizierte Zusammenfassung numerischer Werte. Die Anzahl an Indizes, die man benötigt, um einen Eintrag dieser Anordnung zu identifizieren, nennt man den *Rang* des Tensors. Für die Resultate der nächsten Kapitel wird es jedoch von Bedeutung sein, die folgende, etwas abstraktere Definition eines Tensors zu verwenden: [28]

3.1 Definition. Sei $\mathbb{K}$ ein Körper, V ein endlich-dimensionaler $\mathbb{K}$-Vektorraum der Dimension $\dim V = d$ und V^* der zugehörige Dualraum. Desweiteren sei $r \in \mathbb{N}$. Dann ist ein Tensor eine multilineare Abbildung

$$T : \underbrace{V^* \times \cdots \times V^*}_{r\text{-mal}} \longrightarrow \mathbb{K} \, . \tag{3-1}$$

Die Zahl $r \in \mathbb{N}$ heißt der **Rang** des Tensors T. Der Vektorraum aller Tensoren vom Rang r über V wird mit $\mathcal{T}_r(V)$ bezeichnet.

Ein Beispiel ist für $a_1, \ldots, a_r \in V$ gegeben durch den Tensor $a_1 \otimes \cdots \otimes a_r \in \mathcal{T}_r(V)$, welcher für beliebige $\nu_1, \ldots, \nu_r \in V^*$ wie folgt definiert ist:

$$a_1 \otimes \cdots \otimes a_r(\nu_1, \ldots, \nu_r) := \prod_{i=1}^{r} \nu_i(a_i)$$

Dies lässt sich mit Hilfe des sogenannten *allgemeinen Tensorproduktes* noch verallgemeinern. Seien dazu $T \in \mathcal{T}_r(V)$ und $S \in \mathcal{T}_s(V)$, dann ist das Tensorprodukt $T \otimes S \in \mathcal{T}_{r+s}(V)$ wie folgt definiert:

$$T \otimes S(\nu_1, \ldots, \nu_{r+s}) := T(\nu_1, \ldots, \nu_r) S(\nu_{r+1}, \ldots, \nu_{r+s}) \tag{3-2}$$

Betrachten wir eine Basis $e_1, \ldots, e_d$ von V und die zugehörige duale Basis $e_1^*, \ldots, e_d^*$ von V^*, welche durch $e_i^*(e_j) := \delta_{ij}$ gegeben ist, so erhalten wir eine Basis von $\mathcal{T}_r(V)$ durch die Menge aller Tensoren der Form $e_{j_1} \otimes \cdots \otimes e_{j_r}$ für $j_p \in \{1, \ldots, d\}$. Für jeden Tensor dieser Menge gilt folglich

$$e_{j_1} \otimes \cdots \otimes e_{j_r}(e_{k_1}^*, \ldots, e_{k_r}^*) = \delta_{k_1 j_1} \cdot \ldots \cdot \delta_{k_r j_r} \ .$$

Dies liefert schließlich das Resultat, dass sich jeder Tensor $T \in \mathcal{T}_r(V)$ in der Form

$$T = \sum_{j_1, \ldots, j_r = 1}^{d} T_{j_1 \ldots j_r} \ e_{j_1} \otimes \cdots \otimes e_{j_r}$$

darstellen lässt, wobei $T_{i_1 \ldots i_r} = T(e_{i_1}^*, \ldots, e_{i_r}^*)$. Da die Tensoren $e_{j_1} \otimes \cdots \otimes e_{j_r}$ eine Basis von $\mathcal{T}_r(V)$ bilden, erhalten wir für die Dimension des Vektorraums der Tensoren vom Rang r über V folgendes Resultat:

$$\dim \mathcal{T}_r(V) = (\dim V)^r = d^r \tag{3-3}$$

Im weiteren Verlauf werden die sogenannten *irreduziblen* Tensoren vom Rang r von sehr großer Bedeutung sein. Um diese Tensoren jedoch definieren zu können, betrachten wir zunächst die sogenannten *symmetrischen* Tensoren vom Rang r, welche durch folgende Definition gegeben sind:

3.2 Definition. Sei S_r die symmetrische Gruppe aller Permutationen der Menge $\{1, \ldots, r\}$ für ein beliebiges $r \in \mathbb{N}$. Dann heißt ein Tensor $T \in \mathcal{T}_r(V)$ genau dann **symmetrisch**, wenn

$$T_{j_1 \ldots j_r} = T_{\pi(j_1) \ldots \pi(j_r)} \tag{3-4}$$

für alle $j_1, \ldots, j_r \in \{1, \ldots, d\}$ und alle $\pi \in S_r$ gilt. Die Menge aller symmetrischen Tensoren vom Rang r über V ist erneut ein Vektorraum, der im Folgenden mit $\mathcal{S}_r(V)$ bezeichnet wird.

Mit Hilfe von $I := \{1, \ldots, d\}^r$ und der Multiindex-Abbildung

$$m : I \longrightarrow \mathbb{N}_0^d \ , \ i \longmapsto m(i) \ ,$$
$$\text{wobei } m(i)_k := \#\big\{n \in \{1, \ldots, r\} \mid i_n = k\big\} \text{ für } k \in \{1, \ldots, d\} \ ,$$

lassen sich symmetrische Tensoren noch etwas einfacher formulieren. Da zum einen für jedes $\pi \in S_r$ die Eigenschaft $m\big(\pi(i)\big) = m(i)$ für alle $i \in I$ gilt, und zum anderen die Bedingung $m(i) = m(j)$ für $i, j \in I$ äquivalent zu $i = \pi(j)$ für ein $\pi \in S_r$ ist, lassen sich symmetrische Tensoren T auch sehr kompakt durch $T_\alpha = T_i$ für alle $i \in I$ mit $\alpha = m(i)$ beschreiben. Daraus können wir folgende Bemerkung ableiten:

3.3 Bemerkung. Der Vektorraum $\mathcal{S}_r(V)$ der symmetrischen Tensoren vom Rang r über V hat die Dimension

$$\dim \mathcal{S}_r(V) = N(r,d) := \#\{\alpha \in \mathbb{N}_0^d \mid |\alpha| = r\} = \binom{r+d-1}{d-1},$$

wobei $|\alpha| := \sum_{k=1}^{d} \alpha_k$ für $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$.

Die sogenannten **spurfreien** Tensoren sind auf dem Weg zur Definition irreduzibler Tensoren genauso bedeutend wie die symmetrischen Tensoren. Die Eigenschaft der Spurfreiheit eines Tensors ergibt sich aus der Definition der **Spur** eines Tensors, welche eine natürliche Operation beschreibt, die den Rang eines Tensors um 2 verringert. Für einen beliebigen Tensor $T \in \mathcal{S}_r(V)$ mit $r \geq 2$ ist die Spur gegeben durch

$$(\operatorname{spur} T)_\alpha \;:=\; \sum_{i=1}^{d} T_{\alpha+2e_i} \quad , \quad |\alpha| = r-2 \; . \tag{3-5}$$

Betrachtet man (3-5), so ist es offensichtlich, dass die Definition der Spur aufgrund des Multiindex α im Fall $r > 2$ hinsichtlich der Eindeutigkeit nur Sinn für symmetrische Tensoren macht. Deshalb bezeichnen wir einen Tensor $T \in \mathcal{S}_r(V)$ genau dann als spurfrei, wenn $\operatorname{spur} T = 0$ gilt, wobei 0 hier für den Null-Tensor aus $\mathcal{S}_{r-2}(V)$ steht. Die Anzahl an zusätzlichen Nebenbedingungen an einen symmetrischen Tensor, um spurfrei zu sein, beträgt demnach $N(r-2,d)$.

Damit sind wir nun in der Lage für $\boldsymbol{r \geq 2}$ irreduzible Tensoren zu definieren:

3.4 Definition. Ein Tensor $T \in \mathcal{T}_r(V)$ mit $r \geq 2$ heißt genau dann **irreduzibel**, wenn T symmetrisch und spurfrei ist. Die Menge aller irreduzibler Tensoren vom Rang r über V ist erneut ein Vektorraum, der im Folgenden mit $\mathcal{J}_r(V)$ bezeichnet wird.

3.5 Bemerkung. Der Vektorraum $\mathcal{J}_r(V)$ der irreduziblen Tensoren vom Rang r über V hat die Dimension

$$\dim \mathcal{J}_r(V) = N(r,d) - N(r-2,d) \; .$$

Zusammenfassend haben wir nun eine Klassifizierung des Tensorraumes $\mathcal{T}_r(V)$ in die folgenden Unterräume:

$$\mathcal{J}_r(V) \subset \mathcal{S}_r(V) \subset \mathcal{T}_r(V)$$

Für den für uns wichtigen Spezialfall $\boldsymbol{d = \dim V = 3}$ gibt die folgende Tabelle einen Überblick der Dimensionen dieser Tensorräume in Abhängigkeit des Ranges r. Vergleicht man diese Dimensionen mit jenen der Polynomräume aus Tabelle 2.1, so lässt sich der erwähnte Zusammenhang zwischen Tensoren und Polynomen bereits erahnen.

Rang r	allgemein	symmetrisch	irreduzibel
	$\dim \mathcal{T}_r(V) = 3^r$	$\dim \mathcal{S}_r(V) = \frac{1}{2}(r+1)(r+2)$	$\dim \mathcal{J}_r(V) = 2r+1$
2	9	6	5
4	81	15	9
6	729	28	13
8	6 561	45	17
10	59 049	66	21
12	531 441	91	25

Tabelle 3.1: Dimensionen der Tensorräume im Fall $d = 3$

3.2 Das Tensorskalarprodukt

Bezeichnen wir mit V erneut einen $\mathbb{K}$-Vektorraum der Dimension d mit einem Skalarprodukt $\langle \cdot, \cdot \rangle_V$ und mit V^* den zugehörigen Dualraum, so induzieren die Basen von V und V^* mit dem Skalarprodukt auf V durch

$$\langle T, U \rangle := \sum_{\substack{i_1,\dots,i_r,\\ j_1,\dots,j_r=1}}^{d} T(e_{i_1}^*, \dots, e_{i_r}^*) U(e_{j_1}^*, \dots, e_{j_r}^*) \prod_{k=1}^{r} \langle e_{i_k}, e_{j_k} \rangle_V$$

ein kanonisches Skalarprodukt $\langle \cdot, \cdot \rangle$ auf $\mathcal{T}_r(V)$. Man kann zeigen, dass dieses Skalarprodukt unabhängig von der Wahl der Basis $e_1, \dots, e_d$ in V und der zugehörigen dualen Basis $e_1^*, \dots, e_d^*$ in V^* ist. Im Spezialfall einer ONB reduziert sich das Skalarprodukt auf

$$\langle T, U \rangle = \sum_{j_1,\dots,j_r=1}^{d} T(e_{j_1}^*, \dots, e_{j_r}^*) U(e_{j_1}^*, \dots, e_{j_r}^*) = \sum_{j_1,\dots,j_r=1}^{d} T_{j_1 \dots j_r} U_{j_1 \dots j_r} \ .$$

Mit Hilfe des Skalarproduktes können wir eine orthogonale Projektion definieren, mit welcher wir einen beliebigen Tensor $T \in \mathcal{T}_r(V)$ *symmetrisieren* können, was uns später noch von Nutzen sein wird:

$$\langle T \rangle_j \ := \ \frac{1}{r!} \sum_{\pi \in S_r} T_{\pi(j)} \quad , \quad j \in I$$

Ein Blick auf Definition 3.2 zeigt, dass der Tensor $\langle T \rangle$ symmetrisch ist, d.h. dass $\langle T \rangle \in \mathcal{S}_r(V)$ gilt. Insbesondere liefert die Symmetrisierung eines symmetrischen Tensors erneut denselben symmetrischen Tensor, denn es ist

$$\begin{aligned}\langle\langle T\rangle\rangle_j &= \frac{1}{r!}\sum_{\pi\in S_r}\langle T\rangle_{\pi(j)} = \frac{1}{(r!)^2}\sum_{\pi,\tilde{\pi}\in S_r} T_{\tilde{\pi}(\pi(j))} \\ &= \frac{1}{(r!)^2}\sum_{\pi\in S_r}\sum_{\substack{\sigma\in S_r \\ \tilde{\pi}=\sigma\circ\pi^{-1}}} T_{\tilde{\pi}(\pi(j))} = \frac{1}{(r!)^2}\sum_{\pi\in S_r}\sum_{\sigma\in S_r} T_{\sigma(j)} \\ &= \frac{1}{r!}\sum_{\pi\in S_r}\langle T\rangle_j = \langle T\rangle_j \quad , \quad j\in I. \end{aligned} \tag{3-6}$$

Das ist gleichbedeutend mit $\langle T\rangle = T$ für alle $T \in \mathcal{S}_r(V)$. Bezüglich der gewählten ONB kann das Skalarprodukt zwischen zwei Tensoren $T, U \in \mathcal{T}_r(V)$ in der Form $\langle T, U\rangle = \sum_{j\in I} T_j U_j$ geschrieben werden. Damit erhalten wir folgende nutzvolle Eigenschaft:

$$\begin{aligned}\langle\langle T\rangle, U\rangle &= \sum_{j\in I}\frac{1}{r!}\sum_{\pi\in S_r} T_{\pi(j)}U_j = \sum_{i\in I}\frac{1}{r!}\sum_{\pi\in S_r} T_i U_{\pi^{-1}(i)} \\ &= \sum_{i\in I}\frac{1}{r!}\sum_{\substack{\sigma\in S_r \\ \pi=\sigma^{-1}}} T_i U_{\pi^{-1}(i)} = \sum_{i\in I} T_i \frac{1}{r!}\sum_{\sigma\in S_r} U_{\sigma(i)} = \langle T, \langle U\rangle\rangle \end{aligned} \tag{3-7}$$

Dies zeigt, dass die Symmetrisierung einer orthogonalen Projektion des Tensorraumes $\mathcal{T}_r(V)$ auf den Unterraum $\mathcal{S}_r(V)$ entspricht, denn für alle $T \in \mathcal{T}_r(V)$ und alle $U \in \mathcal{S}_r(V)$ gilt mit Hilfe von (3-6) und (3-7)

$$\langle\langle T\rangle - T, U\rangle = \langle\langle T\rangle, U\rangle - \langle T, U\rangle = \langle T, \langle U\rangle\rangle - \langle T, U\rangle = \langle T, U\rangle - \langle T, U\rangle = 0\ .$$

3.3 Tensoren und Polynome

Im Folgenden sei $B = (e_1, \ldots, e_d)$ eine ONB des Vektorraums V bezüglich $\langle\cdot,\cdot\rangle_V$, c_B die zugehörige Koordinatenabbildung und $\langle\cdot,\cdot\rangle$ das induzierte Tensorskalarprodukt auf $\mathcal{T}_r(V)$. Desweiteren sei für ein beliebiges $x \in V$ der Tensor $x^{\otimes r} \in \mathcal{T}_r(V)$ durch $x^{\otimes r} := x \otimes \cdots \otimes x$ definiert. Somit ist der Tensor $x^{\otimes r}$ symmetrisch, d.h. es gilt $x^{\otimes r} \in \mathcal{S}_r(V)$. Mit Hilfe der folgenden Definition ist es nun möglich einen einfachen Zusammenhang zwischen Tensoren und Polynomen herzustellen, indem wir einem beliebigen Tensor $T \in \mathcal{T}_r(V)$ ein homogenes Polynom p_T vom Grad r zuordnen:

$$p_T\big(c_B(x)\big) := \langle x^{\otimes r}, T\rangle \quad , \quad x \in V \tag{3-8}$$

Wir werden sehen, dass der Vektorraum $\mathcal{S}_r(\mathbb{R}^d)$ der symmetrischen Tensoren vom Rang r über $\mathbb{R}^d$ isomorph zum Vektorraum $\mathcal{V}_r(\mathbb{R}^d)$ der homogenen Polynome vom

Grad r über $\mathbb{R}^d$ ist. Da wir mit der obigen Definition zunächst jedem beliebigen Tensor $T \in \mathcal{T}_r(V)$ ein homogenes Polynom vom Grad r zuordnen können, müssen wir, um die Isomorphie zwischen dem Vektorraum der symmetrischen Tensoren und dem Vektorraum der entsprechenden homogenen Polynome zu erhalten, die Symmetriebedingung der Tensoren in Zusammenhang mit den Polynomen bringen. Dies gelingt unter Verwendung von (3-6) und (3-7), denn damit gilt

$$p_T\big(c_B(x)\big) = \langle x^{\otimes r}, T\rangle = \langle\langle x^{\otimes r}\rangle, T\rangle = \langle x^{\otimes r}, \langle T\rangle\rangle = p_{\langle T\rangle}\big(c_B(x)\big).$$

Somit erhalten wir $p_T = p_{\langle T\rangle}$, weshalb die erwähnte Isomorphie nur zwischen $\mathcal{V}_r(\mathbb{R}^d)$ und dem Unterraum $\mathcal{S}_r(\mathbb{R}^d) \subset \mathcal{T}_r(\mathbb{R}^d)$ der symmetrischen Tensoren erwartet werden kann. Dies liefert uns nun folgenden Satz:

3.6 Satz. *Der Vektorraum $\mathcal{S}_r(\mathbb{R}^d)$ der symmetrischen Tensoren vom Rang r über $\mathbb{R}^d$ ist isomorph zum Vektorraum $\mathcal{V}_r(\mathbb{R}^d)$ der homogenen Polynome vom Grad r über $\mathbb{R}^d$.*

Beweis. Mit Hilfe von (3-8) können wir eine bijektive lineare Abbildung φ zwischen den beiden Vektorräumen definieren:

$$\begin{aligned}\varphi : \mathcal{S}_r(\mathbb{R}^d) &\longrightarrow \mathcal{V}_r(\mathbb{R}^d)\\ T &\longmapsto p_T \ , \text{ wobei } p_T\big(c_B(x)\big) := \langle x^{\otimes r}, T\rangle \text{ für } x \in \mathbb{R}^d\end{aligned}$$

Die Linearität dieser Abbildung ergibt sich direkt aus der Linearität des Skalarprodukts. Um die Bijektivität der Abbildung φ zu zeigen, zeigen wir zunächst einmal die Surjektivität. Sei dazu $H : \mathbb{R}^d \longrightarrow \mathbb{R}$ ein beliebiges homogenes Polynom vom Grad r über $\mathbb{R}^d$. Folglich können wir schreiben $H(y) = \sum_{|\alpha|=r} b_\alpha y^\alpha$, wobei $\alpha \in \mathbb{N}_0^d$ ein Multiindex und $b_\alpha \in \mathbb{R}$ der Koeffizient des Monoms $y^\alpha := y_1^{\alpha_1} y_2^{\alpha_2} \cdot \ldots \cdot y_d^{\alpha_d}$ ist, dessen Grad durch $|\alpha| = \sum_{k=1}^d \alpha_k = r$ gegeben ist. Definieren wir $\alpha! := \prod_{k=1}^d \alpha_k!$ für einen Multiindex $\alpha \in \mathbb{N}_0^d$ und

$$T_{j_1 j_2 \ldots j_r} := \frac{\alpha!}{|\alpha|!}\, b_\alpha \quad , \quad \alpha_k := |\{i \mid j_i = k\}| = m(j)_k \tag{3-9}$$

für $j \in I$, so gilt, dass T ein symmetrischer Tensor ist mit $p_T = H$. Dies wird ersichtlich unter Verwendung von $x := \sum_{i=1}^d y_i e_i$ bezüglich der eingangs dieses Abschnitts gewählten ONB, d.h. $c_B(x) = y$, und

$$\begin{aligned} p_T(y) = \langle x^{\otimes r}, T\rangle &= \sum_{j\in I} y_{j_1} \cdot \ldots \cdot y_{j_r} \langle e_{j_1} \otimes \cdots \otimes e_{j_r}, T\rangle \\ &= \sum_{j\in I} T_j y^{m(j)} = \sum_{|\alpha|=r} \Big(\sum_{\substack{j\in I:\\ m(j)=\alpha}} T_j \Big) y^\alpha = \sum_{|\alpha|=r} \frac{\alpha!}{|\alpha|!}\, b_\alpha \Big(\sum_{\substack{j\in I:\\ m(j)=\alpha}} 1 \Big) y^\alpha \ . \end{aligned} \tag{3-10}$$

Um die zweite Summe der rechten Seite von (3-10) ausrechnen zu können, betrachten wir den Tensor $E = e^{\otimes r}$ mit $e = \sum_{k=1}^{d} e_k$. Gemäß (3-10) erhalten wir dann zum einen

$$p_E(y) = \sum_{|\alpha|=r} \Big(\sum_{\substack{j \in I: \\ m(j)=\alpha}} 1 \Big) y^\alpha$$

und zum anderen

$$p_E(y) = \langle x^{\otimes r}, e^{\otimes r} \rangle = \langle x, e \rangle^r = (y_1 + \ldots + y_d)^r$$

gemäß (3-8) (wobei hier das kanonische Skalarprodukt auf $\mathbb{R}^d$ ebenfalls mit $\langle \cdot, \cdot \rangle$ bezeichnet wird). Deshalb erhalten wir für einen beliebigen Multiindex $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = r$ zum einen

$$\begin{aligned} \nabla^\alpha p_E(y) &:= \Big(\frac{\partial}{\partial y_1}\Big)^{\alpha_1} \cdots \Big(\frac{\partial}{\partial y_d}\Big)^{\alpha_d} p_E(y) \\ &= \Big(\frac{\partial}{\partial y_1}\Big)^{\alpha_1} \cdots \Big(\frac{\partial}{\partial y_d}\Big)^{\alpha_d} \Big(\Big(\sum_{\substack{j \in I: \\ m(j)=\alpha}} 1 \Big) y_1^{\alpha_1} \cdot \ldots \cdot y_d^{\alpha_d} \Big) \\ &= \Big(\sum_{\substack{j \in I: \\ m(j)=\alpha}} 1 \Big)\, \alpha! \end{aligned}$$

und zum anderen

$$\nabla^\alpha p_E(y) = r! = |\alpha|! \ .$$

Kombinieren wir nun diese zwei Resultate, so erhalten wir schließlich

$$\sum_{\substack{j \in I: \\ m(j)=\alpha}} 1 = \frac{|\alpha|!}{\alpha!} \ . \tag{3-11}$$

Insgesamt wird (3-10) dadurch vereinfacht zu $p_T(y) = \sum_{|\alpha|=r} b_\alpha y^\alpha = H(y)$. Somit gilt $\varphi(T) = H$, was gleichbedeutend mit der Surjektivität von φ ist.

Für die Injektivität von φ seien $T, U \in \mathcal{S}_r(\mathbb{R}^d)$ zwei beliebig gewählte symmetrische Tensoren mit $p_T = p_U$, d.h. $\varphi(T) = \varphi(U)$. Somit gilt $\nabla^\alpha p_T(y) = \nabla^\alpha p_U(y)$ für jeden Multiindex $\alpha \in \mathbb{N}_0^d$. Sei nun $\alpha \in \mathbb{N}_0^d$ ein beliebiger Multiindex mit $|\alpha| = r$, so liefert diese Gleichheit

$$\alpha! \sum_{\substack{j \in I: \\ m(j)=\alpha}} T_j \ = \ \alpha! \sum_{\substack{j \in I: \\ m(j)=\alpha}} U_j \ .$$

Die Symmetrie von T und U vereinfacht dies umgehend zu

$$|\alpha|!\, T_\alpha = |\alpha|!\, U_\alpha \ .$$

Somit gilt $T_\alpha = U_\alpha$ für jedes beliebige $\alpha \in \mathbb{N}_0^d$ mit $|\alpha| = r$, sprich $T = U$. Dies liefert die Injektivität und somit insgesamt die Bijektivität der Abbildung φ. □

3.7 Bemerkung. Die Dimension des Vektorraumes $\mathcal{V}_r(\mathbb{R}^d)$ der homogenen Polynome vom Grad r über $\mathbb{R}^d$ entspricht der Dimension von $\mathcal{S}_r(\mathbb{R}^d)$ und ist deshalb gegeben durch Bemerkung 3.3.

Betrachten wir den Unterraum $\mathcal{H}_r(\mathbb{R}^d) \subset \mathcal{V}_r(\mathbb{R}^d)$ der homogenen, harmonischen Polynome vom Grad r über $\mathbb{R}^d$, können wir ein weiteres Resultat erzielen, welches die Wichtigkeit der irreduziblen Tensoren zusätzlich hervorhebt. Sei dazu Q ein beliebiges homogenes, harmonisches Polynom vom Grad r über $\mathbb{R}^d$. Aufgrund der Homogenität können wir demnach erneut $Q(y) = \sum_{|\alpha|=r} b_\alpha y^\alpha$ schreiben. Berechnen wir die zweiten partiellen Ableitungen von Q, so erhalten wir

$$\begin{aligned}\partial_i^2 Q(y) &= \sum_{|\alpha|=r} b_\alpha \alpha_i(\alpha_i - 1) y^{\alpha - 2e_i} \\ &= \sum_{|\beta|=r-2} b_{\beta+2e_i}(\beta_i + 2)(\beta_i + 1) y^\beta \ .\end{aligned}$$

Da Q nach Voraussetzung zusätzlich harmonisch sein soll, muss es die Bedingung $\Delta Q := \sum_{i=1}^d \partial_i^2 Q = 0$ erfüllen. Dies führt zur Gleichung

$$\Delta Q(y) = \sum_{|\beta|=r-2} \Big(\sum_{i=1}^d b_{\beta+2e_i}(\beta_i + 2)(\beta_i + 1) \Big) y^\beta = 0$$

und liefert die Bedingung

$$\sum_{i=1}^d b_{\beta+2e_i}(\beta_i + 2)(\beta_i + 1) = 0$$

für alle Multiindizes $\beta \in \mathbb{N}_0^d$ mit $|\beta| = r - 2$. Aus dieser Bedingung können wir über den Isomorphismus φ aus Satz 3.6 unter Benutzung von (3-9) die entsprechende äquivalente Bedingung an die Tensorkomponenten des zu Q gehörenden Tensors T herleiten:

$$\sum_{i=1}^d \frac{r!}{(\beta + 2e_i)!} \, (\beta_i + 2)(\beta_i + 1) \, T_{\beta+2e_i} \ = \ \frac{r!}{\beta!} \, \sum_{i=1}^d T_{\beta+2e_i} \ = \ 0$$

Schließlich erhalten wir für den Tensor T die Bedingung $\sum_{i=1}^d T_{\beta+2e_i} = 0$ für alle Multiindizes $\beta \in \mathbb{N}_0^d$ mit $|\beta| = r - 2$. Diese Bedingung beschreibt genau die Eigenschaft der Spurfreiheit des Tensors T, d.h. dass $T \in \mathcal{J}_r(\mathbb{R}^d)$ gelten muss.

Insgesamt erhalten wir demnach mit Hilfe von Satz 3.6 und der eben gemachten Beobachtung bereits den Beweis des folgenden Satzes:

3.8 Satz. *Der Vektorraum $\mathcal{J}_r(\mathbb{R}^d)$ der irreduziblen Tensoren vom Rang r über $\mathbb{R}^d$ ist isomorph zum Vektorraum $\mathcal{H}_r(\mathbb{R}^d)$ der homogenen, harmonischen Polynome vom Grad r über $\mathbb{R}^d$.*

3.9 Bemerkung. Die Dimension des Vektorraumes $\mathcal{H}_r(\mathbb{R}^d)$ der homogenen, harmonischen Polynome vom Grad r über $\mathbb{R}^d$ entspricht der Dimension von $\mathcal{J}_r(\mathbb{R}^d)$ und ist deshalb gegeben durch Bemerkung 3.5.

3.4 Konstruktion einer Basis der irreduziblen Tensoren

Ein Blick auf die Dimensionstabelle 3.1 zeigt, dass es aus numerischer Sicht äußerst sinnvoll ist, alle Tensorrechnungen direkt in der Basis irreduzibler Tensoren durchzuführen. Somit erspart man sich aufgrund der deutlich weniger unabhängigen Komponenten im Vergleich zum allgemeinen Tensorraum erheblichen Mehraufwand an Rechenzeit. Um einen irreduziblen Tensor aus $\mathcal{T}_r(\mathbb{R}^d)$ auf diejenigen unabhängigen Komponenten zu reduzieren, die notwendig sind, um den Tensor in irreduzibler Dimension darzustellen, reduzieren wir den Tensor zunächst einmal auf die bekannten Komponenten symmetrischer Dimension. Die Bedingung, die einen symmetrischen Tensor irreduzibel macht, ist die Spurfreiheit. Nach (3-5) ist die Spur eines beliebigen Tensors $T \in \mathcal{S}_r(\mathbb{R}^d)$ mit $r \geq 2$ durch

$$(\operatorname{spur} T)_\alpha \;=\; \sum_{i=1}^{d} T_{\alpha+2e_i} \quad , \quad |\alpha| = r-2$$

gegeben. Die zusätzliche Anzahl an Nebenbedingungen an einen symmetrischen Tensor, um spurfrei zu sein, ist nach Bemerkung 3.3 durch $N(r-2,d)$ gegeben. Diese $N(r-2,d)$ Bedingungen sind durch linear unabhängige, lineare Funktionale aus $\mathcal{S}_r(\mathbb{R}^d)^*$ gegeben, die, angewandt auf einen symmetrischen Tensor, dessen Spurkomponenten ergeben. Im Falle eines irreduziblen Tensors also jeweils 0. Diese linearen Funktionale können wir als Zeilenvektoren in eine Matrix der Dimension $N(r-2,d) \times N(r,d)$ schreiben. Füllen wir diese Matrix nach oben auf bis zur vollen Dimension $N(r,d) \times N(r,d)$, indem wir die linear unabhängigen, linearen Funktionale mit weiteren linearen Funktionalen zu einer Basis des Dualraumes $\mathcal{S}_r(\mathbb{R}^d)^*$ ergänzen, so erhalten wir eine invertierbare Matrix C, deren $N(r-2,d)$ untersten Zeilen die linearen Funktionale enthalten, welche die einzelnen Spurkomponenten eines symmetrischen Tensors berechnen. Invertieren wir diese Matrix C, so erhalten wir mit den ersten $N(r,d)-N(r-2,d)$ Spalten von C^{-1} linear unabhängige Koordinatenvektoren irreduzibler Tensoren in symmetrischer Dimension und haben somit eine Basis des Vektorraumes $\mathcal{J}_r(\mathbb{R}^d)$ der irreduziblen Tensoren vom Rang r über $\mathbb{R}^d$ in symmetrischer Dimension gefunden. Im Folgenden sei diese Vorgehensweise für den Fall $d = 3$, $r = 4$ skizziert:

Die Multiindizes, mit denen man einen symmetrischen Tensor T vom Rang $r = 4$ in symmetrischer Dimension für $d = 3$ komplett beschreiben kann, sind der Reihe nach von links nach rechts und von oben nach unten gegeben durch

$$(400), (310), (301), (220), (211), (202), (130), (121),$$
$$(112), (103), (040), (031), (022), (013), (004).$$

Um die Spur von T berechnen zu können, benötigen wir noch alle Multiindizes zum Rang 2, welche der Reihe nach durch $(200), (110), (101), (020), (011), (002)$ gegeben sind. Somit lautet die erste Komponente des Spurtensors zum Multiindex $\alpha = (200)$

$$(\operatorname{spur} T)_{(200)} \;=\; \sum_{i=1}^{3} T_{(200)+2e_i} \;=\; T_{(400)} + T_{(220)} + T_{(202)} \;.$$

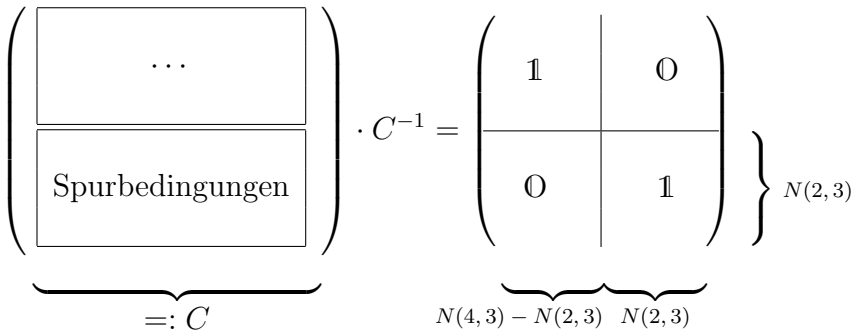
Formulieren wir die restlichen Spurbedingungen analog (untere Blockmatrix) und füllen diese durch entsprechende Zeilen der kanonischen Dualbasis nach oben auf (obere Blockmatrix), so erhalten wir in diesem Fall folgende invertierbare Matrix C und folgende Beziehung:

$$\underbrace{\left(\begin{array}{c}
\boxed{\begin{array}{cccccccccccccccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0
\end{array}}\\
\boxed{\begin{array}{cccccccccccccccc}
1&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&1&0&1&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&1&0&1&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&1&0&1&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&1&0&1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&0\\
0&0&0&0&0&1&0&0&0&0&0&0&1&0&1&1
\end{array}}
\end{array}\right)}_{=:\,C}
\begin{pmatrix}
T_{(400)}\\ T_{(310)}\\ T_{(301)}\\ T_{(220)}\\ T_{(211)}\\ T_{(202)}\\ T_{(130)}\\ T_{(121)}\\ T_{(112)}\\ T_{(103)}\\ T_{(040)}\\ T_{(031)}\\ T_{(022)}\\ T_{(013)}\\ T_{(004)}
\end{pmatrix}
=
\left.\begin{array}{c}
\begin{pmatrix}\vdots\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ \vdots\end{pmatrix}\\
\hline
\begin{pmatrix}\vdots\\ \vdots\\ \vdots\\ \vdots\\ \vdots\\ \vdots\end{pmatrix}
\end{array}\right\}\ \operatorname{spur} T$$

Aufgrund von $CC^{-1} = \mathbb{1}$ ist es ersichtlich, dass wir mit den ersten

$$N(4,3) - N(2,3) = 15 - 6 = 9$$

Spalten von C^{-1} eine Basis von $\mathcal{J}_4(\mathbb{R}^3)$ in symmetrischer Dimension erhalten. Die Spurfreiheit dieser Basistensoren ist in der folgenden Veranschaulichung direkt am Nullblock unten links in der Einheitsmatrix abzulesen:



Betrachten wir nun die Matrix U der Dimension $N(r,d) \times \big(N(r,d) - N(r-2,d)\big)$, welche genau diese Basis des $\mathcal{J}_r(\mathbb{R}^d)$ in symmetrischer Dimension in ihren Spalten enthält, so gilt für einen beliebigen **irreduziblen** Tensor T folgender Zusammenhang:

$$T_{\mathcal{S}} = UT_{\mathcal{J}} \ ,$$

wobei $T_{\mathcal{S}}$ die Reduktion von T auf die unabhängigen Komponenten in symmetrischer Dimension beschreibt, und $T_{\mathcal{J}}$ entsprechend die Reduktion von T auf die unabhängigen Komponenten in irreduzibler Dimension. Über diesen Zusammenhang erhalten wir demnach die Information, welche Komponenten aus einem irreduziblen Tensor in symmetrischer Dimension herauszugreifen sind, um den Tensor in irreduzibler Dimension angeben zu können.

3.4.1 Berechnung des Tensorskalarproduktes

Mit den Überlegungen des vorherigen Abschnittes ist es natürlich aus Rechenaufwandsgründen ebenso sinnvoll, das Skalarprodukt zweier symmetrischer Tensoren bzw. zweier irreduzibler Tensoren auf die jeweils notwendigen unabhängigen Tensorkomponenten zu beschränken. Dazu betrachten wir nun zwei beliebige irreduzible Tensoren $S, T \in \mathcal{T}_r(\mathbb{R}^d)$, die wir in symmetrischer Dimension erneut durch $S_{\mathcal{S}}$ und $T_{\mathcal{S}}$ bzw. in irreduzibler Dimension entsprechend durch $S_{\mathcal{J}}$ und $T_{\mathcal{J}}$ darstellen können. Mit Hilfe von (3-11) erhalten wir jeweils die Anzahl an Tensorindizes, die zum selben Multiindex gehören. Definieren wir eine Diagonalmatrix M in symmetrischer Dimension $N(r,d) \times N(r,d)$, die diese Anzahlen, entsprechend der Reihenfolge der Multiindizes sortiert, auf der Diagonalen trägt, so erhalten wir folgenden Zusammenhang zwischen den Skalarprodukten der irreduziblen Tensoren S und T in voller bzw. symmetrischer Dimension:

$$\langle S, T \rangle = \langle MS_{\mathcal{S}}, T_{\mathcal{S}} \rangle_{\mathcal{S}_r(\mathbb{R}^d)}$$

Mit der Matrix U des vorherigen Abschnittes gilt $S_{\mathcal{S}} = US_{\mathcal{J}}$ bzw. $T_{\mathcal{S}} = UT_{\mathcal{J}}$ und somit

$$\langle S, T \rangle = \langle MS_{\mathcal{S}}, T_{\mathcal{S}} \rangle_{\mathcal{S}_r(\mathbb{R}^d)} = \langle MUS_{\mathcal{J}}, UT_{\mathcal{J}} \rangle_{\mathcal{S}_r(\mathbb{R}^d)}$$

$$= \langle \underbrace{U^T M U}_{=:W} S_{\mathcal{J}}, T_{\mathcal{J}} \rangle_{\mathcal{J}_r(\mathbb{R}^d)} = \langle W S_{\mathcal{J}}, T_{\mathcal{J}} \rangle_{\mathcal{J}_r(\mathbb{R}^d)}$$

Mit den Gewichtsmatrizen M bzw. W lässt sich das jeweilige Skalarprodukt in symmetrischer bzw. irreduzibler Dimension demnach sehr effizient berechnen. Da die Gewichtsmatrizen unabhängig von den Tensoren S und T sind, sondern lediglich von r und d abhängen, können diese Matrizen im Vorfeld einer Rechnung einmalig berechnet und abgespeichert werden. Im Folgenden wird aus Einfachheitsgründen weiterhin das Standardskalarprodukt auf $\mathcal{T}_r(\mathbb{R}^d)$ ohne Gewichtsmatrizen verwendet, denn die Gewichtsmatrizen dienen lediglich einer schnelleren Berechnung.

3.5 Darstellung von $SO(d)$ auf $\mathcal{T}_r(\mathbb{R}^d)$

Im Folgenden betrachten wir eine Darstellung der kompakten Gruppe $SO(d)$ auf $\mathcal{T}_r(\mathbb{R}^d)$, welche für beliebige $Q \in SO(d)$, $T \in \mathcal{T}_r(\mathbb{R}^d)$ und $\nu_i \in (\mathbb{R}^d)^*$ für $i = 1, \ldots, r$ definiert ist durch

$$\begin{aligned} D : SO(d) \times \mathcal{T}_r(\mathbb{R}^d) &\longrightarrow \mathcal{T}_r(\mathbb{R}^d) \\ (Q, T) &\longmapsto D_Q T, \text{ wobei } (D_Q T)(\nu_1, \ldots, \nu_r) := T(\nu_1 Q, \ldots, \nu_r Q) \ . \end{aligned} \tag{3-12}$$

Man beachte, dass zum einen $D_{\mathbb{1}} T = T$ gilt und zum anderen die Bedingung $D_{QR} T = D_Q D_R T$ erfüllt ist[1]. Für $i \in I$ können wir $(D_Q T)_i$ mit Hilfe der Tensorkomponenten $T_i = T(e_{i_1}^*, \ldots, e_{i_r}^*)$ bezüglich der ONB $e_1, \ldots, e_d$ schreiben als $(D_Q T)_i = T(e_{i_1}^* Q, \ldots, e_{i_r}^* Q)$. Aufgrund von

$$(e_{i_1}^* Q)(x) = (Qx)_{i_1} = \sum_{j_1=1}^{d} Q_{i_1 j_1} x_{j_1} = \sum_{j_1=1}^{d} Q_{i_1 j_1} e_{j_1}^*(x) \quad \text{für } x \in \mathbb{R}^d$$

und der Multilinearität eines Tensors erhalten wir schließlich

$$(D_Q T)_i = \sum_{j \in I} Q_{i_1 j_1} \cdot \ldots \cdot Q_{i_r j_r} T_j =: (Q * T)_i \ , \tag{3-13}$$

wobei wir mit $Q * T$ das sogenannte Rayleigh-Produkt einer orthogonalen Matrix $Q \in SO(d)$ und einem Tensor $T \in \mathcal{T}_r(\mathbb{R}^d)$ abkürzen. Das Rayleigh-Produkt $Q * T$ ist auch als die Drehung des Tensors T mit der orthogonalen Matrix Q zu interpretieren. Im Folgenden tragen wir einige wichtige Eigenschaften dieser Darstellung zusammen.

3.10 Lemma. *Die Darstellung D ist unitär bezüglich dem kanonischen Skalarprodukt auf $\mathcal{T}_r(\mathbb{R}^d)$, d.h. für beliebige $Q \in SO(d)$ und beliebige $S, T \in \mathcal{T}_r(\mathbb{R}^d)$ gilt:*

$$\langle D_Q S, D_Q T \rangle = \langle S, T \rangle$$

[1] $(D_Q D_R T)(\nu_1, \ldots, \nu_r) = (D_R T)(\nu_1 Q, \ldots, \nu_r Q) = T(\nu_1 QR, \ldots, \nu_r QR) = (D_{QR} T)(\nu_1, \ldots, \nu_r)$

Beweis.

$$\begin{aligned}\langle D_Q S, D_Q T\rangle &= \sum_{i\in I}(D_Q S)_i (D_Q T)_i = \sum_{i,j,k\,\in\, I} Q_{i_1 j_1} Q_{i_1 k_1} \cdot \ldots \cdot Q_{i_r j_r} Q_{i_r k_r} S_j T_k \\ &= \sum_{k\in I}\Big(\sum_{j\in I}\delta_{k_1 j_1}\cdot\ldots\cdot\delta_{k_r j_r} S_j\Big) T_k = \sum_{k\in I} S_k T_k = \langle S, T\rangle\end{aligned}$$

□

3.11 Korollar. *Für beliebige $Q \in SO(d)$ gilt $D_Q^T = D_{Q^T}$.*

Beweis. Sei $Q \in SO(d)$ beliebig gewählt, so gilt für beliebige $S, T \in \mathcal{T}_r(\mathbb{R}^d)$ unter Verwendung von Lemma 3.10 die Bedingung

$$\begin{aligned}\langle D_Q S, T\rangle &= \langle D_Q S, D_{\mathbb{1}} T\rangle = \langle D_Q S, D_{QQ^T} T\rangle \\ &= \langle D_Q S, D_Q D_{Q^T} T\rangle = \langle S, D_{Q^T} T\rangle\ .\end{aligned}$$

Somit folgt die behauptete Aussage $D_Q^T = D_{Q^T}$. □

3.12 Lemma. *Für beliebige $Q \in SO(d)$ und beliebige $x \in \mathbb{R}^d$ gilt:*

$$D_Q x^{\otimes r} = (Qx)^{\otimes r}$$

Beweis. Seien $Q \in SO(d)$ und $x \in \mathbb{R}^d$ beliebig gewählt, dann gilt für beliebige $\nu_1, \ldots, \nu_r \in (\mathbb{R}^d)^*$:

$$\begin{aligned}(D_Q x^{\otimes r})(\nu_1, \ldots, \nu_r) &= x^{\otimes r}(\nu_1 Q, \ldots, \nu_r Q) \\ &= \nu_1(Qx)\cdot\ldots\cdot\nu_r(Qx) = (Qx)^{\otimes r}(\nu_1, \ldots, \nu_r)\end{aligned}$$

□

Berechnen wir das homogene Polynom, welches dem Tensor $D_Q T$ für $Q \in SO(d)$ und $T \in \mathcal{T}_r(\mathbb{R}^d)$ gemäß (3-8) zugeordnet wird, so erhalten wir folgende nützliche Aussage:

3.13 Lemma. *Es sei $Q \in SO(d)$ und $T \in \mathcal{T}_r(\mathbb{R}^d)$ beliebig gewählt, so gilt für das dem Tensor $D_Q T \in \mathcal{T}_r(\mathbb{R}^d)$ zugeordnete homogene Polynom $p_{D_Q T}$ für alle $x \in \mathbb{R}^d$:*

$$p_{D_Q T}\big(c_B(x)\big) = p_T\big(c_B(Q^T x)\big)$$

Wählen wir im $\mathbb{R}^d$ die kanonische ONB, d.h. gilt $c_B(x) = x$, so gilt folglich

$$p_{D_Q T} = p_T \circ Q^T\ .$$

Beweis. Seien $Q \in SO(d)$ und $T \in \mathcal{T}_r(\mathbb{R}^d)$ beliebig gewählt, so gilt mit Hilfe von Korollar 3.11 und Lemma 3.12 für alle $x \in \mathbb{R}^d$:

$$\begin{aligned} p_{D_Q T}\big(c_B(x)\big) &= \langle x^{\otimes r}, D_Q T\rangle = \langle D_{Q^T} x^{\otimes r}, T\rangle \\ &= \langle (Q^T x)^{\otimes r}, T\rangle = p_T\big(c_B(Q^T x)\big) \end{aligned}$$

□

Im folgenden Abschnitt betrachten wir spezielle Zusammenhänge im Fall $d = 3$, welche jedoch allesamt auch im allgemeinen d-dimensionalen Fall Gültigkeit besitzen.

3.5.1 Zerlegung von $\mathbb{L}^2\big(SO(3)\big)$

Ausgehend von den homogenen, harmonischen Polynomen vom Grad r über der Einheitssphäre S^2 aus Kapitel 2.3.1 und der dortigen Darstellung

$$\begin{aligned} \varrho : SO(3) \times \mathcal{H}_r(S^2) &\longrightarrow \mathcal{H}_r(S^2) \\ (Q, f) &\longmapsto \varrho_Q f \text{ mit } (\varrho_Q f)(x) := f(Q^T x),\ x \in S^2 \end{aligned}$$

berechnen wir die zu ϱ äquivalente Darstellung $\overline{\varrho} : SO(3) \times \mathcal{J}_r(\mathbb{R}^3) \longrightarrow \mathcal{J}_r(\mathbb{R}^3)$, welche nach Definition 2.7 mit Hilfe der unitären Abbildung $A : \mathcal{H}_r(S^2) \longrightarrow \mathcal{J}_r(\mathbb{R}^3)$ aus Satz 3.8 für alle $Q \in SO(3)$ definiert ist durch $\overline{\varrho}_Q := A\varrho_Q A^{-1}$.

