Fakultät für Informatik
der
Technischen Universität München

Ein rekursiver Algorithmus zur adaptiven numerischen Quadratur mehrdimensionaler Funktionen

Thomas Bonk

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. Heinz-Gerd Hegering

Prüfer der Dissertation:

1. Univ.-Prof. Dr. Christoph Zenger

2. Univ.-Prof. Dr. Arndt Bode

Die Dissertation wurde am 03.11.1994 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 24.03.1995 angenommen.

Ein rekursiver Algorithmus zur adaptiven numerischen Quadratur mehrdimensionaler Funktionen

Zusammenfassung

Die Archimedische Ausschöpfung und das Cavalierische Prinzip führen in natürlicher Weise zu einem auf dünnen Gittern basierenden, adaptiven, mehrdimensionalen Quadraturverfahren. Im Unterschied zu den klassischen Verfahren der numerischen Quadratur wächst bei diesem Verfahren, ähnlich wie bei Monte-Carlo-Verfahren und Verfahren der Zahlentheorie, der Aufwand nicht exponentiell mit der Dimension des Raumes. Die multivariate Verallgemeinerung der Romberg-Quadratur hingegen führt zu einem nicht-adaptiven Verfahren höherer Ordnung auf regulären dünnen Gittern. Hierfür wird eine Fehlerabschätzung angegeben. Schließlich wird ein sich an der rekursiven Struktur dünner Gitter orientierendes Verfahren vorgestellt, das die Vorteile der beiden obigen, nämlich Gebietsadaptivität und höhere Ordnung, vereinigt. Die Eigenschaften des Verfahrens werden anhand numerischer Experimente demonstriert und mit Ergebnissen bekannter Verfahren verglichen.

Meiner lieben Frau
und unseren Kindern ...

Inhalt

1 Einleitung **2**

2 Die Archimedische Quadratur **6**

2.1 Das Verfahren im eindimensionalen Fall 6
2.2 Das Verfahren im zweidimensionalen Fall 7
2.3 Das Verfahren im mehrdimensionalen Fall 9
2.4 Eigenschaften des Verfahrens 12

3 Ein nicht-adaptives Quadraturverfahren höherer Ordnung **20**

3.1 Die Romberg-Quadratur 20
3.2 Numerische Quadratur höherer Ordnung im mehrdimensionalen Fall 22
3.3 Abschätzung des Verfahrensfehlers 27
3.3.1 Eine kompakte Notation 27
3.3.2 Die Euler-Maclaurinsche Summenformel für den mehrdimensionalen Fall . 28
3.3.3 Auslöschung von Fehlergliedern 36
3.3.4 Fehlerabschätzung des Restgliedes 39
3.4 Implementation des Verfahrens 58
3.4.1 Datenstruktur 59
3.4.2 Realisierung 60
3.4.3 Verfahrensvarianten 63
3.5 Numerische Experimente 64

4 Adaptive Quadratur höherer Ordnung **68**

4.1 Datenstruktur 68
4.2 Kompilation 72
4.3 Aufbau der Datenstruktur 75
4.4 Extrapolation 76
4.5 Festlegung der Halbierungsrichtungen 81
4.6 Adaptivität 82
4.6.1 Die fehlerorientierte Adaptivitätssteuerung 82
4.6.2 Die kostenorientierte Adaptivitätssteuerung 84
4.7 Richtungsabhängige Begrenzung der Maximalverfeinerung 85
4.8 Kosten der Datenstruktur 89
4.9 Numerische Experimente 96

5 Vergleich mit Konkurrenzverfahren **107**

5.1 Die Routine `d01fcf` 107
5.2 Numerische Experimente 109

6 Schlußbemerkungen **115**

Literatur **116**

1. Einleitung. Die numerische Quadratur gehört zu den ältesten Aufgabenstellungen der Mathematik. So befaßte sich bereits Archimedes mit diesem Problem, als er die Kreisfläche näherungsweise berechnete, um obere und untere Schranken für die Zahl π angeben zu können. Später waren es Newton, Gauß und Maxwell, die wesentliche Beiträge zur numerischen Quadratur leisteten.

Heute läuft die Lösung vieler Probleme aus verschiedensten wissenschaftlichen Disziplinen auf die numerische Berechnung von Integralen hinaus. In diesem Zusammenhang seien nur die Quantenfeldtheorie (Feynman-Integrale) und die Statistik (multivariate Normalverteilung) erwähnt. Dabei treten immer häufiger Quadraturprobleme mit mehreren Veränderlichen auf, die an die verwendeten Integrationsroutinen hohe Anforderungen stellen. Um so verständlicher ist es, daß gerade die Entwicklung leistungsfähiger Verfahren zur numerischen Quadratur im mehrdimensionalen Raum einen breiten Raum in den wissenschaftlichen Anstrengungen der vergangenen Zeit eingenommen hat und auch in Zukunft einnehmen wird.

Für die numerische Quadratur im mehrdimensionalen Raum findet der Anwender heute eine breite Unterstützung vor. So sind zu diesem Thema zahlreiche Bücher erschienen ([DR84], [Eng80], [Str71]). Das Buch von Davis und Rabinowitz beispielsweise enthält ein Kapitel, das die Ergebnisse mehrdimensionaler Quadraturverfahren, die bis 1984 bekannt waren, überblicksmäßig zusammenfaßt. Des weiteren finden sich viele Algorithmen in mathematisch-numerischen Programmbibliotheken wie IMSL [IMS91] und NAG [NAG91] wieder, wobei letztere die reichhaltigste Auswahl an mehrdimensionalen Quadraturroutinen aufzuweisen hat.

In dieser Arbeit werden mehrere Verfahren zur näherungsweisen Berechnung des Wertes $I(f) \in \mathbb{R}$ eines bestimmten Integrals

$$I(f) \;=\; \int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_d}^{b_d} f(x_1, x_2, \ldots, x_d)\; dx_d \ldots dx_2\, dx_1$$

für einen gegebenen Integranden

$$f : [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d] \subseteq \mathbb{R}^d \to \mathbb{R}$$

vorgestellt. Dabei ist vorausgesetzt, daß die Riemann-integrierbare Funktion f zur maschinellen Auswertung in Form eines Unterprogramms vorliegt.

Ein Blick in Datenbanken für numerische Software wie GAMS (Guide to Available Mathematical Software) zeigt, daß hierzu schon etwa ein Dutzend Verfahren allgemein verfügbar sind. Diese lassen sich in fünf Klassen unterteilen. Zum einen sind die Verfahren zu nennen, die sich als kartesisches Produkt eindimensionaler Verfahren, meist der Gauß-Quadratur, ergeben (beispielsweise `NAG/d01fbf` und `IMSL/qand`). Sie stellen sozusagen die einfachste Verallgemeinerung vom eindimensionalen auf den mehrdimensionalen Fall dar. Nachteilig an diesen Methoden ist, daß der Aufwand, gemessen in Integrandenauswertungen, exponentiell mit der Dimension ansteigt, weshalb diese Verfahren oft schon für Probleme mit $d > 10$ ungeeignet sind.

Eine weitere Klasse stellen die Monte-Carlo-Verfahren dar. Sie werten den Integranden an n gleichmäßig über das Integrationsgebiet verteilten Stützstellen aus und bilden darüber den Mittelwert $\bar{f}$. Das *starke Gesetz der großen Zahlen* besagt, daß $\bar{f}$ für $n \to \infty$ mit der Wahrscheinlichkeit 1 gegen $I(f)$ konvergiert, wenn als Integrationsgebiet der Einheitswürfel gewählt wird. Da der Aufwand nicht von der Dimension des Integranden abhängt, kommt den Monte-Carlo-Verfahren eine besondere Bedeutung bei hochdimensionalen und krummlinig berandeten Problemen zu, wo viele andere Verfahren keine brauchbaren Ergebnisse mehr liefern. Allerdings sind Näherungen hoher Genauigkeit mit diesen Verfahren aufgrund der geringen Konvergenzrate von $O(h^{\frac{1}{2}})$ praktisch nicht erreichbar. Eine adaptive Variante aus dieser Klasse von Verfahren stellt die Routine `NAG/d01gbf` ([Lau71]) dar.

Sehr verwandt mit den Monte-Carlo-Methoden sind die zahlentheoretischen Verfahren (beispielsweise `NAG/d01gcf`, [Kor63]). Anstatt statistisch gleichmäßig verteilte Stützstellen zu verwenden, basieren sie auf determinierten Punktfolgen mit kleiner Diskrepanz. Optimale Folgen hierfür sind im mehrdimensionalen Fall nicht bekannt, allerdings liegt die Vermutung nahe, daß der Verfahrensfehler in Abhängigkeit von n, der Anzahl der verwendeten Punkte, mit $O(n^{-1} \cdot (\log n)^d)$ abfällt.

Aus der Klasse der Transformationsverfahren ist die Methode von *Sag* und *Szekeres* weit verbreitet (`NAG/d01fdf`,[SS64]). Hierbei wird das Integral derart transformiert, daß der Integrand und alle seine Ableitungen am Rand verschwinden. Für die numerische Berechnung des Integrals wird eine Produkt-Trapezregel verwendet.

Als letzte Klasse sind noch die global adaptiven multivariaten Verfahren zu erwähnen (beispielsweise `NAG/d01fcf` [GM80] oder `ADAPT` [vDdR76]). Sie basieren auf Gauß-artigen Quadraturformeln sowie einer dynamischen Gebietszerlegungsstrategie, die sich den lokalen Ungleichmäßigkeiten des Integranden anpaßt. Die Ordnung dieser Verfahren hängt von der zugrunde gelegten Quadraturregel ab.

Mehrdimensionale Verfahren mit konvergenzbeschleunigenden Maßnahmen wie Extrapolation oder ε-Algorithmus sind zwar untersucht worden (beispielsweise [Gen74] und [Gen75]), haben aber nicht den Weg in kommerziell verfügbare Programmbibliotheken gefunden.

Aus der Komplexitätstheorie ist bekannt, daß der Aufwand n zur Lösung eines Quadraturproblems bei einer geforderten Genauigkeit ε exponentiell mit der Dimension ansteigt, was in vielen Fällen dazu führt, daß das Problem praktisch nicht mehr berechenbar ist. Andererseits können die Kosten gemildert werden, wenn f glatt ist. Nach [TWW88] gilt nämlich:

$$n \;=\; O\left(\varepsilon^{-\frac{d}{r}}\right),$$

wobei r angibt, wie oft f stetig differenzierbar ist. Aufgrund dieser Eigenschaft ist es bei der Entwicklung von Algorithmen zur numerischen Quadratur mehrdimensionaler Funktionen erstrebenswert, nicht nur Verfeinerungsadaptivität vorzusehen, sondern auch Ordnungsadaptivität, um eine vorhandene Glattheit der Integranden nutzen zu können ([Bon93]). Keines der zuvor vorgestellten Verfahren weist beide Adaptivitätsstrategien auf. An diesem Punkt soll die vorliegende Arbeit anknüpfen.

Eine wesentliche Rolle werden hierbei *dünne Gitter* spielen, die bereits in [Zen90] vorgestellt wurden. Sie besitzen im Gegensatz zu vollen Gittern den Vorteil, daß die

Anzahl der Gitterpunkte bei gegebener Randfeinheit h nicht mehr mit $O(h^{-d})$ also exponentiell in der Dimension ansteigt, sondern nur noch mit $O(h^{-1} \cdot (\mathrm{ld}(h^{-1}))^{d-1})$. Die Approximationsgenauigkeit bei stückweise d-linearen Interpolanten liegt mit $O(h^2 \cdot (\mathrm{ld}(h^{-1}))^{d-1})$ nur geringfügig schlechter als $O(h^2)$, dem Fehler bei vollen Gittern. Zur rechnerinternen Darstellung und Verarbeitung von dünnen Gittern treten häufig rekursive Daten- und Programmstrukturen auf, was auch der Grund dafür sein mag, daß sie in der von FORTRAN geprägten Computer-Numerik noch keine breite Akzeptanz gefunden haben. Sämtliche Algorithmen wurden in der Programmiersprache C implementiert, in der Rekursion bequem ausgedrückt werden kann.

In Kapitel 2 stellen ich ein Verfahren vor, das auf dem Archimedischen Ausschöpfungsprinzip beruht und im eindimensionalen Fall in eine adaptive Variante der Trapezregel übergeht. Für die Verallgemeinerung des Verfahrens auf mehrdimensionale Funktionen wird das Cavalierische Prinzip verwendet. Die mathematische Spezifikation des Verfahrens, die sich auf diese Weise ergibt, kann unmittelbar in ein rekursives Programm umgesetzt werden ([Zen93]). Die Kombination der beiden Prinzipien führt in natürlicher Weise zu dünnen Gittern und zur Approximation von Funktionen durch d-lineare Interpolanten mittels hierarchischer Basen. Anhand von Beispielen wird gezeigt, daß ein zyklischer Tausch der Halbierungsrichtung im Gegensatz zu fortgesetzter Halbierung entlang der gleichen Richtung zu einer gleichmäßigeren Abtastung des Integranden und mitunter zu besseren Ergebnissen führt.

Kapitel 3 beschreibt ein Quadraturverfahren höherer Ordnung. Hierbei wird die Eigenschaft ausgenutzt, daß sich reguläre dünne Gitter aus der Überlagerung voller Gitter unterschiedlicher Maschenweiten entlang der Koordinatenachsen ergeben. Die den Teilgittern zugeordneten Mittelpunktsummen werden nach einem Extrapolationsverfahren miteinander verrechnet. Man erhält eine Menge von Näherung gemischter Ordnungen für $I(f)$, die mit Hilfe der in [GSZ90] vorgestellten Kombinationstechnik zum Endresultat verrechnet werden. Für den Verfahrensfehler wird eine Abschätzung hergeleitet. Dazu wird eine mehrdimensional verallgemeinerte Form der Euler-Maclaurinschen-Summenformel vorgestellt. Es wird gezeigt, daß das Verfahren Fehlerglieder bestimmter Ordnungen auslöscht und daß der von den Restgliedern herrührende Fehler beschränkt ist, sofern der Integrand hinreichend glatt ist. Das Verfahren kann als eine multivariate Verallgemeinerung der Romberg-Quadratur angesehen werden. Ein hierzu äquivalentes Verfahren wurde bereits in [Gen74] beschrieben. Im Gegensatz dazu erlaubt der hier vorgestellte Algorithmus wesentlich mehr Varianten.

Aufbauend auf den gewonnenen Erkenntnissen wird in Kapitel 4 ein drittes Verfahren vorgestellt, das sowohl Verfeinerungsadaptivität als auch Ordnungsadaptivität aufweist. Hierzu wird das Integrationsgebiet durch reguläre dünne Gitter zunehmender Gitterfeinheit abgetastet. Die Gitterpunkte werden in einer Baumdatenstruktur organisiert, die es erlaubt, auch für Teilintegrale des betrachteten Problems schrittweise Näherungen zunehmender Ordnung zu produzieren. Teilgebiete, in denen schon Näherungen hinreichender Genauigkeit errechnet wurden, werden aus dem weiteren Verfeinerungsprozeß herausgenommen. Das Verfahren terminiert, sobald keine Verfeinerung mehr möglich ist. Es werden zwei Adaptivitätsstrategien vorgestellt, von denen die erste darauf abzielt, die vorgegebene Fehlerschranke gleichmäßig über alle Teilintegrale zu verteilen, die zweite hingegen die Einhaltung eines vorgegebenen Kosten-Nutzen-Verhältnisses überwacht. In der praktischen Anwendung des Verfahrens traten noch einige Schwächen zutage, die durch Modifikationen beseitigt werden konnten. So wurden Maßnahmen ergriffen, um die Maximalverfeinerung richtungsabhängig zu begrenzen sowie die Halbierungsrichtung der Gebietszerlegungsstrategie angepaßt auf das Integrationsproblem zu wählen. Es wird gezeigt, daß der Aufwand zur rechnerinternen Repräsentation des Baumes nicht wesentlich ansteigt, wenn die für die Extrapolation benötigten Datenstrukturen für jeden Knoten vorgesehen werden.

Die numerischen Eigenschaften aller drei Verfahren werden jeweils im Anschluß an ihre Beschreibung durch Beispielintegranden demonstriert, wobei die acht Varianten des letzten Verfahrens bezüglich des Verhaltens bei Integranden mit speziellen Glattheitseigenschaften eingehender untersucht werden.

In Kapitel 5 wird schließlich das zuletzt beschriebene Verfahren mit allgemein verfügbaren Konkurrenzverfahren verglichen. Alle Berechnungen wurden auf einer RISC-Workstation des Typs HP9000/755 der Firma Hewlett-Packard durchgeführt, deren Hauptspeicher auf 128 MByte ausgebaut war. Die Konfiguration des Betriebssystems erlaubte die Nutzung eines virtuellen Datenspeichers von 256 MByte.

An dieser Stelle möchte ich mich bei meinem Kollegen Dr. Ulrich Rüde für die Korrektur des Manuskripts dieser Arbeit bedanken.

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr. Christoph Zenger für die Überlassung des Themas sowie die Betreuung meiner Arbeit.

2. Die Archimedische Quadratur. Der erste Teil dieser Arbeit stellt ein Quadraturverfahren vor, das sich aus der mehrdimensionalen Verallgemeinerung der Archimedischen Ausschöpfung ergibt. Zunächst wird eine rekursive Rechenvorschrift hergeleitet. Danach werden die numerischen Eigenschaften des Verfahrens anhand ausgewählter Beispiele untersucht.

2.1. Das Verfahren im eindimensionalen Fall. Wir wollen die Fläche F_1 berechnen, die eine Funktion f mit der Abszisse und den linken und rechten Begrenzungen a und b einschließt:

$$F_1\left(f\left(\mathbf{x}\right), a, b\right) = \int_a^b f\left(\mathbf{x}\right) d\mathbf{x}.$$

(aus Gründen der Lesbarkeit werden ungebundene Variablen für die Beschreibung des Verfahrens fettgedruckt dargestellt). Wir zerlegen das Integral, wie in Abbildung 1 (links) gezeigt, in zwei Teilprobleme, nämlich ein Trapez der Fläche

$$T_1 = \tfrac{1}{2}\left(f\left(a\right) + f\left(b\right)\right) \cdot \left(b - a\right) \tag{1}$$

und ein Funktionssegment S_1, das sich über das Intervall $x \in [a, b]$ erstreckt und dessen Fläche mit $S_1\left(f\left(\mathbf{x}\right), a, b\right)$ bezeichnet werden soll. Wir erhalten die Flächenäquivalenz:

$$F_1\left(f\left(\mathbf{x}\right), a, b\right) = \tfrac{1}{2}\left(f\left(a\right) + f\left(b\right)\right) \cdot \left(b - a\right) + S_1\left(f\left(\mathbf{x}\right), a, b\right). \tag{2}$$

Um $S_1\left(f(\mathbf{x}), a, b\right)$ zu berechnen, nähern wir das Segment durch ein einbeschriebenes Dreieck an (Abbildung 1 (rechts)). Dessen Fläche ergibt sich zu

$$D_1\left(f\left(\mathbf{x}\right), a, b\right) = \left(f\left(\frac{a+b}{2}\right) - \tfrac{1}{2}\left(f\left(a\right) + f\left(b\right)\right)\right) \cdot \frac{b-a}{2}. \tag{3}$$

Die verbliebenen Teilflächen S_1' und S_1'' stellen wiederum Funktionssegmente dar, so daß sich die rekursive Beziehung

$$\begin{aligned} S_1\left(f\left(\mathbf{x}\right), a, b\right) &= g_1\left(f\left(\mathbf{x}\right), a, b\right) \cdot \frac{b-a}{2} \\ &+ S_1\left(f\left(\mathbf{x}\right), a, \frac{a+b}{2}\right) + S_1\left(f\left(\mathbf{x}\right), \frac{a+b}{2}, b\right) \end{aligned} \tag{4}$$

mit

$$g_1\left(f\left(\mathbf{x}\right), a, b\right) = f\left(\frac{a+b}{2}\right) - \tfrac{1}{2}\left(f\left(a\right) + f\left(b\right)\right)$$

ergibt. $g_1\left(f\left(\mathbf{x}\right), a, b\right)$, die Höhe des einbeschriebenen Dreiecks im Punkt $x = \frac{a+b}{2}$, bezeichnet man auch als *hierarchischen Überschuß* der Funktion $f\left(\mathbf{x}\right)$ auf dem Träger $[a, b]$. Den Punkt $x = \frac{a+b}{2}$ nennt man auch *Basispunkt*.

Die vorgestellte Flächenzerlegung läßt sich ebenfalls auf S_1' und S_1'' sowie alle weiteren Segmente anwenden, die infolge von (4) entstehen. Das Funktionssegment S_1 wird

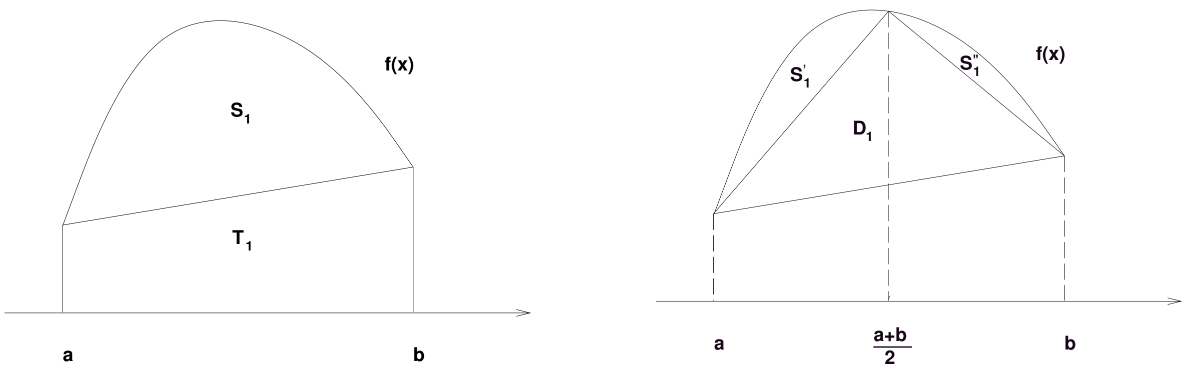

ABB. 1. *Die Archimedische Ausschöpfung.*

somit durch einbeschriebene Dreiecke ausgefüllt. Begrenzt man die rekursive Verschachtelung von (4) auf $m > 0$ Inkarnationen, so wird der Integrand durch einen Polygonzug mit $2^m + 1$ äquidistanten Stützstellen angenähert. Das Integral dieses Interpolanten ergibt die Trapezsumme über der Funktion f mit Schrittweite $(b-a) \cdot 2^{-m}$.

Die Gleichungen (2) und (4) können zur Implementation eines Quadraturverfahrens verwendet werden. Lediglich für die in (4) enthaltene Rekursion muß noch eine Abbruchalternative gefunden werden. Wir wollen die Rekursion abbrechen, sobald der Betrag des hierarchischen Überschusses kleiner als eine vorgegebene Schranke ε_{abs} wird. Zusätzlich zu (4) definieren wir also:

$$S_1\left(f\left(\mathbf{x}\right), a, b\right) \;=\; g_1\left(f\left(\mathbf{x}\right), a, b\right) \cdot \frac{b-a}{2} \quad \text{falls} \quad \left|g_1\left(f\left(\mathbf{x}\right), a, b\right)\right| \leq \varepsilon_{\text{abs}}. \tag{5}$$

Mit den Gleichungen (2), (4) und (5) erhalten wir ein Quadraturverfahren, das als eine adaptive Variante der Trapezregel angesehen werden kann. Das hier angewandte Prinzip, Flächen unbekannter Größe durch Regelflächen anzunähern, ist schon seit der Antike bekannt und wird nach ihrem Erfinder als *Archimedische Ausschöpfung* bezeichnet. Erste Anwendungen waren die exakte Berechnung der Fläche eines Parabelsegments sowie die näherungsweise Bestimmung der Kreiszahl π.

2.2. Das Verfahren im zweidimensionalen Fall. Das *Cavalierische Prinzip* stellt eine gängige Methode dar, die Dimension von Problemen zu reduzieren, um sie mit einfacheren Verfahren zu lösen. So kann etwa die Berechnung von Volumen auf die von Flächen zurückgeführt werden, indem der Körper gleichmäßig in ,,dünne Scheiben“ zerlegt wird, deren Flächen summiert und mit der Dicke der Scheiben multipliziert werden.

Mit Hilfe des Cavalierischen Prinzips soll nun das vorgestellte Verfahren auf den zweidimensionalen Fall erweitert werden.

Wir wollen das Volumen berechnen, das eine Funktion $f\left(\mathbf{x_1}, \mathbf{x_2}\right)$ mit der

x_1-x_2-Ebene auf dem Gebiet $[a_1, b_1] \times [a_2, b_2]$ einschließt:

$$F_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right) \;=\; \int_{a_1}^{b_1} \int_{a_2}^{b_2} f\left(\mathbf{x_1}, \mathbf{x_2}\right)\; d\mathbf{x_2}\; d\mathbf{x_1}.$$

Wir betrachten das Volumenstück als aus unendlich vielen Scheiben zusammengesetzt, die durch Bindung der Variablen x_1 des Integranden entstehen. Abbildung 2 (links) zeigt drei solche Schnittflächen für $x_1 \in \{a_1, \frac{a_1+b_1}{2}, b_1\}$, die nun in Trapez und Segment unterteilt werden. Die Gesamtheit aller Trapeze, deren Fläche sich zu

$$T_2\left(\mathbf{x_1}\right) \;=\; \tfrac{1}{2}\left(f\left(\mathbf{x_1}, a_2\right) + f\left(\mathbf{x_1}, b_2\right)\right) \cdot \left(b_2 - a_2\right)$$

ergibt, spannen über $x_1 \in [a_1, b_1]$ einen Körper mit Volumen $F_1\left(T_2\left(\mathbf{x_1}\right), a_1, b_1\right)$ auf. Ebenso ergeben die Segmentflächen einen Körper, dessen Volumen wir mit

$$S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right)$$

bezeichnen wollen. Der erste Zerlegungsschritt kann somit durch

$$F_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right) \;=\; F_1\left(T_2\left(\mathbf{x_1}\right), a_1, b_1\right) + S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right) \tag{6}$$

beschrieben werden.

Im zweiten Schritt werden die verbliebenen Segmente durch einbeschriebene Dreiecke der Fläche

$$D_2\left(\mathbf{x_1}\right) \;=\; \left(f\left(\mathbf{x_1}, \frac{a_2+b_2}{2}\right) - \tfrac{1}{2}\left(f\left(\mathbf{x_1}, a_2\right) + f\left(\mathbf{x_1}, b_2\right)\right)\right) \cdot \frac{b_2 - a_2}{2}$$

angenähert (Abbildung 2 [rechts]). Integriert über den Bereich $x_1 \in [a_1, b_1]$ spannen sie einen Körper vom Volumen $F_1\left(D_2\left(\mathbf{x_1}\right), a_1, b_1\right)$ auf. Die Volumenanteile der neu entstandenen Segmentkörper ergeben $S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, \frac{a_2+b_2}{2}\right)$ und $S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, \frac{a_2+b_2}{2}, b_1, b_2\right)$. Insgesamt gilt:

$$\begin{aligned} S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right) \;=\;& F_1\left(D_2\left(\mathbf{x_1}\right), a_1, b_1\right) \\ &+\; S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, \frac{a_2+b_2}{2}\right) \\ &+\; S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, \frac{a_2+b_2}{2}, b_1, b_2\right). \end{aligned} \tag{7}$$

Dieser Schritt kann rekursiv auf die noch nicht berechneten Volumenanteile angewendet werden.