Sei dazu $T \in \mathcal{J}_r(\mathbb{R}^3)$ beliebig gewählt, so gilt mit Hilfe von (3-8) und Lemma 3.13 für alle $Q \in SO(3)$ und alle $x \in S^2$ zunächst

$$(\varrho_Q p_T)(x) = p_T(Q^T x) = p_{D_Q T}(x)\ .$$

Damit erhalten wir schließlich

$$\overline{\varrho}_Q T = A\varrho_Q A^{-1} T = A\varrho_Q p_T = A p_{D_Q T} = D_Q T\ . \tag{3-14}$$

Daraus erhalten wir die Information, dass die Unterräume $\mathcal{S}_r(\mathbb{R}^3) \subset \mathcal{T}_r(\mathbb{R}^3)$ und $\mathcal{J}_r(\mathbb{R}^3) \subset \mathcal{S}_r(\mathbb{R}^3)$ jeweils D-invariant sind. Dies hat zur Folge, dass wir mit D ab sofort für den Rest dieser Arbeit die auf $\mathcal{J}_r(\mathbb{R}^3)$ eingeschränkte Darstellung

$$\begin{aligned} D : SO(3) \times \mathcal{J}_r(\mathbb{R}^3) &\longrightarrow \mathcal{J}_r(\mathbb{R}^3) \\ (Q, T) &\longmapsto D_Q T := Q * T \end{aligned} \tag{3-15}$$

bezeichnen. Diese eingeschränkte Darstellung D wird im Laufe dieser Arbeit noch von sehr großer Bedeutung sein.

Die Beziehung (3-14) zeigt uns desweiteren, dass die Darstellungen ϱ und D zueinander äquivalent sind. Mit Hilfe von Satz 2.11 erhalten wir die Irreduzibilität von D

demnach direkt aus der bereits aus Kapitel 2.3 bekannten Irreduzibilität von ϱ. Aufgrund der Zugehörigkeit zur selben Äquivalenzklasse generieren ϱ und D dieselben Unterräume $U_r \subset \mathbb{L}^2\big(SO(3)\big)$ und somit eine identische Zerlegung von $\mathbb{L}^2\big(SO(3)\big)$. Im Gegensatz zu $\mathcal{H}_r(S^2)$ hat der Raum $\mathcal{J}_r(\mathbb{R}^3)$ jedoch keine traditionell ausgezeichnete Basis wie etwa die Kugelflächenfunktionen. Trotzdem können wir eine Basis der Unterräume U_r generieren, indem wir im Falle der Darstellung D durch eine geeignete Wahl von Tensorkomponenten wie in Kapitel 3.4 eine Alternative zu den WIGNER-Funktionen konstruieren.[6]

Definieren wir zunächst ausgehend von den Kugelflächenfunktionen Y_r^m mit Hilfe der obigen unitären Abbildung A die irreduziblen Tensoren $T^m := AY_r^m \in \mathcal{J}_r(\mathbb{R}^3)$ für $m \in \{-r, \dots, r\}$, so erhalten wir durch die Matrixeinträge b_{mn} der linearen Abbildung ϱ_Q in der Form

$$\begin{aligned} b_{mn}(Q) &= \langle \varrho_Q Y_r^n, Y_r^m \rangle_{\mathbb{L}^2(S^2)} = \langle A^{-1} D_Q A Y_r^n, Y_r^m \rangle_{\mathbb{L}^2(S^2)} \\ &= \langle D_Q A Y_r^n, A Y_r^m \rangle = \langle D_Q T^n, T^m \rangle \\ &= \langle Q * T^n, T^m \rangle = \langle T^m, Q * T^n \rangle \end{aligned}$$

für alle $Q \in SO(3)$ jene Funktionen, die den $\mathcal{D}$-invarianten Unterraum U_r von $\mathbb{L}^2\big(SO(3)\big)$ erzeugen. Die Basisfunktionen von U_r sind demnach von der Form

$$\begin{aligned} d_{S,T} : SO(3) &\longrightarrow \mathbb{R} \\ Q &\longmapsto d_{S,T}(Q) := \langle S, Q * T \rangle \end{aligned}$$

mit $d_{S,T} \in \mathbb{L}^2\big(SO(3)\big)$ für $S, T \in \mathcal{J}_r(\mathbb{R}^3)$. Mit Hilfe des folgenden Satzes gelingt es uns, eine ONB der $\mathcal{D}$-invarianten Unterräume U_r anzugeben. Desweiteren liefert er eine Aussage, wie man die Unterräume U_r noch weiter zerlegen kann, und zwar in orthogonale, invariante und irreduzible Teilräume von $\mathbb{L}^2\big(SO(3)\big)$ bezüglich der regulären Darstellung $\mathcal{D}$. Dieses Resultat ist ebenso auf den d-dimensionalen Fall übertragbar und lässt sich sogar allgemein für kompakte Gruppen G formulieren.

3.14 Satz. *Sei $n := 2r + 1$ und $E_1, \dots, E_n$ eine ONB von $\mathcal{J}_r(\mathbb{R}^3)$. Desweiteren sei $m_{ij}(Q) := \langle E_i, D_Q E_j \rangle$ für $Q \in SO(3)$ die Matrixdarstellung der linearen Abbildung D_Q in dieser Basis. Dann spannen die Funktionen m_{ij} n n-dimensionale, orthogonale, invariante und irreduzible Unterräume von $U_r \subset \mathbb{L}^2\big(SO(3)\big)$ bezüglich der regulären Darstellung $\mathcal{D}$ auf. Dabei bilden die $\{m_{ij} \mid i = 1, \dots, n\}$ für ein festes $j \in \{1, \dots, n\}$ jeweils eine orthogonale Basis eines solchen Unterraumes, mit Normierungsfaktor $\sqrt{n}$ sogar eine ONB, d.h. es gilt*

$$\langle \sqrt{n}\, m_{ij}, \sqrt{n}\, m_{kl} \rangle_{\,\mathbb{L}^2(SO(3))} \;=\; \delta_{ik}\delta_{jl} \;.$$

Beweis. Sei also $n := 2r+1$ und $E_1, \dots, E_n$ eine ONB von $\mathcal{J}_r(\mathbb{R}^3)$. Dann ist zunächst einmal zu beachten, dass $U_r = \operatorname{span}\{m_{ij} \mid i, j = 1, \dots, n\}$ mit

$$m_{ij}(Q) := \langle E_i, D_Q E_j \rangle = \langle E_i, Q * E_j \rangle = d_{E_i, E_j}(Q)$$

gilt. Mit Hilfe des Buches von SIMON[33] erhalten wir direkt die Orthogonalität der m_{ij}, bzw. dass $\langle \sqrt{n}\ m_{ij}, \sqrt{n}\ m_{kl} \rangle_{\ \mathbb{L}^2(SO(3))} = \delta_{ik}\delta_{jl}$ gilt. Mit den normierten m_{ij} haben wir somit sogar eine ONB des Unterraumes U_r gefunden. Daraus können wir direkt folgern, dass $\dim U_r = n^2 = (2r+1)^2$ gilt, U_r also ein $(2r+1)^2$-dimensionaler Unterraum von $\mathbb{L}^2\big(SO(3)\big)$ ist.

Betrachten wir nun die reguläre Darstellung $\mathcal{D} : SO(3) \times \mathbb{L}^2\big(SO(3)\big) \longrightarrow \mathbb{L}^2\big(SO(3)\big)$, die wie bereits bekannt durch $(\mathcal{D}_R f)(Q) := f(R^T Q)$ gegeben ist, und für ein beliebiges $j \in \{1, \ldots, n\}$ ein beliebiges Element $m \in U_r^j := \operatorname{span}\{m_{1j}, \ldots, m_{nj}\}$, d.h. es ist $m(Q) := \langle T, D_Q E_j \rangle = d_{T,E_j}(Q)$ für alle $Q \in SO(3)$ und ein $T \in \mathcal{J}_r(\mathbb{R}^3)$, so gilt

$$(\mathcal{D}_R m)(Q) = m(R^T Q) = \langle T, D_{R^T Q} E_j \rangle = \langle \underbrace{D_R T}_{\in \mathcal{J}_r(\mathbb{R}^3)}, D_Q E_j \rangle =: \overline{m}(Q)$$

mit $\overline{m} \in U_r^j$. Die n Unterräume $U_r^j \subset U_r$ sind folglich jeweils n-dimensional, zueinander orthogonal aufgrund der paarweisen Orthogonalität der m_{ij}, $\mathcal{D}$-invariant und entsprechen demnach den von den Spalten der darstellenden Matrix erzeugten Unterräume von $\mathbb{L}^2\big(SO(3)\big)$.

Bleibt lediglich die Irreduzibilität dieser Unterräume zu zeigen, d.h. dass es keine nichttrivialen $\mathcal{D}$-invarianten Unterräume von U_r^j gibt. Dazu nehmen wir an, dass $W_j \subset U_r^j$ ein $\mathcal{D}$-invarianter Unterraum sei. Definieren wir

$$V_j := \{T \in \mathcal{J}_r(\mathbb{R}^3) \mid m_T : Q \longmapsto \langle T, D_Q E_j \rangle \ \in \ W_j\} \ ,$$

so ist V_j ein Unterraum von $\mathcal{J}_r(\mathbb{R}^3)$ mit $V_j \neq \emptyset$, denn es ist $0 \in V_j$ und mit $S, T \in V_j$ folgt $m_{S+T} = m_S + m_T \in W_j$, sprich $S+T \in V_j$. Desweiteren gilt für ein $T \in V_j$ und ein beliebiges $\lambda \in \mathbb{R}$ auch $\lambda m_T \in W_j$ und daher $m_{\lambda T} = \lambda m_T \in W_j$, also $\lambda T \in V_j$. Außerdem ist V_j D-invariant, denn für ein beliebiges $T \in V_j$ ist auch $D_R T \in V_j$ für jedes $R \in SO(3)$, da

$$m_{D_R T}(Q) = \langle D_R T, D_Q E_j \rangle = \langle T, D_{R^T Q} E_j \rangle = m_T(R^T Q) = (\mathcal{D}_R m_T)(Q) \ ,$$

und somit $m_{D_R T} \in W_j$ aufgrund der vorausgesetzten $\mathcal{D}$-Invarianz von W_j. Da die Darstellung D irreduzibel ist, gilt $V_j \in \big\{\{0\}, \mathcal{J}_r(\mathbb{R}^3)\big\}$ und somit $W_j \in \big\{\{0\}, U_r^j\big\}$. Dies ist aber gleichbedeutend mit der Irreduzibilität der Unterräume U_r^j. □

Mit Hilfe dieses Satzes erhalten wir demnach eine weitere Zerlegung der Unterräume U_r der folgenden Form:

$$U_r = U_r^1 \oplus \ldots \oplus U_r^{2r+1}$$

Dabei sind die Unterräume U_r^j für $j = 1, \ldots, 2r+1$ jeweils von der Dimension $2r+1$ und irreduzibel bezüglich $\mathcal{D}$, wobei die Zerlegung von U_r in diese irreduziblen Unterräume im Allgemeinen nicht eindeutig ist. Dies ist leicht nachvollziehbar, denn

je nach Basiswahl im Ausgangsvektorraum zweier irreduziblen Darstellungen von $SO(3)$ aus derselben Äquivalenzklasse kann man unterschiedliche Matrixeinträge in den darstellenden Matrizen der entsprechenden linearen Abbildungen erhalten. Dies kann wiederum zur Folge haben, dass die jeweiligen Spalten unterschiedliche Räume erzeugen. Der $\mathcal{D}$-invariante Oberraum U_r ist jedoch wie bereits erläutert für alle irreduziblen Darstellungen derselben Äquivalenzklasse identisch. Der Beweis dieses Satzes zeigt ebenso, dass für $n := 2r + 1$

$$\begin{aligned} U_r &= \operatorname{span}\left\{ d_{E_i,E_j} \in \mathbb{L}^2\big(SO(3)\big) \mid \{E_1, \ldots, E_n\} \text{ ist ONB von } \mathcal{J}_r(\mathbb{R}^3) \right\} \\ &= \operatorname{span}\left\{ d_{S,T} \in \mathbb{L}^2\big(SO(3)\big) \mid S, T \in \mathcal{J}_r(\mathbb{R}^3) \right\} \end{aligned}$$

gilt, denn es ist $d_{S,T} = \sum_{i,j=1}^{n} c_{ij}(S,T) d_{E_i,E_j}$ für alle $S, T \in \mathcal{J}_r(\mathbb{R}^3)$ mit den entsprechenden Koordinaten $c_{ij}(S,T)$ bezüglich der Basis d_{E_i,E_j} von U_r. Hierbei ist anzumerken, dass dies jeweils nur für den Fall $r \geq 2$ gilt, da die irreduziblen Tensoren aufgrund der Definition der Spur ja nur in diesen Fällen definiert sind. Der Zusammenhang zwischen Tensoren und homogenen Polynomen zeigt jedoch, dass wir in den beiden Fällen $r = 0$ und $r = 1$ (d.h. in den Fällen von konstanten bzw. linearen Polynomen) entsprechend

$$U_0 = \operatorname{span}\left\{1\right\} \quad \text{und} \quad U_1 = \operatorname{span}\left\{ d_{S,T} \in \mathbb{L}^2\big(SO(3)\big) \mid S, T \in \mathcal{T}_1(\mathbb{R}^3) \right\}$$

erhalten. Man beachte, dass nach wie vor $\dim U_r = (2r+1)^2$ für alle $r \in \mathbb{N}_0$ gilt.

Wie bereits erwähnt wurde, ermöglicht es uns die Verwendung einer tensoriellen Basis auf einfache Art und Weise, Unterräume von U_r mit zusätzlichen Symmetrien zu konstruieren. Dies werden wir benötigen, um die entsprechend vorliegende Kristallsymmetrie des betrachteten Metalls in die später gesuchte codf einbauen zu können. Dazu betrachten wir zunächst einmal für eine Untergruppe H von $SO(3)$ den Unterraum $\mathbb{L}^2_H\big(SO(3)\big) \subset \mathbb{L}^2\big(SO(3)\big)$, welcher durch

$$\mathbb{L}^2_H\big(SO(3)\big) := \left\{ f \in \mathbb{L}^2\big(SO(3)\big) \mid \forall R \in H : \ \triangle_R f = f \right\}$$

mit der Symmetriebedingung $(\triangle_R f)(Q) := f(QR) \stackrel{!}{=} f(Q)$ für alle $R \in H$ und alle $Q \in SO(3)$ gegeben ist, und somit all diejenigen Funktionen aus $\mathbb{L}^2\big(SO(3)\big)$ beinhaltet, die zum Beispiel die gewünschte Kristallsymmetrie aufweisen. Zunächst einmal wollen wir zeigen, dass sich diese Symmetriebedingung direkt auf die Unterräume U_r übertragen lässt. Dazu verwenden wir nun die Zerlegung

$$\mathbb{L}^2\big(SO(3)\big) = \bigoplus_{r \in \mathbb{N}_0} U_r$$

in die Unterräume U_r, welche in der Form der obigen linearen Hüllen gegeben sind. Einerseits wissen wir bereits, dass diese Unterräume $\mathcal{D}$-invariant sind, andererseits können wir zeigen, dass sie sogar $\triangle_R$-invariant sind für alle $R \in SO(3)$, denn für ein beliebiges $d_{S,T} \in U_r$ gilt:

$$\begin{aligned}(\triangle_R d_{S,T})(Q) &= d_{S,T}(QR) = \langle S, (QR) * T\rangle = \langle S, D_{QR}T\rangle \\ &= \langle S, D_Q D_R T\rangle = \langle S, Q * D_R T\rangle = d_{S,D_RT}(Q)\end{aligned}$$

Bezeichnen wir in $\mathbb{L}^2\big(SO(3)\big)$ mit P_r die zugehörige Projektion auf den Unterraum U_r, so können wir für ein beliebiges $f \in \mathbb{L}^2\big(SO(3)\big)$ und alle $R \in SO(3)$ mit Hilfe der $\triangle_R$-Invarianz das Folgende festhalten:

$$\begin{aligned}P_r\triangle_R f = P_r\triangle_R \sum_j P_j f &= P_r \sum_j \underbrace{\triangle_R P_j f}_{=P_j\triangle_R P_j f} \\ &= \sum_j \underbrace{P_r P_j}_{=\delta_{rj}P_r}\triangle_R P_j f = P_r \underbrace{\triangle_R P_r f}_{\in U_r} = \triangle_R P_r f\end{aligned}$$

Es ist also $P_r\triangle_R = \triangle_R P_r$. Zerlegen wir nun ein $f \in \mathbb{L}^2_H\big(SO(3)\big)$ in $f = \sum_{j\in\mathbb{N}_0} f_j$ mit $f_j := P_j f$, so gilt für alle $R \in H$ die Bedingung $\triangle_R f_r = f_r$, denn es ist

$$\triangle_R f_r = \triangle_R P_r f = P_r \triangle_R f \overset{f\in\mathbb{L}^2_H}{=} P_r f = f_r \ .$$

Die Symmetriebedingung lässt sich demnach entsprechend auf die Unterräume U_r übertragen. Somit genügt es anstelle des Symmetrieraumes $\mathbb{L}^2_H\big(SO(3)\big)$ die Räume $\mathbb{L}^2_H\big(SO(3)\big) \cap U_r$ für $r \in \mathbb{N}_0$ zu untersuchen. Im Folgenden suchen wir nun für $r \geq 2$ ein $f \in \mathbb{L}^2_H\big(SO(3)\big) \cap U_r$ von der speziellen Form $f = d_{S,T}$ mit $S, T \in \mathcal{J}_r(\mathbb{R}^3)$. Die Bedingung $\triangle_R f = f$ für alle $R \in H$ liefert aufgrund von

$$\begin{aligned}(\triangle_R f)(Q) = (\triangle_R d_{S,T})(Q) &= d_{S,D_RT}(Q) \\ &\overset{!}{=} f(Q) = d_{S,T}(Q)\end{aligned}$$

die Bedingung $d_{S,D_RT} = d_{S,T}$ bzw. $d_{S,D_RT-T} = 0$ für alle $R \in H$, denn es ist

$$\begin{aligned}d_{S,D_RT}(Q) - d_{S,T}(Q) &= \langle S, D_Q D_R T\rangle - \langle S, D_Q T\rangle \\ &= \langle S, D_Q(D_R T - T)\rangle = d_{S,D_RT-T}(Q) \ .\end{aligned}$$

Dies liefert bereits den Beweis des folgenden Satzes:

3.15 Satz. *Gibt es einen Tensor $\mathbb{T} \in \mathcal{J}_r(\mathbb{R}^3)$ mit der Eigenschaft $D_R\mathbb{T} = \mathbb{T}$ für alle $R \in H$, so ist der Raum*

$$\big\{d_{S,\mathbb{T}} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3)\big\} \ \subset \ \mathbb{L}^2_H\big(SO(3)\big) \cap U_r$$

ein $(2r+1)$-dimensionaler Teilraum.

Mit dem folgenden Satz können wir für $r \geq 2$ den Unterraum von U_r, der diejenigen Funktionen $f \in U_r$ enthält, welche die Symmetriebedingung $\triangle_R f = f$ für alle $R \in H$ erfüllen, etwas genauer charakterisieren:

3.16 Satz. *Sei H eine Untergruppe von $SO(3)$ und*

$$U_r^{\triangle} := \{f \in U_r \mid \triangle_R f = f \text{ für alle } R \in H\} ,$$

dann gilt

$$U_r^{\triangle} = \operatorname{span}\left\{d_{S,E} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3) \ , \ E \in \mathcal{E}\right\} ,$$

wobei

$$\mathcal{E} = \{E \in \mathcal{J}_r(\mathbb{R}^3) \mid D_R E = E \text{ für alle } R \in H\} .$$

Beweis. Sei $R \in H$ beliebig gewählt und damit $U_R := \{f \in U_r \mid \triangle_R f = f\}$, dann gilt trivialerweise

$$U_r^{\triangle} = \bigcap_{R \in H} U_R .$$

Da die lineare Abbildung D_R unitär ist[12], ist sie normal, d.h. es existiert eine Basis des $\mathcal{J}_r(\mathbb{R}^3)$ aus Eigenvektoren $E_1, \ldots, E_n$ von D_R mit $n := 2r + 1$. Somit lässt sich jedes $f \in U_R$ in der Form $f = \sum_{i,j=1}^{n} a_{ij} d_{E_i,E_j}$ schreiben und aus der Symmetriebedingung folgt

$$\begin{aligned} \triangle_R f &= \sum_{i,j=1}^{n} a_{ij} \triangle_R d_{E_i,E_j} = \sum_{i,j=1}^{n} a_{ij} d_{E_i,D_R E_j} = \sum_{i,j=1}^{n} a_{ij} \lambda_j d_{E_i,E_j} \\ &\stackrel{!}{=} f = \sum_{i,j=1}^{n} a_{ij} d_{E_i,E_j} \ , \end{aligned}$$

wobei $\lambda_j \in \mathbb{R}$ den Eigenwert zum Eigenvektor E_j von D_R bezeichnet. Ein Koeffizientenvergleich liefert, dass $a_{ij} = 0$ sein muss, falls $\lambda_j \neq 1$. Damit folgt

$$f = \sum_{j:\lambda_j=1} \sum_{i=1}^{n} a_{ij} d_{E_i,E_j} = \sum_{j:\lambda_j=1} d_{\sum_{i=1}^{n} a_{ij} E_i, E_j} \ ,$$

und somit $f \in \operatorname{span}\left\{d_{S,E_j} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3) \ , \ \lambda_j = 1\right\}$. Umgekehrt gilt auch $d_{S,E_j} \in U_R$, denn unter der Voraussetzung $S \in \mathcal{J}_r(\mathbb{R}^3)$ und $\lambda_j = 1$ folgt $\triangle_R d_{S,E_j} = d_{S,D_R E_j} = d_{S,E_j}$. Zusammenfassend gilt demnach

$$U_R = \operatorname{span}\left\{d_{S,E_j} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3) \ , \ \lambda_j = 1\right\} .$$

Aufgrund von

$$(\mathcal{D}_R d_{S,E_j})(Q) = d_{S,E_j}(R^T Q) = \langle S, D_{R^T Q} E_j \rangle = \langle D_R S, D_Q E_j \rangle = d_{D_R S, E_j}(Q)$$

für alle $Q \in SO(3)$ ist U_R $\mathcal{D}$-invariant, somit insbesondere auch $U_r^{\triangle}$. Mit Hilfe von Satz 3.14 erhalten wir mit $W_i := \operatorname{span}\left\{d_{S,E_i} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3)\right\}$

für $i = 1, \ldots, n$ eine irreduzible Zerlegung von U_r durch $U_r = W_1 \oplus \ldots \oplus W_n$. Bezeichnen wir mit P_i die Projektion auf den Raum W_i, so können wir ein $f \in U_r$ in der Form $f = \sum_{j=1}^{n} f_j$ schreiben, wobei $f_j := P_j f$. Damit können wir zeigen, dass für $i = 1, \ldots, n$ die Kommutativität $\mathcal{D}_R P_i = P_i \, \mathcal{D}_R$ gilt, denn es ist

$$(\mathcal{D}_R P_i f)(Q) = (\mathcal{D}_R f_i)(Q) = f_i(R^T Q) = P_i f(R^T Q) = (P_i \mathcal{D}_R f)(Q)$$

für alle $Q \in SO(3)$. Trivialerweise gilt $U_r^{\triangle} = P_1 U_r^{\triangle} \oplus \ldots \oplus P_n U_r^{\triangle}$, wobei die Unterräume $P_i U_r^{\triangle} \subseteq W_i$, aufgrund der Kommutativität $\mathcal{D}_R P_i = P_i \, \mathcal{D}_R$ und der $\mathcal{D}$-Invarianz von $U_r^{\triangle}$, ebenso $\mathcal{D}$-invariant sind. Da die Räume W_i irreduzibel sind, gilt entweder $P_i U_r^{\triangle} = \{0\}$ oder $P_i U_r^{\triangle} = W_i$, d.h. es ist $U_r^{\triangle} = W_{i_1} \oplus \ldots \oplus W_{i_p}$ für ein $p \leq n$. Da $W_{i_k} \subset U_r^{\triangle}$ und $d_{S,E_{i_k}} \in W_{i_k}$ für alle $S \in \mathcal{J}_r(\mathbb{R}^3)$ gilt, folgt $\triangle_R d_{S,E_{i_k}} = d_{S,E_{i_k}}$, d.h. $d_{S,D_R E_{i_k} - E_{i_k}} = 0$ für alle $S \in \mathcal{J}_r(\mathbb{R}^3)$ und alle $R \in H$. Demnach folgt $D_R E_{i_k} = E_{i_k}$ für alle $R \in H$ und somit $E_{i_k} \in \mathcal{E}$, d.h.

$$U_r^{\triangle} \subseteq \operatorname{span}\left\{ d_{S,E} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3) \ , \ E \in \mathcal{E} \right\} .$$

Die Inklusion "$\supseteq$" ist trivial, da für ein beliebiges f aus der linearen Hülle

$$\operatorname{span}\left\{ d_{S,E} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{J}_r(\mathbb{R}^3) \ , \ E \in \mathcal{E} \right\} ,$$

aufgrund der Definition von $\mathcal{E}$, direkt $\triangle_R f = f$ für alle $R \in H$ folgt. Dies wiederum ist gleichbedeutend mit $f \in U_r^{\triangle}$. □

Für $r = 0$ bzw. $r = 1$ erhalten wir analog

$$U_0^{\triangle} = \operatorname{span}\{1\} \quad \text{und} \quad U_1^{\triangle} = \operatorname{span}\left\{ d_{S,E} \in \mathbb{L}^2\big(SO(3)\big) \mid S \in \mathcal{T}_1(\mathbb{R}^3) \ , \ E \in \mathcal{E} \right\} ,$$

wobei hier nun $\mathcal{E} = \{E \in \mathcal{T}_1(\mathbb{R}^3) \mid D_R E = E \text{ für alle } R \in H\}$ gilt. Somit erhalten wir für den Raum $\mathbb{L}^2_H\big(SO(3)\big)$, welcher alle $f \in \mathbb{L}^2\big(SO(3)\big)$ beinhaltet, die zusätzlich die Symmetrie $\triangle_R f = f$ für alle $R \in H$ erfüllen, folgende Zerlegung:

$$\mathbb{L}^2_H\big(SO(3)\big) = \bigoplus_{r \in \mathbb{N}_0} U_r^{\triangle} \tag{3-16}$$

4 Die crystalline orientation distribution function (codf)

Mit den vorherigen Kapiteln haben wir nun alle theoretischen Grundlagen zusammen, um die codf wie gewünscht nach tensoriellen Darstellungsfunktionen entwickeln zu können. Der Theorie nach erhalten wir die codf im Allgemeinen zunächst einmal als eine Funktion der Form

$$\begin{aligned} f : SO(3) &\longrightarrow \mathbb{R} \\ Q &\longmapsto 1 + \sum_{r=1}^{\infty} d_{\mathbb{S}^r,\mathbb{T}^r}(Q) \ , \end{aligned} \tag{4-1}$$

mit geeigneten Tensoren $\mathbb{S}^r, \mathbb{T}^r \in \mathcal{J}_r(\mathbb{R}^3)$, sodass die Darstellungsfunktionen $d_{\mathbb{S}^r,\mathbb{T}^r}$ aus den Unterräumen U_r der Zerlegung von $\mathbb{L}^2\big(SO(3)\big)$ sind, und zusätzlich die Positivitätsbedingung sowie die Normierungsbedingung

$$f(Q) \geq 0 \quad \forall\, Q \in SO(3) \quad , \quad \int\limits_{SO(3)} f(Q)\, dQ \ = \ 1 \tag{4-2}$$

erfüllt sind.[1] Dies ist zunächst sehr allgemein formuliert, denn f ist nur dann definiert, wenn alle Tensoren $\mathbb{S}^r, \mathbb{T}^r \in \mathcal{J}_r(\mathbb{R}^3)$ gegeben sind. Um die Frage beantworten zu können, ob dies der Fall ist, betrachten wir zunächst noch einige weitere Aspekte.

Der Wunsch, dass die vorliegende Kristallsymmetrie des zu untersuchenden Metalls durch die codf wiedergespiegelt werden soll, wurde bereits geäußert. Ein Grund für diesen Wunsch liegt in der Tatsache, dass wir bei der späteren Integration der codf über $SO(3)$ damit einen großen Vorteil haben, wie wir in Kapitel 5.3 sehen werden. Doch wie lässt sich diese Symmetrie auf die codf übertragen? Um eine bestimmte Kristallsymmetrie (d.h. eine bestimmte Grundform einer Elementarzelle eines Einkristalls des Metalls) mathematisch zu erfassen, fassen wir alle Rotationen aus $SO(3)$, welche eine Elementarzelle dieser Symmetrie aus einer Ausgangsorientierung wieder ununterscheidbar in sich selbst überführen, zu einer Menge zusammen. Diese Menge besitzt eine Gruppenstruktur und bildet demnach eine Untergruppe von

[1] $\mathbb{S}^r, \mathbb{T}^r \in \mathcal{J}_r(\mathbb{R}^3)$ gilt nach Definition 3.4 der irreduziblen Tensoren nur für $r \geq 2$. Im Fall $r = 1$ ist dies durch die Bedingung $\mathbb{S}^1, \mathbb{T}^1 \in \mathcal{T}_1(\mathbb{R}^3)$ zu ersetzen. Aus diesem Grund definieren wir $\mathcal{J}_1(\mathbb{R}^3) := \mathcal{T}_1(\mathbb{R}^3)$, um somit obige Schreibweise beibehalten zu können. Wir werden jedoch sehen, dass der Summand im Fall $r = 1$ bei der später betrachteten kubischen Kristallsymmetrie ohnehin nicht in der Reihe vorkommen wird.

$SO(3)$, die sogenannte *Rotations-Symmetriegruppe* H des Kristalls. Darauf werden wir jedoch noch genauer im folgenden Kapitel 4.1 eingehen. Da die codf Auskunft darüber gibt, wieviele Kristalle des zu untersuchenden Metalls bezüglich einer Referenzelementarzelle dieselbe Orientierung $Q \in SO(3)$ haben, muss diese Verteilung unverändert bleiben, wenn man die Referenzelementarzelle mit einer beliebigen Rotation $R \in H$ dieser Rotations-Symmetriegruppe dreht. Demnach ist nun auch klar, dass sich die Kristallsymmetrie mit Hilfe der Forderung $f(QR) = f(Q)$ für alle $Q \in SO(3)$ und alle $R \in H$ auf die codf, d.h. auf die Darstellungsfunktionen $d_{\mathbb{S}^r,\mathbb{T}^r}$, übertragen lässt. In Kapitel 4.2 werden wir sehen, dass diese Forderung gleichbedeutend mit der Bedingung $D_R\mathbb{T}^r = \mathbb{T}^r$ für alle $R \in H$ an die Tensoren $\mathbb{T}^r$ ist. Dieses Resultat liefert somit einen Bezug zu Satz 3.16, sodass die codf wie oben angegeben sogar als eine Funktion aus $\mathbb{L}^2_H\big(SO(3)\big)$ aufzufassen ist. Dies bedeutet wiederum, dass die Tensoren $\mathbb{S}^r, \mathbb{T}^r \in \mathcal{J}_r(\mathbb{R}^3)$ sogar so gewählt werden müssen, dass die Darstellungsfunktionen $d_{\mathbb{S}^r,\mathbb{T}^r}$ in den Unterräumen $U_r^{\triangle}$ für $r \geq 1$ liegen.

Im Folgenden machen wir einen kurzen Ausflug in die Kristallographie, um eine Vorstellung davon zu bekommen, was für verschiedene Kristallsymmetrien eines Metalls überhaupt vorliegen können. Danach werden wir uns dann mit der genaueren Wahl der Tensoren $\mathbb{S}^r, \mathbb{T}^r \in \mathcal{J}_r(\mathbb{R}^3)$ beschäftigen.

4.1 Kristallsysteme

In der Kristallographie gibt es sieben verschiedene Kristallsysteme, d.h. sieben verschiedene Grundformen, in denen Elementarzellen verschiedener Kristalle vorkommen können. In der folgenden Grafik[27] werden diese Grundformen veranschaulicht:

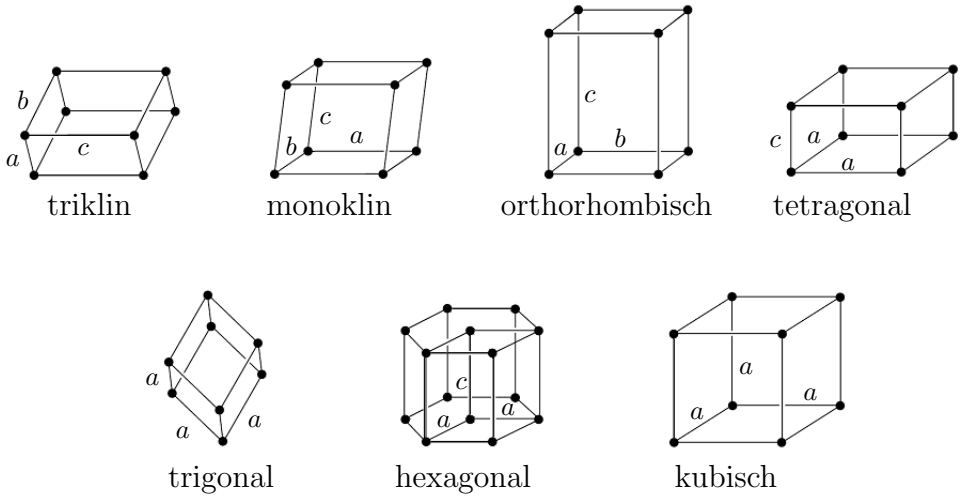


Abbildung 4.1: Übersicht der 7 Kristallsysteme

So unterscheiden sich die verschiedenen Kristallsysteme nicht nur durch die geometrische Struktur der jeweiligen Elementarzellen (Details siehe Tabelle 4.1), sondern auch durch die sich daraus ergebenden unterschiedlichen Rotationssymmetrien. Suchen wir alle Rotationen aus $SO(3)$, die eine Elementarzelle einer bestimmten Symmetrie aus einer Ausgangsorientierung wieder ununterscheidbar in sich selbst überführen, so ist diese Menge, wie bereits erwähnt wurde, eine Untergruppe von $SO(3)$ und wird als Rotations-Symmetriegruppe H des Kristalls bezeichnet. Im Falle von kubischer Symmetrie besteht die Rotations-Symmetriegruppe aus folgenden 24 Rotationen:

- 90°, 180°, 270°-Drehung um die 3 vierzähligen Drehachsen (durch gegenüber liegende Flächenmittelpunkte)
- 120°, 240°-Drehung um die 4 dreizähligen Drehachsen (durch gegenüber liegende Ecken)
- 180°-Drehung um die 6 zweizähligen Drehachsen (durch gegenüber liegende Kantenmittelpunkte)
- die Identität

Die konkreten Rotationen der Rotations-Symmetriegruppen der anderen Kristallsysteme sind der Literatur zu entnehmen.[9] Die folgende Tabelle gibt einen Überblick über die geometrische Beschreibung der einzelnen Kristallsysteme, und wieviele Rotationen in der jeweiligen Rotations-Symmetriegruppe des Kristalls enthalten sind. Der Zusammenhang zwischen den Winkeln und Kanten ist dabei durch $\alpha = \sphericalangle(b, c)$, $\beta = \sphericalangle(a, c)$ und $\gamma = \sphericalangle(a, b)$ gegeben:

Kristallsystem	Geometrie	$\lvert H \rvert$
triklin	allgemeiner Spat $a \neq b \neq c\ ,\ \alpha \neq \beta \neq \gamma$	1
monoklin	Parallelogrammzylinder $a \neq b \neq c\ ,\ \alpha = \gamma = 90° \neq \beta$	2
orthorhombisch	Quader $a \neq b \neq c\ ,\ \alpha = \beta = \gamma = 90°$	4
tetragonal	Quadratzylinder $a = b \neq c\ ,\ \alpha = \beta = \gamma = 90°$	8
trigonal	Rhomboeder $a = b = c\ ,\ \alpha = \beta = \gamma \neq 90°$	6
hexagonal	Sechseckzylinder $a = b \neq c\ ,\ \alpha = \beta = 90°\ ,\ \gamma = 120°$	12
kubisch	Würfel $a = b = c\ ,\ \alpha = \beta = \gamma = 90°$	24

Tabelle 4.1: Mächtigkeit der Rotations-Symmetriegruppen

Da die wichtigsten Metalle, wie z.B. Aluminium, Blei, Eisen, Gold, Calcium, Strontium, Kupfer, Nickel, Palladium, Platin, Rhodium oder Silber, eine kubische Kristallstruktur aufweisen, werden wir uns für den Rest dieser Arbeit nur auf kubische Symmetrien beziehen. Alle weiteren Folgerungen, die lediglich aus der Existenz einer Kristallsymmetrie gezogen werden und nicht aus der konkreten Symmetrie selbst, lassen sich demnach analog in jeder der sechs anderen Kristallsysteme problemlos formulieren.

4.2 Bestimmung der Tensoren $\mathbb{T}^r$ und $\mathbb{S}^r$

Wie in der Einführung dieses Kapitels bereits erwähnt wurde, soll die codf die Kristallsymmetrie des zu untersuchenden Metalls wiederspiegeln, d.h. für alle Rotationen $R \in H$ der entsprechenden Rotations-Symmetriegruppe H muss gelten:

$$f(QR) = f(Q) \quad \forall\, Q \in SO(3)$$

Dies ist jedoch gleichbedeutend mit $d_{\mathbb{S}^r,\mathbb{T}^r}(QR) = d_{\mathbb{S}^r,\mathbb{T}^r}(Q)$, was $(QR) * \mathbb{T}^r \stackrel{!}{=} Q * \mathbb{T}^r$ zur Folge hat. Für die Tensoren $\mathbb{T}^r \in \mathcal{J}_r(\mathbb{R}^3)$ erhalten wir aufgrund von

$$D_Q D_R \mathbb{T}^r = D_{QR}\mathbb{T}^r = (QR) * \mathbb{T}^r = Q * \mathbb{T}^r = D_Q \mathbb{T}^r$$

für alle $Q \in SO(3)$ und alle $R \in H$ demnach die Bedingung $D_R\mathbb{T}^r = R * \mathbb{T}^r = \mathbb{T}^r$ für alle $R \in H$. Mit Hilfe der jeweils entsprechenden Einheitsmatrix erhalten wir somit die Kristallsymmetrie-Bedingungen in der Form

$$(D_R - \mathbb{1})\mathbb{T}^r = 0 \quad \forall\, R \in H\ ,$$

wobei mit 0 hierbei der entsprechende Nulltensor aus $\mathcal{J}_r(\mathbb{R}^3)$ bezeichnet wird. Für jedes $R \in H$ liefert demnach jede Zeile der entsprechenden Matrix $D_R - \mathbb{1}$ eine Kristallsymmetrie-Bedingung, sprich ein lineares Funktional. Diese Funktionale nutzen wir nun, um damit die Dualbasis der symmetrischen Tensoren aus Kapitel 3.4 sukzessive zu aktualisieren. Dies gelingt uns durch einen Basisaustausch, bei dem wir ausgehend von den in der Matrix C enthaltenen Spurbedingungen, die restlichen Zeilen der Matrix C durch eben diese Kristallsymmetrie-Bedingungen ersetzen, ohne dabei die Invertierbarkeit der Matrix zu verlieren. Auf diese Weise erhalten wir eine neue Dualbasis der symmetrischen Tensoren, welche in den Zeilen der folgenden Matrix B steht und zusätzlich zu den Spurbedingungen auch die Kristallsymmetrie-Bedingungen enthält. Im Allgemeinen Fall ist diese Matrix B also von folgendem Typ:

$$B \;=\; \left(\begin{array}{|c|} \hline \text{freie Bedingungen} \\ \hline \text{Kristallsymmetrie-Bedingungen} \\ \hline \text{Spurbedingungen} \\ \hline \end{array} \right)$$

Schränkt man die Matrix B auf die Zeilen der Kristallsymmetrie-Bedingungen und der Spurbedingungen ein, und nennt diese Teilmatrix $\tilde{B}$, so erfüllt jeder irreduzible Tensor $T \in \mathcal{J}_r(\mathbb{R}^3)$, der zusätzlich die Bedingung $D_R T = T$ für alle $R \in H$ erfüllt, die Bedingung

$$\tilde{B}T = 0 \ ,$$

sofern ein solcher irreduzibler Tensor für den entsprechenden Rang r existiert. Die Frage nach der Existenz eines solchen Tensors wird durch die Anzahl der Zeilen mit den sogenannten freien Bedingungen beantwortet. Gibt es keine freien Bedingungen, so besteht die Dualbasis der symmetrischen Tensoren lediglich aus Spurbedingungen und Kristallsymmetrie-Bedingungen, d.h. aus

$$BT = \tilde{B}T = 0$$

folgt aufgrund der Invertierbarkeit von B direkt $T = 0$. In diesem Fall gibt es also keinen irreduziblen Tensor $T \neq 0$, der zusätzlich die Kristallsymmetrie-Bedingungen erfüllt. Demnach verschwindet der entsprechende Summand in der Reihendarstellung der codf. Gibt es in B hingegen $k \in \mathbb{N}$ viele Zeilen mit freien Bedingungen, so liefern die ersten k Spalten von B^{-1} aufgrund von $BB^{-1} = \mathbb{1}$ genau k linear unabhängige Koordinatenvektoren irreduzibler Tensoren in symmetrischer Dimension, die zusätzlich die Kristallsymmetrie-Bedingungen erfüllen. Diese zu den k Koordinatenvektoren gehörende Tensoren bilden also eine Basis des Raumes $\mathcal{E}$ aus Satz 3.16. Durch diese Vorgehensweise können wir also die Tensoren $\mathbb{T}^r$ aus der Reihendarstellung der codf bestimmen. Die Tensoren $\mathbb{T}^r$ können noch entsprechend normiert werden, ohne dabei die Positivität und die Normierungseigenschaft der codf zu verletzen. Denn diese Eigenschaften werden erst durch die entsprechende Wahl der Tensoren $\mathbb{S}^r$ realisiert. Greifen wir aus den normierten Tensoren $\mathbb{T}^r$, wie in Kapitel 3.4 beschrieben, die entsprechenden unabhängigen Komponenten heraus, so können wir diese Tensoren schließlich auch in irreduzibler Dimension darstellen.

Im Falle von kubischer Kristallsymmetrie zeigt die folgende Auflistung bis zum Rang 20, zu welchen Rängen überhaupt bzw. wieviele irreduzible Tensoren $\mathbb{T}^{r_i} \in \mathcal{J}_{r_i}(\mathbb{R}^3)$

existieren, die zusätzlich die Kristallsymmetrie-Bedingungen erfüllen:

$$r_i \in \{4, 6, 8, 9, 10, 12_1, 12_2, 13, 14, 15, 16_1, 16_2, 17, 18_1, 18_2, 19, 20_1, 20_2, \ldots\}$$

Der erste Tensor existiert demnach für $r = 4$ und die Fälle $r = 12, 16, 18, 20$ sind somit die ersten, in welchen sogar zwei linear unabhängige Tensoren die gewünschte Symmetrie aufweisen. Dadurch reduziert sich die Reihe der codf entsprechend auf die zugehörigen Summanden:

$$f(Q) \;=\; 1 + \sum_{i=1}^{\infty} d_{\mathbb{S}^{r_i},\mathbb{T}^{r_i}}(Q) \;=\; 1 + \sum_{i=1}^{\infty} \langle \mathbb{S}^{r_i}, Q * \mathbb{T}^{r_i} \rangle$$

Mit Hilfe der Orthogonalitätsrelation

$$\int_{SO(3)} (Q * \mathbb{T}^{r_i}) \otimes (Q * \mathbb{T}^{r_j})\; dQ \;=\; 0 \quad \forall\, i \neq j \quad \text{und}$$
$$\int_{SO(3)} (Q * \mathbb{T}^{r_i}) \otimes (Q * \mathbb{T}^{r_i})\; dQ \;=\; \frac{1}{2r_i + 1}\; \mathbb{1}^{2r_i}\;,$$

wobei 0 erneut den entsprechenden Nulltensor und $\mathbb{1}^{2r_i}$ die Identität auf $\mathcal{J}_{2r_i}(\mathbb{R}^3)$ beschreibt, erhalten wir unter Berücksichtigung von (4-2) mit

$$\mathbb{S}^{r_i} \;=\; (2r_i + 1) \int_{SO(3)} f(Q)(Q * \mathbb{T}^{r_i})\; dQ \tag{4-3}$$

eine Bestimmungsgleichung der sogenannten *tensoriellen Texturkoeffizienten* $\mathbb{S}^{r_i}$.[6] Da die codf die Tensoren $\mathbb{S}^{r_i}$ selbst auch enthält, ist diese Bestimmungsgleichung implizit. Um nun an die Tensoren $\mathbb{S}^{r_i}$ zu kommen, approximieren wir das Maß $f\; dQ$ durch eine endliche Summe gleichgewichteter Punktmaße, welche zu diskreten Kristallorientierungen $Q_j \in SO(3)$ gehören, die z.B. als Messwerte des zu untersuchenden Metalls detektiert wurden (siehe Einleitung, Seite 2). Somit erhalten wir

$$f\; dQ \;\approx\; \frac{1}{N} \sum_{j=1}^{N} \delta_{Q_j}\;. \tag{4-4}$$

Da wir jedoch nicht auf der Suche nach einer Approximation eines Maßes sind, bezüglich welchem (4-3) erfüllt ist, sondern die Motivation verfolgen, eine Dichte bezüglich dem gegebenen Haar-Maß zu finden, sodass (4-3) gilt, sind wir mit (4-4) noch nicht am Ziel. Mit Hilfe von (4-3) erhalten wir damit jedoch eine nun explizite Bestimmungsgleichung der tensoriellen Texturkoeffizienten $\mathbb{S}^{r_i}$:

$$\mathbb{S}^{r_i} \;=\; \frac{2r_i + 1}{N} \sum_{j=1}^{N} Q_j * \mathbb{T}^{r_i} \tag{4-5}$$

Man beachte, dass die Texturkoeffizienten somit durch die Daten Q_j bestimmt werden, und dass $\mathbb{S}^{r_i} \in \mathcal{J}_{r_i}(\mathbb{R}^3)$ sichergestellt ist. Kommen wir nun zurück zur Frage, ob somit alle Tensoren $\mathbb{S}^{r_i}$ und $\mathbb{T}^{r_i}$ gegeben sind, um eine sinnvolle Approximation von (4-3) zu erzielen, so ist die Antwort darauf leider negativ. Denn aufgrund der Tatsache, dass die $\mathbb{S}^{r_i}$ lediglich mit Hilfe endlich vieler Kristallorientierungen bestimmt wurden, können in Abhängigkeit von N auch nur endlich viele sinnvolle Approximationen der $\mathbb{S}^{r_i}$ im Sinne von (4-3) experimentell ermittelt werden. Dies bedeutet, dass wir in Abhängigkeit von N nur für endlich viele $\mathbb{S}^{r_i}$ in der Lage sind, eine Dichte zu finden, für welche für eben diese $\mathbb{S}^{r_i}$ (4-3) jeweils sehr gut approximiert wird. Ab einem bestimmten Tensorrang wird (4-3) bei dieser Vorgehensweise also schlecht approximiert werden. Dies entspricht faktisch jedoch einer Approximation $\tilde{f}$ der codf (4-1) durch Abbruch der Reihe ab diesem Tensorrang, d.h. es gilt

$$\tilde{f}(Q) \;=\; 1 + \sum_{i=1}^{L} \langle \mathbb{S}^{r_i}, Q * \mathbb{T}^{r_i} \rangle \;.$$

Mit Hilfe der endlich vielen Messdaten ist die Approximation $\tilde{f}$ der codf also bereits durch Angabe der Tensoren $\mathbb{S}^{r_i}, \mathbb{T}^{r_i} \in \mathcal{J}_{r_i}(\mathbb{R}^3)$ ohne großen Rechenaufwand bestimmt, da die Texturkoeffizienten ja bereits durch die Messdaten gegeben sind. In Zeiten geringer Rechnerleistung war das mit Sicherheit sehr sinnvoll. Diese Vorgehensweise hat jedoch einen großen Nachteil, auf den wir in Zeiten von großen Rechnerleistungen heute nicht mehr eingehen müssen. Denn im Allgemeinen kann man bei der abgebrochenen codf nicht mehr von der Gültigkeit der Bedingungen (4-2) ausgehen, d.h. für ein endliches $L \in \mathbb{N}$ und ein $Q \in SO(3)$ kann

$$\tilde{f}(Q) \;=\; 1 + \sum_{i=1}^{L} \langle \mathbb{S}^{r_i}, Q * \mathbb{T}^{r_i} \rangle \;\not\geq\; 0$$

gelten. Da wir aber generell daran interessiert sind, eine Verteilungsfunktion zu bestimmen, ist die Positivität eine nicht zu vernachlässigende Eigenschaft. Das Problem, eine Verteilungsfunktion basierend auf unvollständigen Daten zu approximieren, hat keine eindeutige Lösung und ist daher schlecht gestellt.[6] Daher wählen wir für die Approximation $\tilde{f}$ der codf einen anderen Ansatz[24], die sogenannte **Maximum Entropie Methode**, auf welche wir im folgenden Kapitel genauer eingehen werden.