Die aus algorithmischer Sicht notwendige Terminierungsalternative für (7) kann nach den gleichen Gesichtspunkten wie im eindimensionalen Fall festgelegt werden. Wir definieren:

$$\begin{gathered} S_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right) = F_1\left(D_2\left(\mathbf{x_1}\right), a_1, b_1\right) \\ \text{falls} \quad \left|g_2\left(f\left(\mathbf{x_1}, \mathbf{x_2}\right), a_1, a_2, b_1, b_2\right)\right| \le \varepsilon_{\text{abs}}, \end{gathered} \tag{8}$$

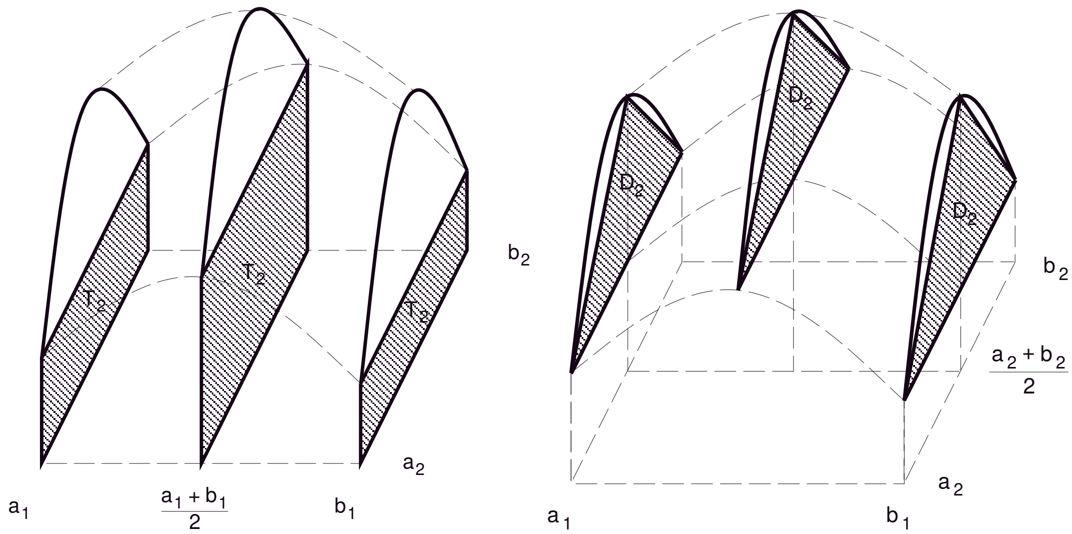

ABB. 2. *Anwendung des Cavalierischen Prinzips.*

wobei sich $g_2(f(\mathbf{x_1}, \mathbf{x_2}), a_1, a_2, b_1, b_2)$, der zweidimensionale hierarchische Überschuß, durch kartesische Produktbildung wie folgt ergibt:

$$\begin{aligned} g_2(f(\mathbf{x_1}, \mathbf{x_2}), a_1, a_2, b_1, b_2) &= g_1\left(f\left(\mathbf{x_1}, \frac{a_2 + b_2}{2}\right), a_1, b_1\right) \\ &\quad -\tfrac{1}{2}\left(g_1(f(\mathbf{x_1}, a_2), a_1, b_1) + g_1(f(\mathbf{x_1}, b_2), a_1, b_1)\right). \end{aligned}$$

Mit den Gleichungen (6) bis (8) wird die Quadratur zweidimensionaler Funktionen auf den eindimensionalen Fall zurückgeführt. Für die Implementation ist zu beachten, daß hier Funktionen als Argumente zu übergeben sind, da F_1 einmal mit $F_1(T_2(\mathbf{x_1}), a_1, b_1)$, ein andermal mit $F_1(D_2(\mathbf{x_1}), a_1, b_1)$ aufgerufen wird.

2.3. Das Verfahren im mehrdimensionalen Fall. Das Cavalierische Prinzip kann nun rekursiv fortgesetzt werden, um ein Verfahren zur Berechnung von

$$F_d(f(\mathbf{x_1}, \ldots, \mathbf{x_d}), a_1, \ldots, a_d, b_1, \ldots, b_d) = \int_{a_1}^{b_1} \cdots \int_{a_d}^{b_d} f(\mathbf{x_1}, \ldots, \mathbf{x_d}) \; d\mathbf{x_d} \ldots \; d\mathbf{x_1}$$

zu erhalten.

Wir nehmen zunächst an, daß alle Variablen des Integranden bis auf x_d an einen Wert gebunden sind. Hierdurch erhalten wir ein eindimensionales Quadraturproblem, das durch die bekannte Zerlegung in Trapez und Funktionssegment gelöst wird. Die Trapezflächen, die

$$T_d(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}) = \tfrac{1}{2}(f(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}, a_d) + f(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}, b_d)) \cdot (b_d - a_d)$$

ergeben, spannen über $x_i \in [a_i, b_i], i = 1, \ldots, d-1$ ein d-dimensionales Hypervolumen auf, das durch das $(d-1)$-dimensionale Integral

$$F_{d-1}\left(T_d\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}\right), a_1, \ldots, a_{d-1}, b_1, \ldots, b_{d-1}\right)$$

berechnet wird. Ebenso ergeben die Funktionssegmente ein Hypervolumen, das wir mit

$$S_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_d, b_1, \ldots, b_d\right)$$

bezeichnen. Wir erhalten die auf beliebige Dimensionen d verallgemeinerte Form der Gleichungen (2) und (6):

$$\begin{aligned} & F_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_d, b_1, \ldots, b_d\right) = \\ (9) \qquad & \quad F_{d-1}\left(T_d\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}\right), a_1, \ldots, a_{d-1}, b_1, \ldots, b_{d-1}\right) \\ & + \; S_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_d, b_1, \ldots, b_d\right) \quad \text{für} \quad d > 0 \end{aligned}$$

sowie

$$F_0\left(T_1\right) = T_1.$$

Die nach diesem Schritt verbliebenen Funktionssegmente werden durch einbeschriebene Dreiecke der Fläche

$$\begin{aligned} D_d\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}\right) = & \left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}, \frac{a_d + b_d}{2}\right) \right. \\ & \left. -\tfrac{1}{2}\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}, a_d\right) + f\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}, b_d\right)\right) \right) \cdot \frac{b_d - a_d}{2} \end{aligned}$$

angenähert. Zusammen mit den beiden, sich in diesem Zerlegungsprozeß neu ergebenden Funktionssegmenten werden drei $(d-1)$-dimensionale Hypervolumina aufgespannt, die formal durch Integration über die Variablen $x_1, \ldots, x_{d-1}$ berechnet werden können. Die Verallgemeinerung von (4) und (7) lautet somit:

$$\begin{aligned} & S_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1 \ldots, a_d, b_1 \ldots, b_d\right) = \\ & \quad F_{d-1}\left(D_d\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}\right), a_1, \ldots, a_{d-1}, b_1, \ldots, b_{d-1}\right) \\ (10) \qquad & + \; S_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_d, b_1, \ldots, b_{d-1}, \frac{a_d + b_d}{2}\right) \\ & + \; S_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_{d-1}, \frac{a_d + b_d}{2}, b_1, \ldots, b_d\right) \quad \text{für} \quad d > 0. \end{aligned}$$

Als Terminierungsalternative für (10) legen wir fest:

$$\begin{aligned} & S_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_d, b_1, \ldots, b_d\right) = \\ (11) \qquad & \quad F_{d-1}\left(D_d\left(\mathbf{x_1}, \ldots, \mathbf{x_{d-1}}\right), a_1, \ldots, a_{d-1}, b_1, \ldots, b_{d-1}\right) \\ & \quad \text{falls} \quad \left|g_d\left(f\left(\mathbf{x_1}, \ldots, \mathbf{x_d}\right), a_1, \ldots, a_d, b_1, \ldots, b_d\right)\right| \le \varepsilon_{\text{abs}}, \end{aligned}$$

wobei sich der hierarchische Überschuß berechnet zu

$$\begin{aligned}
&g_d\left(f\left(\mathbf{x_1},\ldots,\mathbf{x_d}\right),a_1,\ldots,a_d,b_1,\ldots,b_d\right) = \\
&\qquad g_{d-1}\left(f\left(\mathbf{x_1},\ldots,\mathbf{x_{d-1}},\frac{a_d+b_d}{2}\right),a_1,\ldots,a_{d-1},b_1,\ldots,b_{d-1}\right) - \\
&\tfrac{1}{2}(\quad g_{d-1}\left(f\left(\mathbf{x_1},\ldots,\mathbf{x_{d-1}},a_d\right),a_1,\ldots,a_{d-1},b_1,\ldots,b_{d-1}\right) + \\
&\qquad g_{d-1}\left(f\left(\mathbf{x_1},\ldots,\mathbf{x_{d-1}},b_d\right),a_1,\ldots,a_{d-1},b_1,\ldots,b_{d-1}\right)) \quad \text{für} \quad d>0
\end{aligned}$$

und

$$g_0\left(f\right) \;=\; f.$$

Durch die Beziehungen (9) bis (11) ist das Quadraturverfahren hinreichend spezifiziert, so daß eine Implementation in einer Programmiersprache vorgenommen werden kann. Wir wollen diesen Algorithmus mit `arch` bezeichnen. Neben Integranden und Integrationsgebiet benötigt `arch` eine Genauigkeitsschranke ε_{abs}, die den maximal erlaubten hierarchischen Überschuß angibt, für den (10) in die nicht-rekursive Alternative (11) übergeht.

Es sei noch auf einige Eigenschaften des Verfahrens hingewiesen, die sich bereits aus der Darstellung des Algorithmus gemäß den obigen Gleichungen erkennen lassen. Zum einen beachte man, daß sich F_j und S_j, $j = 1, 2, \ldots, d$ auf F_{j-1} stützen. Hierin drückt sich das Cavalierische Prinzip aus. F_{j-1} wird von F_j und S_j mit verschiedenen Integranden im Argument aufgerufen, was in einigen Programmiersprachen unter Umständen nicht direkt umgesetzt werden kann. Des weiteren sei darauf hingewiesen, daß die Gleichungen bereits eine Gebietszerlegungsstrategie festlegen. Mit jeder erneuten Anwendung von (10) wird das betrachtete Gebiet entlang der Koordinatenrichtung x_d halbiert. Schließlich ist noch zu erwähnen, daß `arch` Mehrfachauswertungen von Gitterpunkten ausführt. Im Fall $d = 1$ sind von den drei Stützstellen, die zur Berechnung der Dreiecksfläche nach (3) notwendig sind, die beiden äußeren zuvor ermittelt worden. Diese Ineffizienz kann im eindimensionalen Fall durch Parametrisierung der Integrationsgrenzen behoben werden. Im mehrdimensionalen Fall müssen für die Speicherung der Stützstellen geeignete Datenstrukturen vorgesehen werden. Läßt man Mehrfachauswertungen zu, so gilt folgender Zusammenhang zwischen der Zahl von Gitterpunkten n und der Zahl von Funktionsauswertungen n_A:

$$n_A \;\approx\; 3^d \cdot n.$$

Im eindimensionalen Fall sahen wir, daß `arch` den Integranden durch stückweise lineare Funktionen approximiert. Mit Hilfe der hierarchischen Basisdarstellung von Funktionen kann gezeigt werden, daß der Integrand im mehrdimensionalen Fall durch einen stückweise d-linearen Interpolanten angenähert wird, dessen exaktes Integral die Näherung des gesuchten Wertes $I\left(f\right)$ bildet. Ähnlich wie die Newton-Cotes-Formeln ist `arch` somit ein interpolierendes Quadraturverfahren.

`arch` kann leicht modifiziert werden, so daß auch andere Interpolanten als stückweise d-lineare Funktionen verwendet werden. Wendet man beispielsweise das Cavalierische

Prinzip nicht, wie zuvor ausgeführt, auf die Rekursionsgleichungen

$$
\begin{aligned}
T(a,b) &= \tfrac{1}{2}\left(f(a)+f(b)\right)\cdot(b-a)+S(a,b)\\
S(a,b) &= \tfrac{1}{2}\left(f\left(\frac{a+b}{2}\right)-\tfrac{1}{2}\left(f(a)+f(b)\right)\right)\cdot(b-a)\\
&+ S\left(a,\frac{a+b}{2}\right)+S\left(\frac{a+b}{2},b\right),
\end{aligned}
$$

sondern auf

$$
\begin{aligned}
M(a,b) &= (b-a)\cdot f\left(\frac{a+b}{2}\right)+M_l\left(a,\frac{a+b}{2}\right)+M_r\left(\frac{a+b}{2},b\right)\\
M_l(a,b) &= (b-a)\cdot\left(f\left(\frac{a+b}{2}\right)-f(b)\right)+M_l\left(a,\frac{a+b}{2}\right)+M_r\left(\frac{a+b}{2},b\right)\\
M_r(a,b) &= (b-a)\cdot\left(f\left(\frac{a+b}{2}\right)-f(a)\right)+M_l\left(a,\frac{a+b}{2}\right)+M_r\left(\frac{a+b}{2},b\right)
\end{aligned}
$$

beziehungsweise

$$
\begin{aligned}
S(a,b) &= \frac{b-a}{6}\cdot\left(f(a)+4\cdot f\left(\frac{a+b}{2}\right)+f(b)\right)\\
&+ S_l\left(a,\frac{a+b}{2}\right)+S_r\left(\frac{a+b}{2},b\right)\\
S_l(a,b) &= \frac{b-a}{12}\cdot\left(-3\cdot f(a)+8\cdot f\left(\frac{a+b}{2}\right)-6\cdot f(b)+f(2\cdot b-a)\right)\\
&+ S_l\left(a,\frac{a+b}{2}\right)+S_r\left(\frac{a+b}{2},b\right)\\
S_r(a,b) &= \frac{b-a}{12}\cdot\left(f(2\cdot a-b)-6\cdot f(a)+8\cdot f\left(\frac{a+b}{2}\right)-3\cdot f(b)\right)\\
&+ S_l\left(a,\frac{a+b}{2}\right)+S_r\left(\frac{a+b}{2},b\right)
\end{aligned}
$$

an, so wird der Integrand durch stückweise konstante beziehungsweise stückweise d-quadratische Funktionen angenähert.