4.3 Die Maximum Entropie Methode

Wie der Name bereits beinhaltet, gehen wir bei dieser Methode von der Entropie einer Funktion $f \in \mathbb{L}^2\big(SO(3)\big)$ aus, welche durch

$$E(f) := -\int_{SO(3)} f(Q) \ln\big(f(Q)\big)\, dQ$$

gegeben ist und somit die Positivität von f bereits fordert. Diese wollen wir nun maximieren unter den zunächst ganz beliebig formulierten Nebenbedingungen

$$\int\limits_{SO(3)} f(Q)m_k(Q)\ dQ \ = \ \beta_k \quad \text{für } k = 1, \ldots, n \ . \tag{4-6}$$

Wir haben es also auch bei dieser Problemstellung mit einem Momentenproblem für eine positive Dichte f zu tun. Man beachte jedoch, dass in dieser allgemeineren Formulierung die Momentenfunktionen m_k und die Momente β_k auch mehrdimensional bzw. ein von einer reellen Zahl unterschiedliches Objekt, wie zum Beispiel ein Tensor, sein können. Dies macht hinsichtlich der codf Sinn, wie wir direkt an (4-2) und (4-3) ablesen können. Auch hier werden wir im Allgemeinen immer davon ausgehen, dass die Momente mit den Momentenfunktionen kompatibel sind, d.h. dass die Momente für mindestens ein $f \in \mathbb{L}^2\big(SO(3)\big)$ mit $f \geq 0$ gemäß (4-6) darstellbar sind.

Um nun die Lösung dieses allgemeinen Problems zu berechnen, bedienen wir uns erneut des Lagrange-Formalismus, bei welchem wir das Extremum der folgenden Lagrange-Funktion bestimmen:

$$L(f, \mu) \ := \ - \int\limits_{SO(3)} f \ln(f)\ dQ \ + \ \sum_{k=1}^{n} \mu_k \cdot \Big[\int\limits_{SO(3)} f m_k\ dQ \ - \ \beta_k \Big]$$

Mit μ bezeichnen wir die vektorielle Gesamtheit aller Lagrange-Multiplikatoren μ_k, wobei ein einzelnes μ_k vom Objekttyp her identisch mit dem von m_k bzw. β_k ist. Der notierte Malpunkt steht für das den Objekten entsprechende Skalarprodukt.

Für das Optimum $\tilde{f}$ dieses Problems muss demnach für alle Testfunktionen gelten:

$$\frac{d}{d\varepsilon}\ L(\tilde{f} + \varepsilon h, \mu)\ \Big|_{\varepsilon=0} \ \stackrel{!}{=} \ 0$$

Im Detail bedeutet dies:

$$\begin{aligned}
\frac{d}{d\varepsilon}\ L(\tilde{f} + \varepsilon h, \mu)\ \Big|_{\varepsilon=0} \ &= \ - \ \frac{d}{d\varepsilon} \left(\int\limits_{SO(3)} (\tilde{f} + \varepsilon h) \ln(\tilde{f} + \varepsilon h)\ dQ \right) \Bigg|_{\varepsilon=0} \\
&\quad + \ \sum_{k=1}^{n} \left(\mu_k \cdot \frac{d}{d\varepsilon} \ \Big[\int\limits_{SO(3)} (\tilde{f} + \varepsilon h) m_k\ dQ \ - \ \beta_k \Big] \right) \Bigg|_{\varepsilon=0} \\
&= \ - \left(\int\limits_{SO(3)} h \ \Big(\ln(\tilde{f} + \varepsilon h) + 1 \Big)\ dQ \right) \Bigg|_{\varepsilon=0} \\
&\quad + \ \sum_{k=1}^{n} \left(\mu_k \ \cdot \int\limits_{SO(3)} h m_k\ dQ \right)
\end{aligned}$$

$$= \; - \left(\int\limits_{SO(3)} h \, \Big(\ln(\tilde{f}) + 1 - \sum_{k=1}^{n} \mu_k \cdot m_k \Big) \, dQ \right) \; \stackrel{!}{=} \; 0$$

Da dies für alle Testfunktionen im Sinne des Fundamentallemmas der Variationsrechnung[19] gelten soll, folgt unmittelbar

$$\ln(\tilde{f}) + 1 - \sum_{k=1}^{n} \mu_k \cdot m_k \; = \; 0 \; .$$

Somit erhalten wir für das Optimum $\tilde{f}$ dieses Problems, analog zu Seite 9 der Einleitung, folgendes Ergebnis:

$$\tilde{f} \; = \; \exp\Big(-1 + \sum_{k=1}^{n} \mu_k \cdot m_k \Big)$$

Die LAGRANGE-Multiplikatoren μ_k sind analog zur Einleitung die Lösung des zugehörigen dualen Optimierungsproblems, welches wie folgt lautet:

$$\text{minimiere} \quad \Phi(\omega) := \int\limits_{SO(3)} \exp\Big(-1 + \sum_{k=1}^{n} \omega_k \cdot m_k(Q) \Big) \, dQ \; - \; \sum_{k=1}^{n} \omega_k \cdot \beta_k$$

Auch hier sieht man sofort, dass die Forderung $\nabla\Phi(\omega) = 0$ direkt die Nebenbedingungen (4-6) liefert.

Übertragen wir dies nun auf unseren Fall, so erhalten wir mit Hilfe des beschriebenen LAGRANGE-Formalismus eine **positive Approximation $\tilde{f} \in \mathbb{L}^2_H\big(SO(3)\big)$ der codf** in der Form[5]

$$\begin{aligned} \tilde{f}(Q) \; &= \; \exp\Big(-1 + \mu_0 + \sum_{k=1}^{n} d_{\mu_k, \mathbb{T}^{r_k}}(Q) \Big) \\ &= \; \exp\Big(-1 + \mu_0 + \sum_{k=1}^{n} \langle \mu_k, Q * \mathbb{T}^{r_k} \rangle \Big) \end{aligned} \tag{4-7}$$

als Lösung eines tensoriellen Maximum Entropie Momentenproblems mit der zusätzlichen Normierungsbedingung

$$\int\limits_{SO(3)} \tilde{f}(Q) \, dQ \; = \; 1 \tag{4-8}$$

und den tensoriellen Nebenbedingungen

$$\int\limits_{SO(3)} \tilde{f}(Q)(Q * \mathbb{T}^{r_k})\, dQ \;=\; \frac{1}{2r_k+1}\, \mathbb{S}^{r_k} \quad \text{für } k = 1, \ldots, n\ , \tag{4-9}$$

bei denen die tensoriellen Momente, bis auf den Dimensionsvorfaktor, gerade den Texturkoeffizienten $\mathbb{S}^{r_k}$ entsprechen. Die in $\tilde{f}$ auftauchenden und zu bestimmenden LAGRANGE-Multiplikatoren μ_k sind für $k = 1, \ldots, n$ in diesem Fall irreduzible Tensoren vom Rang r_k über $\mathbb{R}^3$, d.h. es gilt $\mu_k \in \mathcal{J}_{r_k}(\mathbb{R}^3)$. Für den LAGRANGE-Multiplikator μ_0 gilt $\mu_0 \in \mathbb{R}$, da er aus der skalaren Normierungsbedingung resultiert. Die Gesamtheit $\mu = [\mu_0; \mu_1; \ldots; \mu_n]$ aller LAGRANGE-Multiplikatoren ist demnach die Minimalstelle der folgenden Funktion:

$$\Phi(\omega) = \int\limits_{SO(3)} \exp\Big(-1 + \omega_0 + \sum_{k=1}^{n} \langle \omega_k, Q * \mathbb{T}^{r_k} \rangle \Big)\, dQ \;-\; \omega_0 \;-\; \sum_{k=1}^{n} \frac{\langle \omega_k, \mathbb{S}^{r_k} \rangle}{2r_k+1} \tag{4-10}$$

Der Zusammenhang dieses tensoriellen Momentenproblems mit dem in der Einleitung vorgestellten skalaren Momentenproblem ist nun offensichtlich, denn für unseren Fall ist das skalar formulierte Momentenproblem von folgender Form:

Finde eine positive Dichte $f \in \mathbb{L}^2_H\big(SO(3)\big)$, die die Normierungsbedingung und für ein gegebenes $n \in \mathbb{N}$ die skalaren Momentenbedingungen

$$\int\limits_{SO(3)} a_k(Q) f(Q)\, dQ \;=\; b_k \quad \textit{für} \quad k = 1, \ldots, n$$

erfüllt, für gegebene Momentenfunktionen $a_k \in \mathbb{L}^2_H\big(SO(3)\big)$ und gegebene Momente $b_k \in \mathbb{R}$. Dabei sind die Momentenfunktionen von der Form $a_k = d_{\mathbb{V}^{r_k}, \mathbb{T}^{r_k}}$ mit $\mathbb{V}^{r_k}, \mathbb{T}^{r_k} \in \mathcal{T}_{r_k}(\mathbb{R}^3)$, sodass $d_{\mathbb{V}^{r_k}, \mathbb{T}^{r_k}} \in U^{\triangle}_{r_k}$ für $r_k \in \mathbb{N}$ gilt.

Bestimmt man in der tensoriellen Formulierung komplette Tensoren als LAGRANGE-Multiplikatoren, so muss in der skalaren Formulierung geklärt werden, wie die Tensoren $\mathbb{V}^{r_k}$ zu wählen sind. Wählen wir in $\mathcal{J}_{r_k}(\mathbb{R}^3)$ jeweils die Standard-ONB $E^{r_k}_1, \ldots, E^{r_k}_{2r_k+1}$, so gilt

$$d_{\mathbb{V}^{r_k}, \mathbb{T}^{r_k}}(Q) = \langle \mathbb{V}^{r_k}, Q * \mathbb{T}^{r_k} \rangle = \sum_{j=1}^{2r_k+1} \alpha_j \langle E^{r_k}_j, Q * \mathbb{T}^{r_k} \rangle = \sum_{j=1}^{2r_k+1} \alpha_j\, d_{E^{r_k}_j, \mathbb{T}^{r_k}}(Q)\ .$$

Da die Tensoren $\mathbb{V}^{r_k}$, und somit die Koeffizienten $\alpha_j \in \mathbb{R}$, jedoch unbekannt sind, formulieren wir das skalare Momentenproblem wie folgt um:

Finde eine positive Dichte $f \in \mathbb{L}^2_H\big(SO(3)\big)$*, die die Normierungsbedingung und für ein gegebenes* $n \in \mathbb{N}$ *die skalaren Momentenbedingungen*

$$\int_{SO(3)} a_{jk}(Q) f(Q)\, dQ \;=\; b_{jk} \quad \textit{für} \quad k = 1, \dots, n \quad \textit{und} \quad j = 1, \dots, 2r_k + 1$$

erfüllt, für gegebene Momentenfunktionen $a_{jk} \in \mathbb{L}^2_H\big(SO(3)\big)$ *und gegebene Momente* $b_{jk} \in \mathbb{R}$*. Dabei sind die Momentenfunktionen von der Form* $a_{jk} = d_{E_j^{r_k}, \mathbb{T}^{r_k}}$ *mit* $\mathbb{T}^{r_k} \in \mathcal{T}_{r_k}(\mathbb{R}^3)$*, sodass* $d_{E_j^{r_k}, \mathbb{T}^{r_k}} \in U_{r_k}^{\triangle}$ *für* $r_k \in \mathbb{N}$ *gilt.*

Mit Hilfe der Maximum Entropie Methode erhalten wir für dieses skalare Momentenproblem demnach für $Q \in SO(3)$ die Lösung

$$\begin{aligned} f_{\mathrm{sk}}(Q) \;&=\; \exp\Big(-1 + \lambda_0 + \sum_{k=1}^{n} \sum_{j=1}^{2r_k+1} \lambda_{jk} a_{jk}(Q) \Big) \\ &=\; \exp\Big(-1 + \lambda_0 + \sum_{k=1}^{n} \sum_{j=1}^{2r_k+1} \lambda_{jk} \langle E_j^{r_k}, Q * \mathbb{T}^{r_k} \rangle \Big) \end{aligned}$$

mit den Lagrange-Multiplikatoren $\lambda_0, \lambda_{jk} \in \mathbb{R}$. Wählen wir nun beim skalaren Momentenproblem für $k = 1, \dots, n$ und $j = 1, \dots, 2r_k + 1$ die speziellen Momente

$$b_{jk} \;:=\; \frac{1}{2r_k + 1} \, \langle E_j^{r_k}, \mathbb{S}^{r_k} \rangle \;,$$

so sieht man direkt, dass das zugehörige skalare Momentenproblem und das tensorielle Momentenproblem (4-8),(4-9) zueinander äquivalent sind, da die einzelnen Nebenbedingungen im skalaren Fall gerade den einzelnen Komponenten der tensoriellen Nebenbedingungen im tensoriellen Fall entsprechen. Demnach gilt $f_{\mathrm{sk}} = \tilde{f}$, und desweiteren gelten folgende Beziehungen zwischen den jeweiligen Lagrange-Multiplikatoren:

$$\mu_0 = \lambda_0 \quad \text{und} \quad \mu_k = \sum_{j=1}^{2r_k+1} \lambda_{jk} E_j^{r_k}$$

Das Existenzresultat einer Lösung des Maximum Entropie Momentenproblems[24] von Seite 9 der Einleitung (Satz 1.10) lässt sich demnach direkt vom skalaren auf den tensoriellen Fall übertragen. Deshalb werden wir das Maximum Entropie Momentenproblem im Folgenden nur noch direkt in der tensoriellen Form (4-8),(4-9) mit der Lösung (4-7) betrachten. Um diese Lösung jedoch zu berechnen, bedarf es einiges an Rechenaufwand. In den folgenden Kapiteln widmen wir uns nun den einzelnen Bausteinen, die zur Berechnung der codf in der Form (4-7) notwendig sind.

4.4 Auswertung des Rayleigh-Produktes

In Kapitel 3.5.1 haben wir die Darstellung D von $SO(3)$ auf dem Vektorraum $\mathcal{J}_r(\mathbb{R}^3)$ der irreduziblen Tensoren vom Rang r über $\mathbb{R}^3$ eingeführt, welche durch

$$\begin{aligned} D : SO(3) \times \mathcal{J}_r(\mathbb{R}^3) &\longrightarrow \mathcal{J}_r(\mathbb{R}^3) \\ (Q, T) &\longmapsto D_Q T := Q * T \end{aligned}$$

gegeben ist. Die Aufgabe, das dabei auftretende RAYLEIGH-Produkt $Q*T$ möglichst effizient berechnen zu können, ist demnach für eine effiziente Auswertung dieser Darstellung, und somit für eine effiziente Auswertung der codf von größter Bedeutung, wie man an (4-7) sofort erkennen kann. Gemäß (3-13) ist eine einzelne Komponente $(Q * T)_i$ des RAYLEIGH-Produkts für ein $i \in I = \{1, 2, 3\}^r$ durch

$$(Q * T)_i = \sum_{j \in I} Q_{i_1 j_1} \cdot \ldots \cdot Q_{i_r j_r} T_j \tag{4-11}$$

gegeben. Somit ist das RAYLEIGH-Produkt $Q * T$ für ein festes $Q \in SO(3)$ jeweils eine lineare Abbildung auf $\mathcal{J}_r(\mathbb{R}^3)$, und kann somit, durch Anwendung der zu dieser linearen Abbildung gehörenden darstellenden Matrix D_Q auf den als Vektor aufgefassten Tensor T, berechnet werden. Das RAYLEIGH-Produkt kann also als schlichte Matrixmultiplikation $Q * T = D_Q T$ aufgefasst werden. Die Matrix D_Q ist im allgemeinen Fall, in welchem der Tensor $T \in \mathcal{J}_r(\mathbb{R}^3)$ zunächst in voller Dimension dargestellt wird, von der Dimension $3^r \times 3^r$. Ausgehend von einer beliebigen Matrix $Q \in SO(3)$, lässt sich die Matrix D_Q durch eine sehr einfache Vorgehensweise generieren. Da es sich um einen rekursiven Aufbau der Matrix handelt, bei dem man zur Erstellung der Matrix die entsprechende Darstellungsmatrix aus dem Fall des um eins niederen Ranges benötigt, sei hier zunächst beschrieben, wie man D_Q zum Rang r konkret aus jener Darstellungsmatrix zum Rang $r-1$ konstruiert. Bezeichnen wir der besseren Übersicht wegen die Darstellungsmatrix zum Rang $r-1$ mit $D_Q^{(r-1)}$ für $r \geq 2$, so gilt $D_Q^{(r-1)} \in \mathbb{R}^{3^{r-1} \times 3^{r-1}}$. Für die Rekursion setzen wir $D_Q^{(1)} := Q$. Nun generieren wir eine $3^{r-1} \times 3^{r-1}$ - Blockmatrix, die in jedem Blockeintrag die Matrix Q enthält, und somit insgesamt von der Dimension $3^r \times 3^r$ ist. Die gewünschte Matrix D_Q erhalten wir nun, indem wir jeden Matrixeintrag der Matrix des Blockeintrages (kl) mit $k, l \in \{1, \ldots, 3^{r-1}\}$ mit dem entsprechenden Matrixeintrag $(D_Q^{(r-1)})_{kl}$ multiplizieren. Diese Vorgehensweise wird im Folgenden zum besseren Verständnis für die Fälle $r = 2$ und $r = 3$ noch etwas anschaulicher dargestellt.

Struktur von D_Q in Abhängigkeit des Ranges r:

$$Q = \begin{pmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{21} & Q_{22} & Q_{23} \\ Q_{31} & Q_{32} & Q_{33} \end{pmatrix} \overset{r=2}{\curvearrowright} \quad D_Q = D_Q^{(2)} = \begin{pmatrix} Q_{11}Q & Q_{12}Q & Q_{13}Q \\ Q_{21}Q & Q_{22}Q & Q_{23}Q \\ Q_{31}Q & Q_{32}Q & Q_{33}Q \end{pmatrix} \in \mathbb{R}^{9 \times 9}$$

$$\overset{r=3}{\curvearrowright} \quad D_Q = D_Q^{(3)} = \begin{pmatrix} Q_{11}Q_{11}\mathcal{Q} & Q_{11}Q_{12}\mathcal{Q} & Q_{11}Q_{13}\mathcal{Q} & & \\ Q_{11}Q_{21}\mathcal{Q} & Q_{11}Q_{22}\mathcal{Q} & Q_{11}Q_{23}\mathcal{Q} & \cdots & \cdots \\ Q_{11}Q_{31}\mathcal{Q} & Q_{11}Q_{32}\mathcal{Q} & Q_{11}Q_{33}\mathcal{Q} & & \\ & \vdots & & & \\ & \vdots & & & \end{pmatrix} \in \mathbb{R}^{27\times 27}$$

Die Matrix D_Q besteht also, wie auch schon in (4-11) ersichtlich ist, aus Q-Monomen vom Grad r, die beim Aufbau der Matrix für ein bestimmtes $Q \in SO(3)$ alle ausgewertet werden müssen. Da identische Monome jedoch an verschiedenen Matrixeinträgen vorkommen können, muss dasselbe Monom mehrmals an der gleichen Stelle ausgewertet werden. Deshalb gilt es bei einer effizienten Berechnung der Matrix D_Q gerade diese Mehrfachberechnungen zu vermeiden. In MATLAB lässt sich die Matrix D_Q mit dem `kron`-Befehl direkt erzeugen. Diese numerische Umsetzung ist jedoch sehr ineffizient. Zwar kann das Skalarprodukt in der Auswertung einer Darstellungsfunktion $d_{S,T}(Q) = \langle S, Q * T\rangle$ an einer Stelle Q für zwei Tensoren $S, T \in \mathcal{J}_r(\mathbb{R}^3)$ aufgrund der Eigenschaft, dass auch $Q * T$ irreduzibel ist, mit der Matrix W aus Kapitel 3.4.1 auf irreduzible Dimension reduziert werden, um den Tensor $Q * T$ jedoch berechnen zu können, muss die Matrix D_Q bei Verwendung des `kron`-Befehls zunächst in voller Dimension erzeugt werden, auch wenn man die anschließende Matrixmultiplikation $D_Q T$ ebenfalls auf irreduzible Komponenten reduzieren kann. Die Mehrfachberechnungen eines identischen Monoms können also nicht verhindert werden. Dieser Tatsache ist es geschuldet, dass diese Vorgehensweise sehr ineffizient ist. Dies macht sich bei ansteigendem Rang, wie es bei den einzelnen Summanden der codf der Fall ist, unmittelbar stärker bemerkbar, da die Dimension $3^r \times 3^r$ von D_Q exponentiell mit dem Rang r steigt. Deshalb werden in den folgenden Abschnitten alternative Möglichkeiten vorgestellt, die Auswertung des RAYLEIGH-Produktes durch Ausnutzung von Symmetrien und Tensoreigenschaften, wie zum Beispiel der Irreduzibilität, effizienter zu machen. Im Anschluss daran folgt in Kapitel 4.4.3 noch ein zeitlicher Vergleich dieser unterschiedlichen Methoden bei numerischer Umsetzung in MATLAB.

4.4.1 Auswertung durch geschicktes Verwalten der Q-Monome

Da das RAYLEIGH-Produkt die Irreduzibilität eines Tensors $T \in \mathcal{J}_r(\mathbb{R}^3)$ erhält (siehe (3-15)), ist der Tensor $F(Q) := Q * T$ für jedes beliebige $Q \in SO(3)$ ebenfalls symmetrisch und spurfrei. Diese Eigenschaften machen wir uns nun zunutze, indem wir das RAYLEIGH-Produkt zunächst auf symmetrische Dimension reduzieren, d.h.

wir schreiben $F(Q)$ in symmetrischer Basis in der Form

$$F(Q)_\alpha = \sum_{|\beta|=r} M(Q)_{\alpha\beta}\, T_\beta$$

mit einer reellen Matrix $M(Q) = \big(M(Q)_{\alpha\beta}\big)$, wobei α und β für die entsprechenden Multiindizes stehen und $M(Q)$ mit $d^r_{\text{sym}} := \frac{1}{2}(r+1)(r+2)$ somit von Dimension $d^r_{\text{sym}} \times d^r_{\text{sym}}$ ist (siehe Tabelle 3.1). Die Matrixeinträge $M(Q)_{\alpha\beta}$ sind demnach ausgewertete Polynome in 9 Unbekannten, den Einträgen der Matrix Q. Im Folgenden wird exemplarisch beschrieben, wie wir für ein gegebenes $Q \in SO(3)$ einen einzelnen Matrixeintrag $M(Q)_{\alpha\beta}$ zu gegebenen Multiindizes α und β berechnen können. Das entsprechende Polynom ist die Summe von jenen Q-Monomen, die bei der Matrixmultiplikation $D_Q T$ aufgrund der Symmetrie von T auf einen identischen Tensoreintrag von T treffen. Die entsprechenden Q-Monome bekommen wir, indem wir ausgehend von einem beliebigen zu α gehörenden Tensorindex $i \in I = \{1,2,3\}^r$ (ausreichend aufgrund der Symmetrie von F) alle zum gegebenen Multiindex β gehörenden Tensorindizes $j \in I$ bestimmen, und die jeweiligen Q-Monome (siehe (4-11)) zunächst durch 9-dimensionale Multiindizes identifizieren. Nummerieren wir die Matrixeinträge von Q spaltenweise durch, d.h. wie in MATLAB automatisch in der Form

$$Q \;=\; \begin{pmatrix} Q_1 & Q_4 & Q_7 \\ Q_2 & Q_5 & Q_8 \\ Q_3 & Q_6 & Q_9 \end{pmatrix} \;,$$

so wird beispielsweise das Q-Monom $Q_1 \cdot Q_3 \cdot Q_7^3 \cdot Q_9^2$ vom Grad 7 durch den zugehörigen 9-dimensionalen Multiindex (1 0 1 0 0 0 3 0 2) beschrieben. Hat man für die gegebenen Multiindizes α und β nun alle zugehörigen Q-Monome bestimmt, wir nehmen an es sind n viele, so kann es auch hier vorkommen, dass dabei identische Q-Monome auftauchen. Da wir dasselbe Q-Monom aber nicht überflüssig mehrmals berechnen wollen, zählen wir von jedem Q-Monom die Häufigkeit seines Auftretens, denn so können wir die zu berechnenden Q-Monome auf die $m \leq n$ vielen paarweise verschiedenen Q-Monome reduzieren. Benennen wir diese Häufigkeiten mit N_k für $k = 1, \ldots, m$, so bekommen wir eine Zuordnung der paarweise verschiedenen Q-Monome zu ihrer jeweiligen Häufigkeit, d.h. insbesondere eine Zuordnung $\gamma_k \leftrightarrow N_k$ zwischen den zu den Q-Monomen gehörenden 9-dimensionalen Multiindizes γ_k und den entsprechenden Häufigkeiten N_k für $k = 1, \ldots, m$. Damit können wir den Matrixeintrag $M(Q)_{\alpha\beta}$ nun wie folgt berechnen:

$$M(Q)_{\alpha\beta} = \sum_{k=1}^{m} N_k Q^{\gamma_k} \tag{4-12}$$

Man beachte, dass diese Vorgehensweise insofern von der Matrix Q unabhängig ist, dass die 9-dimensionalen Multiindizes und ihre Häufigkeiten nicht von Q, sondern

nur vom gegebenen Rang r, abhängig sind. Dies ermöglicht eine allgemeine Implementierung des Aufbaus der Matrix $M(Q)$ in Abhängigkeit des Ranges r. Im Folgenden werden verschiedene Möglichkeiten der numerischen Umsetzung bei der Implementierung der Matrix $M(Q)$ beschrieben.

Pauschale Berechnung aller Q-Monome

Bei dieser Vorgehensweise liegt die Idee darin, durch die Erzeugung aller zum Rang r gehörenden 9-dimensionalen Multiindizes, alle Q-Monome vom Grad r für ein gegebenes $Q \in SO(3)$ pauschal zu berechnen und in einem Spaltenvektor v^r_{mon} anzuordnen. Nach Bemerkung 3.3 sind das

$$d^r_Q \;:=\; \binom{r+8}{8}$$

viele Q-Monome, d.h. es gilt $v^r_{\text{mon}} \in \mathbb{R}^{d^r_Q}$. Durch Multiplikation mit einer entsprechenden Matrix $V_r \in \mathbb{R}^{(d^r_{\text{sym}})^2 \times d^r_Q}$, welche die tatsächlich benötigten Q-Monome für die Berechnung des RAYLEIGH-Produktes aus dem Vektor v^r_{mon} aller Q-Monome auswählt, erhalten wir demnach mit Hilfe des MATLAB-Befehls `reshape`

$$M(Q) \;=\; \texttt{reshape}(V_r \,{}^{*}\, v^r_{\text{mon}} \;,\; d^r_{\text{sym}} \;,\; d^r_{\text{sym}}) \;.$$

Da die Matrix V_r nur vom Rang r abhängig ist und nicht von Q, braucht man sie im Vorfeld für jeden gewünschten Rang jeweils nur einmal zu berechnen und abzuspeichern. Nun stellt sich jedoch trotzdem die Frage, ob man durch die pauschale Berechnung aller Q-Monome zum jeweiligen Rang nicht viel zu viel Rechenaufwand betreibt. Dies ist sicherlich davon abhängig, wieviele der Q-Monome davon letztlich gebraucht werden. Je mehr verschiedene und je öfter das Einzelne gebraucht wird, desto rentabler wird diese Vorgehensweise sein. Da die codf in jedem Summanden für ein gegebenes $Q \in SO(3)$ ein vom Rang abhängiges RAYLEIGH-Produkt enthält, kann man bei der Berechnung aller Q-Monome zum jeweiligen Rang sehr effizient vorgehen. Betrachten wir beispielsweise das Q-Monom $Q_1 \cdot Q_3^2 \cdot Q_4^2$ vom Grad 5, so stellen wir fest, dass wir bei der Berechnung aller Q-Monome vom Grad 6, dieses Q-Monom vom Grad 5 zwangsläufig 9 weitere Male berechnen müssen, bei einer jeweils weiteren Multiplikation mit einem Q_k für $k \in \{1, 2, 3, \ldots, 9\}$. Durch rekursives Aufbauen der 9-dimensionalen Multiindizes zum Rang r aus denjenigen zum Rang $r-1$ können wir dies jedoch verhindern und somit jeweils alle Q-Monome bis zum gewünschten Maximalrang berechnen, ohne dabei eine einzige dieser Doppelberechnungen durchführen zu müssen. Denn mit Hilfe der 9-dimensionalen Multiindizes können wir alle Q-Monome vom Grad r aus den Q-Monomen vom Grad $r-1$ berechnen, indem wir die für den Rang $r-1$ bereits durchgeführten Multiplikationen der Matrixeinträge Q_k konservieren, und sie somit nicht erneut berechnen müssen. Diese Vorgehensweise wird verhältnismäßig umso mehr Rechenzeitersparnis bringen,

je größer der gewünschte Maximalrang ist. Nichtsdestotrotz ist der gesamte Rechenaufwand natürlich vom gewünschten Maximalrang $r_{\max}$ abhängig, und zwar in der Größenordnung

$$\binom{r_{\max}+8}{8} = \frac{(r_{\max}+1)\cdot(r_{\max}+2)\cdot\ldots\cdot(r_{\max}+8)}{8!} = \mathcal{O}(r_{\max}^8)$$

an zu berechnenden Q-Monomen. Es wird sich letztlich herausstellen, dass es wesentlich bessere Möglichkeiten gibt, das RAYLEIGH-Produkt effizient auszuwerten.

Zum einen lässt sich die Matrix $M(Q)$ bei der im nächsten Abschnitt vorgestellten Vorgehensweise noch auf die entsprechende irreduzible Dimension $d_{\text{irr}}^r \times d_{\text{irr}}^r$ mit $d_{\text{irr}}^r := 2r+1$ reduzieren (siehe Tabelle 3.1). Dies ist zwar auch bei der eben vorgestellten Vorgehensweise möglich, was aber nichts daran ändert, dass man hierbei aufgrund der rekursiven Bauart zunächst alle Q-Monome berechnen muss und somit durch die erst anschließende Dimensionsreduktion keine Einsparung an Q-Monom-Berechnungen erzielen kann. Zum anderen besitzen die bei der codf am RAYLEIGH-Produkt beteiligten Tensoren $\mathbb{T}^{r_k}$ in kubischer Symmetrie eine nicht zu vernachlässigende Anzahl an Null-Einträgen. Deshalb müssen die entsprechenden Q-Monome, die bei der Matrixmultiplikation $M(Q)\mathbb{T}^{r_k}$ auf eben diese Null-Einträge treffen, natürlich nicht im Vorfeld berechnet werden. Deshalb bringt uns die Methode des folgenden Abschnittes eine enorme Ersparnis an Q-Monom-Berechnungen, wie wir im späteren Vergleich der verschiedenen Methoden sehen werden.

Berechnung einer bestimmten Auswahl an Q-Monomen

Wie wir an (4-12) sehen können, sind die letztlichen Einträge der Matrix $M(Q)$ zwar von Q abhängig, die Struktur, die hinter $M(Q)$ steckt, ist jedoch nur von den Variablen $Q_1, Q_2, \ldots, Q_9$ abhängig und nicht von deren konkretem Wert in einem bestimmten Fall. Da wir daran interessiert sind, eine sehr allgemeine Implementierung dieser Matrix durchzuführen, bei der erst zuletzt ein bestimmtes $Q \in SO(3)$ eingesetzt wird, um die konkrete Matrix $M(Q)$ zu erhalten, müssen wir uns die exakte Struktur dieser Matrix etwas genauer anschauen. Deshalb betrachten wir zunächst eine Strukturmatrix M, deren Matrixeinträge jeweils eine Ansammlung mathematischer Objekte sein können. Dies ist zunächst als reine Speicheranordnung zu verstehen. Für ein bestimmtes $Q \in SO(3)$ kann man sodann aus den Objekten eines Eintrages dieser Strukturmatrix, nach einer im folgenden beschriebenen Vorgehensweise, einen einzelnen skalaren Wert berechnen. Insgesamt entsteht auf diese Weise aus der Strukturmatrix M sodann die Matrix $M(Q)$. Die Strukturmatrix M ist bezogen auf ihre einzelnen Strukturfelder demnach von gleicher Dimension wie $M(Q)$. Die einzelnen Strukturfelder $M_{\alpha\beta}$ beinhalten jeweils ein sogenanntes Indexfeld und einen gegebenen Vektor. Ein solches Indexfeld ist dabei eine zeilenweise Liste, die die entsprechenden m paarweise verschiedenen Q-Monome enthält, die alle zum gleichen

Multiindex β gehören. Der gegebene Vektor beinhaltet die Häufigkeiten N_k der entsprechenden Q-Monome des zugehörigen Indexfeldes. Um die Q-Monome innerhalb einer solchen Liste schnellstmöglich auswerten zu können, verwenden wir nicht die entsprechenden 9-dimensionalen Multiindizes, sondern identifizieren jedes einzelne Q-Monom durch einen entsprechenden Index. So wird beispielsweise das Q-Monom $Q_1 \cdot Q_3^2 \cdot Q_7^3$ vom Grad 6 durch den Index (1 3 3 7 7 7) repräsentiert. Das hat den Vorteil, dass wir später in MATLAB durch den Befehl

```
prod(Q(Ind),2)
```

mit Hilfe des kompletten Indexfeldes `Ind`, d.h. der Liste dieser zeilenweisen Indizes, alle zu einem Strukturfeld gehörenden Q-Monome gleichzeitig an der Stelle Q auswerten können. Die 2 steht dabei für die zeilenweise Multiplikation. Die bei dieser Vorgehensweise benötigten Indexfelder sind also wie die Häufigkeiten der Q-Monome nur vom Rang abhängig und nicht von Q selbst, und können somit für jeden Rang jeweils im Vorfeld einmal berechnet und abgespeichert werden.

Die allgemeine Aufstellung der Strukturmatrix M kann jedoch noch weiter vereinfacht werden, indem wir M von vornherein zunächst auf die Dimension $d_{\text{irr}}^r \times d_{\text{sym}}^r$ reduzieren, $M_{\alpha\beta}$ also nur für jene Multiindizes α berechnen, die zu den unabhängigen Komponenten eines irreduziblen Tensors in irreduzibler Dimension gehören. Für ein beliebiges $Q \in SO(3)$ überträgt sich dies entsprechend auf $M(Q)$, und mit der Matrix U aus Kapitel 3.4 erhalten wir dann

$$F(Q)_{\mathcal{J}} = M(Q)T_{\mathcal{S}} = M(Q)UT_{\mathcal{J}} =: C(Q)T_{\mathcal{J}} \ .$$

Dadurch erhalten wir für ein beliebiges $Q \in SO(3)$ die Matrix $C(Q) = M(Q)U$ der Dimension $d_{\text{irr}}^r \times d_{\text{irr}}^r$. Anstelle der Strukturmatrix M berechnen wir demnach die entsprechende Strukturmatrix C, die aus der entsprechenden Verknüpfung der Strukturmatrix M mit der Matrix U resultiert. Da wir in jedem Indexfeld eines Strukturfeldes der Strukturmatrix C erneut nur paarweise verschiedene Q-Monome repräsentiert haben wollen, müssen die Häufigkeitsvektoren entsprechend angepasst werden.

Schließlich gibt es noch eine weitere, bereits erwähnte Möglichkeit die Dimension der Strukturmatrix C zu reduzieren. Da die in der codf auftauchenden Tensoren $\mathbb{T}^{r_k}$ für jeden Rang fix sind, können wir uns bei der Berechnung von C all diejenigen Spalten sparen, die bei der späteren Multiplikation $C(Q)\mathbb{T}^{r_k}$ auf Null-Einträge des Tensors $\mathbb{T}^{r_k}$ treffen würden. In kubischer Symmetrie haben die Tensoren $\mathbb{T}^{r_k}$ verhältnismäßig viele Null-Einträge, so dass uns deren Berücksichtigung noch einmal eine erhebliche Ersparnis an Q-Monom-Berechnungen liefert. Die so entstehende Strukturmatrix C ist also schließlich von der Dimension $d_{\text{irr}}^{r_k} \times d_{\text{irr}}^{\text{nne}(r_k)}$, wobei $d_{\text{irr}}^{\text{nne}(r_k)}$ für die Anzahl der Nicht-Null-Einträge des Tensors $\mathbb{T}^{r_k}$ steht. Da auch all diese Überlegungen nur vom Rang und nicht von Q abhängig sind, lässt sich die reduzierte Strukturmatrix C für alle gewünschten Ränge einmal im Vorfeld berechnen und abspeichern.

Für beliebiges $Q \in SO(3)$ erhalten wir letztlich die reduzierte Matrix $C(Q)$, indem wir in jedem Strukturfeld von C die entsprechenden Q-Monome mit obigem `prod`-Befehl auswerten, jeweils mit ihrer Häufigkeit multiplizieren, und aufsummieren.

Als Hinweis sei erwähnt, dass die Implementierung der Strukturmatrix C in MATLAB mit Hilfe sogenannter struct-Variablen zwar zunächst sehr übersichtlich möglich ist, das Berechnen einer konkreten Matrix $C(Q)$ unter Verwendung dieser struct-Variablen jedoch unnötig viel Rechenzeit kostet. Alternativ kann man die mit struct-Variablen erzeugte Matrix C so umstrukturieren, dass man komplett auf structs verzichten kann, indem man alle Indexfelder in einer festgelegten Ordnung in einer großen Matrix untereinander schreibt, entsprechend die Häufigkeiten in einem Vektor. Das Auswerten aller notwendigen Q-Monome erfolgt auf diese Weise mit dem `prod`-Befehl dann sogar in einem Schritt, ebenso das Multiplizieren mit den entsprechenden Häufigkeiten. Beim Aufsummieren der mit ihrer Häufigkeit gewichteten Q-Monome muss nun aber klar sein, welche Q-Monome zu welchem späteren Matrixeintrag $C(Q)$ gehören. Mit dem `reshape`-Befehl gelingt schließlich die richtige Dimensionierung von $C(Q)$.

Vergleich des Aufwandes

In der folgenden Tabelle wird aufgelistet, wieviele Q-Monome bei den beiden Vorgehensweisen für die Auswertung des RAYLEIGH-Produktes in Abhängigkeit des Ranges jeweils ausgewertet werden müssen. Bei der zweiten Methode reduzieren wir die Matrix C mit Hilfe der Null-Einträge der Tensoren $\mathbb{T}^{r_k}$, deren Anzahl jedoch von der gewählten Kristallsymmetrie abhängt. Im Falle von kubischer Kristallsymmetrie erhalten wir bis zum Rang 12 folgendes Resultat:

Rang r_k	alle Q-Monome	Auswahl an Q-Monomen
4	495	150
6	3 003	684
8	12 870	2 303
9	24 310	1 784
10	43 758	6 328
12_1	125 970	9 926
12_2	125 970	8 726

Tabelle 4.2: Vergleich der Anzahlen an Q-Monom-Auswertungen

Während die rekursive Q-Monom-Berechnung bei der ersten Methode darauf basiert, die Q-Monome zu einem beliebigen Grad r für die Berechnung der Q-Monome vom Grad $r+1$ zu konservieren, werden bei der zweiten Methode die Q-Monome aufgrund der unterschiedlichen Null-Einträge der Tensoren $\mathbb{T}^{r_k}$ für jeden Rang se-

parat berechnet. Betrachten wir nun die codf mit ausschließlich den Summanden zu den in der Tabelle aufgeführten Rängen, so erhalten wir für eine Auswertung der codf an einer Stelle $Q \in SO(3)$ schließlich folgenden Vergleich an Anzahlen von auszuwertenden Q-Monomen:

Methode 1 (alle Q-Monome):
Die Anzahl der auszuwertenden Q-Monome wird aufgrund der rekursiven Vorgehensweise lediglich durch den maximalen Rang bestimmt. In diesem Fall sind für eine Auswertung der codf also insgesamt 125 970 Q-Monome vom Grad 12 auszuwerten.

Methode 2 (Auswahl an Q-Monomen):
Die Anzahl der auszuwertenden Q-Monome ergibt sich durch die Summe der Anzahlen zu den einzelnen Rängen, da die Q-Monome hierbei für jeden Rang separat berechnet wurden. In diesem Fall sind für eine Auswertung der codf also insgesamt 29 901 Q-Monome auszuwerten. Im Vergleich zu den 125 970 Q-Monomen vom Grad 12 bei Methode 1, sind die 29 901 Q-Monome bei dieser Methode jedoch von unterschiedlichen Graden, was im Vergleich noch eine weitere Zeitersparnis liefert.

Die insgesamt 29 901 Q-Monom-Auswertungen, die für jede einzelne Auswertung der codf notwendig sind, sind aber nach wie vor zu viel, denkt man nur an die Optimierung bei der Maximum Entropie Methode. Dort muss zur Bestimmung der tensoriellen Lagrange-Multiplikatoren die Funktion in (4-10) minimiert werden. Durch die dabei auftretende Integration sind beim Minimierungsvorgang demnach sehr viele Integrandauswertungen notwendig, und somit ein Vielfaches dieser 29 901 Q-Monom-Auswertungen. Deshalb verfolgen wir im kommenden Abschnitt eine weitere Idee, mit der wir das Rayleigh-Produkt schließlich noch effizienter ausrechnen werden können.

4.4.2 Auswertung mit Hilfe von Funktionalen

Bei diesem Ansatz liegt die Idee darin, die in der codf auftretenden Darstellungsfunktionen vom Typ $d_{S,T}$ mit $d_{S,T}(Q) = \langle S, Q * T\rangle$ mit Hilfe von Funktionalen direkt auszuwerten, d.h. das Skalarprodukt und das Rayleigh-Produkt direkt in einem Schritt zu berechnen. Dazu betrachten wir für ein gegebenes $S \in \mathcal{J}_r(\mathbb{R}^3)$ das folgende lineare Funktional Λ auf $\mathcal{J}_r(\mathbb{R}^3)$:

$$\Lambda : \mathcal{J}_r(\mathbb{R}^3) \longrightarrow \mathbb{R} \quad , \quad T \longmapsto \langle S, T\rangle$$

Somit gilt für ein beliebiges $T \in \mathcal{J}_r(\mathbb{R}^3)$ und ein beliebiges $Q \in SO(3)$

$$\Lambda(D_Q T) = \Lambda(Q * T) = \langle S, Q * T\rangle \ .$$

Bezeichnen wir mit $e_1^*, \ldots, e_{2r+1}^*$ eine Basis des Dualraumes $\mathcal{J}_r(\mathbb{R}^3)^*$, so können wir mit geeigneten Koeffizienten $\lambda_1, \ldots, \lambda_{2r+1} \in \mathbb{R}$ das Funktional Λ bezüglich dieser Basis darstellen, d.h. es gilt $\Lambda = \sum \lambda_i e_i^*$, wobei die λ_i jeweils von dem Tensor S abhängig sind. Mit dem Isomorphismus $\psi : \mathcal{J}_r(\mathbb{R}^3) \longrightarrow \mathcal{H}_r(\mathbb{R}^3)$ aus Satz 3.8

zwischen den irreduziblen Tensoren und den homogenen, harmonischen Polynomen können wir sodann folgendes festhalten:

$$\langle S, Q * T\rangle \;=\; \Lambda(D_Q T) \;=\; \sum_{i=1}^{2r+1} \lambda_i \; e_i^*(D_Q T) \;=\; \sum_{i=1}^{2r+1} \lambda_i \; e_i^* \circ \psi^{-1}\big(\psi(D_Q T)\big)$$

Die Idee ist nun, das dem Tensor $D_Q T$ über ψ zugeordnete homogene, harmonische Polynom $p_{D_Q T}$ mit bestimmten Auswertefunktionalen δ_{z_i} an bestimmten Auswertepunkten $z_i \in \mathbb{R}^3$ auszuwerten, um damit $\langle S, Q * T\rangle$ direkt zu berechnen. Mit Hilfe von Lemma 3.13 können wir die Polynomauswertung sogar auf das Polynom p_T zurückführen, was im Hinblick auf die codf den Vorteil hat, dass aufgrund der dortigen fixen Tensoren $\mathbb{T}^{r_k}$ somit auch die Polynome $p_{\mathbb{T}^{r_k}}$ bekannt sind. Konkret erhalten wir demnach die folgende Beziehung:

$$\begin{aligned}\langle S, Q * T\rangle \;&=\; \sum_{i=1}^{2r+1} \lambda_i \; e_i^* \circ \psi^{-1}\big(\psi(D_Q T)\big) \;=\; \sum_{i=1}^{2r+1} \lambda_i \; e_i^* \circ \psi^{-1}(p_{D_Q T}) \\ &\overset{!}{=} \sum_{i=1}^{2r+1} \lambda_i \; \delta_{z_i}(p_{D_Q T}) \;=\; \sum_{i=1}^{2r+1} \lambda_i \; p_{D_Q T}(z_i) \;=\; \sum_{i=1}^{2r+1} \lambda_i \; p_T(Q^T z_i) \qquad (4\text{-}13)\end{aligned}$$

Nach (3-8) gilt $p_T(Q^T z_i) = \langle (Q^T z_i)^{\otimes r}, T\rangle$ bezüglich der Standard-ONB im $\mathbb{R}^3$. Da der Tensor $(Q^T z_i)^{\otimes r}$ zwar symmetrisch ist, aber im Allgemeinen nicht irreduzibel, ist zu beachten, dass zur Berechnung des Skalarproduktes $\langle (Q^T z_i)^{\otimes r}, T\rangle$, unter Verwendung der entsprechenden Gewichtsmatrix aus Kapitel 3.4.1, die Tensoren $(Q^T z_i)^{\otimes r}$ und T jeweils in symmetrischer Dimension verwendet werden müssen.