2.4. Eigenschaften des Verfahrens. Wir betrachten als erstes die Funktion

Beispiel 1: $$f(x_1,x_2)=x_1^2\cdot x_2^2 \quad\text{mit}\quad (x_1,x_2)\in[-1,1]^2,$$

und wenden darauf `arch` mit $\varepsilon_{\text{abs}}=10^{-3}$ an. Abbildung 3 gibt die Gitterpunkte wieder, an denen der Integrand ausgewertet wird. Es handelt sich hier um ein *reguläres dünnes Gitter* ([Zen90]). Dies erscheint auf den ersten Blick erstaunlich, da bei der Konzeption des Algorithmus keine expliziten Maßnahmen ergriffen wurden, ein derart

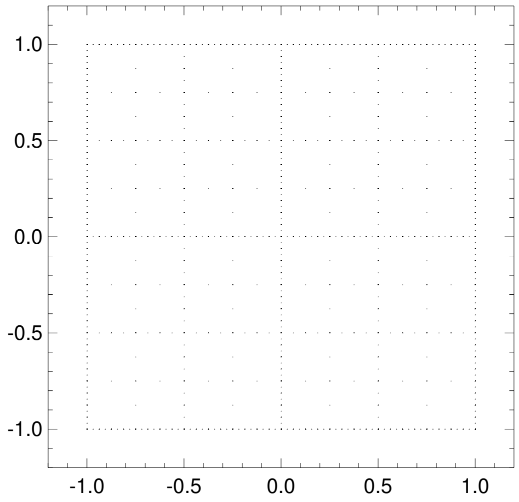

ABB. 3. BEISPIEL 1: *Gitterpunkte für* $\varepsilon_{\text{abs}} = 10^{-3}$.

strukturiertes Gitter auswerten zu lassen. Statt dessen hätte man, ausgehend vom eindimensionalen Fall, eine adaptive Variante der Produkt-Trapezregel erwartet, die an einer anderen Gitterstruktur zu erkennen gewesen wäre.

Es soll nun das Zustandekommen des Gitters erklärt werden. Die Gitterpunkte im Innern des Integrationsgebiets ergeben sich aus der Berechnung der hierarchischen Überschüsse, die sich für den gewählten Integranden zu

$$g_2\left(f\left(x_1, x_2\right), a_1, a_2, b_1, b_2\right) \quad = \quad \left(\frac{b_1 - a_1}{2} \cdot \frac{b_2 - a_2}{2}\right)^2$$

ergeben. Der hierarchische Überschuß in einem Basispunkt ist also proportional zum Quadrat der Fläche des Trägers $[a_1, b_1] \times [a_2, b_2]$. Abbildung 4 (links) zeigt ein Schema verschiedener Zerlegungen des Integrationsgebiets in Trägergebiete mit den entsprechenden Basispunkten. Alle Trägergebiete einer Diagonalen des Schemas sind gleich groß und besitzen damit den gleichen hierarchischen Überschuß. Die Adaptivitätsstrategie ist gerade so angelegt, daß die Trägergebiete fortlaufend halbiert werden, so daß prinzipiell alle in Abbildung 4 (links) dargestellten Trägergebiete behandelt werden und der hierarchische Überschuß im Basispunkt berechnet wird. Da keine weitere Verfeinerung des betrachteten Teilgebiets durchgeführt wird, wenn der hierarchische Überschuß eine vorgegebene Schranke unterschreitet, andererseits die hierarchischen Überschüsse auf den Diagonalen des Schemas gleich groß sind, werden also die hierarchischen Überschüsse für ein Dreiecksschema berechnet. Die Überlagerung all dieser Stützstellen ergibt ein reguläres dünnes Gitter (Abbildung 4 [rechts]).

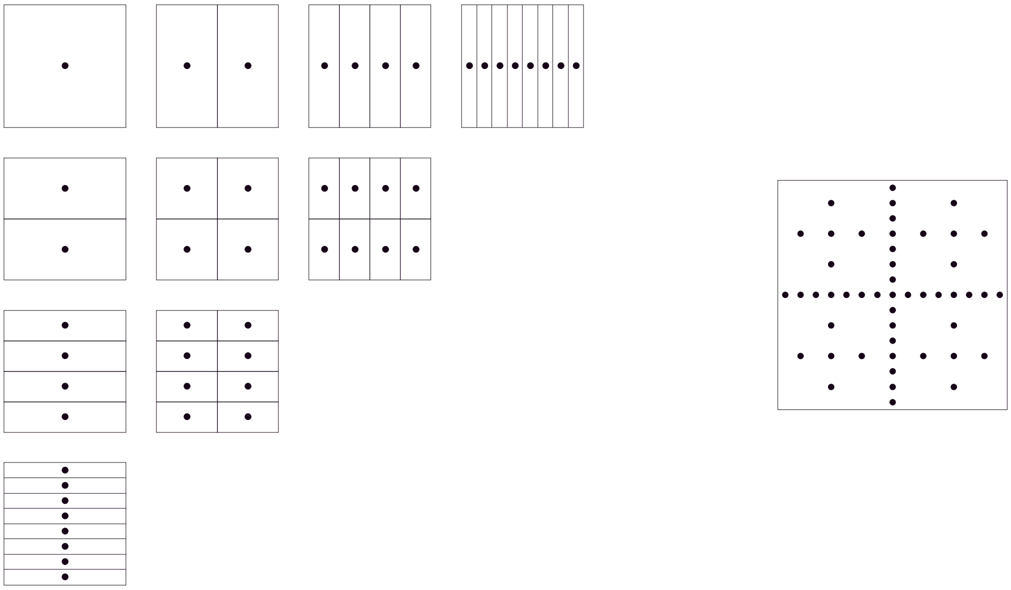

Abb. 4. *Reguläre dünne Gitter (rechts) ergeben sich aus der Überlagerung voller Gitter mit unterschiedlichen Schrittweiten in den Koordinatenrichtungen (links).*

Reguläre dünne Gitter benötigen bei vorgegebener Randfeinheit h wesentlich weniger Gitterpunkte als volle Gitter. Ihre Anzahl fällt von $O\left(h^{-d}\right)$ auf

$$O\left(h^{-1} \cdot \left(\mathrm{ld}\left(h^{-1}\right)\right)^{d-1}\right). \tag{12}$$

In [Bun92] ist gezeigt, daß die Approximationsgenauigkeit durch die verringerte Zahl von Stützstellen aber nur unwesentlich abnimmt, wenn d-lineare Interpolanten verwendet werden. Sie verschlechtert sich lediglich um einen logarithmischen Term von $O\left(h^2\right)$ auf

$$O\left(h^{2} \cdot \left(\mathrm{ld}\left(h^{-1}\right)\right)^{d-1}\right). \tag{13}$$

Diese Beziehung gilt auch für den Verfahrensfehler von `arch`, da es sich um ein interpolierendes Quadraturverfahren handelt, basierend auf stückweise d-linearen Funktionen. Die Eigenschaften (12) und (13) sollen am folgenden Beispiel gezeigt werden. Dazu betrachten wir die Funktion

Beispiel 2: $$f\left(x_1, \ldots, x_d\right) = e^{x_1 + \ldots + x_d} \quad \text{für} \quad \left(x_1, \ldots, x_d\right) \in [0,1]^d,$$

und wenden das Verfahren nicht-adaptiv an, indem eine Tiefenbeschränkung eingeführt wird. Der Algorithmus darf maximal $m > 0$ Inkarnationen von (10) beziehungsweise (11) gleichzeitig eröffnet haben. Dadurch werden nur reguläre dünne Gitter ausgewertet. Abbildung 5 gibt die Ergebnisse für $d = 2, \ldots, 10$ wieder. Die diskreten Meßpunkte wurde durch Linien zu einer Kennlinie verbunden. Auf der Abszisse sind in logarithmischem Maßstab die Anzahl der Gitterpunkte aufgetragen, auf der Ordinate die Genauigkeit in

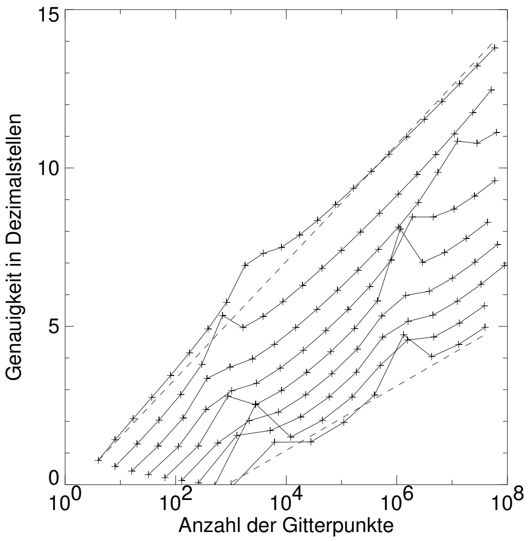

ABB. 5. BEISPIEL 2: *Genauigkeit in Abhängigkeit der Gitterpunktanzahl für* $d = 2, \ldots, 10$ *(von oben nach unten).*

Dezimalstellen. Es gilt: $h = O(2^{-m})$. Aus (12) folgt:

$$O\left(h^{-1} \cdot \left(\mathrm{ld}\left(h^{-1}\right)\right)^{d-1}\right) \quad = \quad O\left(2^m \cdot m^{d-1}\right).$$

Die Anzahl der Gitterpunkte steigt also im wesentlichen exponentiell mit m an. Dies wird durch die äquidistanten Meßpunkte bestätigt, aus denen sich die einzelnen Kurvenzüge zusammensetzen. Für den Verfahrensfehler folgt aus (13):

$$O\left(h^2 \cdot \left(\mathrm{ld}\left(h^{-1}\right)\right)^{d-1}\right) \quad = \quad O\left(4^{-m} \cdot m^{d-1}\right).$$

Der Fehler fällt also im wesentlichen geometrisch mit dem Faktor $\frac{1}{4}$ ab. Das spiegelt sich in den Kennlinien wieder, die gut durch Geraden approximiert werden können. Zum besseren Vergleich sind für $d = 2$ und $d = 10$ gestrichelt die Verläufe eingezeichnet, wenn die Konvergenz $O\left(4^{-m}\right)$ vorliegen würde. Man sieht, daß der logarithmische Term der Fehlerabschätzung nach (13) nur eine untergeordnete Rolle spielt.

Mit dem nächsten Beispiel soll die Verbesserung verdeutlicht werden, die durch die im Verfahren enthaltene Adaptivität erreicht wird. Wir betrachten die Funktion

BEISPIEL 3: $$f\left(x_1, x_2\right) = e^{x_1 + x_2} \quad \text{mit} \quad \left(x_1, x_2\right) \in [-2, 2]^2$$

und wenden `arch` in zwei Varianten an, bei Tiefenbeschränkung auf m Inkarnation beziehungsweise unter Vorgabe einer Abbruchschranke ε_{abs}. Beide Parameter werden variiert, so daß sich die in Abbildung 6 wiedergegebenen Kennlinien ergeben. Der Kurvenzug (1) in Abbildung 6 zeigt das Ergebnis für den nicht-adaptiven Fall. Aus der

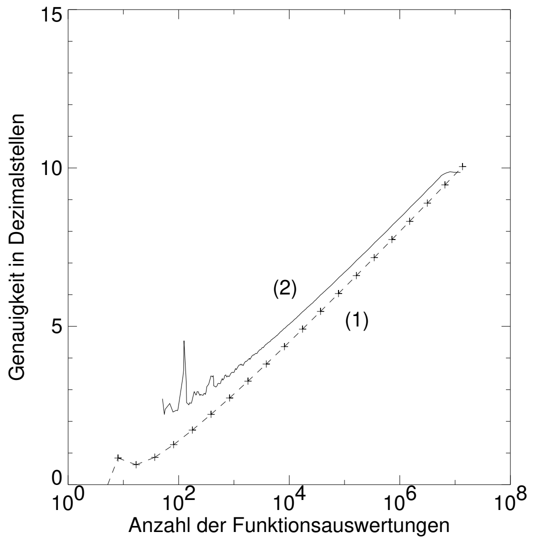

ABB. 6. BEISPIEL 3: *Vergleich von adaptiver* (2) *und nicht-adaptiver* (1) *Variante von* `arch`.

Steigung der Kennlinie kann eine Verfahrensordnug von etwa $O(h^2)$ abgelesen werden. Die Kennlinie (2) gibt die Ergebnisse des adaptiven Verfahrens wieder. Man sieht, daß keine Erhöhung der Konvergenzordnung erreicht wird. Die Ergebnisse verbessern sich bei gleichem Berechnungsaufwand um etwa eine halbe Dezimalstelle.

Anhand des nächsten Beispiels wollen wir das von `arch` erzeugte, adaptiv verfeinerte Gitter genauer untersuchen. Abbildung 7 (links) gibt die Stützstellen wieder, an denen der Integrand

BEISPIEL 4: $$f(x_1, x_2) = e^{x_1+x_2} \quad \text{im Bereich} \quad (x_1, x_2) \in [-8, 8]^2,$$

ausgewertet wird, wenn $\varepsilon_{\text{abs}} = 10^2$ gewählt wird. Man erkennt, daß `arch` den glatten Bereich der Funktion (linker unterer Quadrant) durch deutlich weniger Gitterpunkte auflöst, um die gewählte Genauigkeit zu erreichen, als im unglatten Bereich, zu dessen Spitze in der rechten oberen Ecke hin sich das Gitter stark verdichtet. In kleinen Bereichen erkennt man reguläre Dünn-Gitter-Strukturen wieder.

Der Integrand ist symmetrisch zu einer Achse $x_2 = x_1$. Demnach würde man erwarten, daß die zugehörigen Gitter Symmetrie bezüglich dieser Achse aufweisen würden. Dies ist aber nicht der Fall, wenn man beispielsweise die Punkte mit $x_1 = -4$ und $x_2 = -4$ vergleicht.