Mit der folgenden Gegenüberstellung ist bereits jetzt zu erkennen, worin der Vorteil dieser Vorgehensweise im Vergleich zu jenen aus Kapitel 4.4.1 liegen wird. Dabei werden die jeweiligen Pendants der beiden Methoden gegenübergestellt:

irreduzibler Tensor T	$\overset{\psi}{\longleftrightarrow}$	homogenes, harmonisches Polynom p_T
$Q * T$	$\longleftrightarrow$	$p_T(Q^T \cdot)$
$\langle S, Q * T\rangle$	$\longleftrightarrow$	$\sum \lambda_i \; p_T(Q^T z_i)$
Monome in 9 Variablen	$\longleftrightarrow$	Monome in 3 Variablen
max. Anzahl an Monomen $\binom{\mathrm{r}+8}{8} = \mathcal{O}(r^8)$	$\longleftrightarrow$	max. Anzahl an Monomen $\binom{\mathrm{r}+2}{2} = \mathcal{O}(r^2)$

Damit diese Vorgehensweise jedoch insgesamt funktioniert, muss noch geklärt werden, wie die Auswertepunkte z_i zu wählen sind. Dabei wird klar sein, dass die Auswertepunkte z_i im Allgemeinen zunächst so gewählt werden müssen, dass die zugehörigen Auswertefunktionale δ_{z_i} linear unabhängig sind. Denn wie in der obigen Rechnung vorausgesetzt wurde, muss es sich bei den über $e_i^* := \delta_{z_i} \circ \psi$ letztlich festgelegten Funktionalen auch tatsächlich um eine Basis des Dualraumes $\mathcal{J}_r(\mathbb{R}^3)^*$ handeln. Nach Bestimmung der e_i^* werden wir dann auch in der Lage sein, die Koeffizienten λ_i zu bestimmen.

Für die Wahl der Auswertepunkte z_i gehen wir zunächst erneut von der symmetrischen Dimension als Ausgangslage aus, d.h. wir verwenden zur Berechnung von $\langle S, Q * T\rangle$ in (4-13) eine Summe von $i = 1$ bis $i = d^r_{\mathrm{sym}}$. Das bedeutet nicht nur, dass wir dafür nun auch mehr Auswertepunkte z_i benötigen als im irreduziblen Fall, sondern auch, dass die Koeffizienten λ_i bezüglich einer Basis $e_1^*, \ldots, e_{d^r_{\mathrm{sym}}}^*$ des Dualraumes $\mathcal{S}_r(\mathbb{R}^3)^*$ aufzufassen sind, und wir mit p_T das homogene Polynom bezeichnen, welches dem Tensor T durch den Isomorphismus φ aus Satz 3.6 zugeordnet wird. Dies entspricht also der obigen Herleitung, lediglich für den symmetrischen Fall. Dieser Umweg ist deshalb notwendig, weil wir für die Wahl der Auswertepunkte z_i ein Resultat verwenden werden, welches nur für den symmetrischen Fall von Gültigkeit ist. Es gibt aber numerische Methoden, wie wir uns dieses Resultat auch im irreduziblen Fall von Nutzen machen können. Darauf werden wir später noch eingehen.

Das erwähnte Resultat für die Wahl der Auswertepunkte z_i ist zunächst ein Resultat zur Interpolation auf dem Vektorraum $\mathcal{P}_r(\mathbb{R}^2)$ der reellen Polynome vom Grad $\leq r$ in zwei Variablen. Um es also auf den in unserem Fall von Interesse seienden Vektorraum $\mathcal{V}_r(\mathbb{R}^3)$ der homogenen Polynome vom Grad r in drei Variablen anwenden zu können, betrachten wir folgenden Zusammenhang dieser zwei Vektorräume:

4.1 Satz. *Der Vektorraum $\mathcal{P}_r(\mathbb{R}^2)$ der reellen Polynome vom Grad $\leq r$ in zwei Variablen ist isomorph zum Vektorraum $\mathcal{V}_r(\mathbb{R}^3)$ der homogenen Polynome vom Grad r in drei Variablen.*

Beweis. Die Monome der Menge $\{x_1^i x_2^j x_3^{r-i-j} \mid i,j \in \mathbb{N}_0\ ,\ i+j \leq r\}$ bilden die Monombasis von $\mathcal{V}_r(\mathbb{R}^3)$ und es ist $\dim \mathcal{V}_r(\mathbb{R}^3) = \frac{1}{2}(r+1)(r+2)$. Desweiteren bilden die Monome der Menge $\{x_1^i x_2^j \mid i,j \in \mathbb{N}_0\ ,\ i+j \leq r\}$ die Monombasis von $\mathcal{P}_r(\mathbb{R}^2)$, und deshalb folgt $\dim \mathcal{P}_r(\mathbb{R}^2) = \dim \mathcal{V}_r(\mathbb{R}^3)$. Sei nun $p \in \mathcal{P}_r(\mathbb{R}^2)$ beliebig gewählt, so lässt es sich mit geeigneten Koeffizienten $\alpha_{ij} \in \mathbb{R}$ bezüglich der Monombasis wie folgt darstellen:

$$p(x_1, x_2) \;=\; \sum_{\substack{i,j \\ i+j \leq r}} \alpha_{ij} x_1^i x_2^j$$

Die Abbildung $X : \mathcal{P}_r(\mathbb{R}^2) \longrightarrow \mathcal{V}_r(\mathbb{R}^3)$, unter welcher dieses reelle Polynom p vom

Grad $\leq r$ auf das homogene Polynom $q \in \mathcal{V}_r(\mathbb{R}^3)$ mit

$$q(x_1, x_2, x_3) \;:=\; \sum_{\substack{i,j \\ i+j \,\leq\, r}} \alpha_{ij} x_1^i x_2^j x_3^{r-i-j}$$

abgebildet wird, beschreibt trivialerweise einen Isomorphismus und liefert somit bereits das gewünschte Resultat. □

Kommen wir nun zum angesprochenen Interpolationsresultat auf $\mathcal{P}_r(\mathbb{R}^2)$. Dazu betrachten wir die Anordnung der Knoten $\tilde{z}_i \in \mathbb{R}^2$ der sogenannten nodalen Basis in folgendem Dreieck K (hier für den Fall $r = 4$):

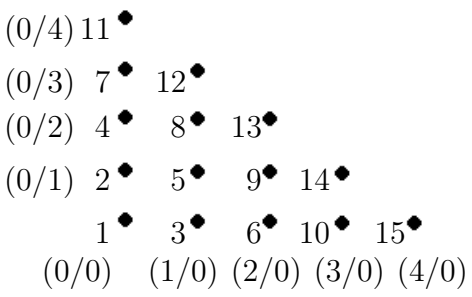


Abbildung 4.2: Knoten der nodalen Basis im Fall $r = 4$

Die Knoten sind demnach auf Linien angeordnet. Für den nächst höheren Grad $r+1$ würde man der Grafik entsprechend alle Knoten zum Grad r identisch übernehmen, und die neuen zusätzlichen Knoten auf der nächsten Diagonalen entsprechend anordnen. Zu dieser Wahl der Knoten erhalten wir schließlich folgendes Interpolationsresultat, dessen Beweis dem Buch von BRAESS[11] zu entnehmen ist:

4.2 Satz. *Sei $r \in \mathbb{N}_0$. In dem Dreieck K seien auf $r+1$ Linien $d^r_{sym} = \frac{1}{2}(r+1)(r+2)$ viele Punkte $\tilde{z}_1, \ldots, \tilde{z}_{d^r_{sym}}$ angeordnet. Dann gibt es zu jeder Funktion $g \in C(K)$ genau ein Polynom $p \in \mathcal{P}_r(\mathbb{R}^2)$, das die Interpolationsaufgabe $p(\tilde{z}_i) = g(\tilde{z}_i)$ für $i = 1, \ldots, d^r_{sym}$ löst.*

Nun übertragen wir dieses Resultat auf die homogenen Polynome aus $\mathcal{V}_r(\mathbb{R}^3)$, d.h. wir klären die endgültige Wahl der Auswertepunkte z_i in unserem Fall und zeigen die notwendige lineare Unabhängigkeit der Auswertefunktionale δ_{z_i} in $\mathcal{V}_r(\mathbb{R}^3)^*$. Dazu seien die Polynome $p_j \in \mathcal{P}_r(\mathbb{R}^2)$ für $j = 1, \ldots, d^r_{\text{sym}}$ die jeweils eindeutig existierenden Lösungen der 2-dimensionalen Interpolationsprobleme $p_j(\tilde{z}_i) = \delta_{ij}$ für $i = 1, \ldots, d^r_{\text{sym}}$ (mit $\tilde{z}_i \in \mathbb{R}^2$ wie oben beschrieben). Wählen wir nun im 3-dimensionalen Fall die **Auswertepunkte**

$$z_i \;:=\; (\tilde{z}_i, 1) \in \mathbb{R}^3 \;,$$

und definieren mit Hilfe der Abbildung X aus dem Beweis zu Satz 4.1 die Polynome $q_j \in \mathcal{V}_r(\mathbb{R}^3)$ für $j = 1, \ldots, d^r_{\text{sym}}$ durch $q_j := X(p_j)$, so gilt $q_j(z_i) = p_j(\tilde{z}_i) = \delta_{ij}$. Mit dieser Wahl der Auswertepunkte z_i erhalten wir linear unabhängige Auswertefunktionale δ_{z_i} in $\mathcal{V}_r(\mathbb{R}^3)^*$. Denn angenommen, für eine beliebige Linearkombination der Auswertefunktionale gilt $\sum_{i=1}^{d^r_{\text{sym}}} \beta_i \delta_{z_i}(P) = 0$ für alle homogenen Polynome $P \in \mathcal{V}_r(\mathbb{R}^3)$, so folgt insbesondere mit den Polynomen $q_j \in \mathcal{V}_r(\mathbb{R}^3)$ die Bedingung

$$0 \;=\; \sum_{i=1}^{d^r_{\text{sym}}} \beta_i \delta_{z_i}(q_j) \;=\; \sum_{i=1}^{d^r_{\text{sym}}} \beta_i q_j(z_i) \;=\; \sum_{i=1}^{d^r_{\text{sym}}} \beta_i \delta_{ij} \;=\; \beta_j$$

für $j = 1, \ldots, d^r_{\text{sym}}$. Diese Wahl der Auswertepunkte z_i liefert demnach mit den Auswertefunktionalen δ_{z_i} eine Basis des Dualraumes $\mathcal{V}_r(\mathbb{R}^3)^*$, woraus wir direkt folgern können, dass die Funktionale $e_1^*, \ldots, e_{d^r_{\text{sym}}}^*$ eine Basis des Dualraumes $\mathcal{S}_r(\mathbb{R}^3)^*$ liefern. Denn betrachten wir die zum Isomorphismus $\varphi : \mathcal{S}_r(\mathbb{R}^3) \longrightarrow \mathcal{V}_r(\mathbb{R}^3)$ gehörende duale Abbildung

$$\begin{aligned} \varphi^* : \mathcal{V}_r(\mathbb{R}^3)^* &\longrightarrow \mathcal{S}_r(\mathbb{R}^3)^* \;, \\ v &\longmapsto v \circ \varphi \end{aligned}$$

welche aufgrund der Isomorphie-Eigenschaft von φ ebenfalls ein Isomorphismus ist, so überträgt sich die Basis-Eigenschaft der $\delta_{z_i} \in \mathcal{V}_r(\mathbb{R}^3)^*$ aufgrund von $e_i^* := \delta_{z_i} \circ \varphi = \varphi^*(\delta_{z_i})$ und der Isomorphie-Eigenschaft von φ^* demnach direkt auf die $e_i^* \in \mathcal{S}_r(\mathbb{R}^3)^*$.

Aufgrund der Definition der $e_i^* : \mathcal{S}_r(\mathbb{R}^3) \longrightarrow \mathbb{R}$ sind wir nun in der Lage, diese **Dualbasis** noch etwas genauer zu beschreiben. Denn für alle $T \in \mathcal{S}_r(\mathbb{R}^3)$ gilt:

$$\begin{aligned} e_i^*(T) \;&=\; \delta_{z_i} \circ \varphi\,(T) \;=\; \delta_{z_i}\big(\varphi(T)\big) \;=\; \delta_{z_i}(p_T) \\ &=\; p_T(z_i) \;=\; \langle z_i^{\otimes r}, T \rangle \;=\; \sum_{|\alpha|=r} \frac{r!}{\alpha!}\, T_\alpha\, z_i^\alpha \end{aligned}$$

Bleibt schließlich noch die Bestimmung der Koeffizienten λ_i des Funktionals Λ bezüglich der Dualbasis $e_1^*, \ldots, e_{d^r_{\text{sym}}}^*$ zu klären. Dazu benötigen wir die zur nun bekannten Dualbasis duale Basis $e_1, \ldots, e_{d^r_{\text{sym}}}$ von $\mathcal{S}_r(\mathbb{R}^3)$. Diese wird durch folgende Bedingung bestimmt:

$$e_j^*(e_i) = \delta_{ji} \tag{4-14}$$

Sei nun $b_1, \ldots, b_{d^r_{\text{sym}}}$ eine beliebige, frei wählbare Basis von $\mathcal{S}_r(\mathbb{R}^3)$, so können wir die noch unbekannten $e_i \in \mathcal{S}_r(\mathbb{R}^3)$ in der Form

$$e_i \;=\; \sum_{k=1}^{d^r_{\text{sym}}} \varrho_{ki} b_k \;\in \mathcal{S}_r(\mathbb{R}^3) \tag{4-15}$$

schreiben für alle $i = 1, \ldots, d^r_{\mathrm{sym}}$ mit geeigneten Koeffizienten $\varrho_{ki} \in \mathbb{R}$, welche es nun zu bestimmen gilt. Dies gelingt uns unter Verwendung der geforderten Bedingung (4-14) bei Berücksichtigung von (4-15):

$$\begin{aligned} \delta_{ji} \overset{!}{=} e_j^*(e_i) &= e_j^*\Big(\sum_{k=1}^{d^r_{\mathrm{sym}}} \varrho_{ki} b_k\Big) \\ &= \sum_{k=1}^{d^r_{\mathrm{sym}}} \varrho_{ki} \underbrace{e_j^*(b_k)}_{=:\, a_{jk}} = \sum_{k=1}^{d^r_{\mathrm{sym}}} a_{jk}\varrho_{ki} \end{aligned} \tag{4-16}$$

Bezeichnen wir mit A die Matrix $A = (a_{jk})$ und mit ϱ die Matrix $\varrho = (\varrho_{ki})$, so bedeutet (4-16) gerade $A\varrho = \mathbb{1}$, d.h. $\varrho = A^{-1}$. Da wir A berechnen können, ist uns demnach auch ϱ bekannt, und somit nach (4-15) auch die **Tensoren** $e_1, \ldots, e_{d^r_{\mathrm{sym}}}$:

$$\begin{pmatrix} | & & | \\ e_1 & \cdots & e_{d^r_{\mathrm{sym}}} \\ | & & | \end{pmatrix} = \underbrace{\begin{pmatrix} | & & | \\ b_1 & \cdots & b_{d^r_{\mathrm{sym}}} \\ | & & | \end{pmatrix}}_{=:\, B} \varrho$$

Bei den Tensoren $e_1, \ldots, e_{d^r_{\mathrm{sym}}} \in \mathcal{S}_r(\mathbb{R}^3)$ handelt es sich aufgrund der Invertierbarkeit von B und ϱ also tatsächlich um eine Basis von $\mathcal{S}_r(\mathbb{R}^3)$, und zwar um die zur Basis $e_1^*, \ldots, e_{d^r_{\mathrm{sym}}}^* \in \mathcal{S}_r(\mathbb{R}^3)^*$ duale Basis.

Da die Berechnung der Koeffizienten ϱ_{ki} von den Einträgen der Matrix A und somit von den Funktionalen e_i^* der Dualbasis abhängt, welche wiederum von der Wahl der Auswertepunkte z_i abhängen, gilt es folgendes zu beachten: Das Dreieck der Auswertepunkte aus Abbildung 4.2 muss eventuell entsprechend skaliert werden, sodass sichergestellt ist, dass die Auswertepunkte weit genug voneinander entfernt sind. Andernfalls kann die Matrix A bei der numerischen Berechnung singulär werden und somit die Berechnung der Inversen A^{-1}, d.h. die Bestimmung von ϱ, unmöglich machen.

Die bezüglich der Dualbasis zu bestimmenden **Koeffizienten** λ_i des Funktionals $\Lambda = \langle S, \cdot \rangle$ zu einem fixen Tensor $S \in \mathcal{S}_r(\mathbb{R}^3)$ erhalten wir schließlich wie folgt:

$$\begin{gathered} \langle S, e_i \rangle = \Lambda(e_i) = \sum_{j=1}^{d^r_{\mathrm{sym}}} \lambda_j e_j^*(e_i) = \sum_{j=1}^{d^r_{\mathrm{sym}}} \lambda_j \delta_{ji} = \lambda_i \\ (i = 1, \ldots, d^r_{\mathrm{sym}}) \end{gathered} \tag{4-17}$$

Bleibt zu klären, wie wir diese Vorgehensweise aus dem symmetrischen Fall nun wieder, wie bereits zu Beginn dieses Kapitels eingeführt, auf den **irreduziblen Fall** übertragen können. Dabei ist folgendes zu beachten: Die Wahl der Auswertepunkte

z_i resultiert aus dem Interpolationsresultat, welches im Allgemeinen nicht auf den irreduziblen Fall übertragbar ist. Wenn es uns jedoch gelingt, aus den d^r_{sym} vielen Auswertepunkte z_i nach einem bestimmten Kriterium d^r_{irr} viele auszuwählen, ohne schließlich die Invertierbarkeit der entsprechenden Matrix A zu verlieren, so werden wir diese Vorgehensweise auch im irreduziblen Fall durchführen können.

Wählen wir als Basis $b_1, \ldots, b_{d^r_{\text{irr}}}$ des $\mathcal{J}_r(\mathbb{R}^3)$ hierbei die Standardbasis, und entscheiden uns für die folgende Wahl von d^r_{irr} vielen Auswertepunkten (rot eingefärbt) aus den d^r_{sym} vielen des symmetrischen Falls, so wird uns dies gelingen (hier erneut für den Fall $r = 4$):

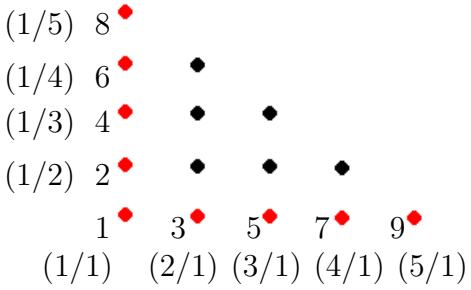


Abbildung 4.3: Auswahl an Knoten der nodalen Basis
im irreduziblen Fall für $r = 4$

Die Angaben der Koordinaten der Auswertepunkte beziehen sich in dieser Grafik lediglich auf die x_1- und x_2-Koordinate. Auch hier setzen wir erneut $x_3 := 1$ für alle Auswertepunkte. Die Tatsache, dass wir im Vergleich zum symmetrischen Fall jedoch andere Koordinaten für die Auswertepunkte gewählt haben, hat einen einfachen Grund: Im irreduziblen Fall benötigen wir lediglich die rot eingefärbten Auswertepunkte. Würden wir also dieselben Koordinaten verwenden wie im symmetrischen Fall, so hätten alle Auswertepunkte mindestens eine Null-Koordinate, was in der Berechnung

$$a_{jk} \;=\; e_j^*(b_k) \;=\; \ldots \;=\; \langle z_j^{\otimes r}, b_k \rangle \;=\; \sum_{|\alpha|=r} \frac{r!}{\alpha!}\, (b_k)_\alpha \; z_j^\alpha \tag{4-18}$$

der entsprechenden Matrix A zu Nullspalten führen kann. In solch einem Fall wäre A also nicht invertierbar. Im symmetrischen Fall tritt dies nicht auf, da die dort ebenfalls benötigten, schwarz eingefärbten Auswertepunkte unproblematische Koordinaten haben. Im irreduziblen Fall können wir diesem Effekt jedoch entgegen wirken, indem wir das Dreieck der Auswertepunkte in der Ebene $x_3 = 1$ einfach verschieben, denn es kommt nur auf die Lage der Auswertepunkte zueinander an und nicht auf die Position des Dreiecks in dieser Ebene. Die Invertierbarkeit der so aufgestellten Matrix A wurde mit MATLAB numerisch überprüft. Gegebenenfalls muss man auch hier das Dreieck entsprechend skalieren.

Die Berechnung der Koeffizientenmatrix $\varrho = A^{-1}$ verläuft bei dieser Wahl der Auswertepunkte demnach analog zum symmetrischen Fall. Wie in (4-18) bereits abzulesen ist, ist bei der Berechnung der Einträge a_{jk} jedoch erneut darauf zu achten, dass aufgrund der im Allgemeinen nicht vorhandenen Irreduzibilität von $z_j^{\otimes r}$ das auftretende Skalarprodukt nicht in irreduzibler Dimension berechnet werden kann.

Auch die Berechnung der Koeffizienten λ_i erfolgt in irreduzibler Dimension analog zur Vorgehensweise in (4-17). Somit ist es uns gelungen, den symmetrischen Fall auf den irreduziblen Fall zu übertragen. Deshalb werden wir im Folgenden nur noch auf den irreduziblen Fall Bezug nehmen.

Da wir es im Hinblick auf die codf in den einzelnen Summanden $\langle \mu_k, Q * \mathbb{T}^{r_k} \rangle$ mit fixen Tensoren μ_k bzw. $\mathbb{T}^{r_k}$ zu tun haben, ist es auch bei dieser Vorgehensweise möglich, die Berechnung der Koeffizienten λ_i einmalig im Voraus für jeden Rang durchzuführen. Denn auch sie sind nicht von Q, sondern lediglich vom Rang und den fixen Tensoren μ_k bzw. $\mathbb{T}^{r_k}$ und den pro Rang fixen Auswertepunkten z_i abhängig. Dasselbe gilt selbstverständlich auch für die Tensorbasis $e_1, \ldots, e_{d_{\text{irr}}^{r_k}}$. Das erleichtert die mehrfache Auswertung der codf ungemein.

Bevor wir die Tensoren μ_k jedoch kennen, müssen wir diese als Lösung eines Optimierungsproblems bestimmen, d.h. als Minimalstelle der zu minimierenden Funktion Φ in (4-10). Um diese Minimalstelle zu berechnen, suchen wir mit einem entsprechenden Optimierungsalgorithmus jene LAGRANGE-Multiplikatoren $\mu = [\mu_0; \mu_1; \ldots; \mu_n]$, für die $\nabla\Phi(\mu) = 0$ gilt. Dies bedeutet jedoch, dass wir auf dem Weg zur Minimalstelle μ in jedem Optimierungsschritt eine Approximation $\tilde{\mu}$ der Minimalstelle erhalten, sodass wir demnach in jedem Optimierungsschritt neue Koeffizienten λ_i bestimmen müssten, um die Summanden $\langle \tilde{\mu}_k, Q * \mathbb{T}^{r_k} \rangle$ zu berechnen. Da aber in den Nebenbedingungen (4-9) des Maximum Entropie Momentenproblems, welche im Gradienten von Φ auftauchen, das RAYLEIGH-Produkt $Q * \mathbb{T}^{r_k}$ auch separat vorkommt, d.h. ohne gleichzeitige Auswertung mit dem Skalarprodukt $\langle \tilde{\mu}_k, Q * \mathbb{T}^{r_k} \rangle$, sollte noch geklärt werden, wie wir, mit der in diesem Kapitel vorgestellten Vorgehensweise, das RAYLEIGH-Produkt $Q * \mathbb{T}^{r_k}$ auch separat berechnen können.

Dies betrachten wir am allgemeinen Fall für gegebene Tensoren $S, T \in \mathcal{J}_r(\mathbb{R}^3)$. Um zu sehen, wie wir die Vorgehensweise zur direkten Berechnung der Darstellungsfunktion $d_{S,T}$ mit $d_{S,T}(Q) = \langle S, Q * T \rangle$ für ein beliebiges $Q \in SO(3)$ verwenden können, um damit auch das RAYLEIGH-Produkt $Q * T$ separat effizient zu berechnen, betrachten wir nun analog zu (4-17) folgenden Zusammenhang in irreduzibler Dimension:

$$\lambda := \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{d_{\text{irr}}^r} \end{pmatrix} = \begin{pmatrix} \langle WS_{\mathcal{J}}, e_{1_{\mathcal{J}}} \rangle_{\mathcal{J}_r(\mathbb{R}^3)} \\ \vdots \\ \langle WS_{\mathcal{J}}, e_{d_{\text{irr}}^r{}_{\mathcal{J}}} \rangle_{\mathcal{J}_r(\mathbb{R}^3)} \end{pmatrix} = \underbrace{\begin{pmatrix} — & e_{1_{\mathcal{J}}}^T & — \\ & \vdots & \\ — & e_{d_{\text{irr}}^r{}_{\mathcal{J}}}^T & — \end{pmatrix}}_{=:\, E^T} WS_{\mathcal{J}}$$

Definieren wir noch den Vektor $P \in \mathbb{R}^{d^r_{\text{irr}}}$, der die Werte des Polynoms p_T an den gedrehten Auswertepunkten $Q^T z_i$ beinhalten soll, d.h. es gilt $P_i := p_T(Q^T z_i)$ für $i = 1, \ldots, d^r_{\text{irr}}$, so erhalten wir mit Hilfe von (4-13) für das RAYLEIGH-Produkt $D_Q T = Q * T$ entsprechend in irreduzibler Dimension

$$(WS_{\mathcal{J}})^T (D_Q T)_{\mathcal{J}} \;=\; \langle S, D_Q T \rangle \;=\; \sum_{i=1}^{d^r_{\text{irr}}} \lambda_i \; p_T(Q^T z_i) \;=\; \lambda^T P \;=\; (WS_{\mathcal{J}})^T E P \; .$$

Da dies aufgrund der Unabhängigkeit der Größen $D_Q T, W, E$ und P von S für alle $S \in \mathcal{J}_r(\mathbb{R}^3)$ gilt, erhalten wir schließlich

$$(Q * T)_{\mathcal{J}} \;=\; (D_Q T)_{\mathcal{J}} \;=\; EP \; .$$

Somit werden die Koeffizienten λ_i für die separate Berechnung des RAYLEIGH-Produktes nicht benötigt, sondern lediglich die Tensorbasis $e_1, \ldots, e_{d^r_{\text{irr}}}$ und die Auswertepunkte z_i, welche jeweils für jeden Rang nur einmal berechnet werden müssen.

Das Ergebnis des so berechneten RAYLEIGH-Produktes verwenden wir dann auch jeweils dazu, das im Exponentialteil der Zielfunktion Φ in jedem Optimierungsschritt auftretende Skalarprodukt $\langle \tilde{\mu}_k, Q * \mathbb{T}^{r_k} \rangle$ auf dem herkömmlichen Weg, wie in Kapitel 3.4.1 beschrieben, zu berechnen. Das erspart uns Rechenzeit im Vergleich zu der Alternative, für das jeweilige Skalarprodukt erneut die in diesem Kapitel vorgestellte Vorgehensweise anzuwenden. Somit kommen wir auch um die in diesem Fall sonst notwendige Neuberechnung der jeweiligen Koeffizienten λ_i herum.

Rechnen wir das RAYLEIGH-Produkt auf diese Weise aus, kommt es ebenso darauf an, Monome effizient auswerten zu können - im Falle der codf die Monome des Polynoms

$$p_{\mathbb{T}^{r_k}}(x) \;=\; \sum_{|\alpha|=r_k} \frac{r_k!}{\alpha!} \; \mathbb{T}^{r_k}_{\alpha} \; x^{\alpha} \quad , \quad x \in \mathbb{R}^3 \tag{4-19}$$

an den Stellen $Q^T z_i$. Der Vorteil hierbei ist, dass es sich im Vergleich zu den Methoden aus Kapitel 4.4.1 lediglich um Monome in 3 Variablen handelt. Im Folgenden seien auch hier verschiedene Methoden betrachtet, das Polynom $p_{\mathbb{T}^{r_k}}$ auszuwerten.

Berechnung aller Monome in 3 Variablen

Bei dieser Methode berechnen wir alle Monome in 3 Variablen für die gewünschten Ränge durch analoges Vorgehen wie in Kapitel 4.4.1. Da wir dies, aufgrund der verschiedenen Auswertepunkte z_i, pro auszuwertendem RAYLEIGH-Produkt jedoch d^r_{irr}-mal durchführen müssen, beläuft sich der insgesamte Rechenaufwand bei gewünschtem Maximalrang $r_{\max}$ auf die Größenordnung

$$d^{r_{\max}}_{\text{irr}} \cdot \binom{r_{\max}+2}{2} \;=\; (2r_{\max}+1) \cdot \frac{(r_{\max}+1)(r_{\max}+2)}{2} \;=\; \mathcal{O}(r^3_{\max})$$

an auszuwertenden Monomen in 3 Variablen. Im Vergleich zu der folgenden Auswertemethode benötigen wir für diese Vorgehensweise jedoch erneut mehr Rechenaufwand, wie anhand von Tabelle 4.3 ersichtlich ist.

Berechnung einer bestimmten Auswahl an Monomen in 3 Variablen

Der Unterschied zur vorherigen Vorgehensweise liegt bei dieser Methode schlicht und ergreifend darin, die Anzahl der auszuwertenden Monome erneut durch Ausnutzung der Null-Einträge der fixen Tensoren $\mathbb{T}^{r_k}$ zu reduzieren. Demnach brauchen wir in (4-19) nur diejenigen Monome x^α pro gewünschtem Rang r_k auszuwerten, die bei der Auswertung von $p_{\mathbb{T}^{r_k}}$ nicht auf einen Null-Eintrag von $\mathbb{T}^{r_k}$ treffen.

Im Folgenden betrachten wir zwei Möglichkeiten, wie die Monome $x^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3}$ numerisch effizient ausgewertet werden können:

Möglichkeit 1 (Multiindizes):
Bezeichnen wir mit `multis` die matrixwertige Liste, welche zeilenweise alle zum Rang r_k gehörenden Multiindizes α beinhaltet, für welche $\mathbb{T}^{r_k}_\alpha \neq 0$ gilt, so können wir mit dem folgenden MATLAB-Code

```
(x(1).^multis(:,1)).*(x(2).^multis(:,2)).*(x(3).^multis(:,3))
```

alle zu den Multiindizes zugehörigen Monome gleichzeitig an einer Stelle $x \in \mathbb{R}^3$ auswerten. Durch Normierung einer Komponente von x kann zusätzlich Rechenzeit gespart werden. Die dritte Komponente der Auswertepunkte z_i ist jeweils identisch 1. Da wir das Polynom $p_{\mathbb{T}^{r_k}}$ jedoch an den Stellen $x = Q^T z_i$ auswerten müssen, geht diese Eigenschaft bei x zunächst im Allgemeinen verloren. Normieren wir die dritte Komponente von x durch $\frac{1}{x_3}x$ und nutzen die Eigenschaft $q(x) = \frac{1}{c^m} q(cx)$ homogener Polynome q vom Grad m für $c \neq 0$ aus, so können wir $p_{\mathbb{T}^{r_k}}(x)$ mit Hilfe von $p_{\mathbb{T}^{r_k}}(\frac{1}{x_3}x)$ schneller berechnen, denn bei der Berechnung von $(\frac{1}{x_3}x)^\alpha$ können wir auf die Potenzen der dritten Komponente komplett verzichten. Benennen wir den Auswertepunkt mit der normierten dritten Komponente in MATLAB mit dem Namen `cx`, so ergibt sich für diese Berechnung nun folgender Code:

```
(cx(1).^multis(:,1)).*(cx(2).^multis(:,2))
```

Möglichkeit 2 (`prod`-Befehl):
Verwenden wir zu den Multiindizes aus *Möglichkeit 1* jeweils einen stellvertretenden Tensorindex und speichern diese zeilenweise in der matrixwertigen Liste `indizes` ab, so gelingt uns in MATLAB durch

```
prod(x(indizes),2)
```

eine noch schnellere, gleichzeitige Auswertung aller zu den Multiindizes gehörenden Monome an einer Stelle $x = Q^T z_i$. Aufgrund der sehr effizienten Programmierweise

des `prod`-Befehls lohnt es sich hier nicht, eine Komponente von x zu normieren. Im Gegenteil, die Normierung kostet mehr Zeit als beim `prod`-Befehl die entsprechende Komponente einzusparen.

In der Anzahl an auszuwertenden Monomen unterscheiden sich diese beiden Möglichkeiten nicht. Der zeitliche Gesamtaufwand für die Auswertung des Polynoms $p_{\mathbb{T}^{r_k}}$ mit Hilfe von *Möglichkeit 2* ist jedoch geringer, wie in Kapitel 4.4.3 nachzulesen ist.

Vergleich des Aufwandes

In der folgenden Tabelle wird aufgelistet, wieviele Monome in 3 Variablen bei den beiden Methoden der letzten zwei Abschnitte für die Auswertung des RAYLEIGH-Produktes in Abhängigkeit des Ranges jeweils ausgewertet werden müssen. Auch hier reduzieren wir bei der zweiten Methode die Anzahl an auszuwertenden Monomen durch Berücksichtigung der Null-Einträge der Tensoren $\mathbb{T}^{r_k}$, deren Anzahl jedoch von der gewählten Kristallsymmetrie abhängt. Im Falle von kubischer Kristallsymmetrie erhalten wir bis zum Rang 12 folgende Anzahlen an auszuwertenden Monomen:

Rang r_k	alle Monome in 3 Var.	Auswahl an Monomen in 3 Var.
4	135	54
6	364	130
8	765	204
9	1 045	114
10	1 386	441
12_1	2 275	525
12_2	2 275	550

Tabelle 4.3: Vergleich der Anzahlen an Monom-Auswertungen in 3 Variablen

Während auch hier die Monom-Berechnung bei der ersten Methode darauf basiert, die Monome zu einem beliebigen Grad r für die Berechnung der Monome vom Grad $r+1$ zu konservieren, werden bei der zweiten Methode die Monome aufgrund der unterschiedlichen Null-Einträge der Tensoren $\mathbb{T}^{r_k}$ für jeden Rang separat berechnet. Betrachten wir nun die codf mit ausschließlich den Summanden zu den in der Tabelle aufgeführten Rängen, so erhalten wir für eine Auswertung der codf an einer Stelle $Q \in SO(3)$ schließlich folgenden Vergleich an Anzahlen von auszuwertenden Monomen in 3 Variablen:

Methode 1 (alle Monome in 3 Variablen):
Die Anzahl der auszuwertenden Monome wird aufgrund der rekursiven Vorgehensweise lediglich durch den maximalen Rang bestimmt. In diesem Fall sind für eine Auswertung der codf also insgesamt 2 275 Monome vom Grad 12 auszuwerten.

Methode 2 (Auswahl an Monomen in 3 Variablen):
Die Anzahl der auszuwertenden Monome ergibt sich durch die Summe der Anzahlen zu den einzelnen Rängen, da die Monome hierbei für jeden Rang separat berechnet wurden. In diesem Fall sind für eine Auswertung der codf also insgesamt 2 018 Monome auszuwerten. Im Vergleich zu den 2 275 Monomen vom Grad 12 bei Methode 1, sind die 2 018 Monome bei dieser Methode jedoch von unterschiedlichen Graden, was im Vergleich noch eine weitere Zeitersparnis liefert.

Vergleichen wir die Vorgehensweise unter Verwendung der Funktionale mit jener unter Verwendung der Q-Monome aus Kapitel 4.4.1, so stellen wir bei einem Blick auf die Tabellen 4.3 und 4.2 noch einmal eine große Reduktion der Anzahlen an auszuwertenden Monomen zu Gunsten der Funktionalmethode fest. Wie sich dieser Unterschied im zeitlichen Aufwand bei einer Implementierung dieser Methoden bemerkbar macht, wird im folgenden Kapitel betrachtet.

4.4.3 Zeitlicher Vergleich

In diesem Abschnitt vergleichen wir die in den Kapiteln 4.4.1 und 4.4.2 vorgestellten Methoden, eine Darstellungsfunktion bzw. das RAYLEIGH-Produkt auszuwerten, hinsichtlich des zeitlichen Aufwands bei einer Implementierung mit MATLAB. Im Detail vergleichen wir die folgenden Methoden:

- Erzeugung der Matrix D_Q mit Hilfe des `kron`-Befehls
- Berechnung aller Q-Monome
- Berechnung einer Auswahl an Q-Monomen, ausgewertet mit Hilfe des `prod`-Befehls
- Berechnung aller Monome in 3 Variablen
- Berechnung einer Auswahl an Monomen in 3 Variablen, ausgewertet mit Hilfe von Multiindizes
- Berechnung einer Auswahl an Monomen in 3 Variablen, ausgewertet mit Hilfe des `prod`-Befehls

Um die Methoden sinnvoll miteinander vergleichen zu können, messen wir für jede Methode jeweils den gesamten zeitlichen Aufwand bei Auswertung der Summanden

$$1 + \sum_{k=1}^{n} \langle \mathbb{S}^{r_k}, Q * \mathbb{T}^{r_k} \rangle$$

der theoretischen codf (4-1) an 10 000 paarweise verschiedenen Stellen $Q \in SO(3)$. Dies führen wir bei jeder Methode jeweils für unterschiedliche $n \in \mathbb{N}$ durch. Bei den

Tensoren $\mathbb{S}^{r_k}$ handelt es sich dabei um die nach (4-5) bestimmten, auf realen Daten basierenden tensoriellen Texturkoeffizienten. Die Tensoren $\mathbb{T}^{r_k}$ weisen in diesem Fall die kubische Kristallsymmetrie auf. Für $n = 1, \ldots, 10$ ergeben sich in kubischer Kristallsymmetrie die folgenden zu betrachtenden Ränge r_k:

$$r_k \in \{4, 6, 8, 9, 10, 12_1, 12_2, 13, 14, 15\}$$

Die gemessenen Zeiten sind keineswegs als absolute Zeiten zu verstehen, sondern sollten lediglich relativ zueinander betrachtet werden. Sie dienen dazu, zu sehen, welche der Methoden relativ gesehen am wenigsten Rechenaufwand benötigt. Bei Verwendung anderer Implementiersprachen sind noch wesentlich schnellere Zeiten möglich.

$n \in \mathbb{N}$	`kron` Zeit [sec]	alle Q-Monome Zeit [sec]	Auswahl Q-Monome `prod` Zeit [sec]
1	2.1	525.2	0.6
2	111.6		1.4
3			3.7
4			5.3
5			11.7
6			26.3
7			43.1
8			80.5
9			155.0

$n \in \mathbb{N}$	alle $3d$-Monome Zeit [sec]	Auswahl $3d$-Monome Multiindizes Zeit [sec]	Auswahl $3d$-Monome `prod` Zeit [sec]
1	23.4	3.7	2.9
2	84.0	8.8	6.7
3	209.8	15.6	11.8
4		22.8	17.1
5		32.0	23.9
6		43.0	32.4
7		54.6	40.5
8		65.4	49.1
9		78.8	59.3
10		93.8	69.4

Tabelle 4.4: Zeitlicher Vergleich der verschiedenen Auswertemethoden

Vergleichen wir nun diese Zeiten, so stellen wir fest, dass wir bei der Implementierung in MATLAB bei Auswertungen einschließlich bis Rang $r_k = 12_1$ ($n = 6$) am schnellsten mit der Auswahl an Q-Monomen arbeiten. Ab Berücksichtigung von Rang $r_k = 12_2$ ($n = 7$) liefert die Methode mit den Funktionalen, d.h. der Auswahl an Monomen in 3 Variablen gekoppelt mit dem `prod`-Befehl, die schnellsten Zeiten. Die Verwendung des `kron`-Befehls bzw. die Berechnung aller jeweiligen Monome stellt sich als zeitlich sehr ungünstig heraus. Doch wie kann es sein, dass sich die Funktionalmethode trotz der wesentlich geringeren Anzahlen an auszuwertenden Monomen zeitlich erst ab $n = 7$ durchsetzt? Die Antwort liegt lediglich in der verwendeten Implementiersprache MATLAB. Während bei der Auswahl an Q-Monomen die Q-Monome mit dem `prod`-Befehl schnell ausgewertet werden können, und zur Berechnung des RAYLEIGH-Produktes anschließend nur noch eine Matrixmultiplikation notwendig ist (worauf MATLAB ausgelegt ist), kommen wir bei der Funktionalmethode aufgrund des `prod`-Befehls und den unterschiedlichen Auswertepunkten $Q^T z_i$ zur Polynomauswertung nicht um eine Schleifenbildung über die Anzahl der Auswertepunkte herum, welche in MATLAB sehr zeitintensiv ist. Dieser zeitliche Nachteil kehrt sich, aufgrund des mit steigendem Rang verhältnismäßig immer größer werdenden Unterschieds der beiden Methoden bezüglich der Anzahlen an auszuwertenden Monomen, erst ab $n = 7$ um. Bei Verwendung einer schleifenfreundlichen Implementiersprache wird die Funktionalmethode aufgrund der wesentlich geringeren Anzahl an auszuwertenden Monomen jedoch für alle $n \in \mathbb{N}$ die zeitlich schnellste Berechnungsmethode sein.

5 Integration über $SO(3)$

In diesem Kapitel beschäftigen wir uns mit der speziellen Integration über $SO(3)$, welche uns zum Beispiel beim Maximum Entropie Momentenproblem in den tensoriellen Nebenbedingungen (4-9) bzw. bei der Optimierung der Zielfunktion

$$\Phi(\omega) = \int\limits_{SO(3)} \exp\Big(-1 + \omega_0 + \sum_{k=1}^{n} \langle \omega_k, Q * \mathbb{T}^{r_k} \rangle \Big)\, dQ \;-\; \omega_0 \;-\; \sum_{k=1}^{n} \frac{\langle \omega_k, \mathbb{S}^{r_k} \rangle}{2r_k + 1}$$

in (4-10) begegnet. Bevor wir uns anschauen, wie wir solche Integrale numerisch lösen können, stellt sich jedoch die Frage nach der zu verwendenden Parametrisierung von $SO(3)$ und dem daraus resultierenden HAAR-Maß.

5.1 Parametrisierungen von $SO(3)$

$SO(3)$ kann auf mehrere verschiedene Arten und Weisen parametrisiert werden, wie zum Beispiel mit der Achse-Winkel-Methode, den EULER-Winkeln, mit Hilfe von Quaternionen oder der Oberfläche der zweidimensionalen und der dreidimensionalen Einheitskugel.[38] In den folgenden Abschnitten gehen wir auf zwei der erwähnten Parametrisierungen etwas genauer ein.

5.1.1 Parametrisierung mit Achse und Winkel

Der folgende Satz zeigt, dass man jede Rotation $R \in SO(3)$ durch Angabe einer Rotationsachse und einem Rotationswinkel exakt beschreiben kann, und liefert gleichzeitig eine mögliche Parametrisierung von $SO(3)$. Der Beweis dieses Satzes ist dem Buch von KOECHER[26] zu entnehmen:

5.1 Satz. *Sei $R_0 \in SO(3)$ beliebig gewählt. Dann existiert zu jeder orthogonalen Matrix $R \in SO(3)$ ein $z \in \mathbb{R}^3$, sodass R mit Hilfe der schiefsymmetrischen Matrix*

$$A(z) \;:=\; \begin{pmatrix} 0 & -z_3 & z_2 \\ z_3 & 0 & -z_1 \\ -z_2 & z_1 & 0 \end{pmatrix}$$

und der Matrix-Exponentialfunktion in der Form $R = \exp\big(A(z)\big)R_0$ dargestellt werden kann. Hierbei gilt die sogenannte RODRIGUEZ*-Formel*

$$\exp\big(A(z)\big) \;=\; \cos\|z\|_2 \cdot \mathbb{1} \;+\; \sin\|z\|_2 \cdot A\big(\tfrac{z}{\|z\|_2}\big) \;+\; \frac{1-\cos\|z\|_2}{\|z\|_2^2} \cdot zz^T \;.$$

Desweiteren entspricht $\|z\|_2$ gerade dem Rotationswinkel und z der Rotationsachse der Rotation RR_0^T.

Setzen wir $R_0 := \mathbb{1}$, so erhalten wir durch die Wahl eines beliebigen Rotationswinkels $\omega \in [0,\pi]$ und einer beliebigen Rotationsachse $n \in S^2$ (d.h. es gilt $\|n\|_2 = 1$) die folgende Parametrisierung von $SO(3)$:

$$\begin{aligned} P:\; [0,\pi]\times S^2 \;&\longrightarrow\; SO(3) \\ (\omega, n) \;&\longmapsto\; \cos\omega\cdot\mathbb{1} \;+\; \sin\omega\cdot A(n) \;+\; (1-\cos\omega)\cdot nn^T \end{aligned}$$

Dabei betrachten wir bei Rotationen um eine Achse n nur Winkel aus $[0,\pi]$, denn Rotationen um n mit einem Winkel aus $[\pi, 2\pi]$ können durch Rotationen mit einem Winkel aus $[0,\pi]$ um die Achse $-n$ dargestellt werden. Die Doppeldeutigkeit einer Rotation, die wir unter anderem durch $P(\pi, n) = P(\pi, -n)$ und $P(0, n) = \mathbb{1}$ für alle $n \in S^2$ dennoch erhalten, hat auf die spätere Integration jedoch keine Auswirkung, da die Menge, welche durch solche Bedingungen charakterisiert wird, eine Nullmenge bezüglich dem Haar-Maß ist.

Das zugehörige Haar-Maß ist in diesem Fall bis auf einen Faktor identisch mit dem entsprechenden Lebesgue-Maß λ.[16] Demnach erhalten wir mit Hilfe des Transformationssatzes und der Gramschen Determinante

$$\int\limits_{SO(3)} d\lambda \;=\; \int\limits_{P([0,\pi]\times S^2)} d\lambda \;=\; \int\limits_{S^2}\int\limits_{0}^{\pi} \sqrt{\mathrm{Gram}(\nabla_\omega P, \nabla_{h_1} P, \nabla_{h_2} P)}\; d\omega\; dO \;, \qquad (5\text{-}1)$$

wobei $\nabla_\omega P(\omega, n) = \partial_\omega P(\omega, n)$, und $\nabla_{h_1} P$ bzw. $\nabla_{h_2} P$ die Richtungsableitung von P in Richtung $h_1 \in \mathbb{R}^3$ bzw. $h_2 \in \mathbb{R}^3$ am entsprechenden Punkt $n \in S^2$ bezeichnet. Die Richtungen h_1 und h_2 sind dabei linear unabhängig und aus dem entsprechenden Tangentialraum an S^2 im Punkt n zu wählen. Zweckmäßigerweise wählen wir zueinander orthogonale und normierte Richtungen h_1 und h_2, d.h. es gilt

$$n^T h_i = 0 \quad , \quad h_i^T h_j = \delta_{ij} \quad \text{für } i,j = 1,2 \;.$$

Die Gramsche Determinante ist dabei gegeben durch

$$\mathrm{Gram}(\nabla_\omega P, \nabla_{h_1} P, \nabla_{h_2} P) \;=\; \det \begin{pmatrix} \langle \nabla_\omega P, \nabla_\omega P\rangle & \langle \nabla_\omega P, \nabla_{h_1} P\rangle & \langle \nabla_\omega P, \nabla_{h_2} P\rangle \\ \langle \nabla_{h_1} P, \nabla_\omega P\rangle & \langle \nabla_{h_1} P, \nabla_{h_1} P\rangle & \langle \nabla_{h_1} P, \nabla_{h_2} P\rangle \\ \langle \nabla_{h_2} P, \nabla_\omega P\rangle & \langle \nabla_{h_2} P, \nabla_{h_1} P\rangle & \langle \nabla_{h_2} P, \nabla_{h_2} P\rangle \end{pmatrix},$$

wobei $\langle \cdot, \cdot \rangle$ in diesem Fall für das Matrixskalarprodukt $\langle X, Y \rangle := \text{spur}(XY^T)$ für $X, Y \in \mathbb{R}^{3\times 3}$ steht. Führen wir diese konkrete Berechnung der GRAMschen Determinante durch, so erhalten wir schließlich

$$\text{Gram}(\nabla_\omega P, \nabla_{h_1} P, \nabla_{h_2} P) \;=\; 32\,(1-\cos\omega)^2\;. \tag{5-2}$$

Für das HAAR-Maß gilt mit einem Faktor $c > 0$, wie bereits erwähnt, die Beziehung $dQ = c\; d\lambda$. Damit und mit Hilfe von (5-1) und (5-2) erhalten wir, bei zusätzlicher Forderung der Normierungsbedingung $\int_{SO(3)} dQ = 1$, aus

$$\begin{aligned}
\int\limits_{SO(3)} dQ \;&=\; \int\limits_{S^2}\int\limits_{0}^{\pi} c\; 4\sqrt{2}\;(1-\cos\omega)\; d\omega\; dO \\
&=\; 16\sqrt{2}\;\pi\;c\int\limits_{0}^{\pi}(1-\cos\omega)\; d\omega \;=\; 16\sqrt{2}\;\pi\;c\;\big[\omega-\sin\omega\big]_0^\pi \\
&=\; 16\sqrt{2}\;\pi^2\;c \;\stackrel{!}{=}\; 1
\end{aligned}$$

die Bedingung $c = (16\sqrt{2}\;\pi^2)^{-1}$, und somit schließlich

$$dQ \;=\; \frac{1}{4\pi^2}\;(1-\cos\omega)\; d\omega\; dO\;.$$

5.1.2 Parametrisierung mit Euler-Winkeln

Mit Hilfe der EULER-Winkel[12] lässt sich jede Rotation $R \in SO(3)$ in drei hintereinander ausgeführte Rotationen zerlegen. Bei der Angabe dieser drei Rotationen gibt es jedoch verschiedene Konventionen. Wir verwenden die sogenannte zxz-Konvention, mit der wir jede beliebige Rotation durch Hintereinanderausführung einer Rotation um die z-Achse, einer um die x-Achse und erneut einer um die z-Achse beschreiben können. Die Winkel, um welche die drei Rotationen durchgeführt werden müssen, um damit die gewünschte Rotation R zu beschreiben, werden als die sogenannten EULER-Winkel bezeichnet. Rotationen um die x-Achse um einen Winkel Φ bzw. Rotationen um die z-Achse um einen Winkel φ werden durch folgende Matrizen beschrieben:

$$R_x(\Phi) \;=\; \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi & \cos\Phi \end{pmatrix}$$

$$R_z(\varphi) \;=\; \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Somit erhalten wir unter Verwendung der EULER-Winkel die folgende Parametrisierung von $SO(3)$:[12]

$$\begin{aligned} P: \ [0,2\pi) \times [0,\pi] \times [0,2\pi) &\longrightarrow SO(3) \\ (\varphi_1, \Phi, \varphi_2) &\longmapsto R_z(\varphi_1)\ R_x(\Phi)\ R_z(\varphi_2) \end{aligned} \tag{5-3}$$

Durch analoge Vorgehensweise bei der Bestimmung des HAAR-Maßes wie in Kapitel 5.1.1 erhalten wir erneut mit Hilfe des Transformationssatzes und der GRAMschen Determinante

$$\int\limits_{SO(3)} d\lambda \ = \ \int\limits_0^{2\pi} \int\limits_0^{\pi} \int\limits_0^{2\pi} \sqrt{\mathrm{Gram}(\nabla_{\varphi_1} P, \nabla_{\Phi} P, \nabla_{\varphi_2} P)}\ d\varphi_1\ d\Phi\ d\varphi_2 \ , \tag{5-4}$$

wobei $\nabla_{\varphi_1} P(\varphi_1, \Phi, \varphi_2) = \partial_{\varphi_1} P(\varphi_1, \Phi, \varphi_2)$, $\nabla_{\Phi} P(\varphi_1, \Phi, \varphi_2) = \partial_{\Phi} P(\varphi_1, \Phi, \varphi_2)$ und $\nabla_{\varphi_2} P(\varphi_1, \Phi, \varphi_2) = \partial_{\varphi_2} P(\varphi_1, \Phi, \varphi_2)$.

Führen wir die konkrete Berechnung der GRAMschen Determinante nun für diesen Fall durch, so erhalten wir schließlich

$$\mathrm{Gram}(\nabla_{\varphi_1} P, \nabla_{\Phi} P, \nabla_{\varphi_2} P) \ = \ 8\ \sin^2 \Phi \ . \tag{5-5}$$

Für das HAAR-Maß gilt mit einem Faktor $c > 0$ erneut $dQ = c\ d\lambda$. Damit und mit Hilfe von (5-4) und (5-5) erhalten wir, bei erneuter Forderung der Normierungsbedingung $\int_{SO(3)} dQ = 1$, aus

$$\begin{aligned} \int\limits_{SO(3)} dQ \ &= \ \int\limits_0^{2\pi} \int\limits_0^{\pi} \int\limits_0^{2\pi} c\ 2\sqrt{2}\ \sin\Phi\ d\varphi_1\ d\Phi\ d\varphi_2 \\ &= \ 8\sqrt{2}\ \pi^2\ c\ \int\limits_0^{\pi} \sin\Phi\ d\Phi \ = \ 8\sqrt{2}\ \pi^2\ c\ \big[-\cos\Phi \big]_0^{\pi} \\ &= \ 16\sqrt{2}\ \pi^2\ c \ \stackrel{!}{=} \ 1 \end{aligned}$$

die Bedingung $c = (16\sqrt{2}\ \pi^2)^{-1}$, und somit schließlich

$$dQ \ = \ \frac{1}{8\pi^2}\ \sin\Phi\ d\varphi_1\ d\Phi\ d\varphi_2 \ . \tag{5-6}$$

5.2 Adaptiver Algorithmus zur Approximation von Mehrfachintegralen

Für die numerische Approximation der sowohl bei den Euler-Winkeln als auch bei der Achse-Winkel-Parametrisierung auftretenden Mehrfachintegrale der Form

$$I[f] \;=\; \int_{a_1}^{b_1}\int_{a_2}^{b_2}\cdots\int_{a_d}^{b_d} f(x)\;dx$$

für Funktionen $f : \mathbb{R}^d \longrightarrow \mathbb{R}^m$ verwenden wir einen adaptiven Algorithmus namens DCUHRE, dessen Vorgehensweise im Folgenden etwas beleuchtet wird. Ausführliche Details finden sich in den speziellen Arbeiten zum Algorithmus von Berntsen, Espelid und Genz.[3,4] Der Algorithmus ist nur für achsenparallele Quader als Integrationsgebiet ausgelegt. Die Idee des Algorithmus zur Integralapproximation beruht darauf, das komplette quaderförmige Integrationsgebiet solange in kleinere, quaderförmige Subregionen zu unterteilen, bis die Anwendung einer Quadraturformel auf jeder einzelnen Subregion eine gewisse Fehlervorgabe erfüllt, und somit einen im vorgegebenen Fehlerrahmen akzeptablen Gesamtwert des Integrals liefert. Durch diese Adaptivität kann das Integrationsgebiet an unterschiedlichen Stellen unterschiedlich fein unterteilt werden, was es ermöglicht, auf lokale Besonderheiten des Integranden einzugehen. Durch die Vorgabe einer Fehlerschranke für den Integralwert über einer Subregion, sowie der indirekten Vorgabe einer Maximalanzahl an Subregionen, wird das Integrationsgebiet bezüglich der Fehlervorgabe also immer nur so grob wie möglich, aber auch so fein wie notwendig unterteilt. Im Folgenden werden die einzelnen Schritte des Algorithmus (jeweils als separate Matlab-Subroutine) detaillierter beschrieben und deren Reihenfolge im Algorithmus in einem grafischen Überblick dargestellt:

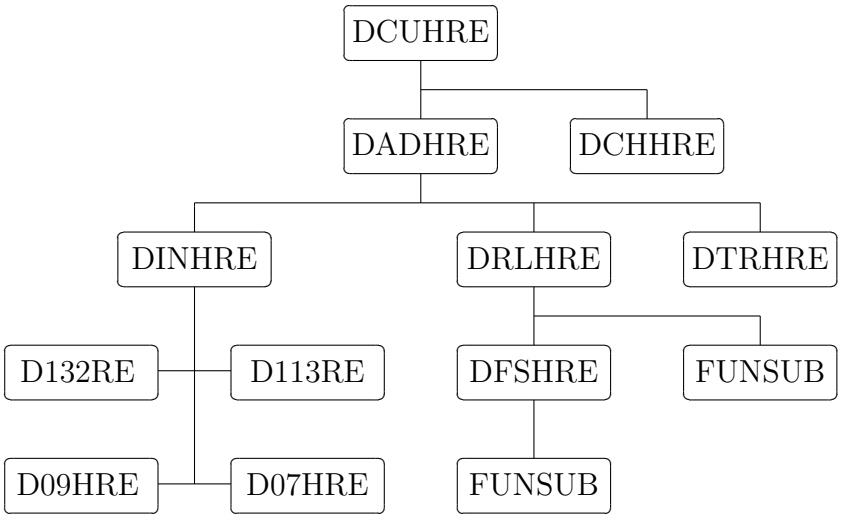


Abbildung 5.1: Überblick der Subroutinen der Integrationsroutine DCUHRE[4]

DCUHRE:
Kopfdatei des Algorithmus, an welche folgende Parameter übergeben werden:

NDIM: Anzahl der Variablen der zu integrierenden Funktion
NUMFUN: Anzahl der Komponenten des vektorwertigen Integrands
A, B: untere bzw. obere Integrationsgrenzen
MINPTS: minimale Anzahl an Auswertungen des Integranden
MAXPTS: maximale Anzahl an Auswertungen des Integranden
FUNSUB: Subroutine, die den Integranden an gegebenem Punkt auswertet
EPSABS, EPSREL: vorgegebener absoluter bzw. relativer Fehler
KEY: wählt die zu verwendende Quadraturformel aus

Als Ausgabeparameter des Algorithmus kann man je nach Wunsch erhalten:

RESULT: vektorwertige Approximation des Integrals
ABSERR: Abschätzung der absoluten Fehler der einzelnen Komponenten des vektorwertigen Integrals
NEVAL: Anzahl an durchgeführten Auswertungen des Integranden
IFAIL: Fehlermeldung (IFAIL = 0 für fehlerfreien Durchlauf)

sowie sämtliche Informationen zu den verwendeten Subregionen (Breite, Tiefe, Höhe, Zentrum) und den jeweils dazugehörenden Abszissen und Gewichten der verwendeten Quadraturformel

DCHHRE:
Überprüft die Konsistenz der Eingabeparameter von DCUHRE.

DADHRE:
Berechnet die Integrale über den gegebenen quaderförmigen Subregionen und unterteilt diese dabei solange in weitere quaderförmige Subregionen, bis der vorgegebene Fehler eingehalten werden kann oder die maximale Anzahl an Subregionen (wird durch MAXPTS ermittelt) erreicht wurde.

DINHRE:
Wählt die entsprechende Quadraturformel aufgrund des Wertes von KEY aus, und berechnet die relativen Abszissen innerhalb eines Quaders und die Gewichte der gewählten Quadraturformel. Desweiteren werden vier Quadraturformeln, sogenannte Nullregeln, für die Fehlerabschätzung bestimmt.

D132RE:
vollsymmetrische Quadraturformel vom Exaktheitsgrad 13 für NDIM = 2 mit 65 Auswertepunkten pro Subregion

D113RE:
vollsymmetrische Quadraturformel vom Exaktheitsgrad 11 für NDIM = 3 mit 127 Auswertepunkten pro Subregion

D09HRE:
vollsymmetrische Quadraturformel vom Exaktheitsgrad 9 für NDIM ≥ 2 mit von der Dimension NDIM abhängend vielen Auswertepunkten pro Subregion

D07HRE:
vollsymmetrische Quadraturformel vom Exaktheitsgrad 7 für NDIM ≥ 2 mit von der Dimension NDIM abhängend vielen Auswertepunkten pro Subregion

DRLHRE:
Berechnet eine Approximation des Integralwerts und den Fehler über jeder Subregion. Desweiteren wird in allen Subregionen, die aufgrund des geschätzten Fehlers weiter unterteilt werden müssen, diejenige Koordinatenachse bestimmt, entlang welcher, aufgrund der größten vierten Differenz des Integranden, die jeweilige Subregion weiter unterteilt wird.

DFSHRE:
Berechnet jeweils die vollsymmetrischen Summen der verwendeten Quadraturformel für einen Vektor an Integrandwerten über den Subregionen.

DTRHRE:
Verwaltet die Menge der Subregionen und ordnet sie nach jedem weiteren Unterteilungsvorgang erneut sinnvoll an.

In den folgenden drei Abschnitten gehen wir noch auf vollsymmetrische Quadraturformeln, das genaue Vorgehen bei der Abschätzung des Integralfehlers, sowie die Unterteilung der Subregionen ein.

5.2.1 Vollsymmetrische Quadraturformeln

Um eine *vollsymmetrische* Quadraturformel definieren zu können, benötigen wir zunächst die Definitionen einer vollsymmetrischen Menge und einer vollsymmetrischen Gewichtsfunktion:[15]

5.2 Definition. Eine Menge $\Omega \subseteq \mathbb{R}^d$ heißt genau dann **vollsymmetrisch**, wenn für alle $x \in \Omega$ automatisch $(\pm x_{\pi(1)}, \pm x_{\pi(2)}, \ldots, \pm x_{\pi(d)}) \in \Omega$ gilt, für alle Permutationen π der Menge $\{1, 2, \ldots, d\}$. Die Menge der permutierten Punkte zu einem festen $\mathbf{x} \in \Omega$ wird mit $\mathbf{x}_{\text{vs}}$ bezeichnet, $\mathbf{x}$ selbst heißt **Generator** von $\mathbf{x}_{\text{vs}}$.

Die bekanntesten vollsymmetrischen Mengen sind der Hyperkubus $[-a, a]^d$, die Hypersphäre und der gesamte Raum.

5.3 Definition. Sei $\Omega \subseteq \mathbb{R}^d$ vollsymmetrisch, so heißt eine Funktion $g : \Omega \longrightarrow \mathbb{R}$ genau dann **vollsymmetrisch**, wenn für alle $x \in \Omega$ und alle Permutationen π der Menge $\{1, 2, \ldots, d\}$ folgende Bedingung gilt:

$$g(x_1, x_2, \ldots, x_d) \;=\; g(\pm x_{\pi(1)}, \pm x_{\pi(2)}, \ldots, \pm x_{\pi(d)})$$

Damit können wir nun ein vollsymmetrische Quadraturformel definieren:[15]

5.4 Definition. Eine Quadraturformel über einer vollsymmetrischen Menge $\Omega \subseteq \mathbb{R}^d$ heißt **vollsymmetrisch**, wenn sie von der Form

$$\int_\Omega g(x)f(x)\,dx \;\approx\; \sum_{i=1}^{n} w_i \sum_{y\in \mathbf{x}_i\mathrm{vs}} f(y) \tag{5-7}$$

mit einer vollsymmetrischen Gewichtsfunktion g ist. Dabei ist die Quadraturformel durch n Gewichte w_i und n Generatoren $\mathbf{x}_i$ festgelegt.

Für die Vorgehensweise zur konkreten Bestimmung einer solchen Quadraturformel, d.h. zur Berechnung der entsprechenden Gewichte und Generatoren, sei auf das Buch von Davis und Rabinowitz[15], sowie auf die Arbeit von Berntsen, Espelid und Genz[3] verwiesen. In Letzterer wird dabei zunächst von $\Omega = [-1,1]^d$ als Integrationsgebiet ausgegangen. Die in jeder der quaderförmigen Subregionen zu verwendenden Auswertepunkte eines beliebigen quaderförmigen Integrationsgebietes werden im vorliegenden Algorithmus dann durch entsprechende Skalierung erzeugt.

5.2.2 Fehlerschätzer[1]

Alle vom Algorithmus verwendeten Quadraturformeln sind ausschließlich vollsymmetrisch, wobei für jede Angabe des Parameters KEY jeweils ein Set von fünf solchen vollsymmetrischen Quadraturformeln ausgewählt wird. In jedem dieser Sets gibt es eine Quadraturformel R vom Exaktheitsgrad $2m+1$ für welche (5-7) mit $g \equiv 1$ wie folgt umformuliert gilt:

$$I[f] \;=\; \int_\Omega f(x)\,dx \;\approx\; R[f] \;=\; \sum_{j=1}^{L} w_j f(x_j)$$

Hierbei sind sowohl die Auswertepunkte $x_j \in \mathbb{R}^d$, als auch die entsprechenden Gewichte $w_j \in \mathbb{R}$ mit $j = 1,\ldots,L$ fortlaufend durchnummeriert. Die restlichen vier vollsymmetrischen Quadraturformeln N_1, N_2, N_3 und N_4 des ausgewählten Sets sind sogenannte *Nullregeln*. Dazu betrachten wir zunächst die folgende Definition:[41]

5.5 Definition. Eine numerische Quadraturformel der Form

$$\int_\Omega f(x)\,dx \;\approx\; \sum_{j} w_j f(x_j)$$

[1] Dieser Abschnitt ist in Anlehnung an die Arbeit von Berntsen, Espelid und Genz[3] formuliert.

wird genau dann als eine **Nullregel** bezeichnet, wenn $\sum_j w_j = 0$ gilt und mindestens ein Gewicht w_j dabei von Null verschieden ist. Man sagt, eine Nullregel ist vom **Grad** k, wenn sie alle Monome vom Grad $\leq k$ zu Null integriert, dies jedoch für mindestens ein Monom vom Grad $k+1$ nicht gelingt.

Die vier Nullregeln N_1, N_2, N_3 und N_4 sind dabei, unter Verwendung derselben Auswertepunkte wie bei der Quadraturformel R, von folgender Struktur:

$$N_i[f] \;=\; \sum_{j=1}^{L} w_j^{(i)} f(x_j) \quad , \quad i = 1,2,3,4$$

Die Nullregeln werden dabei so gewählt, dass sie der Reihe nach vom Grad $2m-1$, $2m-1$, $2m-3$ und $2m-5$ sind. Im Algorithmus wird die Quadraturformel R schließlich zur Approximation des zu berechnenden Integrals über den einzelnen Subregionen verwendet, wohingegen die Nullregeln dazu verwendet werden, über jeder Subregion jeweils eine Abschätzung des Fehlers

$$E[f] \;:=\; R[f] - I[f]$$

zu erzeugen. Dabei ist zu beachten, dass zum Beispiel für jedes $\mu \neq 0$ die Quadraturformel $\mu N_1[f]$ ebenfalls eine Nullregel und vollsymmetrisch ist. Desweiteren ist auch die Quadraturformel $R[f] + \mu N_1[f]$ vollsymmetrisch, jedoch nur noch vom Exaktheitsgrad $2m-1$. Deshalb kann man jede Nullregel als die Differenz zwischen der Quadraturformel R und einer entsprechenden Quadraturformel niederen Exaktheitsgrades auffassen.

Im Folgenden wird die im Algorithmus verwendete Vorgehensweise zur Bestimmung der Fehler über den zwei Hälften einer zu unterteilenden Subregion schematisch vorgestellt (Erklärungen und Erläuterungen zu dieser Vorgehensweise sind der Arbeit von Berntsen, Espelid und Genz[3] zu entnehmen):

Für jede Hälfte der betrachteten Subregion berechnen wir zunächst

$$N_i^*[f] \;:=\; 2^d \max_{\mu_i} \Big(\frac{1}{\|\mu_i N_i + N_{i+1}\|_1} \, |\mu_i N_i[f] + N_{i+1}[f]| \Big) \quad \text{für } i = 1,2,3 \;,$$

bevor wir den folgenden Test durchführen:

```
if c_1 N_1^*[f] ≤ N_2^*[f] and c_2 N_2^*[f] ≤ N_3^*[f] then
   Ê_1^(j)[f] = c_3 N_1^*[f]
else
   Ê_1^(j)[f] = c_4 max(N_1^*[f], N_2^*[f], N_3^*[f])
end
```

$\hat{E}_1^{(1)}[f]$ und $\hat{E}_1^{(2)}[f]$ bezeichnen dabei lokale Fehlerabschätzungen über den beiden Hälften der zu teilenden Subregion, wobei die Konstanten c_i heuristisch gewählt werden, um eine vernünftige Balance zwischen Zuverlässigkeit und Effizienz des Algorithmus herzustellen. Bezeichnen wir mit $R_2[f]$ die Approximation des Integrals über der gegebenen Subregion und mit $R_1^{(j)}[f]$ für $j = 1, 2$ die Approximationen über den beiden Hälften dieser Subregion, so bekommen wir durch

$$\hat{E}_2[f] \;=\; |R_2[f] - (R_1^{(1)}[f] + R_1^{(2)}[f])|$$

eine zweite Fehlerabschätzung und somit für $j = 1, 2$ schließlich die folgenden finalen Abschätzungen über den beiden Hälften:

$$\hat{E}^{(j)}[f] \;=\; \hat{E}_1^{(j)}[f] \;+\; c_5 \,\frac{\hat{E}_1^{(j)}[f]}{\hat{E}_1^{(1)}[f] + \hat{E}_1^{(2)}[f]}\, \hat{E}_2[f] \;+\; c_6 \hat{E}_2[f]$$

5.2.3 Unterteilung der Subregionen[2]

Die Unterteilung einer Subregion, d.h. die Bestimmung jener Koordinatenachse, entlang welcher die entsprechende Subregion weiter unterteilt werden soll, erfolgt durch Betrachtung der vierten Differenzen des Integranden. Bezeichnen wir mit $u \in \mathbb{R}^d$ das Zentrum der ausgewählten Subregion, deren Abmessungen in den Komponenten des Breitenvektors $v \in \mathbb{R}^d$ enthalten sind, und mit $u(\alpha)_i \in \mathbb{R}^d$ jenen Punkt, welcher für einen positiven Parameter α für $i = 1, \ldots, d$ jeweils durch $u(\alpha)_i := u + \alpha \frac{v_i}{2} e_i$ gegeben ist, wobei $e_1, \ldots, e_d$ die kanonische ONB des $\mathbb{R}^d$ bezeichnet, so erhalten wir für zwei positive Parameter α_1 und α_2 den 4te-Differenzen-Operator entlang der i-ten Koordinatenachse wie folgt:

$$\begin{aligned} D_i f \;:=\; \Big\| \; & f\big(u(\alpha_1)_i\big) \;+\; f\big(u(-\alpha_1)_i\big) \;-\; 2f(u) \\ & -\; \frac{\alpha_1^2}{\alpha_2^2} \Big(f\big(u(\alpha_2)_i\big) \;+\; f\big(u(-\alpha_2)_i\big) \;-\; 2f(u) \Big) \; \Big\|_1 \end{aligned}$$

Schließlich wird die Subregion entlang der k-ten Koordinatenachse, für welche

$$D_k f \;=\; \|Df\|_\infty \tag{5-8}$$

gilt, unterteilt. Da die Werte des Integranden, die zur Berechnung dieses Operators verwendet werden, auch in der Berechnung der verwendeten Quadraturformel benötigt werden, werden diese vierten Differenzen zeitgleich mit der Quadraturformel und der Fehlerabschätzung über einer Subregion berechnet und beanspruchen

[2] Dieser Abschnitt ist in Anlehnung an die Arbeit von Berntsen, Espelid und Genz[3] formuliert.

somit keine extra Rechenzeit. Gibt es mehrere Achsen für die (5-8) gilt, so wählen wir k als jene der Achsen, für die die entsprechende Breite v_k maximal ist.

Im Folgenden wird der Teilungsprozess einer Subregion und die jeweils verwendeten Auswertepunkte an einem Beispiel grafisch veranschaulicht. Dabei ist darauf hinzuweisen, dass die 127 verwendeten Auswertepunkte in der zu teilenden Subregion nach der Teilung, jeweils entsprechend verschoben und skaliert, in jeder der beiden neuen Subregionen wiederzufinden sind.

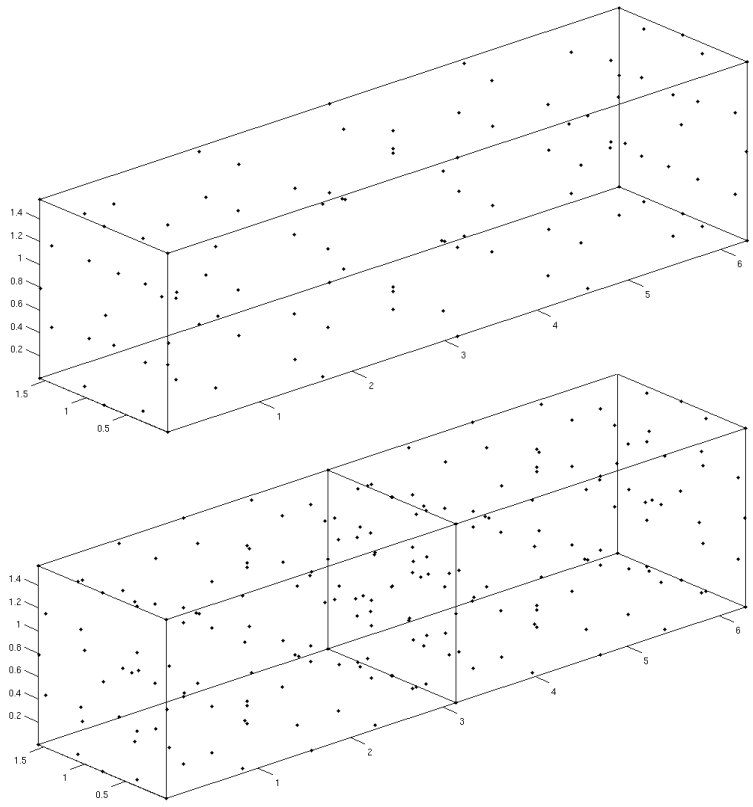


Abbildung 5.2: Teilung einer Subregion

In den folgenden Grafiken wird die Zerteilung des Integrationsgebietes in Abhängigkeit der vorgegebenen Genauigkeitsanforderung EPSABS an die Integralapproximation anhand des kubischen codf Beispiels unter Verwendung von Euler-Winkeln veranschaulicht (lediglich für den Fall $r_k = 4$). In der jeweils linken Grafik ist eine Komponente des mehrdimensionalen Integrands über einem ebenen Schnitt durch die Subregionen des dreidimensionalen, quaderförmigen Integrationsgebietes $[0, 2\pi) \times [0, \frac{\pi}{2}] \times [0, \frac{\pi}{2}]$ dargestellt. Dabei ist zu beachten, dass die Unterteilung in quaderförmige Subregionen von allen Komponenten des Integrands abhängt, d.h. nicht nur von der dargestellten Komponente. Warum sich das Integrationsgebiet in diesem Fall nur über einen Teil der Euler-Winkel erstreckt, wird im folgenden Kapitel

5.3 erläutert. In der jeweiligen rechten Grafik ist die der gewünschten Genauigkeit entsprechende Unterteilung des Integrationsgebietes in die quaderförmigen Subregionen dargestellt. Wenn man bedenkt, dass in diesem Beispiel in jeder einzelnen Subregion jeweils 127 Auswertepunkte des Integranden liegen, so ist es offensichtlich, wie wichtig die Überlegungen aus Kapitel 4 zur effizienten Auswertung des RAYLEIGH-Produktes sind. Die folgenden Grafiken wurden für EPSABS $= 10^{-3}$ bzw. EPSABS $= 10^{-10}$ erstellt:

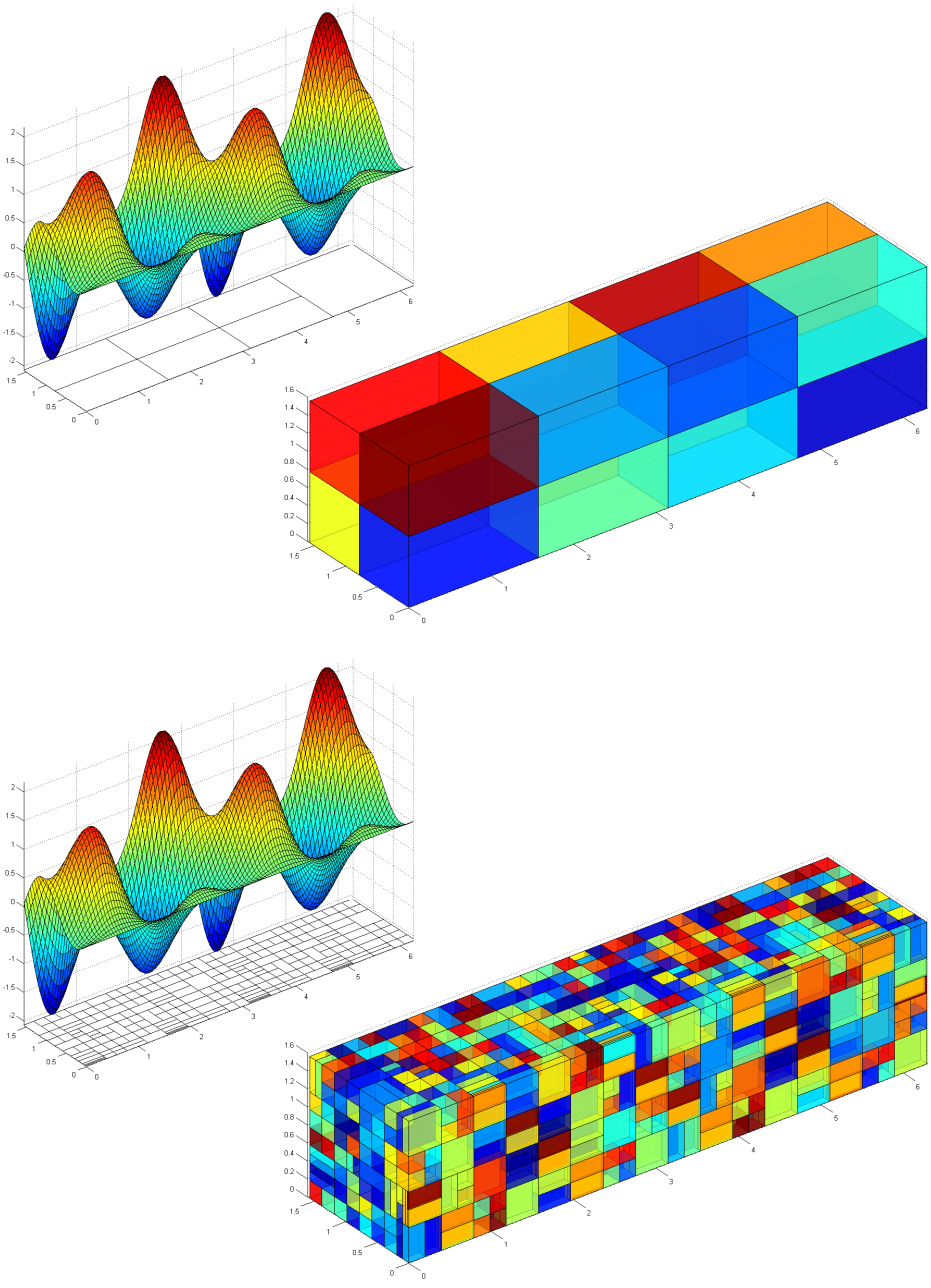


Abbildung 5.3: Teilung des Integrationsgebietes in Subregionen

5.3 Elementarregionen

Wie bereits angekündigt, werden wir in diesem Kapitel die unter anderem bei der codf vorliegende Kristallsymmetrie ausnutzen, um uns damit bei der Integration über $SO(3)$ einen Vorteil zu verschaffen. Dabei werden wir die Integration über $SO(3)$ auf sogenannte *Elementarregionen* einschränken können, was bedeutet, dass es ausreichend sein wird, anstatt über ganz $SO(3)$ über solch eine Elementarregion zu integrieren. Dies hat eine erhebliche Reduktion an Integrandauswertungen zur Folge. Dazu benötigen wir jedoch zunächst folgende Definition:

5.6 Definition. Sei G eine Gruppe und H eine Untergruppe von G. Dann bezeichnen wir eine Menge $E \subset G$ genau dann als **Elementarregion**, wenn folgende Eigenschaften erfüllt sind:

(i) $hE \cap E = \emptyset \quad \forall h \in H \setminus \{\mathbb{1}_H\}$

(ii) $\bigcup\limits_{h \in H} hE = G$

Zum Ausnutzen der Kristallsymmetrie der codf werden wir $G = SO(3)$ und H als die zugehörige Rotations-Symmetriegruppe wählen. Bevor wir uns jedoch dieses konkrete Beispiel etwas genauer anschauen werden, betrachten wir noch folgendes Resultat über Elementarregionen:

5.7 Satz. *Sei G eine Gruppe, H eine Untergruppe von G, $E \subset G$ eine Elementarregion und $\overline{h} \in H$ beliebig gewählt. Dann ist auch $\overline{h}E$ eine Elementarregion.*

Beweis. Sei $\overline{E} := \overline{h}E$. Dann erfüllt $\overline{E}$ die Eigenschaften (i) und (ii) aus Definition 5.6, denn es gilt:

(i) Sei $h \in H \setminus \{\mathbb{1}_H\}$ beliebig gewählt und damit $g := \overline{h}^{-1} h \overline{h} \in H \setminus \{\mathbb{1}_H\}$, so gilt $h\overline{h} = \overline{h}g$ und weiter $h\overline{E} = h\overline{h}E = \overline{h}gE$. Somit folgt schließlich

$$h\overline{E} \cap \overline{E} = \overline{h}gE \cap \overline{h}E = \overline{h}\,(\underbrace{gE \cap E}_{=\,\emptyset}) = \emptyset\,.$$

(ii)

$$\bigcup_{h \in H} h\overline{E} = \bigcup_{h \in H\overline{h}^{-1}} h\overline{E} = \bigcup_{g \in H} g\overline{h}^{-1}\overline{E} = \bigcup_{g \in H} gE = G$$

□

Betrachten wir nun $G = SO(3)$ und wählen als Untergruppe H die Rotations-Symmetriegruppe zur kubischen Kristallsymmetrie, d.h. es ist $|H| = 24$ (siehe Tabelle 4.1), so erhalten wir ausgehend von einer Elementarregion E mit Hilfe von

Definition 5.6 und Satz 5.7 eine disjunkte Zerlegung von $SO(3)$ in 24 Elementarregionen der Form RE für $R \in H$. Dies kann man sich bei der Integration über $SO(3)$ zu Nutze machen, wenn die zu integrierende Funktion f die entsprechende Kristallsymmetrie aufweist, d.h. wenn wie bei der codf für alle $R \in H$ die Bedingung $f(QR) = f(Q)$ gilt für alle $Q \in SO(3)$. Denn in diesem Fall erhalten wir

$$\int_{SO(3)} f(Q)\, dQ \;=\; 24 \int_{E} f(Q)\, dQ \;.$$

Dies zeigt nun, wie bereits erwähnt, dass es ausreichend ist, das Integral lediglich über einer Elementarregion zu berechnen und den Wert des Integrals anschließend mit 24 zu multiplizieren. Im Vergleich zur Integration über ganz $SO(3)$ erspart man sich aufgrund des kleineren Integrationsgebietes dadurch zunächst einmal einige Auswertungen der zu integrierenden Funktion, in unserem Falle der codf. Im Hinblick auf das zur Auswertung der codf zu berechnende RAYLEIGH-Produkt ist dies von großem Vorteil.

Bleibt zu klären, wie solche Elementarregionen von $SO(3)$ aussehen. Zur grafischen Darstellung einer Elementarregion muss man $SO(3)$ zunächst parametrisieren. Demnach sind die geometrischen Formen der Elementarregionen außer von der entsprechenden Kristallsymmetrie auch von der verwendeten Parametrisierung von $SO(3)$ abhängig. Im Folgenden betrachten wir ausschließlich die **kubische** Kristallsymmetrie und parametrisieren $SO(3)$ mit den **Euler-Winkeln**. Die Elementarregionen aus $SO(3)$ übertragen sich demnach auf den Quader $[0, 2\pi) \times [0, \pi] \times [0, 2\pi)$ der EULER-Winkel, in welchem sie nun grafisch darstellbar sind. In einer ersten groben Zerlegung von $SO(3)$ erhalten wir folgende acht, der Form nach identische Teilregionen:[7,21]

$$\varphi_1 \in [0, 2\pi) \;,\; \Phi \in [0, \pi] \;,\; \varphi_2 \in [0, 2\pi)$$

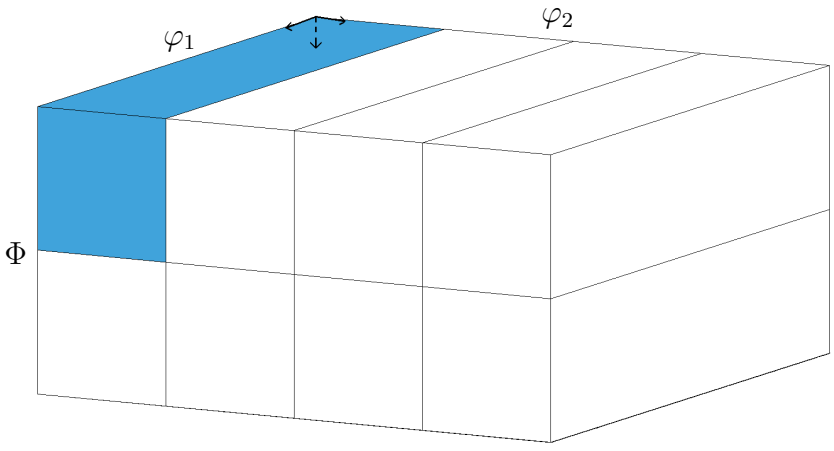


Abbildung 5.4: Grobe Zerlegung der EULER-Winkel in kubische Teilregionen

Jede dieser acht Teilregionen lässt sich in drei Elementarregionen zerlegen. Im Falle der blau eingefärbten Teilregion $[0, 2\pi) \times [0, \frac{\pi}{2}) \times [0, \frac{\pi}{2})$ sehen diese Elementarregionen wie folgt aus:[7,21]

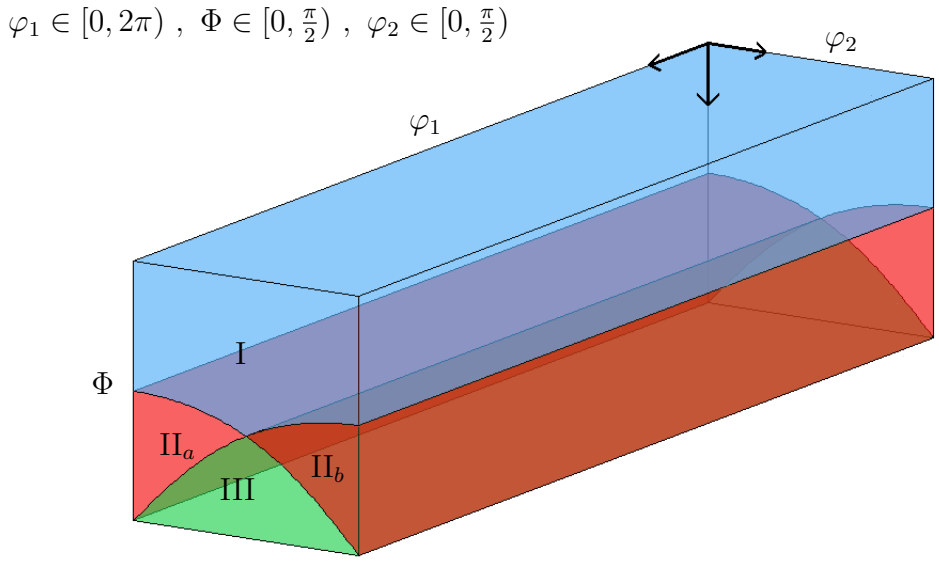


Abbildung 5.5: Zerlegung der Euler-Winkel in kubische Elementarregionen

Dabei setzt sich die Elementarregion II aus den beiden Teilregionen II_a und II_b zusammen. Die Kurve, welche den krummlinigen Teil des Randes der Teilregion III+II_b bzw. der Teilregion II_a+III beschreibt, ist gegeben durch[7,21]

$$\cos\Phi \;=\; \frac{\sin\varphi_2}{\sqrt{1+\sin^2\varphi_2}} \quad \text{bzw.} \quad \cos\Phi \;=\; \frac{\cos\varphi_2}{\sqrt{1+\cos^2\varphi_2}}\,. \tag{5-9}$$

Die Geometrien der drei Elementarregionen in einer der anderen sieben Teilregionen aus Abbildung 5.4 sind im Allgemeinen verschieden zu denen aus Abbildung 5.5. Satz 5.7 besagt zwar, dass wir, ausgehend von einer Elementarregion E, für alle $R \in H$ mit RE erneut eine Elementarregion erhalten, dies bedeutet jedoch nicht, dass eine solche Elementarregion RE gerade der im Raum der Euler-Winkel gedrehten und verschobenen Elementarregion E entsprechen muss. Denn hinter einer solchen Elementarregion RE steckt ein etwas komplizierterer Sachverhalt: Zur Bestimmung der Elementarregion RE zu einem gegebenen $R \in H$ bestimmt man für jedes Euler-Winkel-Tripel aus E die zugehörige Matrix $Q \in SO(3)$ gemäß (5-3), multipliziert diese mit R und bestimmt zu den so erhaltenen Matrizen RQ die zugehörigen Euler-Winkel-Tripel, welche schließlich die Geometrie der Elementarregion RE bestimmen.