Die Ursache hierfür liegt in der Gebietszerlegungsstrategie. Nach Gleichung (10) nämlich wird das Grundgebiet für $d = 2$ in immer schmälere Streifen zerlegt, wie dies in Abbildung 8 veranschaulicht ist. Wir betrachten nun einen solchen Streifen (Abbildung 9 dick gezeichnet). Vom hierarchischen Überschuß im Basispunkt B hängt es nun ab, ob die rekursive Gleichung (10) oder ihre Terminierungsalternative (11) zu verwenden ist und ob damit auch noch weitere Punkte im Innern des Gebiets auszuwerten

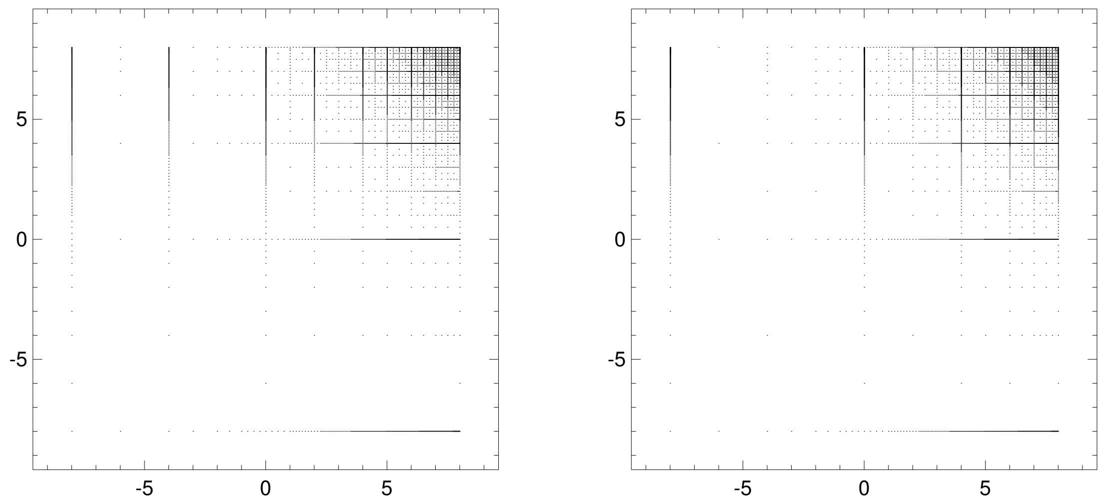

ABB. 7. BEISPIEL 4: *das Gitter, das für* $\varepsilon_{\mathrm{abs}} = 10^2$ *erzeugt wird.*

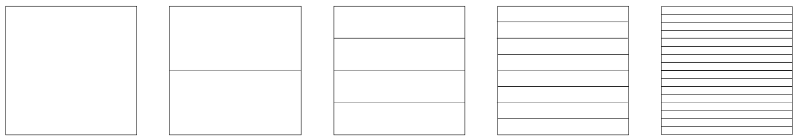

ABB. 8. *Gebietszerlegung durch fortgesetztes Halbieren in der Koordinatenrichtung* x_2.

sind. Für die Entscheidung, ob am Gitterpunkt A eine Auswertung stattfinden soll, ist es aus Lokalitätsgründen besser, den hierarchischen Überschuß im Basispunkt C heranzuziehen, dessen Trägergebiet (gestrichelt gezeichnet) genauso groß ist wie das von B. Um die Robustheit des Verfahrens zu erhöhen, sollte also die Erzeugung von stark entarteten Teilgebieten, wie dies bei `arch` der Fall ist, vermieden werden.

Das Problem kann behoben werden, wenn die Halbierungsrichtung nach jedem Gebietszerlegungsschritt gewechselt wird. Diese Modifikation läßt sich in den Definitionsgleichungen (9) bis (11) berücksichtigen, indem die Variablen $x_1, \ldots, x_d$ (und damit auch die Bereichsgrenzen $a_1, \ldots, a_d$ und $b_1, \ldots, b_d$) vor jeder Anwendung zyklisch getauscht werden. Mathematisch gesehen wird hier lediglich die Integrationsreihenfolge vertauscht. Abbildung 10 gibt die daraus resultierende Gebietszerlegung wieder. Es treten keine schmalen Gebietsstreifen mehr auf. Die Seitenlängen der Teilgebiete differieren höchsten um den Faktor 2.

Wir bezeichnen diese Verfahrensvariante mit `arch_wr` und wenden sie auf den obigen Integranden an. Es ergibt sich das in Abbildung 7 (rechts) gezeigte Gitter. Die Symmetrie des Gitters ist nun stärker ausgeprägt, wie man das insbesondere an den

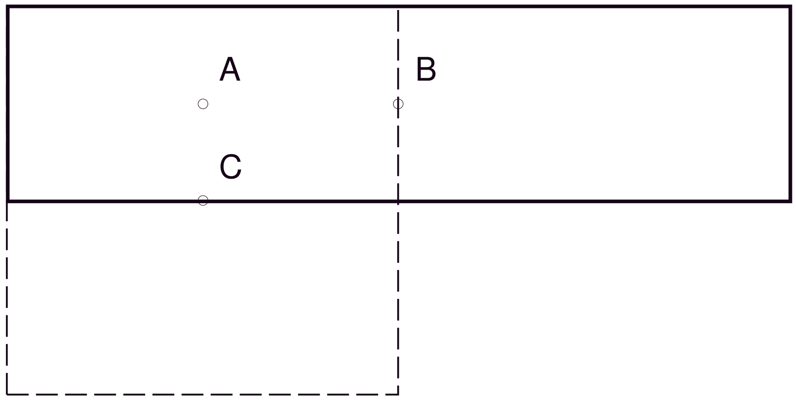

ABB. 9. *Die hierarchischen Überschüsse in den Punkten B oder C können zur Entscheidung herangezogen werden, ob Punkt A berechnet werden soll.*

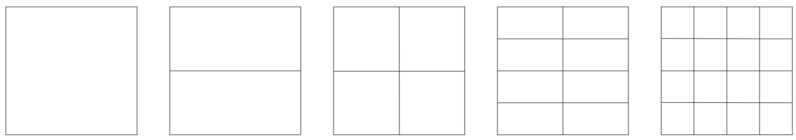

ABB. 10. *Gebietszerlegung bei zyklischem Tausch der Halbierungsrichtung.*

Punkten auf den Geraden $x_1 = -4$ und $x_2 = -4$ erkennt. Eine zusätzlicher Gewinn an Genauigkeit ist in diesem Fall allerdings nicht festzustellen. Mit 46453 Funktionsauswertungen (6575 Gitterpunkte) wird bei `arch` eine Genauigkeit von 3.35 Dezimalstellen erreicht, bei `arch_wr` beträgt sie mit 45745 Stützstellen (6757 Gitterpunkte) 3.49 Dezimalen. Im folgenden Beispiel kann eine leichte Verringerung des Aufwands festgestellt werden.

Wir wenden beide Verfahren auf die Funktion

BEISPIEL 5: $$f(x_1, \ldots, x_d) = \sin(x_1) \cdot \ldots \cdot \sin(x_d) \quad \text{mit} \quad (x_1, \ldots, x_d) \in [0, \pi]^d$$

und $d = 2, \ldots, 5$ unter schrittweiser Verkleinerung von ε_{abs} an. In Abbildung 11 sind die Anzahl der Funktionsauswertungen und die dabei erzielte Genauigkeit für `arch` (durchgezogene Linie) und `arch_wr` (gepunktete Linie) wiedergegeben. Für $d = 2$ liegen beide Linien fast aufeinander. Bei zunehmender Dimension gewinnt `arch_wr` leicht an Überlegenheit.

Als Ergebnis dieses Kapitels halten wir fest: Die Archimedische Ausschöpfung und das Cavalierische Prinzip führen in natürlicher Weise zu dünnen Gittern. Das sich daraus ergebende rekursive Quadraturverfahren approximiert den Integranden durch stückweise d-lineare Funktionen und berechnet darüber das Integral. Der Integrand wird gleichmäßiger abgetastet, wenn die Halbierungsrichtungen zyklisch getauscht werden.

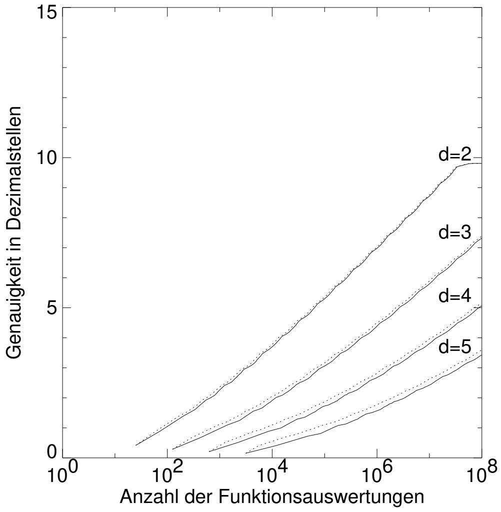

ABB. 11. BEISPIEL 5: *Ergebnisse für* `arch` *(durchgezogen) und* `arch_wr` *(gepunktet) für* $d = 2, \ldots, 5$.

3. Ein nicht-adaptives Quadraturverfahren höherer Ordnung. Die Konvergenzordnung zahlreicher Diskretisierungsverfahren kann mit Hilfe der Extrapolationstechnik nach Richardson ([RG27]) erhöht werden. Die Grundidee besteht darin, die Kenntnis der verfahrensspezifischen Fehlerstruktur auszunutzen, um gezielt Fehlerglieder bestimmter Ordnungen zu eliminieren. Die Romberg-Quadratur stellt eine Anwendung dieser Technik für die eindimensionale numerische Integration dar.

In diesem Abschnitt soll die Romberg-Quadratur für mehrdimensionale Funktionen erweitert werden. Dazu wird nicht, wie in der Literatur allgemein üblich, ein univariater Ansatz gewählt ([Eng80], [Str71]), sondern ein multivariater. Das sich daraus ergebende Verfahren basiert auf dünnen Gittern und ist von höherer Ordnung. Für hinreichend glatte Integranden wird eine Abschätzung des Verfahrensfehlers hergeleitet. Die numerischen Eigenschaften werden anhand zweier Beispiele eingehender untersucht.

Zuvor soll aber noch ein Blick auf den eindimensionalen Fall geworfen werden.

3.1. Die Romberg-Quadratur. Wir betrachten eine Funktion $f(x)$ im Intervall $[a, b]$. Die Trapezsumme $T(h)$ mit der Schrittweite $h = \frac{b-a}{n}$ und $n \in \mathbb{N}$

$$T(h) \;=\; \left(\tfrac{1}{2} \cdot f(a) + \sum_{i=1}^{n-1} f(a + i \cdot h) + \tfrac{1}{2} \cdot f(b)\right) \cdot h$$

stellt eine Näherung für das Integral

$$\tau_0 \;=\; \int_a^b f(x)\, dx$$

dar. Für einen Integranden $f \in C^{2m+2}[a, b]$ ist die Struktur des Verfahrensfehlers $T(h) - \tau_0$ aufgrund der *Euler-Maclaurinschen Summenformel*

$$(14) \quad T(h) \;=\; \tau_0 + \tau_1 \cdot h^2 + \tau_2 \cdot h^4 + \ldots + \tau_m \cdot h^{2m} + \alpha_{m+1}(h) \cdot h^{2m+2} \quad \text{für} \quad m \geq 0$$

bekannt. Zwei Eigenschaften dieser Entwicklung sind für die Extrapolationstechnik von Bedeutung. Zum einen sind die Fehlergliedkoeffizienten $\tau_k, k = 1, \ldots, m$ unabhängig von h

$$\tau_k \;=\; \frac{B_{2k}}{(2k)!} \cdot \left(f^{(2k-1)}(b) - f^{(2k-1)}(a)\right)$$

(B_k bezeichnet die Bernoulli-Zahlen), zum anderen kann der Restgliedkoeffizient

$$\begin{aligned}
\alpha_{m+1}(h) \;&=\; \frac{1}{(2m+2)!} \cdot \int_0^1 \left(\left(B_{2m+2}(t) - B_{2m+2}\right) \cdot \sum_{i=0}^{n-1} f^{(2m+2)}(a + h \cdot (t+i)) \right) dt \\
&=\; \frac{B_{2m+2}}{(2m+2)!} \cdot (b-a) \cdot f^{(2m+2)}(\xi(h)) \qquad \text{mit} \qquad a < \xi(h) < b
\end{aligned}$$

($B_k(t)$ sind die Bernoulli-Polynome) durch eine von h unabhängige Konstante A_{m+1} so abgeschätzt werden, daß für alle $h = \frac{b-a}{n}, n \in \mathbb{N}$ gilt:

$$(15) \qquad |\alpha_{m+1}(h)| \leq A_{m+1} = \left| \frac{B_{2m+2}}{(2m+2)!} \cdot (b-a) \right| \cdot \max_{x \in [a,b]} \left| f^{(2m+2)}(x) \right| .$$

Ist nun eine streng monoton fallende endliche Folge von Schrittweiten $H_m = \{h_0, h_1, \ldots, h_m\}$ gegeben, so können die Trapezsummen $T(h_i), i = 0, 1, \ldots, m$ derart verrechnet werden, daß in (14) die Fehlerglieder der Ordnungen $h^2, h^4, \ldots, h^{2m}$ eliminiert werden. Üblicherweise wird hierzu durch die Stützstellen $(h_i, T(h_i)), i = 0, \ldots, m$ ein Polynom der Form

$$T_m(h) \;=\; \dot{\tau}_0 + \dot{\tau}_1 h^2 + \dot{\tau}_2 h^4 + \ldots + \dot{\tau}_m h^{2m}$$

gelegt und nach $h = 0$ extrapoliert. In der Praxis umgeht man dabei eine explizite Berechnung des Polynoms mit Hilfe des *Algorithmus von Neville und Aitken*, welcher in diesem Fall übergeht zu:

$$\begin{aligned} T_{i,0} &= T(h_i) \\ T_{i,k} &= T_{i,k-1} + \frac{T_{i,k-1} - T_{i-1,k-1}}{\left(\frac{h_{i-k}}{h_i}\right)^2 - 1}, \quad 1 \le k \le i \le m \\ T_m(0) &= T_{m,m}. \end{aligned} \tag{16}$$

Abbildung 12 gibt hierzu die Datenabhängigkeit der $T_{i,k}$ wieder. Aus (16) folgt, daß $T_m(0)$ als gewichtete Summe über $T(h_i), i = 0, 1, \ldots, m$ aufgefaßt werden kann. Sie ergibt sich aus der Interpolationsformel nach *Lagrange* zu:

$$T_m(0) \;=\; \sum_{j=0}^{m} c_{m,j} \cdot T(h_j) \quad \text{mit} \quad c_{m,j} = \prod_{\substack{\varrho=0 \\ \varrho \neq j}}^{m} \frac{h_\varrho^2}{h_\varrho^2 - h_j^2}. \tag{17}$$

Da $\alpha_{m+1}(h)$ nach (15) beschränkt ist, läßt sich zeigen, daß der Verfahrensfehler zu

$$|\tau_0 - T_m(0)| = O\left(h^{2m+2}\right) \tag{18}$$

abgeschätzt werden kann ([Sto83]). Bulirsch [BS64] hat hierfür eine Verschärfung angegeben, sofern die Schrittweitenfolge H genügend stark nach 0 abfällt, etwa

$$H_m \;=\; \left\{h_j \;\middle|\; h_j \le h_0 \cdot b^j, j = 0, 1, \ldots, m\right\} \qquad \text{mit} \qquad 0 < b < 1.$$

Es gilt dann:

$$|\tau_0 - T_m(0)| \;\le\; A_{m+1} \cdot C(b) \cdot h_0^2 \cdot h_1^2 \cdot \ldots \cdot h_m^2, \tag{19}$$

wobei $C(b)$ nur noch von b abhängt. Für die Romberg-Quadratur, die die Schrittweitenfolge mit $b = \frac{1}{2}$ verwendet, läßt sich zeigen:

$$C\left(\tfrac{1}{2}\right) \;\approx\; 1.96926\ldots$$

Auch für die Mittelpunktsumme

$$M(h) \;=\; h \cdot \sum_{i=1}^{n} f\left(a + \left(i - \tfrac{1}{2}\right) \cdot h\right) \quad \text{mit} \quad h = \frac{b-a}{n}$$

$$
\begin{tikzcd}[column sep=small, row sep=tiny]
T(h_0) & = & T_{00} \arrow[dr] & & & \\
 & & & T_{11} \arrow[dr] & & \\
T(h_1) & = & T_{10} \arrow[ur] \arrow[dr] & & T_{22} \arrow[dr] & \\
 & & & T_{21} \arrow[ur] \arrow[dr] & & T_{33} \\
T(h_2) & = & T_{20} \arrow[ur] \arrow[dr] & & T_{32} \arrow[ur] & \\
 & & & T_{31} \arrow[ur] & & \\
T(h_3) & = & T_{30} \arrow[ur] & & &
\end{tikzcd}
$$

ABB. 12. *Berechnungsschema für das Verfahren von Neville und Aitken.*

existiert bei gleichen Glattheitsanforderungen an f eine asymptotische Fehlerentwicklung, ähnlich der Euler-Maclaurinschen Summenformel. Sie läßt sich schnell mit Hilfe der Beziehung

$$M(h) \quad = \quad 2 \cdot T\left(\frac{h}{2}\right) - T(h)$$

aus (14) herleiten. Die Extrapolationstechnik kann also auch auf Mittelpunktsummen angewendet werden, um die Verfahrensordnung zu erhöhen.

3.2. Numerische Quadratur höherer Ordnung im mehrdimensionalen Fall. In diesem Abschnitt soll die Romberg-Quadratur auf den mehrdimensionalen Fall erweitert werden. Wir wählen hierzu einen multivariaten Ansatz.

Zunächst werden einige Notationen und Begriffe eingeführt, die sich für das weitere Arbeiten als hilfreich erwiesen haben. Danach stellen wir die Elementaroperatoren vor, auf die sich das Verfahren stützt. Es folgt die Beschreibung des Quadraturverfahrens, dessen Zwischenschritte anhand eines Beispiels verdeutlicht werden.

Wir bezeichnen das d-Tupel $\vec{\imath} := (i_1, i_2, \ldots, i_d)$ mit $i_k \in \mathbb{N}_0$, und $k = 1, \ldots, d$ als *d-dimensionalen Multiindex* und legen fest:

$$
\begin{aligned}
\vec{0} \quad &:= \quad (0, 0, \ldots, 0) \\
\vec{e}_k \quad &:= \quad \left(0, \ldots, 0, \underbrace{\;1\;}_{\text{Position } k}, 0, \ldots, 0\right) \\
\vec{i}\,[i_k = u] \quad &:= \quad (i_1, i_2, \ldots, i_{k-1}, u, i_{k+1}, \ldots, i_d) \\
|\vec{\imath}| \quad &:= \quad i_1 + i_2 + \ldots + i_d \\
\max(\vec{\imath}) \quad &:= \quad \max\{i_1, i_2, \ldots, i_d\} \\
\vec{\imath} \leq \vec{\jmath} \quad &\Rightarrow \quad \forall\, k = 1, \ldots, d : i_k \leq j_k \\
f^{(\vec{\imath})}(x_1, x_2, \ldots, x_d) \quad &:= \quad \frac{\partial^{|\vec{\imath}|} f(\vec{x})}{\partial x_1^{i_1}\, \partial x_2^{i_2} \cdots \partial x_d^{i_d}}.
\end{aligned}
$$

Die Rechenregeln für Multiindizes ergeben sich analog zu denen von Vektoren im d-dimensionalen Raum:

$$
\begin{aligned}
\vec{\imath} + \vec{\jmath} \quad &= \quad (i_1 + j_1, i_2 + j_2, \ldots, i_d + j_d) \\
a \cdot \vec{\imath} \quad &= \quad (a \cdot i_1, a \cdot i_2, \ldots, a \cdot i_d)\,.
\end{aligned}
$$

Es seien für $i_k \in \mathbb{N}_0, k = 1, 2, \ldots, d$ die Operatoren $X_{i_k}^{[k]}$ sowie die skalaren Größen q_{k,i_k} und p_{i_k} gegeben. Vereinfachend schreiben wir:

$$
\begin{aligned}
X_{\vec{\imath}} f(\vec{x}\,) &:= X_{i_1}^{[1]} X_{i_2}^{[2]} \cdots X_{i_d}^{[d]} f(\vec{x}\,) \\
\vec{p}_{\vec{\imath}} &:= (p_{i_1}, p_{i_2}, \ldots, p_{i_d}) \\
\vec{q}_{\vec{\imath}} &:= (q_{1,i_1}, q_{2,i_2}, \ldots, q_{d,i_d}) \\
p_{\vec{\imath}} &:= p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_d} \\
q_{\vec{\imath}} &:= q_{1,i_1} \cdot q_{2,i_2} \cdot \ldots \cdot q_{d,i_d} \\
p_{\vec{\jmath}}^{\vec{\imath}} &:= p_{j_1}^{i_1} \cdot p_{j_2}^{i_2} \cdot \ldots \cdot p_{j_d}^{i_d} \\
q_{\vec{\jmath}}^{\vec{\imath}} &:= q_{1,j_1}^{i_1} \cdot q_{2,j_2}^{i_2} \cdot \ldots \cdot q_{d,j_d}^{i_d}.
\end{aligned}
$$

Die Elemente h_i einer Folge $H = \{h_0, h_1, \ldots, h_m\}$ können mit Hilfe des Indizes $i \in \mathcal{I} = \{0, 1, \ldots, m\}$ referenziert werden. Die folgende Definition verallgemeinert den Begriff ,,Indexmenge“ auf Multiindizes.

Definition 3.1. *$\mathcal{I}$ ist eine Multiindexmenge der Dimension d, wenn gilt:*

$$
\mathcal{I} \;=\; \left\{ \vec{\imath} = (i_1, i_2, \ldots, i_d) \;\middle|\; i_k \in \mathbb{N}_0, k = 1, \ldots, d \right\}
$$

und $\forall\, \vec{\imath} \in \mathcal{I}, \forall k \in \{1, \ldots, d\} : i_k > 0 \Rightarrow \vec{\imath} - \vec{e}_k \in \mathcal{I}$.

Insbesondere sei die folgende Multiindexmenge definiert:

$$
\mathcal{I}_m^{(d)} \;=\; \left\{ \vec{\imath} \;\middle|\; |\vec{\imath}| \leq m \right\} \quad \textit{mit} \quad m \geq 0.
$$

Eine Menge von Zahlen

$$
(u_{\vec{\imath}})_{\vec{\imath} \in \mathcal{I}} \;:=\; \left\{ u_{\vec{\imath}} \in \mathbb{R} \;\middle|\; \vec{\imath} \in \mathcal{I} \right\} \in \mathbb{R}(\mathcal{I})
$$

auf die mittels eines Multiindizes zugegreifen werden kann, bezeichnen wir als *Schema*. $\mathbb{R}(\mathcal{I})$ sei die Menge aller Schemata reeller Zahlen bezüglich der Multiindexmenge $\mathcal{I}$.

Es werden nun die drei Elementaroperatoren $\Lambda^{[k]}, \Delta^{[k]}$ und Σ definiert, aus denen sich das Quadraturverfahren zusammensetzt. Ein hochgestelltes $[k]$ weist darauf hin, daß der Operator nur in einer Koordinatenrichtung k ,,wirksam“ ist.

Definition 3.2. *Es seien $k \in \{1, \ldots, d\}$, eine streng monoton fallende Schrittweitenfolge $H^{[k]} = \{h_{k,0}, h_{k,1}, \ldots\}$ sowie ein Schema $(u_{\vec{\imath}})_{\vec{\imath} \in \mathcal{I}} \in \mathbb{R}(\mathcal{I})$ gegeben. Die Operatoren $\Lambda^{[k]}$, $\Delta^{[k]}$ und Σ sind wie folgt definiert:*

$$
\begin{aligned}
\Lambda^{[k]} : \mathbb{R}(\mathcal{I}) &\rightarrow \mathbb{R}(\mathcal{I}) \\
\Lambda^{[k]} (u_{\vec{\imath}})_{\vec{\imath} \in \mathcal{I}} &= (\hat{u}_{\vec{\imath}})_{\vec{\imath} \in \mathcal{I}} \\
\hat{u}_{\vec{\imath}} &= \sum_{j=0}^{i_k} c_{i_k,j}^{[k]} \cdot u_{\vec{\imath}[i_k=j]}
\end{aligned}
$$

$$
\begin{aligned}
mit \;\; c^{[k]}_{i_k,j} &= \prod_{\substack{\varrho=0\\ \varrho\neq j}}^{i_k} \frac{h^2_{k,\varrho}}{h^2_{k,\varrho}-h^2_{k,j}} \\
\Delta^{[k]} : \mathbb{R}(\mathcal{I}) &\rightarrow \mathbb{R}(\mathcal{I}) \\
\Delta^{[k]}(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} &= (\bar{u}_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} \\
mit \;\; \bar{u}_{\vec{\imath}} &= \begin{cases} u_{\vec{\imath}} & falls \;\; i_k = 0 \\ u_{\vec{\imath}} - u_{\vec{\imath}-\vec{e}_k} & falls \;\; i_k > 0 \end{cases} \\
\Sigma : \mathbb{R}(\mathcal{I}) &\rightarrow \mathbb{R} \\
\Sigma(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} &= \sum_{\vec{\imath}\in\mathcal{I}} u_{\vec{\imath}}.
\end{aligned}
$$

$\Lambda^{[k]}$ führt auf allen Werten des Eingabeschemas eindimensionale Extrapolationen bezogen auf die Koordinatenrichtung k durch. Man erkennt an der Definition des Operators sehr schnell die Ähnlichkeit zur Interpolationsformel nach Lagrange (17). $\Delta^{[k]}$ errechnet auf dem Schema die Rückwärtsdifferenzen in Richtung k, und Σ bildet die Summe aller Werte eines Schemas.

Alle drei Operatoren sind linear:

LEMMA 3.3. *Es seien eine Multiindexmenge $\mathcal{I}$, zwei Schemata $(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} \in \mathbb{R}(\mathcal{I})$ und $(v_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} \in \mathbb{R}(\mathcal{I})$ sowie $a \in \mathbb{R}$ gegeben. Für $\Theta \in \left\{\Lambda^{[k]}, \Delta^{[k]}, \Sigma\right\}$ und $k \in \{1,\ldots,d\}$ gilt:*

$$
\begin{aligned}
\Theta(u_{\vec{\imath}} + v_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} &= \Theta(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} + \Theta(v_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} \\
\Theta(a \cdot v_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} &= a \cdot \Theta(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}}.
\end{aligned}
$$

Beweis. Die Gültigkeit der Behauptungen läßt sich schnell durch Einsetzen der Definition 3.2 beweisen. □

Es folgt nun das Quadraturverfahren, das Gegenstand der weiteren Betrachtungen dieses Kapitels ist:

DEFINITION 3.4. *Es seien eine nichtleere d-dimensionale Multiindexmenge $\mathcal{I}$ und die Schrittweitenfolgen*

$$
H^{[k]} = \{h_{k,0}, h_{k,1}, \ldots\}
$$

in den Richtungen $k = 1, 2, \ldots, d$ gegeben. Es sei $Q_{\vec{\imath}} = Q\left(\vec{h}_{\vec{\imath}}\right)$ eine Integrationsformel mit multivariater Schrittweite $\vec{h}_{\vec{\imath}} = (h_{1,i_1}, h_{2,i_2}, \ldots, h_{d,i_d})$, dann ist das Quadraturverfahren wie folgt definiert:

$$
\Phi(Q_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} = \Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\Lambda^{[d]}\Lambda^{[d-1]}\cdots\Lambda^{[1]}(Q_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}}. \tag{20}
$$

Mögliche Integrationsformeln für $Q_{\vec{\imath}}$ sind Trapezsummen und Mittelpunktsummen mit der Schrittweite h_{k,i_k} in der Richtung $k = 1, 2, \ldots, d$.