Möchten wir nun die Integration über $SO(3)$ auf das Integral über einer Elementarregion beschränken und dazu den vorgestellten Algorithmus verwenden, so muss die verwendete Elementarregion erneut über einem Quader parametrisiert werden. Da wir nach der Parametrisierung beim Integrieren den entsprechenden JACOBI-Faktor mitberücksichtigen müssen, d.h. bei jeder Integrandauswertung mit auswerten müssen, ist die grün eingefärbte Elementarregion III den Elementarregionen I und II vorzuziehen (Elementarregion I ist von größerem Volumen als Elementarregion III und bedarf deshalb mehr Auswertepunkte, Elementarregion II ist am aufwändigsten zu parametrisieren). Mit Hilfe von (5-9) lässt sich Elementarregion III wie folgt beschreiben:

$$\varphi_1 \in [0, 2\pi) \ , \ \Phi \in [\Phi_l, \tfrac{\pi}{2}) \ , \ \varphi_2 \in [0, \tfrac{\pi}{2})$$

$$\text{mit} \quad \Phi_l \ = \ \arccos\left(\min\left[\frac{\sin\varphi_2}{\sqrt{1+\sin^2\varphi_2}} \ , \ \frac{\cos\varphi_2}{\sqrt{1+\cos^2\varphi_2}} \right] \right)$$

Aufgrund der folgenden Äquivalenzen über den entsprechenden Winkelbereichen (jeweils unter Beibehaltung der ursprünglichen Vorzeichen beim Wurzelziehen)

$$\begin{aligned} \cos\Phi \ = \ \frac{\sin\varphi_2}{\sqrt{1+\sin^2\varphi_2}} \quad &\Leftrightarrow \quad (1+\sin^2\varphi_2)\cos^2\Phi \ = \ \sin^2\varphi_2 \\ &\Leftrightarrow \quad \cos^2\Phi \ = \ (1-\cos^2\Phi)\sin^2\varphi_2 \\ &\Leftrightarrow \quad \varphi_2 \ = \ \arcsin\left(\frac{\cos\Phi}{\sqrt{1-\cos^2\Phi}}\right) \\ \text{bzw.} \quad \cos\Phi \ = \ \frac{\cos\varphi_2}{\sqrt{1+\cos^2\varphi_2}} \quad &\Leftrightarrow \quad \varphi_2 \ = \ \arccos\left(\frac{\cos\Phi}{\sqrt{1-\cos^2\Phi}}\right) \end{aligned}$$

erhalten wir folgende Parametrisierung P_3 der Elementarregion III:

$\varphi_1 \in [0, 2\pi) \ , \ \Phi \in [\Phi_l, \tfrac{\pi}{2}) \ , \ \varphi_2 \in [0, \tfrac{\pi}{2})$

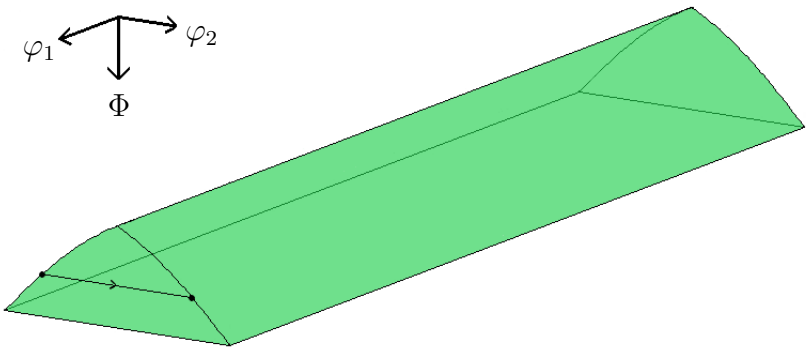


Abbildung 5.6: Parametrisierung der Elementarregion III

$$P_3 : [0, 2\pi) \times [\arccos(\tfrac{1}{3}\sqrt{3}), \tfrac{\pi}{2}) \times [0, 1] \;\longrightarrow\; \mathbb{R}^3$$

$$(\varphi_1, \Phi, t) \;\longmapsto\; \begin{pmatrix} \varphi_1 \\ \Phi \\ (1-t)\arcsin\left(\frac{\cos\Phi}{\sqrt{1-\cos^2\Phi}}\right) \;+\; t\arccos\left(\frac{\cos\Phi}{\sqrt{1-\cos^2\Phi}}\right) \end{pmatrix}$$

Berechnen wir die zugehörige JACOBI-Matrix ∂P_3, so erhalten wir mit derer Determinante den folgenden, bei der Integration über Elementarregion III zu berücksichtigenden JACOBI-Faktor:

$$|\det \partial P_3| \;=\; \arccos\left(\frac{\cos\Phi}{\sqrt{1-\cos^2\Phi}}\right) \;-\; \arcsin\left(\frac{\cos\Phi}{\sqrt{1-\cos^2\Phi}}\right)$$

Somit erhalten wir für das Integral einer Funktion $f : SO(3) \longrightarrow \mathbb{R}^m$, welche die kubische Kristallsymmetrie aufweist, unter zusätzlicher Verwendung der allgemeinen Parametrisierung P aus (5-3) in EULER-Winkeln, dem HAAR-Maß (5-6) und der Integrationsgrenze $a := \arccos(\frac{1}{3}\sqrt{3})$, folgenden Zusammenhang:

$$\begin{aligned} \int\limits_{SO(3)} f(Q)\, dQ \;&=\; \int\limits_0^{2\pi}\int\limits_0^{\pi}\int\limits_0^{2\pi} \underbrace{f\big(P(\varphi_1,\Phi,\varphi_2)\big)\,\frac{\sin\Phi}{8\pi^2}}_{=:\; g(\varphi_1,\Phi,\varphi_2)}\, d\varphi_1\, d\Phi\, d\varphi_2 \\ &=\; 24 \int\limits_{\mathrm{III}} g(\varphi_1,\Phi,\varphi_2)\, d\varphi_1\, d\Phi\, d\varphi_2 \\ &=\; 24 \int\limits_0^1\int\limits_a^{\frac{\pi}{2}}\int\limits_0^{2\pi} g\big(P_3(\varphi_1,\Phi,t)\big)\,|\det\partial P_3|\, d\varphi_1\, d\Phi\, dt \end{aligned}$$

Dabei ist jedoch noch das Folgende zu beachten: Je nachdem wie aufwändig es ist, einen Funktionswert von f zu berechnen, kann es vorkommen, dass es zeitlich schneller geht, das Integral von f nur auf das größere Integrationsgebiet des kompletten Quaders an EULER-Winkeln aus Abbildung 5.5 zu reduzieren (d.h. man integriert über die Gesamtheit der drei Elementarregionen I+II+III und multipliziert das Integral anschließend mit dem Faktor 8). Denn bei besonders einfach auszuwertenden Integranden benötigt der vorgestellte Integrationsalgorithmus weniger Auswertepunkte als bei komplizierteren Integranden, um die gewünschte Genauigkeit zu

erzielen. Reduziert man in solch einem Fall das Integrationsgebiet jedoch bis auf die Elementarregion III, so wird der tatsächliche Integrand, aufgrund des zusätzlichen JACOBI-Faktors, wieder aufwändiger zum Auswerten, d.h. der Integrationsalgorithmus benötigt zum Erreichen derselben Genauigkeitsvorgabe eine feinere Zerlegung in Subregionen, und somit trotz kleinerem Integrationsgebiet weitaus mehr Auswertepunkte. Die Entscheidung, ob man quasi über $1/8$ oder über $1/24$ von $SO(3)$ integriert, ist demnach in Abhängigkeit des Integranden zu treffen.

6 Numerische Ergebnisse des Maximum Entropie Momenten-problems am Beispiel der codf

In diesem Kapitel betrachten wir die numerischen Ergebnisse des Maximum Entropie Momentenproblems (4-7), (4-8), (4-9) der codf, für dessen Lösung man zur Bestimmung der erforderlichen LAGRANGE-Multiplikatoren die Funktion (4-10)

$$\Phi(\omega) = \int\limits_{SO(3)} \exp\Big(-1 + \omega_0 + \sum_{k=1}^{n} \langle \omega_k, Q * \mathbb{T}^{r_k} \rangle \Big)\, dQ \;-\; \omega_0 \;-\; \sum_{k=1}^{n} \frac{\langle \omega_k, \mathbb{S}^{r_k} \rangle}{2r_k + 1} \quad (6\text{-}1)$$

minimieren muss. Dabei gilt $\omega = [\omega_0; \omega_1; \ldots; \omega_n]$ mit $\omega_0 \in \mathbb{R}$ und $\omega_k \in \mathcal{J}_{r_k}(\mathbb{R}^3)$ für $k = 1, \ldots, n$. Bei den Tensoren $\mathbb{T}^{r_k} \in \mathcal{J}_{r_k}(\mathbb{R}^3)$ (siehe Kapitel 4.2) gehen wir erneut von **kubischer** Kristallsymmetrie aus, wobei wir die Tensoren $\mathbb{S}^{r_k} \in \mathcal{J}_{r_k}(\mathbb{R}^3)$, basierend auf einem Datensatz von 1000 im realen Experiment detektierten Kristallorientierungen einer kubischen Kristallprobe, gemäß (4-5) bestimmen. In kubischer Kristallsymmetrie sind die ersten zu berücksichtigenden Tensorränge durch

$$r_k \in \{4, 6, 8, 9, 10, 12_1, 12_2, 13, 14, 15, 16_1, 16_2, 17, 18_1, 18_2, 19, 20_1, 20_2, \ldots\}$$

gegeben. Um die Überlegungen des vorherigen Kapitels anwenden zu können, verwenden wir zur Parametrisierung von $SO(3)$ bei der Integration erneut die EULER-Winkel.

Für die durchzuführende Minimierung der Funktion Φ werden wir verschiedene Optimierungsverfahren heranziehen und miteinander vergleichen. Im Detail werden das die MATLAB-interne Routine `fminunc` für nicht restringierte Optimierungsprobleme, das BFGS-Verfahren und das mehrdimensionale NEWTON-Verfahren sein, auf welche wir in den folgenden Abschnitten kurz eingehen werden.

6.1 Optimierungsverfahren

6.1.1 fminunc

Die Routine `fminunc` ist eine MATLAB-interne Routine zur Bestimmung eines lokalen Minimums einer skalaren Funktion $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ mehrerer Variabler. Durch den

Befehl `X=fminunc(FUN,X0)` wird die Funktion `FUN` ausgehend von einem Startpunkt `X0` entlang iterativ bestimmter Abstiegsrichtungen sukzessive minimiert. Dabei kann `fminunc` situationsbedingt auf verschiedenste Abstiegsverfahren zurückgreifen, und somit ein lokales Minimum von `FUN`, sofern ein solches existiert, bei günstiger Wahl des Startpunktes `X0` bestimmen. Alternativ kann man diese Routine auch über den Befehl `X=fminunc(FUN,X0,OPTIONS)` aufrufen, wobei man hier der Strukturvariable `OPTIONS` verschiedene Parameter für die Ausführung des Minimierungsprozesses mitgeben kann. Diese können unter anderem die Folgenden sein:

- `Display`
  `'off'` Routine zeigt keine Ausgabe an
  `'iter'` bzw. `'iter-detailed'` Routine zeigt Ausgabe nach jeder Iteration an
  `'final'` Routine zeigt eine finale Ausgabe an

- `TolX`
  Abbruchtoleranz für Iterationspunkte `X`

- `TolFun`
  Abbruchtoleranz für Funktionswerte von `FUN`

- `LargeScale` `'on'` bzw. `'off'`
  Die Large-Scale Option legt eine Präferenz fest, welcher Minimierungsalgorithmus von `fminunc` bei Möglichkeit verwendet werden soll. Dies ist jedoch von der Angabe der Parameter abhängig, da die Large-Scale Methode gewisse Parameter, wie zum Beispiel den Gradienten der Zielfunktion `FUN`, benötigt.

- `GradObj` `'on'` bzw. `'off'`
  Routine verwendet benutzerdefinierten Gradienten der Zielfunktion `FUN`, welcher somit in der `FUN`-Routine entsprechend als zweites Argument ausgegeben werden muss. Diese Option ist notwendig für die Large-Scale Methode.

- `DerivativeCheck` `'on'` bzw. `'off'`
  Vergleicht die benutzerdefinierten Ableitungen von `FUN` mit Finite Differenzen Ableitungen und ersetzt diese bei zu großen Abweichungen.

- `Hessian`
  `'on'` Routine verwendet benutzerdefinierte Hesse-Matrix der Zielfunktion `FUN`, welche somit in der `FUN`-Routine entsprechend als drittes Argument ausgegeben werden muss.
  `'off'` Hesse-Matrix wird durch Finite Differenzen approximiert.

- `HessUpdate`
  Methode für das Update der Hesse-Matrix als Alternative zu derer exakten Berechnung. Dabei stehen folgende Methoden zur Auswahl:
  `'bfgs'` Methode nach Broyden, Fletcher, Goldfarb und Shanno
  `'dfp'` Methode nach Davidon, Fletcher und Powell
  `'steepdesc'` Gradientenverfahren, d.h. Verfahren des steilsten Abstiegs

Zu weiteren Details sei auf das Benutzerhandbuch der MATLAB - Optimization Toolbox hingewiesen.[1] Bei der Anwendung dieser Routine auf unser Maximum Entropie Momentenproblem verwenden wir schließlich folgende Optionen:

```
OPTIONS = optimset('GradObj','on','DerivativeCheck','off',
                   'HessUpdate','bfgs','LargeScale','off',
                   'Display','iter-detailed',
                   'TolX',EPSOPT,'TolFun',EPSOPT^2);
```

Da wir bei der Verwendung der `fminunc`-Routine jedoch keine Möglichkeit haben, in den Minimierungsalgorithmus so einzugreifen, dass wir das Maximum Entropie Momentenproblem etwas problemspezifischer behandeln können, werden wir zur Bestimmung der numerischen Lösung dieses Problems alternativ noch einen individuellen Minimierungsalgorithmus verwenden. Dieser wird zur Minimierung zwar auch wahlweise entweder auf das BFGS-Verfahren oder das mehrdimensionale NEWTON-Verfahren zurückgreifen, jedoch mit der Möglichkeit, gezielt in den Minimierungsvorgang eingreifen zu können.

6.1.2 Newton-Verfahren

Mit Hilfe des mehrdimensionalen NEWTON-Verfahrens kann man Nullstellen einer skalaren Funktion mehrerer Variabler approximieren. Wir werden das NEWTON-Verfahren dazu verwenden, eine Funktion $f : \mathbb{R}^d \longrightarrow \mathbb{R}$ zu minimieren, d.h. eine lokale Minimalstelle $x^* \in \mathbb{R}^d$ als Nullstelle des Gradienten dieser Funktion zu approximieren, also über die Bedingung $\nabla f(x^*) = 0$. Dies geschieht, ausgehend von einem Startpunkt $x_0 \in \mathbb{R}^d$, auf eine iterative Art und Weise. Dabei bestimmen wir die jeweilige nächste Approximation $x_{k+1} \in \mathbb{R}^d$ der Minimalstelle $x^* \in \mathbb{R}^d$ als Minimalstelle des lokalen, quadratischen Modells $m_k : \mathbb{R}^d \longrightarrow \mathbb{R}$ von f in einer Umgebung von $x_k \in \mathbb{R}^d$ mit

$$m_k(x) \;:=\; f(x_k) \;+\; \nabla f(x_k)^T(x - x_k) \;+\; \frac{1}{2}(x - x_k)^T \underbrace{\nabla^2 f(x_k)}_{=:\, H_f(x_k)}(x - x_k) \;.$$

Ist die dabei auftauchende HESSE-Matrix $H_f(x_k) = \nabla^2 f(x_k)$ von f positiv definit, so ist die Minimalstelle x_{k+1} von m_k die eindeutige Lösung des Problems $\nabla m_k(x) = 0$. Folglich gilt

$$0 \;=\; \nabla f(x_k) \;+\; H_f(x_k)(x_{k+1} - x_k) \;, \tag{6-2}$$

und somit erhalten wir die neue Approximation x_{k+1} durch

$$x_{k+1} \;=\; x_k \;-\; H_f(x_k)^{-1} \nabla f(x_k) \;.$$

Um die Invertierung der HESSE-Matrix zu vermeiden, berechnen wir zunächst die Lösung y des linearen Gleichungssystems $H_f(x_k)y = -\nabla f(x_k)$ und setzen anschließend $x_{k+1} = y + x_k$. Aufgrund der strikten Konvexität der zu minimierenden Funktion Φ des Maximum Entropie Momentenproblems, liegt in unserem Falle der codf Konvergenz des NEWTON-Verfahrens vor. Allgemeine Aussagen zur Konvergenz dieses Verfahrens sind der Literatur zu entnehmen.[18,25]

Da die in jeder Iteration vorkommende Neuberechnung der HESSE-Matrix in höheren Dimensionen extrem aufwändig sein kann - man denke nur an unser Maximum Entropie Momentenproblem der codf, bei welchem für jeden Matrixeintrag eine Integration über $SO(3)$ und die Auswertung des RAYLEIGH-Produktes durchzuführen ist (siehe Anhang A.1) - betrachten wir im Folgenden noch einen Vertreter eines sogenannten *Quasi-Newton-Verfahrens*, bei welchen die jeweilige Neuberechnung der exakten HESSE-Matrix durch eine einfacher zu berechnende Update-Formel ersetzt wird. Ob man mit Hilfe eines solchen Verfahrens bei identischer Genauigkeitsforderung an die Optimierung jedoch zeitlich schneller sein wird, hängt letztendlich auch von der Anzahl der benötigten Iterationen der unterschiedlichen Verfahren ab. Auf diese Vergleiche werden wir später noch genauer eingehen.

6.1.3 BFGS-Verfahren

Ausgehend von der Bedingung (6-2) des NEWTON-Verfahrens lässt sich die Vorgehensweise eines Quasi-NEWTON-Verfahrens einfach erläutern:[18,30] Die HESSE-Matrix $H_f(x_k)$ wird durch eine Approximation $B_k \in \mathbb{R}^{d\times d}$ ersetzt, mit deren Hilfe man durch Lösen des linearen Gleichungssystems $B_k d_k = -\nabla f(x_k)$ eine Abstiegsrichtung $d_k \in \mathbb{R}^d$ zur Berechnung der nächsten Approximation x_{k+1} der gesuchten Minimalstelle von f bestimmt. Diese neue Approximation x_{k+1} erhält man nach geeigneter Wahl einer Schrittweite $h_k > 0$ durch $x_{k+1} := x_k + h_k d_k$ (siehe (6-4)). Bleibt für die darauffolgende Iteration lediglich die Frage zu klären, wie die Approximation $B_{k+1} \in \mathbb{R}^{d\times d}$ der HESSE-Matrix $H_f(x_{k+1})$ zu wählen ist. Je nach Quasi-NEWTON-Verfahren gibt es hierfür unterschiedliche Möglichkeiten. Die Rang-2-Update-Formel von BROYDEN, FLETCHER, GOLDFARB und SHANNO liefert das nach ihnen benannte BFGS-Verfahren:

BFGS-Schritt $(x_k, B_k) \curvearrowright (x_{k+1}, B_{k+1})$:

Löse $B_k d_k = -\nabla f(x_k)$, wähle $h_k > 0$ und setze

$$x_{k+1} := x_k + h_k d_k \,, \quad B_{k+1} := B_k + \frac{q_k q_k^T}{p_k^T q_k} - \frac{B_k p_k p_k^T B_k}{p_k^T B_k p_k} \,, \tag{6-3}$$

wobei $p_k := x_{k+1} - x_k$ und $q_k := \nabla f(x_{k+1}) - \nabla f(x_k)$.

Zum Startpunkt $x_0 \in \mathbb{R}^d$ wählt man am besten entweder $B_0 := \mathbb{1}$ oder $B_0 := H_f(x_0)$. Bei der Wahl der jeweiligen Schrittweite $h_k > 0$ gibt es generell den Unterschied zwischen der sogenannten *'Exact Line Search'* und der sogenannten *'Inexact Line Search'*. Bei der exakten Liniensuche entlang der Abstiegsrichtung d_k wählt man die Schrittweite $h_k > 0$ durch Lösen des Problems

$$f(x_k + h_k d_k) \;=\; \min_{h \geq 0} f(x_k + h d_k) \;.$$

Im Normalfall ist es jedoch zu aufwändig, in jeder Iteration diese exakte Schrittweite zu berechnen. Daher ist eine tolerantere Suche nach einer geeigneten Schrittweite wesentlich sinnvoller. Eine gängige Methode zur Bestimmung einer solchen sinnvollen Schrittweite liefert die inexakte Liniensuche nach Armijo, welche wir beim Maximum Entropie Momentenproblem der codf für $\sigma = 0.001$ und $\alpha = 0.25$ anwenden werden:[18]

Schrittweitenwahl nach Armijo:
Mit einem von k unabhängigen festen $\sigma > 0$ sei ein $h_0^* \geq \sigma \|\nabla f(x_k)\|_2 / \|d_k\|_2$ gewählt. Bestimme dann das kleinste $j \in \mathbb{N}_0$ derart, dass für $h_j^* := h_0^* / 2^j$

$$f(x_k + h_j^* d_k) \;\leq\; f(x_k) \;+\; \alpha h_j^* \nabla f(x_k)^T d_k \qquad (6\text{-}4)$$

erfüllt ist und setze damit $h_k := h_j^*$.

Für einen detaillierteren Hintergrund zu dieser Schrittweitenwahl sei erneut auf die Literatur verwiesen.[18,30] Bei der numerischen Umsetzung dieser Schrittweitenwahl werden wir noch eine weitere Überlegung mit einbauen. So werden wir, um unnötig kleine Schrittweiten zu vermeiden, in jeder Iteration die Schrittweite der vorherigen Iteration als Startschrittweite h_0^* verwenden. Sollte (6-4) für dieses h_0^* auf Anhieb erfüllt sein, so überprüfen wir, ob dies auch für die doppelte Schrittweite $2h_0^*$ der Fall ist. Ist auch dies der Fall, so wählen wir $h_k := 2h_0^*$, andernfalls bleiben wir bei der Wahl $h_k := h_0^*$. Sollte (6-4) hingegen für h_0^* nicht erfüllt sein, so halbieren wir h_0^* sukzessive bis (6-4) erfüllt ist.

6.2 Adaptive vs. nicht-adaptive Integration

Zur Berechnung der auftauchenden Integrale in der zu minimierenden Funktion Φ (6-1), deren Gradienten $\nabla\Phi$ und deren Hesse-Matrix H_Φ verwenden wir im Allgemeinen den in Kapitel 5.2 vorgestellten adaptiven Integrationsalgorithmus. Dabei sollte beim Minimierungsverfahren beachtet werden, dass es hinsichtlich konsistenter Daten von großer Bedeutung ist, dass der Funktionswert, der Gradient und die Hesse-Matrix der Funktion Φ an einer selben Stelle ω immer zusammenpassen. Wäre

dies nicht der Fall, so würden die Abstiegsrichtungen nichts mehr mit der eigentlich zu minimierenden Funktion zu tun haben, die ganze Minimierung hätte keine Aussagekraft mehr. Um also zu gewährleisten, dass Funktionswert, Gradient und HESSE-Matrix in jedem Minimierungsschritt zusammenpassen, muss sichergestellt werden, dass diese immer innerhalb derselben Routine in ein und derselben Integration berechnet werden. Denn würde man den Funktionswert, den Gradienten und die HESSE-Matrix von Φ an der gleichen Stelle ω mit einer jeweils eigenständigen Routine berechnen, so müsste der adaptive Integrationsalgorithmus, um bei allen dieselbe gewünschte Genauigkeit erzielen zu können, im Allgemeinen unterschiedliche Unterteilungen in Subregionen vornehmen, womit man den Funktionswert, den Gradienten und die HESSE-Matrix letztendlich mit unterschiedlichen Quadraturformeln berechnen würde. Benötigt das verwendete Minimierungsverfahren also den Gradienten von Φ, so können wir den zugehörigen Funktionswert von Φ aus der ersten Komponente des Gradienten gewinnen, was sicherstellt, dass Funktionswert und Gradient jeweils zusammenpassen. Verwenden wir ein Verfahren, welches zusätzlich die HESSE-Matrix von Φ benötigt, so können wir den zugehörigen Gradienten von Φ aus der ersten Spalte der HESSE-Matrix gewinnen, aus welcher man aus der ersten Komponente wiederum den zugehörigen Funktionswert gewinnen kann. Somit ist auch in diesem Fall gesichert, dass Funktionswert, Gradient und HESSE-Matrix zusammenpassen. Die Berechnung der HESSE-Matrix kann man aufgrund der Symmetrie der Matrix auf die entsprechenden Komponenten reduzieren, welche man vektoriell zusammengefasst alle gleichzeitig mit dem Integrationsalgorithmus bestimmen kann.

Durch die gleichzeitige Berechnung des Funktionswertes und des Gradienten von Φ kommt es bei der Durchführung der ARMIJO-Schrittweitenwahl beim BFGS-Verfahren zu eigentlich überflüssigen Gradientenberechnungen. Zur Überprüfung der ARMIJO-Bedingung (6-4) wird an den Teststellen $x_k + h_j^* d_k$ nämlich lediglich der Funktionswert, jedoch nicht der Gradient benötigt. Diese überflüssigen Berechnungen müssen wir jedoch in Kauf nehmen, denn würden wir sie weglassen, d.h. die Funktionswerte $f(x_k + h_j^* d_k)$ $\big($also inklusive $f(x_k + h_k d_k) = f(x_{k+1})\big)$ mit dem adaptiven Integrationsalgorithmus in einer separaten Routine berechnen, so würden wir nach der Akzeptanz einer geeigneten Schrittweite ein Problem bekommen. Denn um nach der Akzeptanz einer Schrittweite die nächste BFGS-Iteration durchführen zu können, würden wir nicht nur den Funktionswert $f(x_{k+1})$ benötigen, sondern auch den Gradienten $\nabla f(x_{k+1})$. Spätestens an dieser Stelle würden wir also gezwungen werden, Funktionswert und Gradient erneut mit derselben Routine zu berechnen. Im Allgemeinen würden wir dann aufgrund der unterschiedlichen Unterteilung in Subregionen, einen an der Stelle x_{k+1}, im Vergleich zum bei der Durchführung der ARMIJO-Bedingung (6-4) berechneten Funktionswert, unterschiedlichen Funktionswert erhalten. Dies könnte dann dazu führen, dass die ARMIJO-Bedingung (6-4) mit dem nun neuen Funktionswert an der Stelle x_{k+1} doch nicht mehr erfüllt ist. Die akzeptierte Schrittweite wäre demnach nicht sinnvoll und das BFGS-Verfahren würde keine verlässlichen Ergebnisse liefern.

Eine andere Möglichkeit beim Minimierungsprozess Zeit zu sparen, liefert jedoch die Idee, die adaptive Integration zur Berechnung der Funktionswerte, Gradienten und HESSE-Matrizen (je nach Verfahren) zwischenzeitlich durch eine nicht-adaptive Integration zu ersetzen. Das Einsetzen von adaptiver und nicht-adaptiver Integration mit dem DCUHRE-Algorithmus innerhalb des Minimierungsprozesses hat gegenüber der rein adaptiven Integration den Vorteil des geringeren Zeitaufwands aufgrund deutlich weniger Rechenoperationen. Nach einem adaptiven Schritt werden die Integrationsstützpunkte samt zugehöriger Gewichte gespeichert und mit eben dieser Quadraturformel weitere, nun nicht-adaptive Schritte durchgeführt. Hierbei entfallen dann alle Fehlerabschätzungen und Unterteilungen in Subregionen, was sich zeitlich als Vorteil bemerkbar macht. Der Nachteil daran ist jedoch, dass man in den nicht-adaptiven Schritten nicht das eigentliche Problem mit der in unserem Fall gegebenen Zielfunktion (6-1)

$$\Phi(\omega) = \int\limits_{SO(3)} \exp\Big(-1 + \omega_0 + \sum_{k=1}^{n} \langle \omega_k, Q * \mathbb{T}^{r_k} \rangle \Big)\, dQ \;-\; \omega_0 \;-\; \sum_{k=1}^{n} \frac{\langle \omega_k, \mathbb{S}^{r_k} \rangle}{2r_k + 1}$$

löst, sondern eine Hilfsfunktion ψ minimiert, bei der das in Φ auftauchende Integral durch die entsprechende Quadraturformel aus gespeicherten Stützpunkten und Gewichten eines vorherigen adaptiven Schritts ersetzt wird. Dieses Hilfsproblem wird jedoch nur in den ersten nicht-adaptiven Schritten nach einem adaptiven Schritt geeignet sein, solange die iterierten LAGRANGE-Multiplikatoren nicht zu weit auseinanderliegen. Nach einigen nicht-adaptiven Schritten wird das Hilfsproblem zu ungenau sein, um das eigentliche Problem damit zu beschreiben. In diesem Fall muss ausgehend vom aktuellen Iterationspunkt erneut ein adaptiver Schritt durchgeführt werden, um damit das ursprüngliche Problem wieder besser zu approximieren. Streng genommen liefert die adaptive Integration ebenso lediglich eine Quadraturformel und somit eine Hilfsfunktion für die eigentliche Zielfunktion Φ, jedoch können wir in diesem Fall aufgrund der gestellten Genauigkeitsanforderung EPSABS dafür sorgen, dass sich Hilfsfunktion und Zielfunktion numerisch kaum unterscheiden. Diese Kontrollmöglichkeit des Integrationsfehlers ist bei der nicht-adaptiven Variante jedoch nicht mehr möglich.

Damit man durch diese Vorgehensweise beim Minimierungsprozess von Φ jedoch nicht wild und unnötig im Definitionsgebiet von Φ umherspringt, muss der Wechsel zwischen adaptiver und nicht-adaptiver Integration (bzw. umgekehrt) gezielt und sinnvoll gesteuert werden. Denn löst man nach einem Wechsel zur nicht-adaptiven Integration das zugehörige Hilfsproblem bis zur gewünschten Endgenauigkeit des Ausgangsproblems, so kann es passieren, dass man sich mit dem Hilfsproblem so weit von der eigentlichen Minimalstelle wegbewegt, dass einen der nachfolgende adaptive Schritt wieder weit von der gefundenen Minimalstelle des Hilfsproblems wegführt. Diese Vorgehensweise würde so also keinen Sinn machen. Betrachten wir sowohl beim Ausgangsproblem als auch beim Hilfsproblem die strikt konvexe Zielfunktion in der Nähe des Minimums, so können wir diese in einer Umgebung der

Minimalstelle lokal durch ein quadratisches Modell approximieren. Betrachten wir ein eindimensionales quadratisches Modell, d.h. eine Parabel cx^2, so bewirkt ein Voranschreiten um 10^{-4} in x-Richtung nur noch eine Änderung der Funktionswerte in der Größenordnung 10^{-8}. Dies können wir nun auf das mehrdimensionale quadratische Modell übertragen, um damit eine Heuristik zu erstellen, mit der wir die Anzahl der aufeinanderfolgenden nicht-adaptiven Schritte in Abhängigkeit der Differenz zweier aufeinanderfolgender Iterationspunkte und der zugehörigen Differenz der entsprechenden Zielfunktionswerte steuern können. Somit wird verhindert, dass zu lange nicht-adaptive Integrationen durchgeführt werden. Der letzte Schritt eines Minimierungsprozesses erfolgt jeweils adaptiv.

In jedem adaptiven Schritt des Minimierungsprozesses passen wir zusätzlich die Integrationsgenauigkeit EPSABS an die gewünschte Endgenauigkeit EPSOPT der Minimierung an, d.h. an die gewünschte Genauigkeit, mit welcher die tensoriellen Nebenbedingungen des Maximum Entropie Momentenproblems erfüllt werden sollen. Je weiter man vom Minimum entfernt ist, desto grober darf integriert werden. In diesem Fall bleibt man einige Schritte beim Hilfsproblem. Je mehr man sich der Minimalstelle nähert, desto exakter muss integriert werden, d.h. das ursprüngliche Problem besser approximiert werden. Diese Vorgehensweise kann unter anderem durch Betrachtung des Verlaufs der Differenzen der bei der Iteration aufeinanderfolgenden Funktionswerte gesteuert werden. Auch durch diese Vorgehensweise wird erneut Rechenzeit eingespart.

Da die Erfüllung der tensoriellen Nebenbedingungen vom jeweiligen Gradienten der Zielfunktion abhängig ist, wir also lediglich daran interessiert sind, dass die Momente in erster Ordnung stimmen und somit höhere Ordnungen nicht berücksichtigt werden müssen, können wir die Berechnung der Hesse-Matrix beim Newton-Verfahren selbst im adaptiven Fall grundsätzlich nicht-adaptiv durchführen und somit sehr viel Rechenzeit sparen. Dabei ist lediglich zu beachten, dass wir die Hesse-Matrix jeweils mit derselben Quadraturformel nicht-adaptiv berechnen, mit welcher wir den entsprechenden Gradienten adaptiv berechnet haben. So ist sichergestellt, dass der Gradient und die Hesse-Matrix der Zielfunktion an der zu berechnenden Stelle erneut zusammenpassen.

In den folgenden Tabellen werden nun die von den verwendeten Verfahren benötigte Zeiten, das Maximum Entropie Momentenproblem der codf in kubischer Kristallsymmetrie numerisch zu lösen, aufgeführt. Dabei werden die Verfahren entsprechend mit dem Zusatz "a" bzw. "a+na" gekennzeichnet, wobei "a" dafür steht, dass ausschließlich adaptiv gerechnet wurde, und "a+na" entsprechend den abwechselnden Einsatz von adaptiver und nicht-adaptiver Integration bezeichnet. Aufgeführt werden die verwendeten Ränge r_k, die geforderte Minimierungsgenauigkeit EPSOPT, d.h. wie klein die Norm des Gradienten der Zielfunktion werden soll, die vom Verfahren benötigte Anzahl an Iterationen und die dafür benötigte Zeit. Bei den Verfahren, bei welchen auch auf nicht-adaptive Integration zurückgegriffen wird, steht die bei

der Anzahl an Iterationen aufgeführte eingeklammerte Zahl für die Anzahl an Iterationen, bei denen nicht-adaptiv gerechnet wurde. Die aufgeführten Zeiten sollten erneut lediglich relativ zueinander betrachtet werden, um zu sehen, welches der Verfahren relativ gesehen am wenigsten Zeit benötigt, und wie bzw. ob sich der Einsatz von nicht-adaptiver Integration bemerkbar macht. Bei Verwendung anderer Implementiersprachen sind, absolut betrachtet, erneut deutlich geringere Zeiten zu erwarten. Um die Verfahren zeitlich sinnvoll miteinander vergleichen zu können, wurde der Minimierungsprozess der Zielfunktion Φ jeweils vom selben Startpunkt aus begonnen, in diesem Fall bei $\omega = 0$. Folgende Zeiten haben sich bei der Implementierung in MATLAB ergeben:

Verfahren	Rang	EPSOPT	# Iterationen	Zeit [sec]	Bemerkung
`fminunc`[a]	4	10^{-3}	18	79	Abbruch, da $h_k <$ EPSOPT
	4	10^{-5}	21	244	Abbruch, da entlang d_k nicht weiter minimiert werden konnte
BFGS[a]	4	10^{-3}	20	94	
	4	10^{-6}	23	115	
	4	10^{-7}	29	212	Abbruch, da B_k singulär. $\nabla\Phi \sim 10^{-7}$
BFGS[a+na]	4	10^{-3}	23(17)	44	
	4	10^{-6}	27(20)	55	
NEWTON[a]	4	10^{-3}	4	20	
	4	10^{-6}	5	26	
	4	10^{-14}	6	31	
NEWTON[a+na]	4	10^{-3}	4(2)	16	
	4	10^{-6}	5(2)	22	
	4	10^{-14}	6(2)	27	

Tabelle 6.1: Zeitlicher Vergleich der verschiedenen Minimierungsverfahren I

Wie anhand der Tabellen 6.1 und 6.2 zu erkennen ist, macht sich der Einsatz von nicht-adaptiver Integration im Wechsel mit adaptiver Integration bezüglich der Zeiten positiv bemerkbar. Von den hier getesteten Verfahren liefert das NEWTON-Verfahren unter Verwendung von nicht-adaptiven Integrationen die besten Zeiten. In der folgenden Tabelle 6.2 ist jedoch ebenso gut zu erkennen, dass bei Hinzunahme der Momentenbedingungen zu höheren Rängen die Zeiten schlagartig zunehmen. Dies ist zum einen durch die daraus resultierende Zunahme der Dimension des Maximum Entropie Momentenproblems (siehe (4-7), (4-9) und Tabelle 3.1) und dem damit

verbundenen steigenden Rechenaufwand zu erklären, zum anderen jedoch auch der verwendeten Implementiersprache MATLAB geschuldet. Wie bereits erwähnt wurde, würde sich die Verwendung einer anderen Implementiersprache wie zum Beispiel C++ in den Zeiten deutlich positiver bemerkbar machen. Eine weitere Möglichkeit Zeit zu sparen, ist der Einsatz von Parallelrechnern - der Integrationsalgorithmus DCUHRE ist dafür bereits ausgelegt.

In Tabelle 6.1 ist deutlich zu erkennen, dass die MATLAB-interne Routine `fminunc` bereits bei geringerer Genauigkeitsforderung Probleme bekommt und deutlich mehr Zeit für den Minimierungsprozess benötigt als das BFGS-Verfahren oder das NEWTON-Verfahren. Dies liegt an der programminternen Vorgehensweise, die wir von außen nicht vollständig kontrollieren können. Deshalb ist das Verfahren für unseren Fall nicht geeignet und wird deshalb bei der Hinzunahme weiterer Ränge in der folgenden Tabelle bereits nicht mehr berücksichtigt:

Verfahren	Ränge	EPSOPT	# Iterationen	Zeit
BFGS[a]	$4, 6$	10^{-5}	83	7h 40min
NEWTON[a]	$4, 6$	10^{-5}	11	37min
	$4, 6$	10^{-10}	12	42min
	$4, 6, 8$	10^{-10}	14	2h 23min
NEWTON[a+na]	$4, 6$	10^{-5}	12(8)	24min
	$4, 6$	10^{-10}	13(8)	29min
	$4, 6, 8$	10^{-10}	14(9)	1h 39min
	$4, 6, 8, 9$	10^{-10}	16(10)	3h 47min
	$4, 6, 8, 9, 10$	10^{-10}	20(13)	13h 57min
	$4, 6, 8, 9, 10, 12_1$	10^{-10}	26(16)	60h 49min

Tabelle 6.2: Zeitlicher Vergleich der verschiedenen Minimierungsverfahren II

Vergleicht man das BFGS-Verfahren mit dem entsprechenden NEWTON-Verfahren, so stellt man fest, dass das NEWTON-Verfahren nicht nur in der Lage ist, das Maximum Entropie Momentenproblem bis zu einer schärferen Genauigkeitsvorgabe lösen zu können, sondern auch, dass es deutlich schnellere Rechenzeiten liefert, obwohl ein einzelner Schritt durch die Berechnung der exakten HESSE-Matrix wesentlich zeitaufwändiger ist als ein entsprechender Schritt des BFGS-Verfahrens. Dass das NEWTON-Verfahren trotzdem bessere Zeiten liefert, liegt daran, dass es bei derselben Genauigkeitsvorgabe insgesamt deutlich weniger Iterationen benötigt als das BFGS-Verfahren, wie man anhand der beiden Tabellen erkennen kann. Dies hat mit den grundlegenden Eigenschaften dieser Verfahren zu tun. Denn das NEWTON-Verfahren löst, wie bereits in Kapitel 6.1.2 beschrieben, in jedem Schritt das zugrundeliegende quadratische Modell exakt, und legt somit absolut gesehen sehr viel größere Schritte als das BFGS-Verfahren zurück (zum Beispiel in der Norm des Gradienten von der Größenordnung 10^{-5} auf 10^{-10} auf 10^{-15}). Somit erreicht

das NEWTON-Verfahren die gewünschte Genauigkeit, ohne sich mehrere Iterationen lang in dem kleinen Normbereich der Genauigkeitsvorgabe aufhalten zu müssen. Das BFGS-Verfahren löst durch die Approximation der HESSE-Matrix das quadratische Modell hingegen nicht exakt und legt im Vergleich zum NEWTON-Verfahren deutlich kleinere Schritte zurück. Dabei kann es vorkommen, dass das BFGS-Verfahren in einer Umgebung der Minimalstelle bis zum Erreichen der gewünschten Genauigkeit nur noch Schritte in der Größenordnung von etwa 10^{-8} zurücklegen kann. Dies ist in einem quadratischen Modell etwa gleichbedeutend mit einem lediglichen Abfall der Zielfunktion im Bereich der Rechengenauigkeit von MATLAB. Die Differenz der Funktionswerte zweier aufeinanderfolgender Iterationspunkte kann somit numerisch nicht mehr aufgelöst werden, was dazu führt, dass das Verfahren stagniert. Dies hat zur Folge, dass eine der gemäß des Rang-2-Updates (6-3) aktualisierten Approximationen B_{k+1} der entsprechenden HESSE-Matrizen irgendwann selbst nur noch von Rang 2 ist, und somit im darauffolgenden Schritt keine Abstiegsrichtung durch Lösen des entsprechenden Gleichungssystems mehr bestimmt werden kann. Das BFGS-Verfahren wird in solch einem Fall den Minimierungsprozess demnach automatisch mit der Fehlermeldung, dass die entsprechende Matrix des zu lösenden Gleichungssystems singulär ist, abbrechen. Da sich dieses Stagnieren des BFGS-Verfahrens einerseits bei schärferen Genauigkeitsforderungen und andererseits auch bei Hinzunahme höherer Ränge zeitlich wesentlich stärker bemerkbar macht, wurden in Tabelle 6.2 mit Ausnahme eines BFGS-Vergleichswert nur noch Zeiten der beiden NEWTON-Varianten betrachtet. Ab der Hinzunahme vom Rang $r_k = 9$ wurde, aufgrund von im Vergleich zu hoher Rechenzeiten der anderen Verfahren, nur noch das zeitschnellste NEWTON-Verfahren in der "a+na"-Version verwendet.

In den folgenden Abbildungen wird das Stagnieren des BFGS-Verfahrens, was im Fall von Rang 4 zum ersten Mal bei der Genauigkeitsvorgabe 10^{-7} auftritt, im Vergleich zum NEWTON-Verfahren bei identischen Vorgaben grafisch illustriert. Zu sehen sind jeweils blaue Linienplots der Zielfunktion Φ entlang der jeweiligen Abstiegsrichtung nach einem adaptiv durchgeführten Schritt. Der jeweilige erste Plot wurde nach einem Schritt zu Beginn des Minimierungsprozesses angefertigt, der jeweilige zweite Plot zeigt den jeweiligen letzten Schritt des Minimierungsprozesses, d.h. beim BFGS-Verfahren den letzten Schritt vor dem Abbruch. Dabei wurde der Ausgangspunkt des aktuellen Schritts auf dem Graphen von Φ jeweils blau markiert, die jeweilige neue Approximation wurde rot markiert. Bei den Plots zum NEWTON-Verfahren wurde zusätzlich das zugrundeliegende quadratische Modell (roter Plot) eingezeichnet. Der Funktionswert der neuen Approximation, von dem das NEWTON-Verfahren aufgrund des quadratischen Modells eigentlich ausgeht, ist dem ebenfalls rot markierten Punkt auf dem Graphen des quadratischen Modells zu entnehmen. Daran ist jeweils schön zu erkennen, dass das quadratische Modell beim NEWTON-Verfahren exakt minimiert wird und gegen Ende des Minimierungsprozesses lokal von der Zielfunktion Φ mit bloßem Auge nicht mehr zu unterscheiden ist.