Es folgt die Linearität von Φ:

LEMMA 3.5. *Es seien eine Multiindexmenge $\mathcal{I}$ und zwei Schemata $(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} \in \mathbb{R}(\mathcal{I})$ und $(v_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}} \in \mathbb{R}(\mathcal{I})$ gegeben, $a \in \mathbb{R}$. Dann gilt:*

$$\begin{aligned}
\Phi\left(u_{\vec{\imath}} + v_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}} &= \Phi\left(u_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}} + \Phi\left(v_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}} \\
\Phi\left(a \cdot u_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}} &= a \cdot \Phi\left(u_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}}.
\end{aligned}$$

Beweis. Die Behauptung ergibt sich sofort aus Lemma 3.3. □

Für $d = 1$ kann man sehr schnell zeigen, daß das Verfahren in die Romberg-Quadratur gemäß Gleichung (17) übergeht, wenn als Integrationsformel die Trapezsumme $T_{\vec{\imath}} := T\left(h_{i_1}\right)$ verwendet wird:

$$\begin{aligned}
\Phi\left(T_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(1)}} &= \Sigma\Delta^{[1]}\Lambda^{[1]}\left(T_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(1)}} \\
&= \Sigma\Delta^{[1]}\left(\sum_{j=0}^{i_1} c_{m,j} \cdot T_j\right)_{\vec{\imath}\in\mathcal{I}_m^{(1)}} \\
&= \sum_{j=0}^{m} c_{m,j} \cdot T\left(h_j\right) = T_m(0).
\end{aligned}$$

Die Zwischenschritte des Quadraturverfahrens sollen anhand eines Beispiels für $d = 2$ illustriert werden. Wir wählen als Integrationsgebiet das Einheitsquadrat und legen die Romberg-Schrittweitenfolgen $H^{[k]} = \left\{h_{k,i} \;\middle|\; h_{k,i} = 2^{-i}, i \in \mathbb{N}\right\}$ in beiden Koordinatenrichtungen $k = 1, 2$ zugrunde. Als Integrationsformel verwenden wir die Mittelpunktsumme

$$M_{\vec{\imath}} = M_{(i_1,i_2)} := M\left(h_{1,i_1}, h_{2,i_2}\right),$$

mit den Schrittweiten h_{1,i_1} und h_{2,i_2} in den Variablen x_1 beziehungsweise x_2. Es sei die Multiindexmenge $\mathcal{I}_3^{(2)}$ zugrunde gelegt. Abbildung 13 veranschaulicht die Schritte des Verfahrens

$$\Phi\left(M_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_3^{(2)}} = \Sigma\Delta^{[2]}\Delta^{[1]}\Lambda^{[2]}\Lambda^{[1]}\left(M_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_3^{(2)}},$$

die sich aus der sukzessiven Anwendung der Elementaroperatoren ergibt.

Oben links ist das Schema $\left(M_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_3^{(2)}}$ dargestellt. Jedes Quadrat symbolisiert eine Mittelpunktsumme, deren Stützstellen durch Punkte markiert sind. Die Multiindizes $\vec{\imath} = (i_1, i_2)$ sind für i_1 aufsteigend von links nach rechts und für i_2 von oben nach unten aufgetragen.

Die beiden folgenden Schemata (oben Mitte und oben rechts) verdeutlichen die Funktionsweise der Extrapolationsoperatoren $\Lambda^{[1]}$ und $\Lambda^{[2]}$. Ein Pfeil entspricht einer

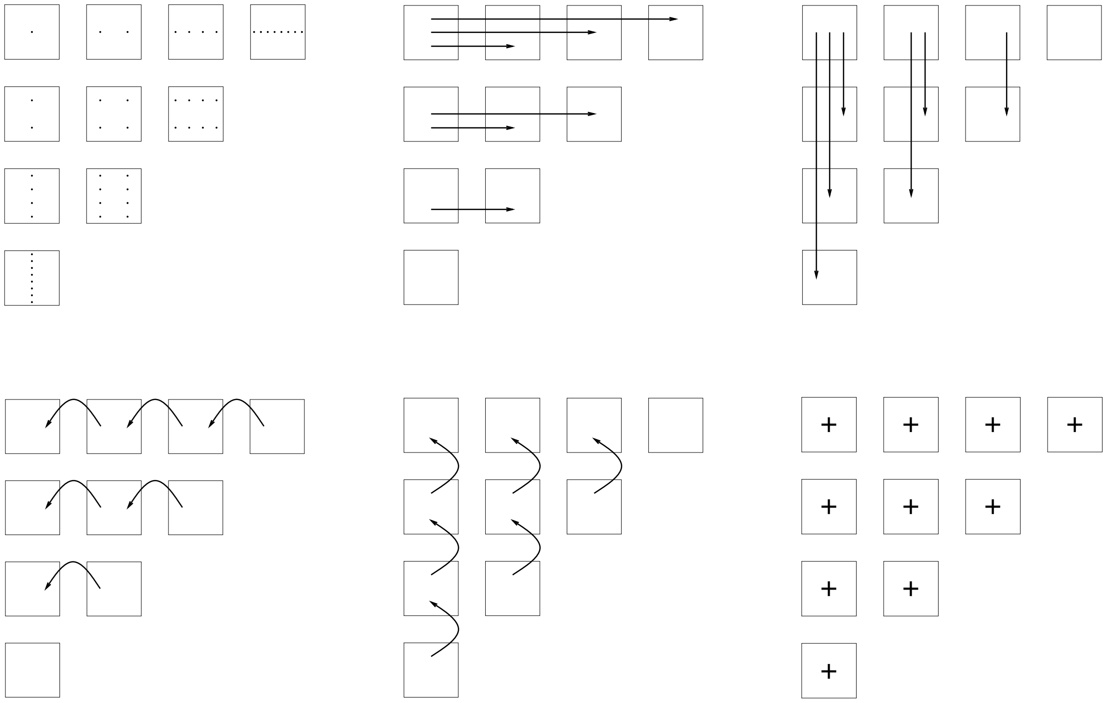

ABB. 13. *Die Schritte des Quadraturverfahrens* $\Phi\left(M\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_3^{(2)}}$.

eindimensionalen Extrapolation, wie sie durch (17) beschrieben wird. Der Wert an der Spitze eines Pfeils wird aus den Werten des Argumentschemas ermittelt, über die sich die Länge der Pfeils erstreckt. Voraussetzung für die Extrapolation ist die Existenz der entsprechenden Werte des Eingabeschemas. Das Schema darf also keine ,,Lücken" aufweisen. Definition 3.1 legt Multiindexmengen gerade so fest, daß dieser Fall nicht auftreten kann.

Die Wirkungsweise von $\Delta^{[1]}$ und $\Delta^{[2]}$ ist in den folgenden beiden Schritten der Abbildung dargestellt. $\Delta^{[1]}$ und $\Delta^{[2]}$ bilden jeweils die Rückwärtsdifferenzen in den Koordinatenrichtungen 1 und 2, die durch gebogene Pfeile angedeutet sind. Danach enthält das Schema die Beiträge, aus denen sich das Endergebnis zusammensetzt. Diese werden, wie in der Abbildung unten rechts dargestellt, durch den Operator Σ zum Endergebnis aufsummiert.

Man beachte, daß die Überlagerung der Integrandenstützstellen für das Schema $(M_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}_3^{(2)}}$ im obigen Beispiel gerade ein dünnes Gitter ergibt. Allgemein gilt, daß bei Verwendung des Schemas $(M_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ die Punkte eines regulären d-dimensionalen dünnen Gitters ausgewertet werden, wenn in jeder Koordinatenrichtung die Romberg-Schrittweitenfolge gewählt wird.

3.3. Abschätzung des Verfahrensfehlers. In diesem Abschnitt soll gezeigt werden, daß das soeben vorgestellte Verfahren für $\mathcal{I} = \mathcal{I}_m^{(d)}$ von der Ordnung $O\left(h^{2m+2}\right)$ ist. Die Vorgehensweise hierzu orientiert sich stark an [Sto83], wo dies bereits für den eindimensionalen Fall gezeigt ist. Im ersten Schritt wird für die Produkt-Trapezregel $T\left(\vec{h}\right)$ mit der multivariaten Schrittweite $\vec{h} = (h_1, h_2, \ldots, h_d)$ eine Fehlerentwicklung ähnlich der Euler-Maclaurinschen Summenformel hergeleitet. Sodann wird gezeigt, daß Fehlerglieder bestimmter Ordnungen durch das Verfahren eliminiert werden. Für den Fall $\mathcal{I} = \mathcal{I}_m^{(d)}$ wird schließlich bewiesen, daß das Verfahren von der Ordnung $O\left(h^{2m+2}\right)$ ist. Außerdem wird dazu eine Abschätzung des Verfahrensfehlers angegeben.

In diesem Abschnitt setzen wir den d-dimensionalen Einheitswürfel $H^d = [0,1]^d$ voraus. Die Resultate lassen sich durch geeignete Integrandentransformation auf beliebige rechteckige Gebiete übertragen.

3.3.1. Eine kompakte Notation. Es hat sich gezeigt, daß mehrdimensionale Verallgemeinerungen der Euler-Maclaurinschen Summenformel sehr schnell zu großen, schwer handhabbaren Ausdrücken führen, wenn sie in aller Ausführlichkeit notiert werden ([Bul64]). Um dennoch mit ihnen arbeiten zu können, benötigt man eine Schreibweise, die einerseits gewisse wiederkehrende Teilausdrücke zusammenfassen kann, andererseits aber die Herleitung der Summenformel nicht unnötig erschwert. Lyness hat in [LM70] eine Notation vorgeschlagen, die diesen Ansprüchen gerecht wird. Wir wollen sie uns zunutze machen, um eine für unsere Zwecke geeignete Form der mehrdimensionalen Euler-Maclaurinschen Summenformel herzuleiten. Die Notation bedient sich einiger Operatoren, die stets nur in einer Koordinatenrichtung wirksam sind. Die Hintereinanderausführung dieser Operatoren erlaubt es, Koordinatenrichtungen unabhängig voneinander zu behandeln. Wir legen fest:

Definition 3.6. *Es sei eine d-dimensionale Funktion $f(\vec{x}) = f(x_1, x_2, \ldots, x_d)$, eine Koordinatenrichtung $k \in \{1, \ldots, d\}$, die Schrittweite $h_k = \frac{1}{n_k}$ sowie $w \in \mathbb{N}$ gegeben. Dann ist:*

$$
\begin{aligned}
T^{[k]} f(\vec{x}) &:= h_k \cdot \left(\tfrac{1}{2} \cdot f(\vec{x}[x_k = 0]) + \sum_{j=1}^{n_k - 1} f(\vec{x}[x_k = j \cdot h_k]) + \tfrac{1}{2} \cdot f(\vec{x}[x_k = 1]) \right) \\
I^{[k]} f(\vec{x}) &:= \int_0^1 f(\vec{x})\, dx_k \\
\delta_w^{[k]} f(\vec{x}) &:= h_k^{2w} \cdot \frac{B_{2w}}{(2w)!} \cdot \int_0^1 \frac{\partial^{2w}}{\partial x_k^{2w}} f(\vec{x})\, dx_k \\
D_w^{[k]} f(\vec{x}) &:= h_k^{2w} \cdot \frac{1}{(2w)!} \\
&\quad \cdot \int_0^1 \left((B_{2w}(t) - B_{2w}) \cdot \sum_{i=0}^{n_k - 1} \left(\frac{\partial^{2w}}{\partial x_k^{2w}} f(\vec{x}[x_k = h_k \cdot (i + t)]) \right) \right) dt.
\end{aligned}
$$

Die Euler-Maclaurinsche Summenformel läßt sich damit alternativ zu (14) darstellen als:

$$T^{[k]}f(\vec{x}) \quad = \quad I^{[k]}f(\vec{x}) + \delta_1^{[k]}f(\vec{x}) + \delta_2^{[k]}f(\vec{x}) + \ldots + \delta_m^{[k]}f(\vec{x}) + D_{m+1}^{[k]}f(\vec{x}).$$

Da die Operatoren der Definition 3.6 im folgenden nur auf die Funktion $f(\vec{x})$ angewendet werden, soll dieses Argument aus Gründen der Übersichtlichkeit weggelassen werden. Wir erhalten also:

$$T^{[k]} \quad = \quad I^{[k]} + \delta_1^{[k]} + \delta_2^{[k]} + \ldots + \delta_m^{[k]} + D_{m+1}^{[k]}.$$

Aus der Herleitung der Euler-Maclaurinschen Summenformel ist bekannt, daß sich die rechte Seite dieser Gleichung unter der Voraussetzung $f \in C^{2m+4}$ mittels zweimaliger partieller Integration von $D_{m+1}^{[k]}$ um ein Glied erweitern läßt zu:

$$T^{[k]} \quad = \quad I^{[k]} + \delta_1^{[k]} + \delta_2^{[k]} + \ldots + \delta_m^{[k]} + \delta_{m+1}^{[k]} + D_{m+2}^{[k]}.$$

Es gilt also der Expansionsschritt

$$D_w^{[k]} \quad = \quad \delta_w^{[k]} + D_{w+1}^{[k]} \quad \text{für} \quad w > 0. \tag{21}$$

Erweitert man Definition 3.6 formal für $w = 0$ durch

$$\begin{aligned} D_0^{[k]} &:= T^{[k]} \\ \delta_0^{[k]} &:= I^{[k]}, \end{aligned}$$

so gilt (21) auch für $w = 0$, denn

$$D_0^{[k]} = T^{[k]} = I^{[k]} + D_1^{[k]} = \delta_0^{[k]} + D_1^{[k]}.$$

Die Euler-Maclaurinsche Summenformel der Form

$$T^{[k]} = D_0^{[k]} = \delta_0^{[k]} + \delta_1^{[k]} + \delta_2^{[k]} + \ldots + \delta_m^{[k]} + D_{m+1}^{[k]}$$

ergibt sich folglich nach $(m+1)$-maliger Anwendung des Expansionsschritts

$$D_w^{[k]} \quad = \quad \delta_w^{[k]} + D_{w+1}^{[k]} \quad \text{für} \quad w \geq 0 \tag{22}$$

auf $T^{[k]} = D_0^{[k]}$.