Vergleicht man nun das BFGS-Verfahren im Fall Rang 4 und EPSOPT $= 10^{-7}$ mit

dem NEWTON-Verfahren, so ist die bereits beschriebene Problematik des BFGS-Verfahrens an der im letzten Schritt zurückgelegten Schrittweite $\|\omega^{(\text{end})} - \omega^{(\text{end-1})}\|_2$ gut zu erkennen:

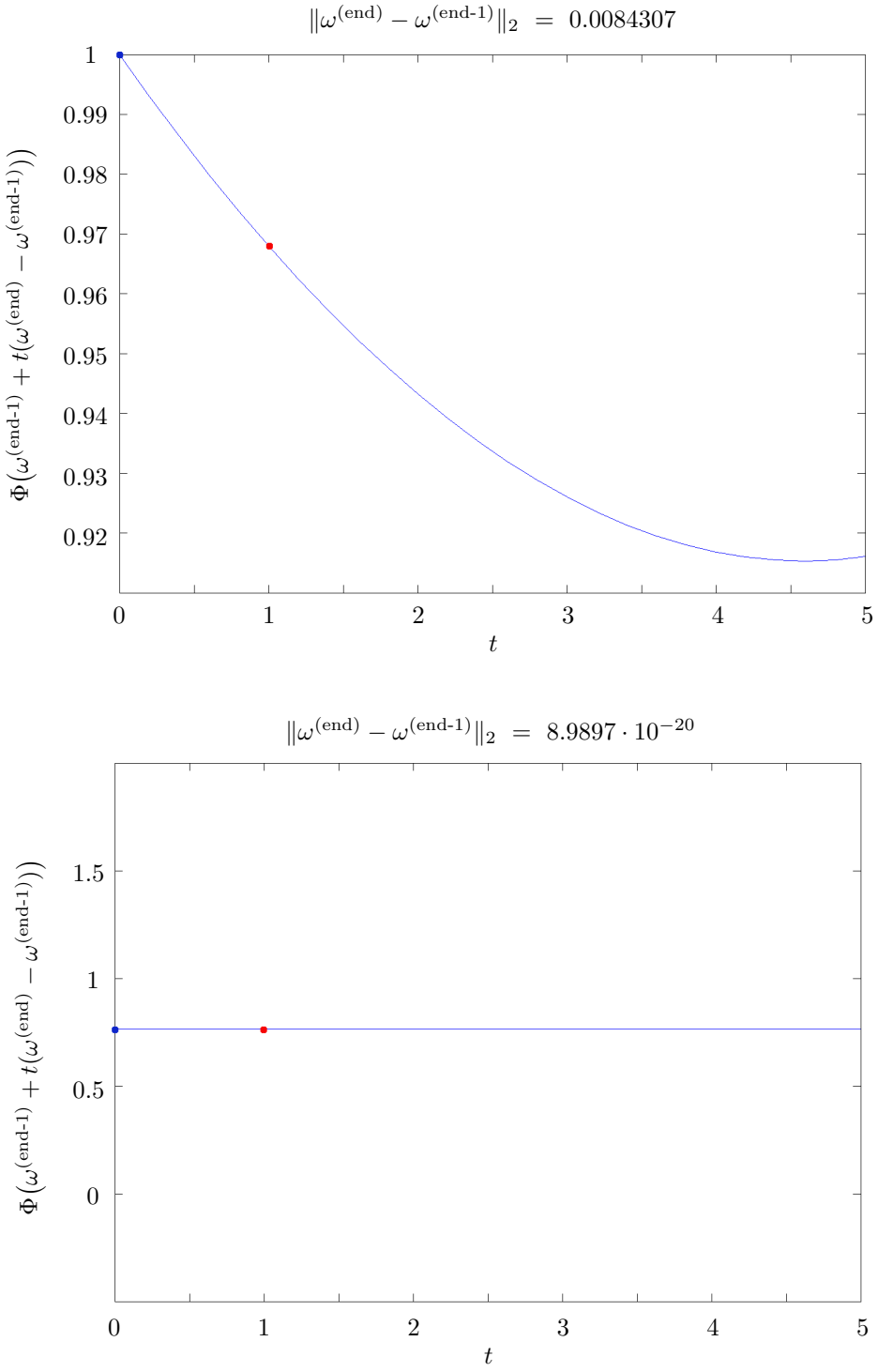


Abbildung 6.1: BFGS adaptiv, Rang 4, EPSOPT = 10^{-7}

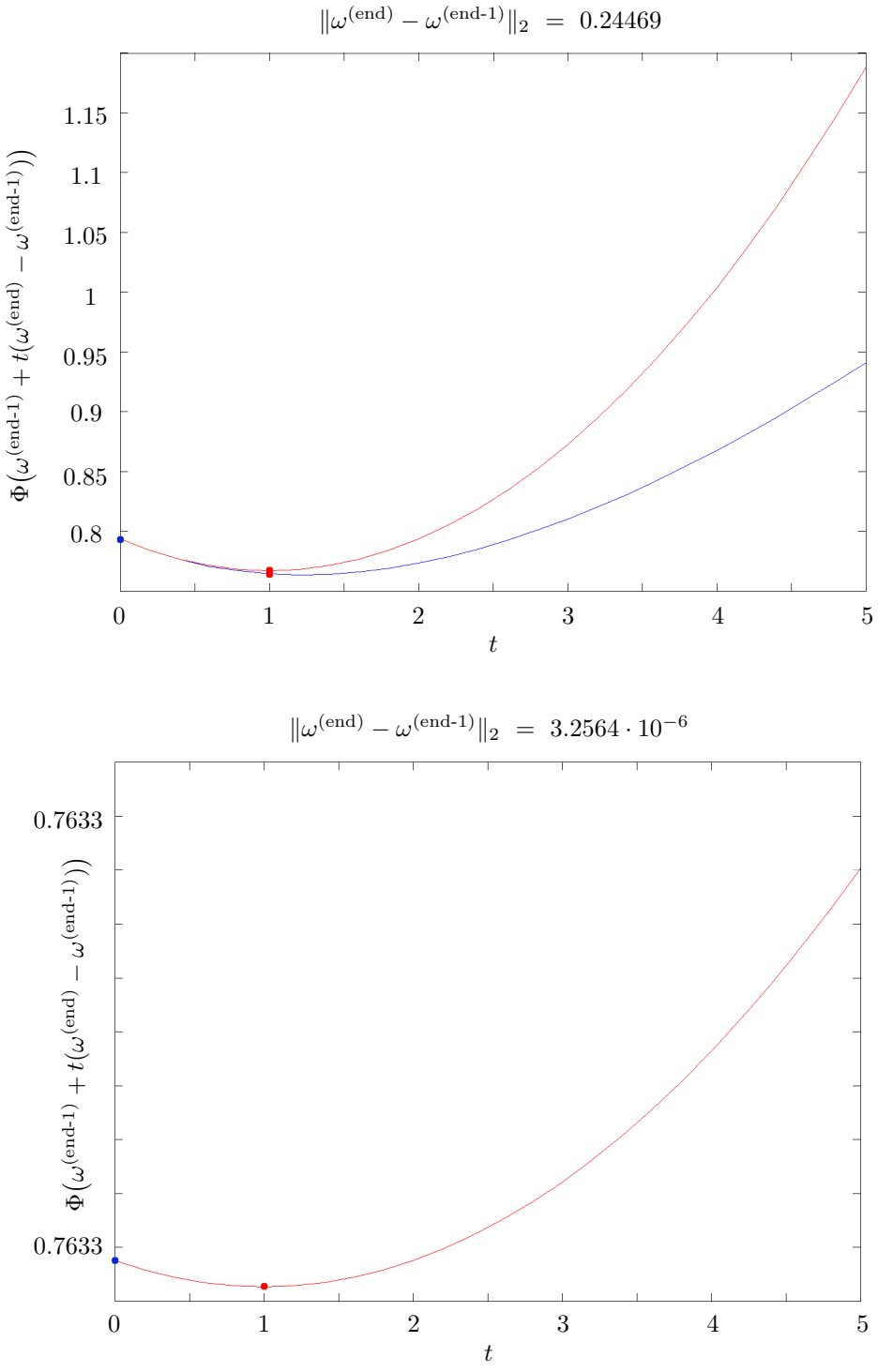


Abbildung 6.2: NEWTON adaptiv, Rang 4, EPSOPT = 10^{-7}

6.3 Test der numerischen Verfahren

Eine Möglichkeit die verwendeten Verfahren zur Lösung des Maximum Entropie Momentenproblems bezüglich der Qualität ihrer jeweils berechneten Lösung zu testen, liefert die Vorgabe von optimalen LAGRANGE-Multiplikatoren $\mu = [\mu_0; \mu_1; \ldots; \mu_n]$, d.h. die Vorgabe der gesuchten Minimalstelle μ der Zielfunktion Φ (6-1). Ausgehend von dieser geforderten Minimalstelle μ können wir mit Hilfe von (4-7), (4-8) und (4-9) die zugehörigen Texturkoeffizienten $\mathbb{S}^{r_k}$ bestimmen, d.h. jenes Maximum Entropie Momentenproblem aufstellen, welches tatsächlich das geforderte μ als Minimalstelle der zu minimierenden Zielfunktion aufweist. Dabei ist zu beachten, dass bei Vorgabe einer beliebigen Minimalstelle μ, die Normierungsbedingung (4-8) zunächst im Allgemeinen nicht erfüllt sein wird. Dies kann jedoch durch Korrektur mit Hilfe des entsprechenden Faktors ausgeglichen werden, was sich lediglich auf den ersten LAGRANGE-Multiplikator μ_0 auswirkt. Lösen wir das auf diese Weise konstruierte Maximum Entropie Momentenproblem mit einem der beschriebenen Verfahren, so sollten all diese Verfahren die bereits bekannte Minimalstelle μ reproduzieren. Die Norm der Differenz von μ und der jeweilig vom verwendeten Verfahren berechneten Minimalstelle $\omega^{(\mathrm{end})}$ liefert somit ein Qualitätsmerkmal für die berechnete Lösung des verwendeten Verfahrens. Doch hierbei ist zu beachten, dass durch Vorgabe beliebiger LAGRANGE-Multiplikatoren, die bei der Reproduktion anschließend zu minimierende Zielfunktion in der Umgebung des Minimums so flach sein kann, dass mit den reproduzierten LAGRANGE-Multiplikatoren zwar die geforderte Gradientengenauigkeit EPSOPT erfüllt sein kann, diese jedoch von den ursprünglich vorgegebenen LAGRANGE-Multiplikatoren in der Norm sehr abweichen können (siehe z.B. Tabelle 6.4, $\text{NEWTON}^{\text{a+na}}$, Rang 4 im Vergleich zu Rang $\{4, 6\}$ und dem zu Rang 4 gehörenden Linienplot in Abbildung 6.3). Insofern ist dieses "Qualitätsmerkmal" der Lösung stark vom betrachteten Beispiel, d.h. von den vorgegebenen LAGRANGE-Multiplikatoren, abhängig.

Möchte man nicht nur die Qualität der Lösung eines einzelnen Verfahrens beurteilen, sondern die einzelnen Lösungen der Verfahren für einen gegebenen Fall sinnvoll miteinander vergleichen, so gelingt dies am besten, wenn man aus der anfänglichen Berechnung der Texturkoeffizienten $\mathbb{S}^{r_k}$ die Informationen aus der bei der Integration verwendeten Unterteilung in Subregionen herausgreift, um damit die numerischen Verfahren mit dieser fixen Quadraturformel komplett nicht-adaptiv laufen zu lassen. Denn so ist aufgrund der identischen Quadraturformeln ein sinnvoller Vergleich erst möglich.

Natürlich lassen sich auch jene Verfahren testen, welche beispielsweise ausschließlich auf die adaptive Integration zurückgreifen. Aufgrund der dabei verwendeten unterschiedlichen Quadraturformeln (aufgrund unterschiedlicher Aufteilungen in Subregionen) ist ein Vergleich dieser Lösungen bezüglich ihrer Qualität dann jedoch wenig aussagekräftig.

In der folgenden Tabelle 6.3 sind die Testergebnisse der nicht-adaptiv durchgeführten Verfahren aufgeführt. Dabei wurde bei allen Verfahren des ersten Blocks (Fall Rang 4) jeweils dieselbe Quadraturformel verwendet. Somit sind diese Verfahren jeweils untereinander vergleichbar. In der anschließenden Tabelle 6.4 sind die Testergebnisse der für sich stehenden, nicht untereinander vergleichbaren Verfahren aufgeführt. In allen Fällen wurde die Minimierung von Φ erneut vom Startpunkt $\omega = 0$ begonnen. Die vorgegebenen optimalen LAGRANGE-Multiplikatoren μ wurden jeweils willkürlich gewählt. Im Fall Rang 4 war dies

$$\mu = [-0.5\ ;\ -0.15\ ;\ 0.06\ ;\ 0.02\ ;\ -0.1\ ;\ 0.05\ ;\ 0.3\ ;\ -0.05\ ;\ -0.1\ ;\ -0.02]\ ,$$

wobei die erste Komponente μ_0 aufgrund der Normierungsbedingung noch entsprechend zu $\mu_0 = -6.762556316029373$ angepasst werden musste. Folgende Ergebnisse konnten dabei in MATLAB erzielt werden:

Verfahren	Ränge	EPSOPT	Abweichung der Lösung
`fminunc`$^{\text{na}}$	4	10^{-3}	$4.2016 \cdot 10^{-2}$
BFGS$^{\text{na}}$	4	10^{-3}	$2.8844 \cdot 10^{-5}$
	4	10^{-6}	$2.2764 \cdot 10^{-8}$
NEWTON$^{\text{na}}$	4	10^{-3}	$1.6736 \cdot 10^{-5}$
	4	10^{-6}	$6.8227 \cdot 10^{-11}$
	4	10^{-10}	$6.8227 \cdot 10^{-11}$
NEWTON$^{\text{na}}$	4, 6	10^{-10}	$1.1746 \cdot 10^{-12}$

Tabelle 6.3: Qualitätstest der nicht-adaptiven Minimierungsverfahren

Verfahren	Ränge	EPSOPT	Abweichung der Lösung
`fminunc`$^{\text{a}}$	4	10^{-3}	$7.4582 \cdot 10^{-2}$
BFGS$^{\text{a}}$	4	10^{-6}	$4.5923 \cdot 10^{-3}$
BFGS$^{\text{a+na}}$	4	10^{-6}	$9.2096 \cdot 10^{-9}$
NEWTON$^{\text{a}}$	4	10^{-10}	$5.3189 \cdot 10^{-11}$
	4, 6	10^{-10}	$1.1285 \cdot 10^{-12}$
NEWTON$^{\text{a+na}}$	4	10^{-10}	$4.5923 \cdot 10^{-3}$
	4, 6	10^{-10}	$2.3626 \cdot 10^{-11}$

Tabelle 6.4: Qualitätstest der adaptiven Minimierungsverfahren

Auch hier wird wiedergespiegelt, dass das NEWTON-Verfahren in den vergleichbaren Fällen (Tabelle 6.3) jeweils die beste Reproduktion der vorgegebenen LAGRANGE-Multiplikatoren liefert. Das angesprochene Problem, das bei diesem Qualitätstest auftreten kann, wird anhand der folgenden Abbildung am erwähnten Beispiel des

NEWTON-Verfahrens in der "a+na"-Version aus Tabelle 6.4 für den Fall Rang 4 illustriert. Hierbei ist die Zielfunktion Φ über der Geraden durch die Punkte μ und $\omega^{(\mathrm{end})}$ gezeichnet, d.h. über der Verbindung der vorgegebenen Minimalstelle μ (der entsprechend zugehörige Punkt auf dem Graphen von Φ wurde rot markiert) und der tatsächlich errechneten Minimalstelle $\omega^{(\mathrm{end})}$ (blaue Markierung auf dem Graphen von Φ). An der Skalierung ist zu erkennen, dass die Zielfunktion Φ in dieser Umgebung sehr flach ist. Beide Gradienten erfüllen jeweils die in diesem Fall geforderte Genauigkeitsvorgabe von 10^{-10} und die Differenz der zugehörigen Funktionswerte beträgt $2.2692 \cdot 10^{-5}$. Dass der Funktionswert der errechneten Minimalstelle kleiner ist als der der eigentlich vorgegebenen Minimalstelle, liegt schlicht und ergreifend daran, dass die zur vorgegebenen Minimalstelle gehörenden Momente über ein Integral zu berechnen sind, welches numerisch mit der adaptiven Quadraturformel approximiert wird. Der dabei entstehende Fehler ist unter anderem einer der Gründe für diese Abweichung:

$$\|\omega^{(\mathrm{end})} - \mu\|_2 \;=\; 0.0045923$$

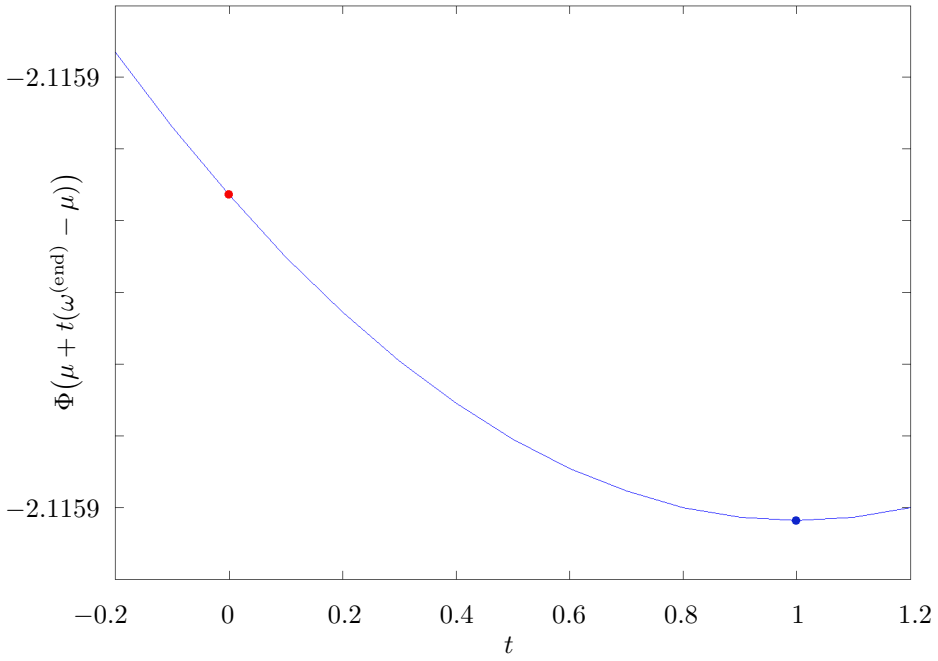


Abbildung 6.3: Zielfunktion über Gerade durch gegebener und errechneter Lösung

7 Weitere Anwendungsbeispiele der Maximum Entropie Methode

In diesem Kapitel wollen wir einen Blick auf weitere Anwendungsbeispiele der Maximum Entropie Methode werfen, genauer gesagt auf polynomiale Momentenprobleme mit Monomen in zwei Variablen als Momentenfunktionen. Diese Momentenprobleme werden demnach für ein $n \in \mathbb{N}$ von der folgenden allgemeinen Form sein:

$$\begin{aligned} &G = (\mathbb{R}^2, +) \quad , \quad \Omega = [-1,1]^2 \\ &a_i \in \mathcal{F}(\Omega, \mathbb{R}) : \text{ Momentenfunktionen } 1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, \dots \\ &b_i \in \mathbb{R} : \text{ gegebene Momente (Datensatz)} \\ &(i = 1, \dots, n) \end{aligned} \tag{7-1}$$

Auch hier ist die zu beantwortende Frage, ob es eine positive Dichte $f \geq 0$ gibt, die für $i = 1, \dots, n$ die folgenden Nebenbedingungen erfüllt:

$$\int\limits_{[-1,1]^2} a_i(x) f(x)\, dx \;=\; b_i \tag{7-2}$$

Sofern eine Lösung existiert, d.h. der Datensatz an Momenten geeignet ist, liefert der LAGRANGE-Formalismus für die Dichte erneut $f_\lambda(x) = \exp\big(-1 + \sum \lambda_i a_i(x)\big)$ mit geeigneten LAGRANGE-Multiplikatoren $\lambda_i \in \mathbb{R}$. Diese sind erneut über die Minimierung der folgenden, bereits bekannten Zielfunktion Φ zu bestimmen:

$$\Phi(\omega) \;:=\; \int\limits_{[-1,1]^2} \exp\Big(-1 + \sum_{i=1}^{n} \omega_i a_i(x)\Big)\, dx \;-\; \sum_{i=1}^{n} \omega_i b_i \quad , \quad \omega \in \mathbb{R}^n \tag{7-3}$$

Im Falle der codf konnten wir aufgrund der im Experiment detektierten Kristallorientierungen unter Verwendung dieser realen Daten immer von der Lösbarkeit des Momentenproblems ausgehen, d.h. von der Existenz einer entsprechenden Dichte. In diesem Fall möchten wir nun jedoch mit Hilfe des Minimierungsalgorithmus

untersuchen, ob zu einem Satz gegebener Momente überhaupt eine positive Dichte existiert, d.h. ob der gegebene Datensatz von einem entsprechenden positiven Maß herkommt oder nicht.

Testen wir dies an einem speziellen Beispiel, für welches bekannt ist, dass es keine kontinuierliche Dichte geben kann, sodass alle Nebenbedingungen damit erfüllt sind, so wird es interessant sein, zu beobachten, wie der Algorithmus sich bei dem Versuch verhält, solch ein Problem zu lösen. Denn in dem Fall, dass keine Dichte existiert, hat das primale Optimierungsproblem der Maximum Entropie Methode keinen zulässigen Punkt, das zugehörige duale Problem ist demnach unbeschränkt. Aufgrund dieser Unbeschränktheit von Φ ist klar, dass die iterierten LAGRANGE-Multiplikatoren $\omega^{(k)}$ beim Versuch Φ zu minimieren ins Unendliche wachsen müssen. Dies lässt sich beim Anwenden der Maximum Entropie Methode bei solch einem Problem durchaus beobachten, doch es bleibt die Frage, ob dadurch die Nebenbedingungen (7-2), d.h. die Momente, von Iteration zu Iteration immer besser approximiert werden können oder nicht.

Um von vornherein beurteilen zu können, ob ein gegebenes Momentenproblem eine Lösung besitzt, d.h. ob eine entsprechende positive Dichte existiert oder nicht, ziehen wir das Definitheitskriterium heran. Denn existiert zu gegebenem Problem eine positive Dichte, so erhalten wir diese mit den entsprechenden LAGRANGE-Multiplikatoren $\lambda_i \in \mathbb{R}$ in obiger exponentieller Form f_λ. Die zugehörige HESSE-Matrix $H_\Phi(\lambda)$ von Φ an dieser Stelle $\lambda \in \mathbb{R}^n$, deren (ij)-ter Eintrag durch

$$\int_\Omega a_i(x)a_j(x)f_\lambda(x)\ dx$$

gegeben ist, ist in diesem Fall aufgrund der strikten Konvexität von Φ positiv definit. Erzeugen wir nun die Momente b_i mit Hilfe des Delta-Maßes $\delta_{\overline{x}}$ in einem Punkt $\overline{x} \in \mathbb{R}^2$ gemäß (7-2) (jedoch über dem gesamten $\mathbb{R}^2$) derart, dass die zugehörige Matrix mit dem (ij)-ten Eintrag

$$\int_{\mathbb{R}^2} a_i(x)a_j(x)\ d\delta_{\overline{x}}(x) \tag{7-4}$$

lediglich positiv semidefinit ist, so werden wir über $\mathbb{R}^2$ keine Exponentialdichte finden können, die das gegebene Momentenproblem löst, also insbesondere auch nicht über $[-1,1]^2$. Diese Eigenschaft werden wir uns in den folgenden zwei Beispielen nun zu eigen machen.

7.1 Beispiel 1: Momente zum Delta-Maß in (0,0)

Beim ersten Anwendungsbeispiel, das wir nun betrachten werden, kommen die Momente vom Delta-Maß im Punkt $(0,0)$, sind also für ein gegebenes $n \in \mathbb{N}$ durch

$b_1 = 1$ und $b_i = 0$ für $i = 2, \ldots, n$ gegeben. Dabei betrachten wir alle polynomialen Momentenfunktionen aus (7-1) bis einschließlich vom Grad 4 und setzen deshalb $n = 15$. Mit Hilfe von (7-4) erhalten wir eine Matrix, welche lediglich an erster Stelle einen von Null verschiedenen Eintrag, nämlich 1, besitzt und somit positiv semidefinit ist. Es wird also keine Exponentialdichte für dieses Problem existieren. Nichtsdestotrotz wird der Algorithmus versuchen, das Delta-Maß mit Hilfe von Exponentialdichten zu approximieren. Anhand der in diesem Fall nach unten unbeschränkten Zielfunktion

$$\Phi(\omega) \;=\; \int\limits_{[-1,1]^2} \exp\Big(-1 + \sum_{i=1}^{n} \omega_i a_i(x) \Big)\, dx \;-\; \omega_1 \quad , \quad \omega \in \mathbb{R}^n \tag{7-5}$$

lässt sich nun schon sehr gut vorhersagen, was beim Minimierungsprozess zu erwarten ist: In jedem Minimierungsschritt wird die Zielfunktion stets einen kleineren Wert annehmen. Um dies realisieren zu können, muss der Integralteil immer mehr verschwinden, und für die erste Komponente der k-ten iterierten Lagrange-Multiplikatoren $\omega^{(k)}$ muss $\omega_1^{(k)} \to \infty$ für $k \to \infty$ gelten. Gleichzeitig wird die Approximation der Momente von Iteration zu Iteration besser werden. Damit das Integral jedoch annähernd verschwinden kann, muss aufgrund der Positivität der Exponentialfunktion sichergestellt sein, dass das im Argument stehende iterierte Polynom $-1 + \sum_{i=1}^{n} \omega_i^{(k)} a_i$ über $[-1,1]^2$ immer negativer wird, mit Ausnahme eines immer positiver werdenden Wertes über $(0,0)$. Solch ein steil abfallendes Polynom lässt sich wiederum nur mit betragsmäßig sehr großen Koeffizienten, d.h. sehr großen iterierten Lagrange-Multiplikatoren $\omega_i^{(k)}$, realisieren. Da das Polynom außerhalb des Peaks in $(0,0)$ dann sehr negativ ist, liefert die Exponentialfunktion von diesem Polynom außerhalb des Peaks quasi keinen Beitrag zum Integral. Da durch den sehr positiven Wert des Polynoms über $(0,0)$ die gesamte Masse in diesen Punkt geschoben wird, lässt sich damit jedoch auch das erste Moment approximieren. Somit wird das Delta-Maß numerisch zunächst immer besser approximiert werden, jedoch nur bis zu einem gewissen Punkt.

Um dies genauer zu beleuchten, stellt sich die Frage, welches der vorgestellten numerischen Verfahren dieses Problem am besten behandelt. Da wir in diesem Fall keinen Wert auf die beste Rechenzeit legen, sondern lediglich an der Qualität der Approximation des Delta-Maßes interessiert sind, werden wir lediglich Verfahren verwenden, welche ausschließlich auf adaptive Integration zurückgreifen, d.h. auf die rein adaptiven Versionen des BFGS-Verfahrens und des Newton-Verfahrens.
Bei der Approximation des Delta-Maßes wird dabei immer folgendes Problem auftreten: Je besser das Delta-Maß approximiert wird, d.h. je steiler die Exponentialdichte wird, desto schlechter werden die Matrizen des BFGS-Verfahrens bzw. die Hesse-Matrizen des Newton-Verfahrens konditioniert sein. Dabei tauchen ab einem gewissen Punkt Konditionszahlen in der Größenordnung von 10^{17} auf. Durch zu große Konditionszahlen kann das mit diesen Matrizen aufgestellte Gleichungssystem zur Bestimmung der nächsten Abstiegsrichtung jedoch nicht mehr gut bzw. nur sehr feh-

lerbehaftet gelöst werden. Im Detail bedeutet dies, dass man unter Umständen eine vermeintliche Abstiegsrichtung bestimmt, die in Wirklichkeit überhaupt keine ist. Dies hat wiederum zur Folge, dass die Momente auf einmal wieder schlechter approximiert werden. Beim BFGS-Verfahren tritt dieser Effekt jedoch erst sehr spät auf, d.h. bis zum Auftreten dieses Effekts ist das Delta-Maß bereits sehr gut approximiert. Im Folgenden ist eine Bilderserie zum Minimierungsprozess der Zielfunktion (7-5) dieses Momentenproblems zu sehen. Dabei wurde das adaptive BFGS-Verfahren ausgehend vom Startpunkt $\omega = 0$ unter der Genauigkeitsforderung EPSOPT $= 10^{-10}$ verwendet. In jeder der Abbildungen ist jeweils ein Plot der aktuellen Exponentialdichte über $[-1,1]^2$, ein Linienplot der Zielfunktion entlang der Abstiegsrichtung $\omega^{(k+1)} - \omega^{(k)}$, ein Plot des aktuellen Polynoms $-1 + \sum_{i=1}^{n} \omega_i^{(k)} a_i$ über $[-1,1]^2$ und ein Plot der Null-Höhenlinien des Polynoms zu sehen:

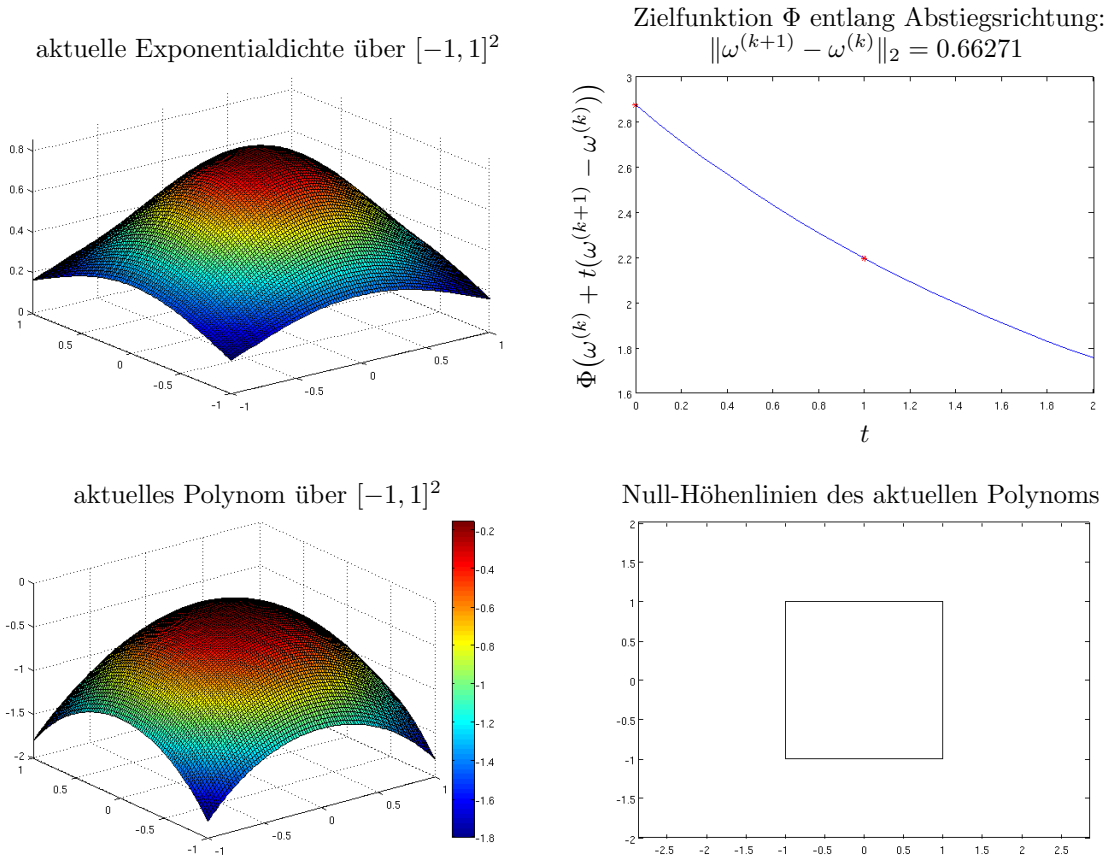
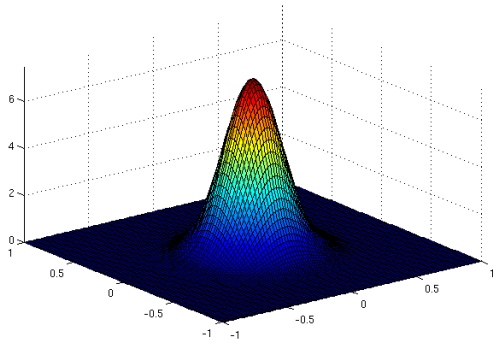
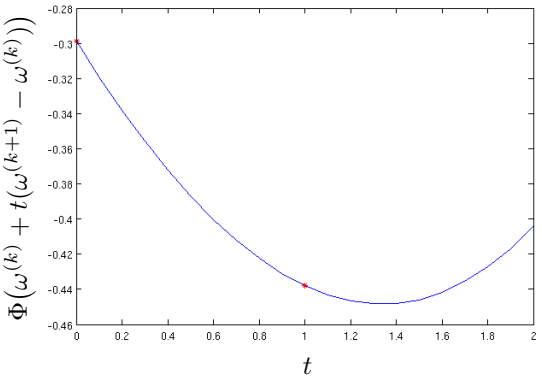


Abbildung 7.1: BFGS-Approximation der Momente zum Delta-Maß in $(0,0)$

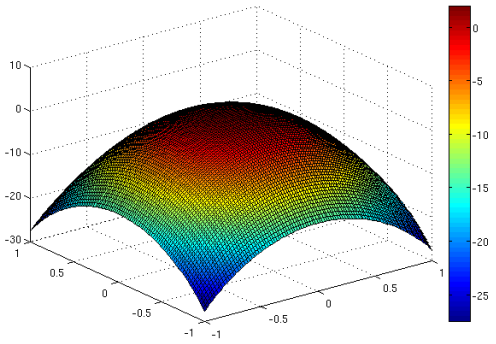
aktuelle Exponentialdichte über $[-1,1]^2$



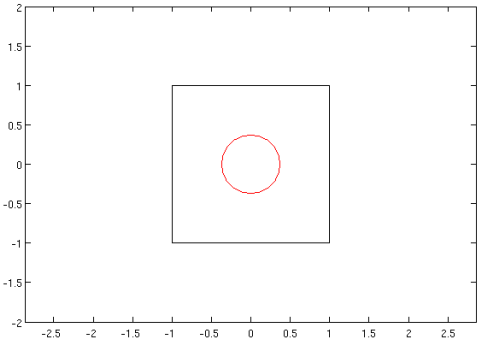
Zielfunktion Φ entlang Abstiegsrichtung:
$\|\omega^{(k+1)} - \omega^{(k)}\|_2 = 4.4292$



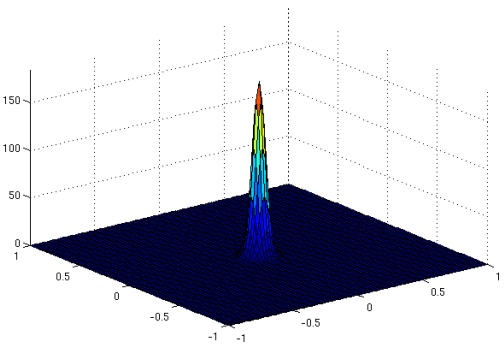
aktuelles Polynom über $[-1,1]^2$



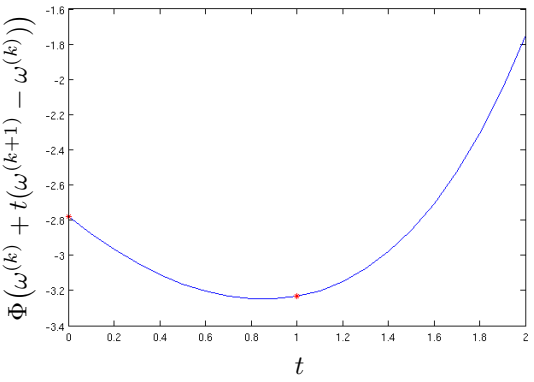
Null-Höhenlinien des aktuellen Polynoms



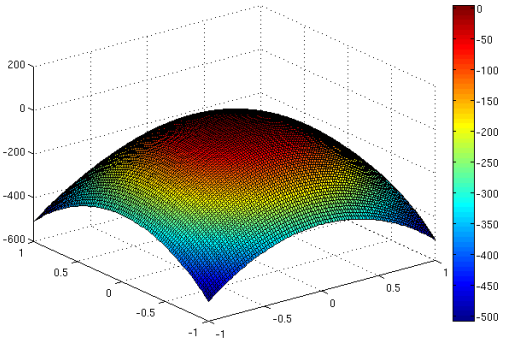
aktuelle Exponentialdichte über $[-1,1]^2$



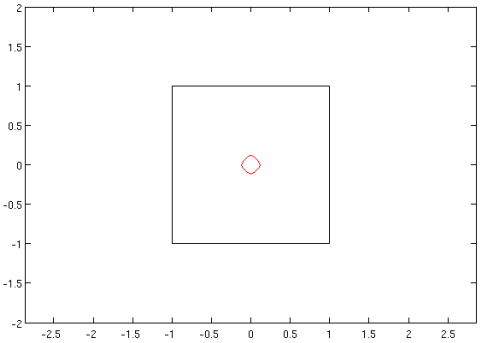
Zielfunktion Φ entlang Abstiegsrichtung:
$\|\omega^{(k+1)} - \omega^{(k)}\|_2 = 220.1226$

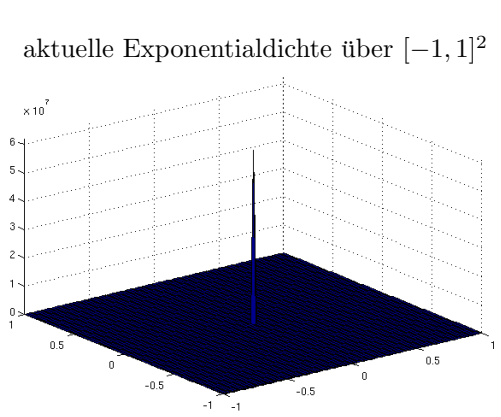


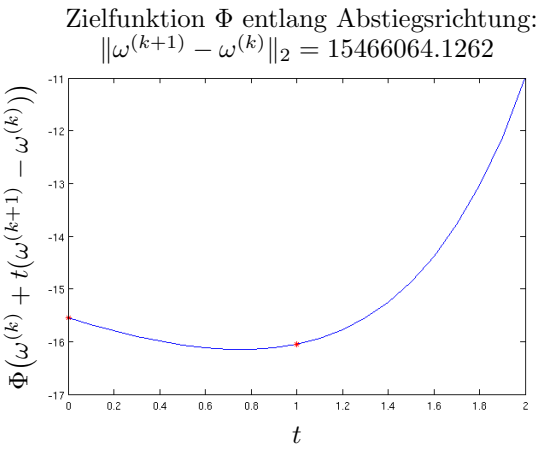
aktuelles Polynom über $[-1,1]^2$

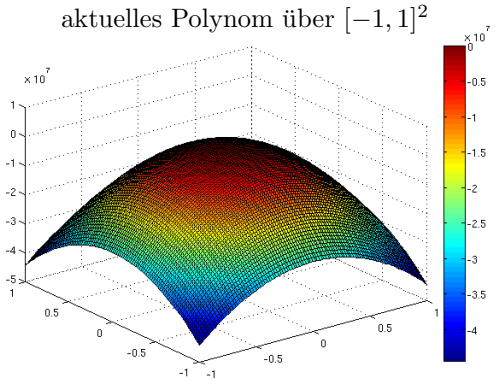


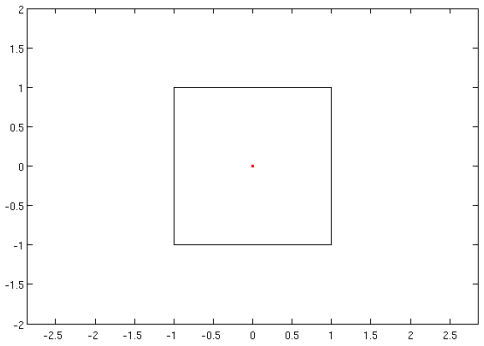
Null-Höhenlinien des aktuellen Polynoms











Anhand der Bilderserie ist sehr gut zu erkennen, wie das aktuelle Polynom von Schritt zu Schritt über $[-1,1]^2$ immer steiler wird, und die Null-Höhenlinie sich dadurch immer weiter um die Stelle $(0,0)$ zusammenzieht. Umso besser man das Delta-Maß approximieren möchte, desto extremere Polynomkoeffizienten sind dazu erforderlich. Dies ist anhand der von Iteration zu Iteration größer werdenden Schrittweite, die in jeder Abbildung jeweils mit angegeben ist, nachzuvollziehen. Kurz bevor der Effekt der schlechten Matrixkonditionierung einsetzt, konnten die vorgegebenen Momente $b_1, \ldots, b_{15}$ wie folgt approximiert werden:

	Momente	Approximation
b_1	1	0.916807578666988
b_2	0	−0.000000000000001
b_3	0	0.000000000000003
b_4	0	0.000000056030521
b_5	0	0.000000000000000
b_6	0	0.000000056030521
b_7	0	−0.000000000000000
b_8	0	0.000000000000000

	Momente	Approximation
b_9	0	−0.000000000000000
b_{10}	0	0.000000000000000
b_{11}	0	0.000000000000010
b_{12}	0	0.000000000000000
b_{13}	0	0.000000000000003
b_{14}	0	0.000000000000000
b_{15}	0	0.000000000000010

Tabelle 7.1: BFGS-Approximation der Momente zum Delta-Maß in $(0,0)$

Die berechnete Dichte kann nachträglich noch problemlos durch anpassen des ersten Moments normiert werden. Insgesamt erhalten wir in diesem Fall mit Hilfe des Minimierungsalgorithmus also trotz der Unlösbarkeit dieses Problems gute Informationen über die Herkunft der Momentdaten.

Wenden wir für dieses Momentenproblem hingegen das adaptive NEWTON-Verfahren unter identischen Voraussetzungen an, d.h. mit dem Startpunkt $\omega = 0$ unter der Genauigkeitsforderung EPSOPT $= 10^{-10}$, so erhalten wir schlechtere Ergebnisse als beim BFGS-Verfahren. Anfänglich liefert auch das NEWTON-Verfahren zunächst brauchbare Approximationen des Delta-Maßes, welche dann jedoch sehr schnell extrem schlecht werden. Zu beobachten ist, dass der Funktionswert der Zielfunktion von einer Iteration zur nächsten auf einmal extrem zunimmt, obwohl der Funktionswert im Minimierungsprozess eigentlich sukzessive verringert werden sollte. Dies kann nur dann vorkommen, wenn das dem NEWTON-Verfahren zugrundeliegende quadratische Modell der Zielfunktion über der Verbindung zweier aufeinanderfolgender Iterationspunkte $\omega^{(k)}$ und $\omega^{(k+1)}$ eine sehr schlechte Approximation der Zielfunktion darstellt. So kann dieses quadratische Modell in einer Umgebung des aktuellen Iterationspunktes $\omega^{(k)}$ zwar lokal geeignet sein, die errechnete Minimalstelle $\omega^{(k+1)}$ des quadratischen Modells jedoch bereits einen größeren Zielfunktionswert liefern.

Doch wie muss das zugehörige Polynom im Argument der Exponentialdichte aussehen, um solche Probleme verursachen zu können? Diese Frage kann mit folgender Überlegung beantwortet werden: Innerhalb von $[-1,1]^2$ wird es Bereiche geben müssen, über denen das zugehörige Polynom so positiv ist, dass das Integral einen nicht zu vernachlässigenden Beitrag zum Funktionswert der Zielfunktion (7-5) liefert. Im Gegensatz zum BFGS-Verfahren schlägt das NEWTON-Verfahren hierbei jedoch einen Weg ein, auf welchem die verwendeten Polynome nicht nur über der Stelle $(0,0)$ extrem positiv sind, sondern auch über den Bereichen in den Ecken von $[-1,1]^2$. Außer der Null-Höhenlinie des Polynoms, welche die Stelle $(0,0)$ einkreist, gibt es in unserem Fall also noch eine zweite Null-Höhenlinie, die zunächst teilweise durch das Innere von $[-1,1]^2$ verläuft. Um das Delta-Maß mit solchen Polynomen approximieren zu können, wird es für den Minimierungsalgorithmus unerlässlich sein, die Polynomkoeffizienten iterativ so zu wählen, dass diese zweite Null-Höhenlinie von Iteration zu Iteration weiter aus dem Integrationsgebiet $[-1,1]^2$ herausgetrieben wird. Ansonsten könnte die Masse über $[-1,1]^2$ nicht im Punkt $(0,0)$ konzentriert werden, was es unmöglich machen würde, damit das Delta-Maß zu approximieren. Da die Beschaffenheit des Polynoms außerhalb von $[-1,1]^2$ bei der Integration jedoch keine Rolle spielt, wird der Minimierungsalgorithmus diese Null-Höhenlinie aber auch nur bis zum Rand des Integrationsgebietes $[-1,1]^2$ heraustreiben und nicht weiter. Und genau das ist es, was das Verfahren so instabil macht. Denn bereits eine kleine Änderung der Polynomkoeffizienten von der einen zur nächsten Iteration kann erhebliche Auswirkungen auf das Polynom haben. Ändert sich das Polynom dadurch so stark, dass der durchgeführte NEWTON-Schritt zu groß ist, d.h. die Zielfunktion plötzlich einen extrem hohen Funktionswert annimmt, so dringt diese Null-Höhenlinie wieder in das Integrationsgebiet $[-1,1]^2$ ein und bringt das Verfahren dadurch zum Scheitern.

Selbst unter Verwendung einer Schrittweitensteuerung, die dafür sorgt, dass in jedem Schritt nur soweit vorangeschritten wird, dass die Verringerung des Zielfunktionswertes garantiert werden kann, erzielt man mit dem NEWTON-Verfahren keine besseren Approximationen. Das Problem eines zu großen Schrittes tritt dann zwar nicht mehr auf, jedoch registriert das NEWTON-Verfahren die sich außerhalb von $[-1,1]^2$ auftürmende Front nicht, da u.a. bei der Bestimmung der HESSE-Matrix das Gebiet außerhalb von $[-1,1]^2$ ebenso wenig berücksichtigt wird. Deshalb wird es weiterhin versuchen, entlang derselben Richtung zu minimieren, dabei jedoch immer kleinere Schrittweiten verwenden und letzten Endes stagnieren.

Ob mit oder ohne Schrittweitensteuerung wird das NEWTON-Verfahren in beiden Fällen sehr schnell an einen kritischen Punkt kommen, ab welchem eine besser werdende Approximation des Delta-Maßes nicht mehr möglich sein wird. Die bis zu diesem Punkt erzielten Approximationen der Momente sind für dieses Beispiel in Tabelle 7.2 aufgeführt. Es ist deutlich zu erkennen, dass die Momente mit Hilfe des NEWTON-Verfahrens wesentlich schlechter approximiert werden konnten als unter Verwendung des BFGS-Verfahrens (siehe Tabelle 7.1). Die folgenden Abbildungen spiegeln die aufgeführten Überlegungen zum NEWTON-Verfahren wieder und zeigen nun auch optisch den Nachteil dieses Verfahrens im Vergleich zum BFGS-Verfahren auf:

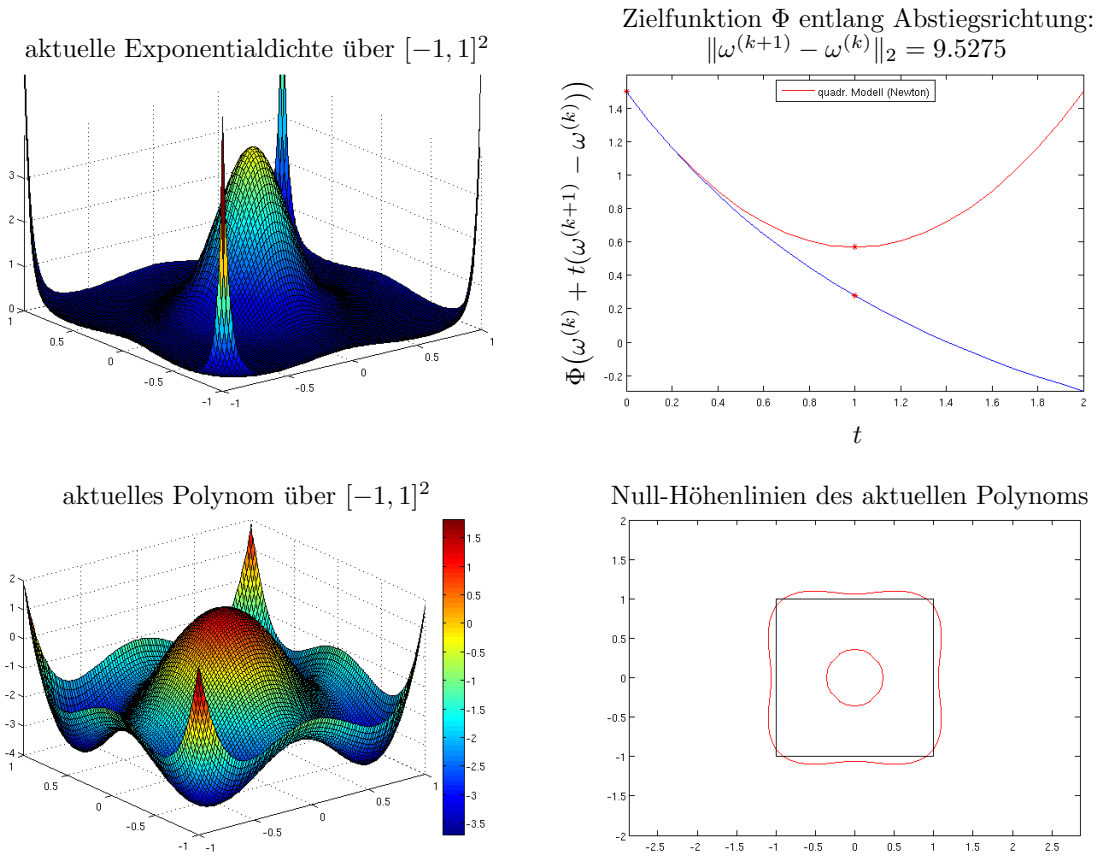
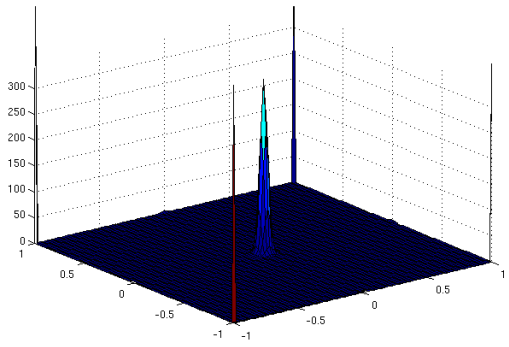
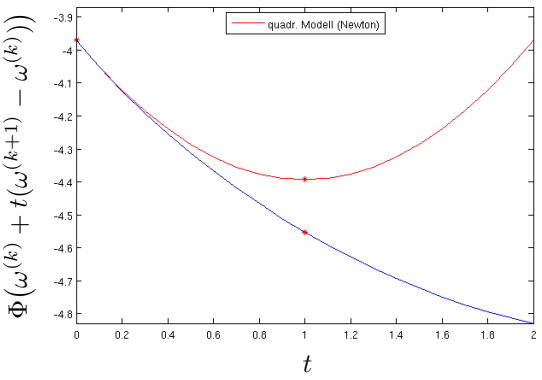


Abbildung 7.2: NEWTON-Approximation der Momente zum Delta-Maß in $(0,0)$

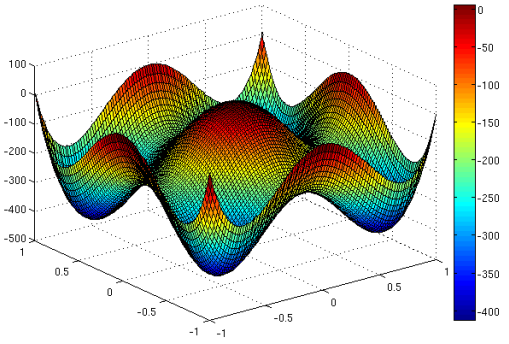
aktuelle Exponentialdichte über $[-1,1]^2$



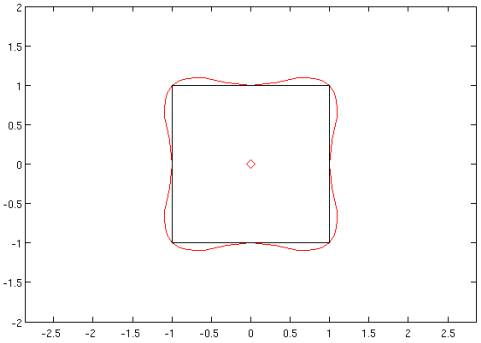
Zielfunktion Φ entlang Abstiegsrichtung:
$\|\omega^{(k+1)} - \omega^{(k)}\|_2 = 737.0381$



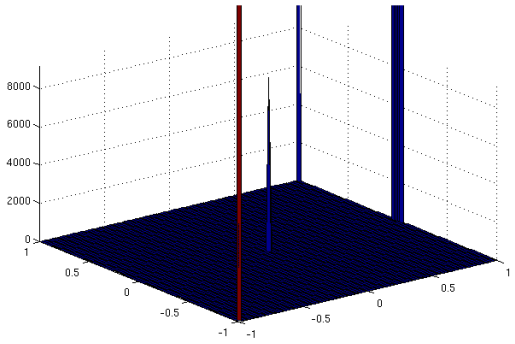
aktuelles Polynom über $[-1,1]^2$



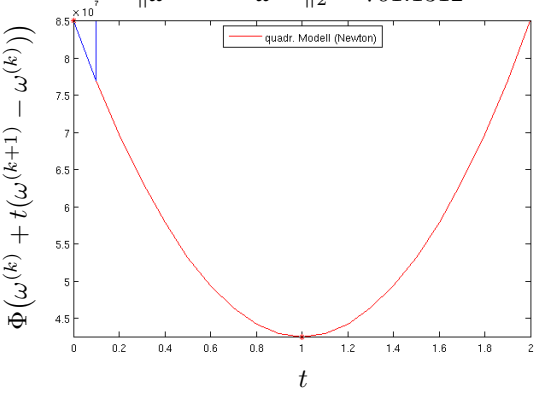
Null-Höhenlinien des aktuellen Polynoms



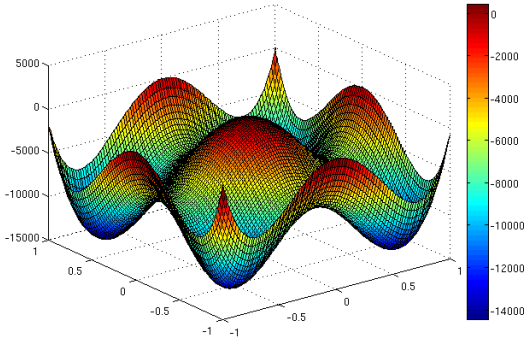
aktuelle Exponentialdichte über $[-1,1]^2$



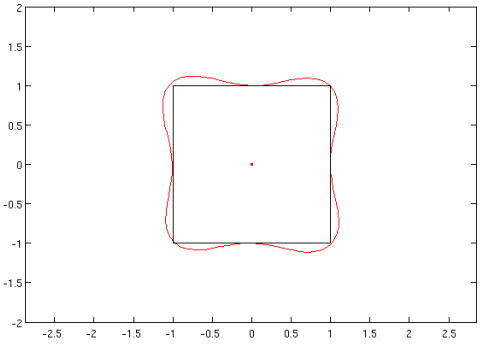
Zielfunktion Φ entlang Abstiegsrichtung:
$\|\omega^{(k+1)} - \omega^{(k)}\|_2 = 761.1312$



aktuelles Polynom über $[-1,1]^2$



Null-Höhenlinien des aktuellen Polynoms



In der letzten Abbildung der Bilderserie ist zu erkennen, wie sehr das quadratische Modell für diese Iteration von der Zielfunktion abweicht. Führt man nun den Schritt zum neu berechneten Iterationspunkt $\omega^{(k+1)}$ durch, so wird das neue Polynom so sehr abgeändert, dass die besagte Null-Höhenlinie erneut in $[-1,1]^2$ eintritt. Dies ist gleichbedeutend mit extrem hohen Funktionswerten der Zielfunktion über manchen Bereichen von $[-1,1]^2$, was dazu führt, dass alle iterierten Momente betragsmäßig ebenso extrem groß werden. Die weiteren Ergebnisse des NEWTON-Verfahrens für den eigentlichen Minimierungsprozess werden demnach sofort unbrauchbar, da die zugehörigen HESSE-Matrizen ab diesem Zeitpunkt sehr schlecht konditioniert sind. In der folgenden Abbildung ist eine solche Situation zu sehen:

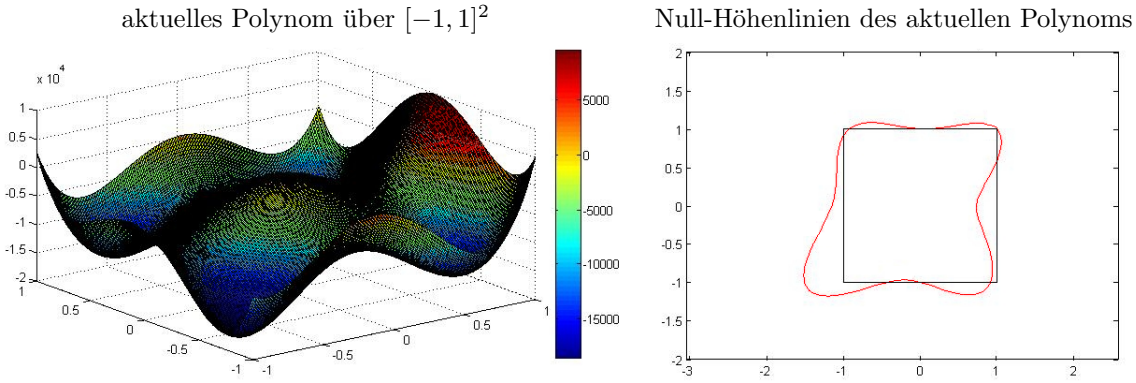


Abbildung 7.3: Rückfluss der Masse bei Verwendung des NEWTON-Verfahrens

Die bis zu dieser problematischen Iteration erzielten Approximationen der Momente sind für dieses Beispiel in der folgenden Tabelle aufgeführt:

	Momente	Approximation
b_1	1	1.263341433635335
b_2	0	0.000009219667067
b_3	0	−0.000009219666410
b_4	0	0.000058600276823
b_5	0	−0.000009219394051
b_6	0	0.000058600274878
b_7	0	0.000009219122401
b_8	0	−0.000009219121718

	Momente	Approximation
b_9	0	0.000009219121055
b_{10}	0	−0.000009219120417
b_{11}	0	0.000017590929699
b_{12}	0	−0.000009218849405
b_{13}	0	0.000009220739003
b_{14}	0	−0.000009218848078
b_{15}	0	0.000017590926725

Tabelle 7.2: NEWTON-Approximation der Momente zum Delta-Maß in $(0,0)$

7.2 Beispiel 2: Momente zum Delta-Maß in (2,0)

Bei diesem zweiten Anwendungsbeispiel betrachten wir Momente, die von einem Delta-Maß im Punkt $(2, 0)$ kommen, d.h. von einem Maß, dessen Träger außerhalb von $[-1, 1]^2$ liegt. Für ein $n \in \mathbb{N}$ bestimmen wir die Momente gemäß (7-2) zu

$$b_i \;=\; \int\limits_{\mathbb{R}^2} a_i(x)\; d\delta_{\overline{x}}(x) \tag{7-6}$$

für $i = 1, \ldots, n$ und $\overline{x} = (2, 0)$. Dabei betrachten wir erneut alle polynomialen Momentenfunktionen a_i aus (7-1) bis einschließlich vom Grad 4 ($n = 15$) und erhalten somit die Momente $b_1 = 1$, $b_2 = 2$, $b_4 = 4$, $b_7 = 8$, $b_{11} = 16$ und $b_i = 0$ für $i \in \{3, 5, 6, 8, 9, 10, 12, 13, 14, 15\}$.

Ziel soll es nun erneut sein, das zu diesen Momenten gehörende Momentenproblem mit Hilfe der Maximum Entropie Methode zu behandeln, d.h. über $[-1, 1]^2$ eine entsprechende Exponentialdichte zu suchen, welche diese Momente erzeugt. Da die entsprechende Matrix (7-4) erneut positiv semidefinit ist, wird auch zu diesem Problem keine Exponentialdichte existieren. Welches Verhalten ist in diesem Fall also vom Algorithmus der Maximum Entropie Methode zu erwarten?

Da zu diesem Beispiel keine Exponentialdichte existiert und der Träger des für die Generierung der Momente verwendeten Delta-Maßes, und somit die gesamte Masse, außerhalb von $[-1, 1]^2$ liegt, wird der Algorithmus im Gegensatz zum vorherigen Beispiel die gesamte Masse nicht über einem einzigen Punkt in $[-1, 1]^2$ konzentrieren können, sondern vielmehr versuchen, alle Masse aus $[-1, 1]^2$ herauszuschieben. Um eine Aussage darüber zu treffen, was dabei für Polynome zu erwarten sind, werfen wir einen Blick auf die Zielfunktion (7-3), welche sich mit Hilfe von (7-6) für alle $\omega \in \mathbb{R}^n$ wie folgt umschreiben lässt:

$$\Phi(\omega) \;=\; \int\limits_{[-1,1]^2} \exp\Big(-1 + \sum_{i=1}^{n} \omega_i a_i(x) \Big)\; dx \;-\; \int\limits_{\mathbb{R}^2} \Big(\sum_{i=1}^{n} \omega_i a_i(x) \Big)\; d\delta_{\overline{x}}(x) \tag{7-7}$$

Da man in dieser Formulierung sehr gut erkennen kann, dass in beiden Integralen dasselbe Polynom auftritt, jedoch einmal linear und einmal als Argument der Exponentialfunktion, kann es für den Algorithmus nur eine Möglichkeit geben, den Funktionswert der auch in diesem Fall unbeschränkten Zielfunktion sukzessive zu verringern: Die Polynomkoeffizienten müssen so gewählt werden, dass das Polynom über dem kompletten Gebiet $[-1, 1]^2$ sehr negativ, und außerhalb von $[-1, 1]^2$ sehr positiv wird. Der Algorithmus wird also auch in diesem Fall versuchen, die Null-Höhenlinie des Polynoms, und somit die gesamte Masse, aus $[-1, 1]^2$ herauszuschieben. Da der Minimierungsalgorithmus bei der Integration über die Exponentialdichte jedoch keinerlei Informationen über den Bereich außerhalb von $[-1, 1]^2$ erhält, wird er die

Null-Höhenlinie erneut nur bis zum Rand von $[-1,1]^2$ schieben können. Die vom Algorithmus zu erwartende Exponentialdichte wird über $[-1,1]^2$ demnach annähernd Null sein, und an den Bereichen des Randes, an welche die Null-Höhenlinie gedrängt wird, annähernd senkrecht ins Unendliche steigen. Dies ist, wie bereits beim vorherigen Beispiel beschrieben wurde, erneut numerisch sehr instabil, da bereits kleine Abänderungen der Polynomkoeffizienten dazu führen können, dass Masse in $[-1,1]^2$ zurückfließt. In der folgenden Bilderserie, welche mit dem adaptiven BFGS-Verfahren mit Startpunkt $\omega = 0$ bei geforderter Genauigkeit EPSOPT $= 10^{-10}$ zu diesem Beispiel generiert wurde, ist dieses Verhalten des Algorithmus sehr gut nachvollziehbar:

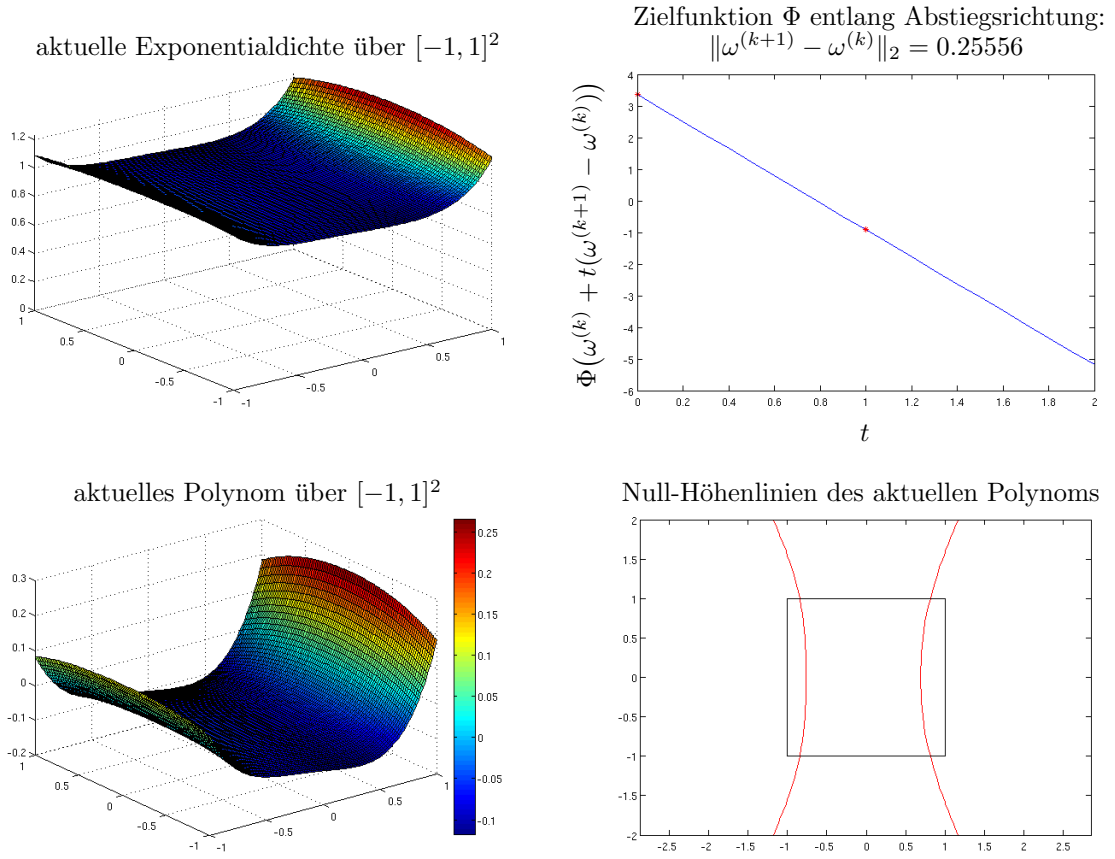
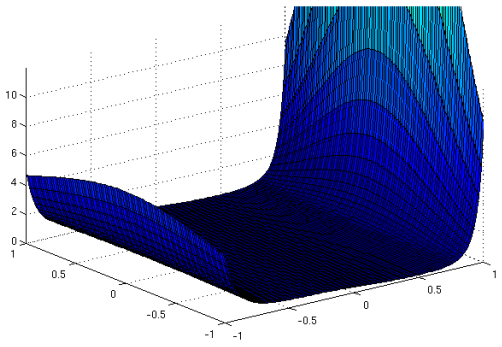
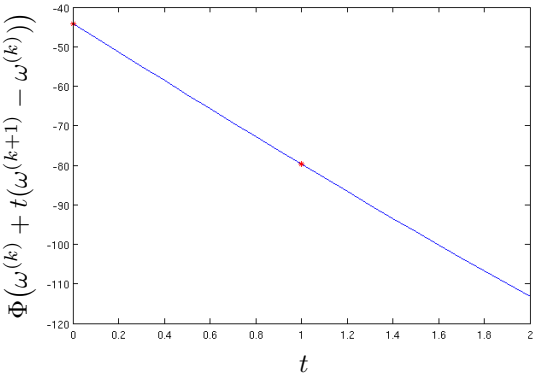


Abbildung 7.4: BFGS-Approximation der Momente zum Delta-Maß in $(2,0)$

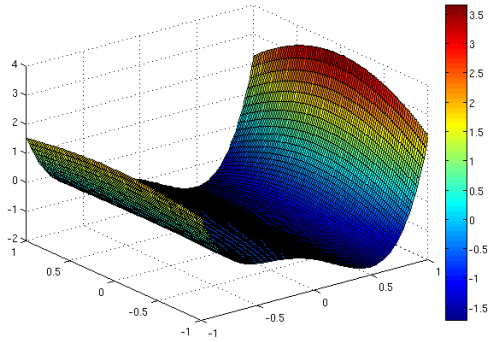
aktuelle Exponentialdichte über $[-1,1]^2$



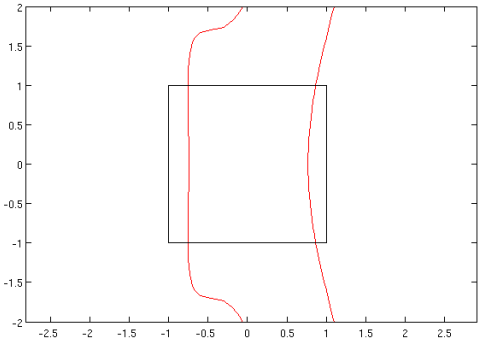
Zielfunktion Φ entlang Abstiegsrichtung:
$\|\omega^{(k+1)} - \omega^{(k)}\|_2 = 2.5816$



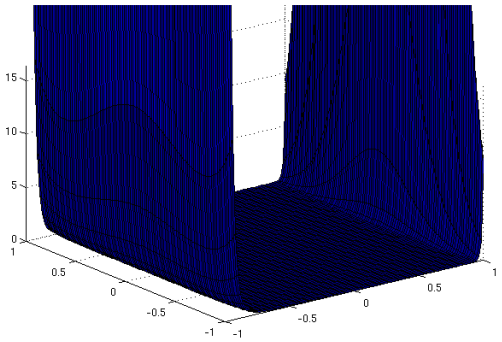
aktuelles Polynom über $[-1,1]^2$



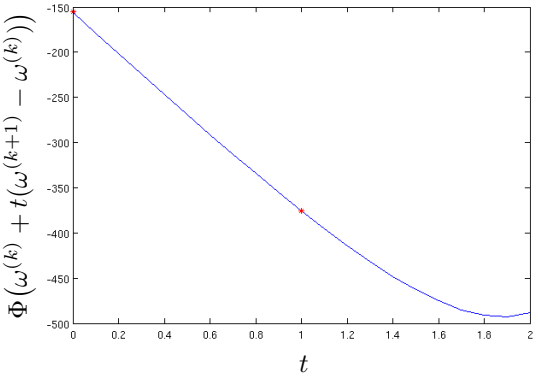
Null-Höhenlinien des aktuellen Polynoms



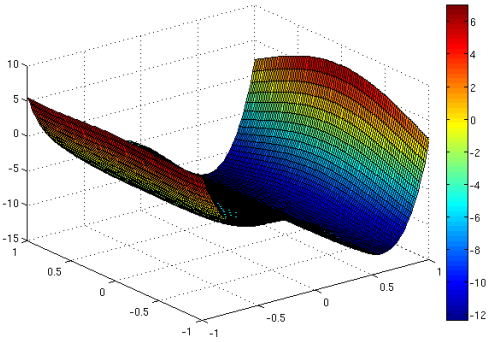
aktuelle Exponentialdichte über $[-1,1]^2$



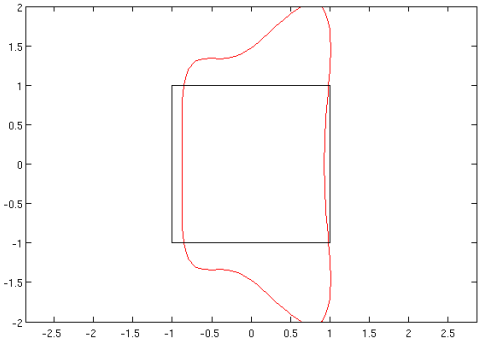
Zielfunktion Φ entlang Abstiegsrichtung:
$\|\omega^{(k+1)} - \omega^{(k)}\|_2 = 19.1989$

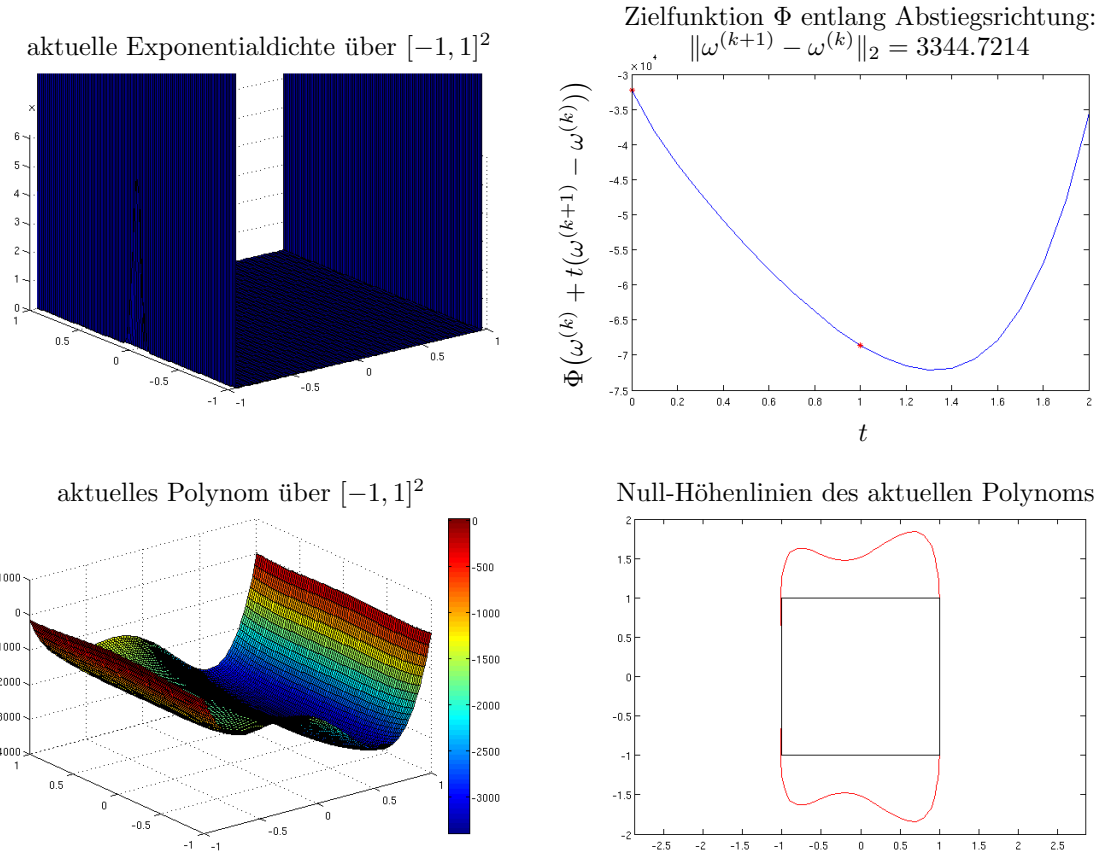


aktuelles Polynom über $[-1,1]^2$



Null-Höhenlinien des aktuellen Polynoms





In diesem Beispiel ist es demnach unmöglich, das außerhalb von $[-1,1]^2$ liegende Delta-Maß numerisch zu approximieren. In solch einem Fall kann demnach, aufgrund der Ergebnisse des Algorithmus, weder der Schluss gezogen werden, dass die gegebenen Momente von einem Punktmaß kommen, noch beurteilt werden, ob die Momente überhaupt von einem positiven Maß kommen oder nicht. Dennoch liefert uns das Verhalten des Algorithmus die wichtige Information, dass der Träger des Maßes, sofern eine entsprechende Dichte existiert, außerhalb von $[-1,1]^2$ liegen muss.

7.3 Fazit

Möchte man für solche polynomialen Momentenprobleme die Frage beantworten, ob gegebene Daten (in unserem Fall Momente) von einem positiven Maß beispielsweise auf $[-1,1]^2$ kommen, so liefert die Behandlung dieser Probleme mit der Maximum Entropie Methode folgende Antworten:

- Kommen die Daten von einem positiven Maß mit Dichte auf $[-1,1]^2$, so wird man mit Hilfe der Maximum Entropie Methode eine entsprechende Dichte

finden können (siehe Existenzresultat in der Arbeit von JUNK, BUDDAY und BÖHLKE[24]).

- Gibt es zu den Daten kein reguläres aber ein singuläres Maß auf $[-1,1]^2$, dann legen die Beispiele nahe, dass im Algorithmus die Momente konvergieren, während die LAGRANGE-Multiplikatoren divergieren.
- Sind die Daten nicht mit einem positiven Maß kompatibel, so wird auch der Algorithmus keine entsprechende Dichte finden können.
- Kommen die Daten von einem positiven Maß, dessen Träger außerhalb von $[-1,1]^2$ liegt, so gibt es, je nach Problem, verschiedene Möglichkeiten an erzielbaren Ergebnissen der Maximum Entropie Methode:

  Bei einem Problem wie dem in Kapitel 7.2 behandelten, d.h. dass die Daten von einem einzelnen Punktmaß außerhalb des betrachteten Gebiets kommen, wird die Maximum Entropie Methode versuchen, die gesamte Masse aus dem betrachteten Gebiet zu schieben.

  Kommen die Daten jedoch beispielsweise von einem positiven Maß, das aus der Summe der beiden Punktmaße in $(-2,0)$ und $(2,0)$ besteht, so ist es bei Betrachtung des Momentenproblems, bis zur einschließlichen Verwendung der linearen Monome als Momentenfunktionen, auch möglich, über $[-1,1]^2$ eine Dichte zu finden, obwohl der Träger des ursprünglichen Maßes außerhalb von $[-1,1]^2$ lag. Denn sind die beiden Punktmaße in der entsprechenden Summe der Maße gleichgewichtet, so wird man über $[-1,1]^2$ eine konstante Dichte finden können, die dieselben Momente erzeugt. Sind die Punktmaße in der Summe hingegen nicht gleichgewichtet, mit einer nicht zu extremen Ungleichgewichtung, so wird man über $[-1,1]^2$ im Allgemeinen eine lineare Dichte finden können. In solchen Fällen wird man mit dieser Methode deshalb keine Aussage mehr darüber treffen können, ob die Daten ursprünglich vielleicht auch von einer Summe von Punktmaßen außerhalb von $[-1,1]^2$ kamen oder nicht.

  Auch im Fall der Summe mehrerer Punktmaße innerhalb von $[-1,1]^2$ kann es möglich sein, dass der Algorithmus eine kontinuierliche Dichte findet, die dieselben Momente erzeugt. Auch hier wird es demnach mit dieser Methode nicht möglich sein, eine Aussage darüber zu treffen, ob die Daten ursprünglich vielleicht auch von einem anderen Maß, wie etwa von einer Summe von Punktmaßen, kamen oder nicht.

Die Tatsache, im Nachhinein nicht mehr sagen zu können, mit welchem Maß die Momente erzeugt wurden, lässt sich dadurch erklären, dass die Abbildung, welche einem Maß seine entsprechenden Momente zuordnet, nicht injektiv ist. Somit lässt sich aus dem erfolgreichen Ablauf des Algorithmus nur indirekt etwas über das Aus-

gangsmaß sagen. Aussagekräftiger ist der Fall der Nichtkonvergenz des Algorithmus. Denn hierbei liegt die Schlussfolgerung nahe, dass das Maß singulär ist, wenn die LAGRANGE-Multiplikatoren bei Konvergenz der Momente divergieren, bzw. dass kein passendes Maß existiert, wenn die Momente nicht konvergieren.

Dieses Kapitel soll lediglich zeigen, dass die Maximum Entropie Methode ebenso bei alternativen Problemstellungen, wie etwa der Frage, ob gewisse gegebene Daten Momente eines positiven Maßes sind oder nicht, Anwendung finden kann. Bei der ausführlichen Behandlung solcher Probleme wird es jedoch nicht ausreichend sein, lediglich die hier verwendeten Verfahren zu benutzen. Alternative Verfahren, wie zum Beispiel die sogenannten *Trust-Region-Verfahren*, könnten hier eventuell noch bessere Ergebnisse erzielen.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Numerik von Maximum Entropie Momentenproblemen in der Texturanalyse. Das überwiegend behandelte Problem der aus der Texturanalysis bekannten Kristallorientierungsverteilungsfunktion (codf) ist dabei ein Spezialfall des folgenden allgemeinen Momentenproblems:

Finde eine positive Dichte $f \geq 0$ bezüglich einem Maß μ, die zu gegebenem $n \in \mathbb{N}$ bezüglich der Momentenfunktionen a_i und der Momente b_i folgende Bedingungen erfüllt:

$$\int_\Omega a_i f \; d\mu \;=\; b_i \quad \textit{für} \quad i = 1, \ldots, n$$

Dabei ist Ω eine Teilmenge einer lokalkompakten topologischen Gruppe G, μ das entsprechende Haar-Maß auf G, und die a_i bezeichnen sogenannte reellwertige Darstellungsfunktionen auf der Gruppe G. Da dieses Problem, sofern es eine Lösung besitzt, im Allgemeinen nicht eindeutig lösbar ist, wird in dieser Arbeit die Maximum Entropie Methode angewandt, um damit eine positive Dichte mit entsprechenden Eigenschaften auszuwählen. Die Maximum Entropie Methode verfolgt dabei die Idee, die statistische Entropie $E(f) = -\int_\Omega f \ln(f) \; d\mu$ unter den Nebenbedingungen $\int_\Omega a_i f \; d\mu = b$ zu maximieren. Mit Hilfe des Lagrange-Formalismus lässt sich zeigen, dass die gesuchte Dichte auf diesem Weg unter den Exponentialdichten zu finden ist, und für ein Set von zu bestimmenden Lagrange-Multiplikatoren $\lambda_i \in \mathbb{R}$ für alle $g \in G$ von folgender Form ist:

$$f_\lambda(g) \;=\; \exp\Big(-1 + \sum_{i=1}^{n} \lambda_i a_i(g) \Big)$$

Diese Lagrange-Multiplikatoren lassen sich zum Beispiel über das zugehörige duale Optimierungsproblem bestimmen, welches folgende Form hat:

$$\begin{aligned} &\text{minimiere} \;\; \Phi : \mathbb{R}^n \longrightarrow \mathbb{R} \;\; \text{mit} \\ &\Phi(\lambda) := \int_\Omega \exp\Big(-1 + \sum_{i=1}^{n} \lambda_i a_i \Big) \; d\mu \;-\; \sum_{i=1}^{n} \lambda_i b_i \end{aligned}$$

Dieses Optimierungsproblem wird in der vorliegenden Arbeit am Beispiel der codf numerisch behandelt. Im Folgenden wird in einer kurzen Übersicht wiedergegeben,

welche Themen bei der numerischen Behandlung dieses Problems von Bedeutung sind und jeweils in einem eigenen Kapitel behandelt werden:

Kapitel 1 - Einleitung
Dieses Kapitel beschreibt das in dieser Zusammenfassung vorgestellte Problem detaillierter und erläutert anhand verschiedener Beispiele die Motivation dieses Vorgehens.

Kapitel 2 - Darstellungstheorie kompakter Gruppen
Da die FOURIER-Entwicklung der codf in engem Zusammenhang mit der Darstellungstheorie kompakter Gruppen steht $\big($in diesem Fall mit $G = SO(3)\big)$, behandelt dieses Kapitel jene Bestandteile dieser Theorie, welche für die Formulierung der FOURIER-Entwicklung der codf von Bedeutung sind. Als wichtigstes Resultat lässt sich hierbei festhalten, dass der Raum $\mathbb{L}^2(G)$ im Falle einer kompakten Gruppe G in eine orthogonale Summe endlich-dimensionaler Unterräume zerfällt, welche in direktem Zusammenhang mit den Äquivalenzklassen der endlich-dimensionalen, irreduziblen Darstellungen von G stehen (Satz von PETER und WEYL). Dieses Resultat liefert schließlich die FOURIER-Entwicklung einer Funktion $f \in \mathbb{L}^2(G)$ nach Darstellungsfunktionen auf G. Dies wird am Ende dieses Kapitels am konkreten Beispiel $G = SO(3)$ behandelt.

Kapitel 3 - Tensoren
Da die codf in dieser Arbeit mit speziellen Darstellungen auf Tensorräumen nach Darstellungsfunktionen auf $SO(3)$ entwickelt wird, gibt dieses Kapitel einen Überblick über Tensorkalkulationen und nutzt die Ergebnisse des Kapitels zur Darstellungstheorie, um mit den Darstellungen von $SO(3)$ auf den Tensorräumen der irreduziblen Tensoren eine Zerlegung des Raumes $\mathbb{L}^2\big(SO(3)\big)$ zu bestimmen.

Kapitel 4 - codf
Dieses Kapitel trägt unter Anwendung der Ergebnisse der vorherigen Kapitel alle notwendigen Informationen zum Aufstellen der codf zusammen und formuliert das Maximum Entropie Momentenproblem der codf in tensorieller Form. Dabei wird unter anderem auch auf die der codf zugrundeliegende Kristallsymmetrie eingegangen, welche bei der Formulierung der codf direkt mit berücksichtigt wird. Ein größerer Abschnitt wird in diesem Kapitel den konkreten Auswertungsmöglichkeiten der Darstellungsfunktionen gewidmet, mit welchen die Approximation der codf numerisch effizient ausgewertet werden soll. Aufgrund des zur Bestimmung der Approximation der codf zu lösenden Optimierungsproblems, bei welchem die Darstellungsfunktionen in jeder Iteration bei jeder durchzuführenden Integration mehrmals ausgewertet werden müssen, ist klar, dass dieser Teil aus numerischen Gesichtspunkten von sehr großer Bedeutung ist. Die Tatsache, dass zusätzlich die Dimension dieses Momentenproblems mit der Berücksichtigung höherer Tensorränge für die Approximation der codf stark zunimmt, liefert einen weiteren Grund, weshalb die Auswertung der Darstellungsfunktionen auf sehr effiziente Art und Weise zu geschehen hat. Hierfür werden schließlich verschiedene Methoden zur Auswertung, wie etwa ein Ansatz

mit Hilfe von Monomen in 9 Variablen $\big($den Einträgen einer orthogonalen Matrix $Q \in SO(3)\big)$ oder ein Ansatz mit Hilfe von Funktionalen aus den Dualräumen der entsprechenden Tensorräume, entwickelt und miteinander verglichen. Für die numerische Umsetzung dieses Optimierungsproblems in MATLAB mussten weit über 100 Unterprogramme mit insgesamt mehreren Tausend Programmzeilen implementiert werden.

Kapitel 5 - Integration über SO(3)
Die in jedem Optimierungsschritt durchzuführende Integration über $SO(3)$ muss numerisch ebenso gut verstanden werden, um insgesamt eine effiziente Berechnung der Approximation der codf gewährleisten zu können. In diesem Zusammenhang wird der adaptive Integrationsalgorithmus DCUHRE zur Bestimmung von Mehrfachintegralen vorgestellt. Der in FORTRAN vorgegebene Programmcode wurde zur einheitlichen Umsetzung in MATLAB portiert. Mit Hilfe der der codf zugrundeliegenden Kristallsymmetrie gelingt es zudem, das Integrationsgebiet $SO(3)$ in sogenannte Elementarregionen zu unterteilen. Dies hat den Vorteil, dass das zu berechnende Integral auf eine solche Elementarregion reduziert werden kann, wodurch eine Vielzahl an Darstellungsauswertungen eingespart werden kann.

Kapitel 6 - Numerische Ergebnisse
In diesem Kapitel werden zum Abschluss die numerischen Ergebnisse zusammengetragen, welche bei der Behandlung des Optimierungsproblems unter Entwicklung einer Heuristik zur Koordination von Adaption und Optimierungsfortschritt bei Verwendung des NEWTON-Verfahrens und des BFGS-Verfahrens erzielt werden konnten.

Kapitel 7 - Weitere Anwendungsbeispiele
Das letzte Kapitel behandelt zwei polynomiale Momentenprobleme, welche ebenfalls mit Hilfe der Maximum Entropie Methode bearbeitet werden. Der Unterschied zum codf-Problem liegt hierbei darin, dass diese Problemstellungen in dem Sinne keine Lösungen besitzen, dass die gesuchten Dichten jeweils zu einem Delta-Maß gehören und somit nicht in exponentieller Form dargestellt werden können. Anhand dieser Probleme soll also getestet werden, inwiefern der Optimierungsalgorithmus auch in solchen Fällen Ergebnisse liefern kann, die diese Sachlage widerspiegeln.

A Ableitungen

A.1 Gradient und Hesse-Matrix der Zielfunktion

Im Folgenden werden der Gradient und die Hesse-Matrix der Zielfunktion (4-10)

$$\Phi(\omega) = \int\limits_{SO(3)} \exp\Big(-1 + \omega_0 + \sum_{k=1}^{n} \langle \omega_k, Q * \mathbb{T}^{r_k} \rangle \Big)\, dQ \;-\; \omega_0 \;-\; \sum_{k=1}^{n} \frac{\langle \omega_k, \mathbb{S}^{r_k} \rangle}{2r_k + 1}$$

der Maximum Entropie Methode aufgeführt. Dabei führen wir für ein beliebiges $Q \in SO(3)$ und ein beliebiges $\omega = [\omega_0; \omega_1; \ldots; \omega_n]$, wobei $\omega_0 \in \mathbb{R}$ und $\omega_k \in \mathcal{J}_{r_k}(\mathbb{R}^3)$ für $k = 1, \ldots, n$ gilt, folgende Abkürzung ein:

$$g(\omega, Q) \;:=\; \exp\Big(-1 + \omega_0 + \sum_{k=1}^{n} \langle \omega_k, Q * \mathbb{T}^{r_k} \rangle \Big)$$

Damit erhalten wir folgenden Gradienten von Φ (die dabei auftauchenden Tensoren sind jeweils als mehrdimensionale Spaltenvektoren aufzufassen):

$$\nabla\Phi(\omega) \;=\; \int\limits_{SO(3)} g(\omega, Q) \begin{pmatrix} 1 \\ Q * \mathbb{T}^{r_1} \\ \vdots \\ Q * \mathbb{T}^{r_n} \end{pmatrix} dQ \;-\; \begin{pmatrix} 1 \\ \frac{1}{2r_1+1}\, \mathbb{S}^{r_1} \\ \vdots \\ \frac{1}{2r_n+1}\, \mathbb{S}^{r_n} \end{pmatrix}$$

Die Nullstelle μ des Gradienten erfüllt somit unmittelbar die Nebenbedingungen (4-8) und (4-9) des Maximum Entropie Momentenproblems der codf. Für die Hesse-Matrix H_Φ von Φ erhalten wir damit folgende Blockmatrix (die in den einzelnen Blöcken auftauchenden Tensoren sind erneut jeweils als mehrdimensionale Spaltenvektoren aufzufassen):

$$H_\Phi(\omega) = \int\limits_{SO(3)} g(\omega, Q) \begin{pmatrix} 1 & (Q * \mathbb{T}^{r_1})^T & \cdots & (Q * \mathbb{T}^{r_n})^T \\ Q * \mathbb{T}^{r_1} & (Q * \mathbb{T}^{r_1})(Q * \mathbb{T}^{r_1})^T & \cdots & (Q * \mathbb{T}^{r_1})(Q * \mathbb{T}^{r_n})^T \\ \vdots & \vdots & \cdots & \vdots \\ Q * \mathbb{T}^{r_n} & (Q * \mathbb{T}^{r_n})(Q * \mathbb{T}^{r_1})^T & \cdots & (Q * \mathbb{T}^{r_n})(Q * \mathbb{T}^{r_n})^T \end{pmatrix} dQ$$

B Sonstiges

B.1 Computer-Informationen

OS	openSUSE 11.4 (64-Bit)
CPU	Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz
RAM	4GB
MATLAB	7.8.0.347 (R2009a)

Tabelle B.1: Informationen über den verwendeten Computer und die für die numerischen Simulationen verwendete Software

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] *User's Guide: Optimization Toolbox*. The MathWorks, 3rd edition.

[2] B.L. Adams, J.P. Boehler, M. Guidi, and E.T. Onat. Group theory and representation of microstructure and mechanical behavior of polycrystals. *J. Mech. Phys. Solids*, 40(4):723–737, 1992.

[3] J. Berntsen, T. Espelid, and A. Genz. An Adaptive Algorithm for the Approximate Calculation of Multiple Integrals. *ACM Transactions on Mathematical Software*, 17, 1991.

[4] J. Berntsen, T. Espelid, and A. Genz. Algorithm 698: DCUHRE: An Adaptive Multidimensional Integration Routine for a Vector of Integrals. *ACM Transactions on Mathematical Software*, 17, 1991.

[5] T. Böhlke. Application of the maximum entropy method in texture analysis. *Computational Materials Science*, 32, 2005.

[6] T. Böhlke. Texture simulation based on tensorial Fourier coefficients. *Comp. Struct.*, 84:1086–1094, 2006.

[7] T. Böhlke, U. Haus, and V. Schulze. Crystallographic texture approximation by quadratic programming. *Acta Materialia*, 54, 2006.

[8] T. Böhlke, K. Jöchen, O. Kraft, D. Löhe, and V. Schulze. Elastic properties of polycrystalline microcomponents. *Mechanics of Materials*, 42:11–23, 2010.

[9] W. Borchardt-Ott. *Kristallographie*. Springer, Heidelberg, 2002.

[10] J. M. Borwein and A. S. Lewis. Partially-finite programming in L_1 and the existence of maximum entropy estimates. *SIAM J. Optimization*, 2:248–267, 1993.

[11] D. Braess. *Finite Elemente*. Springer, Heidelberg, 2000.

[12] T. Bröcker and T. tom Dieck. *Representations of Compact Lie Groups*. Graduate Texts in Mathematics 98, Springer, New York, 1985.

[13] H.-J. Bunge. Zur Darstellung allgemeiner Texturen. *Z. Metallkde.*, 56:872–874, 1965.

[14] H.-J. Bunge. *Texture Analysis in Material Science*. Cuviller Verlag Göttingen, 1993.

[15] P.J. Davis and P. Rabinowitz. *Methods of Numerical Integration*. Academic Press, Computer Science and Applied Mathematics, 2nd edition, 1984.

[16] J. Elstrodt. *Maß- und Integrationstheorie*. Springer, Berlin Heidelberg, 5te Auflage, 2007.

[17] O. Engler and V. Randle. *Introduction to Texture Analysis: Macrotexture, Microtexture and Orientation Mapping*. Springer, 2010.

[18] W. Forst and D. Hoffmann. *Optimization - Theory and Practice*. Springer, 2010.

[19] M. Giaquinta and S. Hildebrandt. *Calculus of Variations I*. Springer, 1996.

[20] M. Guidi, B.L. Adams, and E.T. Onat. Tensorial representation of the orientation distribution function in cubic polycrystals. *Textures Microstruct.*, 19:147–167, 1992.

[21] J. Hansen, J. Pospiech, and K. Lücke. *Tables for texture analysis of cubic crystals*. Springer, Berlin, 1978.

[22] E. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, 1957.

[23] E. Jaynes. Information theory and statistical mechanics ii. *Phys. Rev.*, 108:171–190, 1957.

[24] M. Junk, J. Budday, and T. Böhlke. On the solvability of maximum entropy moment problems in texture analysis. *Mathematical Models and Methods in Applied Sciences*, 22(12), 2012.

[25] C. T. Kelley. *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, 1999.

[26] M. Koecher. *Lineare Algebra und analytische Geometrie*. Springer, Berlin, Heidelberg, New York, 1992.

[27] K. Kopitzki und P. Herzog. *Einführung in die Festkörperphysik*. Vieweg Teubner, Wiesbaden, 2007.

[28] J.M. Lee. *Riemannian Manifolds: An Introduction to Curvature*. Graduate Texts in Mathematics 176, Springer, New York, 1997.

[29] K. Mardia and P. Jupp. *Directional Statistics*. John Wiley and Sons Ltd., Chichester, 2000.

[30] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, Science + Business Media, LLC, 2nd edition, 2006.

[31] H. Schaeben. Diskrete mathematische Methoden zur Berechnung und Interpretation von kristallographischen Orientierungsdichten. *DGM Informationsgesellschaft mbH*, 1994.

[32] A.J. Schwartz, M. Kumar, B.L. Adams, and D.P. Field. *Electron Backscatter Diffraction in Materials Science*. Springer, 2000.

[33] B. Simon. *Representations of Finite and Compact Groups*. Graduate Studies in Mathematics, Volume 10, American Mathematical Society, 1996.

[34] S. Sternberg. *Group theory and physics*. Cambridge University Press, 1994.

[35] P. Van Houtte. The use of a quadratic form for the determination of nonnegative texture functions. *Textures and Microstructures*, 6:1–20, 1983.

[36] P. Van Houtte. A method for the generation of various ghost correction algorithms - the example of the positivity method and the exponential method. *Textures and Microstructures*, 13:199–212, 1991.

[37] N. Wu. *The Maximum Entropy Method*. Springer Series in Information Sciences, Springer, Berlin, 1997.

[38] A. Yershova, S. Jain, S.M. LaValle, and J.C. Mitchell. Generating Uniform Incremental Grids on SO(3) Using the Hopf Fibration. *International Journal of Robotics Research*, 29(7), 2010.

[39] Q.-S. Zheng and Y.-B. Fu. Orientation distribution functions for microstructures of heterogeneous materials: I Directional distribution functions and irreducible tensors. *Appl. Math. Mech.*, 22(8):865–884, 2001.

[40] Q.-S. Zheng and Y.-B. Fu. Orientation distribution functions for microstructures of heterogeneous materials: II Crystal distribution functions and irreducible tensors restricted by various material symmetries. *Appl. Math. Mech.*, 22(8):885–903, 2001.

[41] D. Zwillinger. *Handbook of Integration*. Jones and Bartlett Publishers, Boston, London, 1992.