3.3.2. Die Euler-Maclaurinsche Summenformel für den mehrdimensionalen Fall. Für die multivariate Trapezsumme $T\left(\vec{h}\right) = T(h_1, h_2, \ldots, h_d)$ soll nun bei vorgegebener Multiindexmenge $\mathcal{I}$ eine Fehlerentwicklung der Form

$$T\left(\vec{h}\right) \quad = \quad \sum_{\vec{\imath} \in \mathcal{I}} \kappa_{\vec{\imath}} \cdot h_1^{2i_1} \cdot h_2^{2i_2} \cdot \ldots \cdot h_d^{2i_d} + R(\mathcal{I}) \tag{23}$$

hergeleitet werden, eine mehrdimensionale Verallgemeinerung der Euler-Maclaurinschen Summenformel. Es sei gefordert, daß die Fehlergliedkoeffizienten $\kappa_{\vec{\imath}}, \vec{\imath} \in \mathcal{I}$ nicht von $\vec{h}$

abhängen und die einzelnen Restglieder, die in $R(\mathcal{I})$ aufsummiert sind, betragsmäßig abgeschätzt werden können. Im folgenden stellen wir ein Verfahren zur systematischen Herleitung einer solchen Entwicklung für beliebige Multiindexmengen $\mathcal{I}$ vor. Für den Fall $\mathcal{I} = \mathcal{I}_m^{(d)}$ wird die explizite Form der Entwicklung angegeben.

Wir stützen uns auf folgenden Glattheitsbegriff:

Definition 3.7. *Es sei $f : H^d \to \mathbb{R}$ und eine Multiindexmenge $\mathcal{I} \neq \emptyset$ gegeben, dann gilt:*

$$f \in C^{\mathcal{I}}\left[H^d\right] \quad \Leftrightarrow \quad f^{(2(\vec{\imath}+\vec{e}_k))} \in C\left[H^d\right] \quad \forall \vec{\imath} \in \mathcal{I} \quad \textit{und} \quad k = 1, \ldots, d.$$

Wir betrachten für eine Funktion $f(\vec{x}) \in C^{\mathcal{I}}\left[H^d\right]$ die Trapezsumme

$$T\left(\vec{h}\right) = D_{\vec{0}} = D_0^{[1]} D_0^{[2]} \cdots D_0^{[d]}. \tag{24}$$

Wenden wir hierauf den Extrapolationsschritt (22) auf eine Richtung $r \in \{1, \ldots, d\}$ mit $w = 0$ an, so erhalten wir:

$$\begin{aligned} D_{\vec{0}} &= D_0^{[1]} \cdots D_0^{[r-1]} \delta_0^{[r]} D_0^{[r+1]} \cdots D_0^{[d]} \\ &+ D_0^{[1]} \cdots D_0^{[r-1]} D_1^{[r]} D_0^{[r+1]} \cdots D_0^{[d]}. \end{aligned}$$

Durch fortgesetzte Anwendung dieses Schritts auf weitere Summanden der rechten Seite und geeignete Richtungen kann die Gleichung gezielt zu einer Entwicklung für $T\left(\vec{h}\right)$ gemäß (23) umgeformt werden. Diese Strategie mündet in folgendes Verfahren:

Verfahren 3.8. *Es sei eine endliche Multiindexmenge $\mathcal{I} \neq \emptyset$ gegeben, und es sei $f \in C^{\mathcal{I}}\left[H^d\right]$. Führe folgende Schritte aus:*

Schritt 0: *Gleichung 0 ergibt sich zu:*

$$T\left(\vec{h}\right) = D_0^{[1]} D_0^{[2]} \cdots D_0^{[d]}.$$

Schritt $n > 0$: *Wähle aus der rechten Seite von Gleichung $(n-1)$ einen Summanden $X_{\vec{\imath}} = X_{i_1}^{[1]} X_{i_2}^{[2]} \cdots X_{i_d}^{[d]}$ ($X_{i_k}^{[k]}$ steht für $D_{i_k}^{[k]}$ oder $\delta_{i_k}^{[k]}$) mit $\vec{\imath} \in \mathcal{I}$ sowie eine Koordinatenrichtung $r \in \{1, \ldots, d\}$, für die gelten:*

$$X_{i_r}^{[r]} \equiv D_{i_r}^{[r]}.$$

Dann ergibt sich Gleichung n durch Anwendung des Expansionsschritts

$$\begin{aligned} X_{i_1}^{[1]} \cdots X_{i_{r-1}}^{[r-1]} \; D_{i_r}^{[r]} \; X_{i_{r+1}}^{[r+1]} \cdots X_{i_d}^{[d]} &= \\ X_{i_1}^{[1]} \cdots X_{i_{r-1}}^{[r-1]} \; \delta_{i_r}^{[r]} \; X_{i_{r+1}}^{[r+1]} \cdots X_{i_d}^{[d]} & \\ + \; X_{i_1}^{[1]} \cdots X_{i_{r-1}}^{[r-1]} \; D_{i_r+1}^{[r]} \; X_{i_{r+1}}^{[r+1]} \cdots X_{i_d}^{[d]} & \end{aligned}$$

auf Gleichung $(n-1)$.

Ende: *Das Verfahren ist beendet, wenn gilt:*

$$X_{\vec{\imath}} \equiv \delta_{\vec{\imath}} \quad \forall\, \vec{\imath} \in \mathcal{I}.$$

Um den Expansionsschritt durchführen zu können, muß bei gegebenem $\vec{\imath} \in \mathcal{I}$ und r die Bedingung $f^{(2(\vec{\imath}+\vec{e}_r))} \in C[H^d]$ erfüllt sein. Dies ist aber durch die Voraussetzung $f \in C^{\mathcal{I}}\left[H^d\right]$ gewährleistet. Die Endlichkeit von Indexmenge $\mathcal{I}$ und Dimension d garantieren die Terminierung des Verfahrens.

Das Verfahren liefert eine Formel der Gestalt:

$$T\left(\vec{h}\right) = \sum_{\vec{\imath}\in\mathcal{I}} \delta_{\vec{\imath}} + \sum_{\vec{\imath}\in\tilde{\mathcal{I}}} X_{\vec{\imath}}$$

mit $\mathcal{I} \cap \tilde{\mathcal{I}} = \emptyset$. Diese Formel enthält die gewünschten Eigenschaften, denn einerseits gilt:

$$\delta_{\vec{\imath}} = \kappa_{\vec{\imath}} \cdot h_1^{2i_1} \cdot h_2^{2i_2} \cdot \ldots \cdot h_d^{2i_d},$$

wobei $\kappa_{\vec{\imath}}$ unabhängig von $\vec{h}$ ist, andererseits lassen sich $X_{\vec{\imath}}$ für $\vec{\imath} \in \tilde{\mathcal{I}}$ mit Hilfe des Mittelwertsatzes analog zum eindimensionalen Fall abschätzen zu

$$|X_{\vec{\imath}}| \leq A_{\vec{\imath}} \cdot h_1^{2i_1} \cdot h_2^{2i_2} \cdot \ldots \cdot h_d^{2i_d},$$

wobei $A_{\vec{\imath}}$ unabhängig von $\vec{h}$ ist. Allerdings ist die Form der Restglieder $X_{\vec{\imath}}, \vec{\imath} \in \tilde{\mathcal{I}}$ nach obigem Verfahren nicht eindeutig festgelegt, wie das folgende Beispiel zeigt. Wir betrachten die Indexmenge $\mathcal{I}_2^{(2)}$ und führen im ersten Fall die Expansionsschritte zuerst in Richtung $k = 1$, dann in $k = 2$ aus. Wir erhalten:

$$\begin{aligned}
D_0^{[1]} D_0^{[2]} &= \left(\delta_0^{[1]} + \delta_1^{[1]} + \delta_2^{[1]} + D_3^{[1]}\right) D_0^{[2]} \\
&= \delta_0^{[1]}\delta_0^{[2]} + \delta_0^{[1]}\delta_1^{[2]} + \delta_0^{[1]}\delta_2^{[2]} + \delta_0^{[1]} D_3^{[2]} \\
&+ \delta_1^{[1]}\delta_0^{[2]} + \delta_1^{[1]}\delta_1^{[2]} + \delta_1^{[1]} D_2^{[2]} \\
&+ \delta_2^{[1]}\delta_0^{[2]} + \delta_2^{[1]} D_1^{[2]} \\
&+ D_3^{[1]} D_0^{[2]} \\
&= \sum_{i_1+i_2\leq 2} \delta_{i_1}^{[1]}\delta_{i_2}^{[2]} + \delta_0^{[1]} D_3^{[2]} + \delta_1^{[1]} D_2^{[2]} + \delta_2^{[1]} D_1^{[2]} + D_3^{[1]} D_0^{[2]}.
\end{aligned}$$

Eine Vertauschung der Expansionsrichtungen führt hingegen zu:

$$\begin{aligned}
D_0^{[1]} D_0^{[2]} &= D_0^{[1]} \left(\delta_0^{[2]} + \delta_1^{[2]} + \delta_2^{[2]} + D_3^{[2]}\right) \\
&= \delta_0^{[1]}\delta_0^{[2]} + \delta_0^{[1]}\delta_1^{[2]} + \delta_0^{[1]}\delta_2^{[2]} + D_0^{[1]} D_3^{[2]} \\
&+ \delta_1^{[1]}\delta_0^{[2]} + \delta_1^{[1]}\delta_1^{[2]} + D_1^{[1]} \delta_2^{[2]} \\
&+ \delta_2^{[1]}\delta_0^{[2]} + D_2^{[1]} \delta_1^{[2]} \\
&+ D_3^{[1]} \delta_0^{[2]} \\
&= \sum_{i_1+i_2\leq 2} \delta_{i_1}^{[1]}\delta_{i_2}^{[2]} + D_0^{[1]} D_3^{[2]} + D_1^{[1]} \delta_2^{[2]} + D_2^{[1]} \delta_1^{[2]} + D_3^{[1]} \delta_0^{[2]}.
\end{aligned}$$

Der Nichtdeterminismus des Restgliedes hat seine Ursache also in der nicht näher festgelegten Reihenfolge, in der die Expansionsschritte auf *einen* Summanden angewandt werden, sofern dieser mehrere Operatoren der Form $D_{i_r}^{[r]}$ besitzt.

Für die Abschätzung des Verfahrensfehlers, die wir später angeben werden, benötigen wir Restglieder, deren Operatoren $D_{i_r}^{[r]}$ keine kleinen i_r enthalten. Um dies zu erreichen, soll der Nichtdeterminismus aufgegeben werden.

VERFAHREN 3.9. *Genauso wie Verfahren 3.8, jedoch mit folgender Verschärfung des Expansionsschritts: Existieren für den gewählten Summanden $X_{\vec{\imath}}, \vec{\imath} \in \mathcal{I}$ mehrere Koordinatenrichtungen $r_1 < r_2 < \ldots < r_n$ mit*

$$X_{i_{r_\xi}}^{[r_\xi]} \equiv D_{i_{r_\xi}}^{[r_\xi]}, \quad \xi \in \{1, \ldots, n\},$$

so wird diejenige Richtung r_j ausgewählt, die folgenden Bedingungen genügt:

$$\begin{aligned} &i_{r_j} = \min\left(i_{r_1}, i_{r_2}, \ldots, i_{r_n}\right) \\ &\forall r_\xi \ \textit{mit}\ i_{r_j} = i_{r_\xi} : r_j < r_\xi. \end{aligned} \tag{25}$$

Die Verschärfung (25) wählt also unter allen Operatoren $D_{i_\xi}^{[\xi]}, \xi = 1, \ldots, n$ eines Summanden nur denjenigen mit minimalem Index i_r aus. Existieren mehrere, so wird der bezüglich der Richtungsnumerierung kleinste ausgewählt. Demnach erfolgt die Expansion für $D_2^{[1]}\delta_0^{[2]}D_2^{[3]}$ in der Reihenfolge $1, 3$ und für $D_2^{[1]}\delta_0^{[2]}D_0^{[3]}$ in der Reihenfolge $3, 1$. Für letzteren liefert die umgekehrte Expansionsreihenfolge:

$$\begin{aligned} D_2^{[1]}\delta_0^{[2]}D_0^{[3]} &= \delta_2^{[1]}\delta_0^{[2]}D_0^{[3]} + D_3^{[1]}\delta_0^{[2]}D_0^{[3]} \\ &= \delta_2^{[1]}\delta_0^{[2]}\delta_0^{[3]} + \delta_2^{[1]}\delta_0^{[2]}D_1^{[3]} + D_3^{[1]}\delta_0^{[2]}D_0^{[3]}, \end{aligned}$$

in der der Summand $D_3^{[1]}\delta_0^{[2]}D_0^{[3]}$ wegen $D_0^{[3]}$ ungünstig abgeschätzt werden kann.

Verfahren 3.9 enthält jetzt zwar noch einen Nichtdeterminismus hinsichtlich der Auswahlreihenfolge der Summanden, diese ist aber für die Eindeutigkeit der Formel unerheblich.

Als Ergebnis erhalten wir eine mehrdimensionale Verallgemeinerung der Euler-Maclaurinschen Summenformel der Form:

$$D_{\vec{0}} = \sum_{\vec{\imath} \in \mathcal{I}} \delta_{\vec{\imath}} + \sum_{\vec{\imath} \in \tilde{\mathcal{I}}} X_{\vec{\imath}},$$

wobei $\tilde{\mathcal{I}}$ und $X_{\vec{\imath}}, \vec{\imath} \in \tilde{\mathcal{I}}$ nun eindeutig bestimmt sind. Für den Fall $\mathcal{I} = \mathcal{I}_m^{(d)}$ gibt der folgende Satz explizite Darstellungen für $\tilde{\mathcal{I}}_m^{(d)}$ und $X_{\vec{\imath}}$ mit $\vec{\imath} \in \tilde{\mathcal{I}}_m^{(d)}$ an:

SATZ 3.10. *Es seien $m \geq 0, d > 0, \mathcal{I}_m^{(d)}$ und $f \in C^{\mathcal{I}_m^{(d)}}\left[H^d\right]$ gegeben.*

$$D_{\vec{0}} = \sum_{\vec{\imath} \in \mathcal{I}_m^{(d)}} \delta_{\vec{\imath}} + \sum_{\vec{\imath} \in \tilde{\mathcal{I}}_m^{(d)}} X_{\vec{\imath}}$$

ist nach Verfahren 3.9 hergeleitet worden. Dann gilt:

$$\tilde{\mathcal{I}}_m^{(d)} \;=\; \left\{\vec{\imath} \;\middle|\; |\vec{\imath}| = m+1\right\}. \tag{26}$$

Für $X_{\vec{\imath}}$, $\vec{\imath} \in \tilde{\mathcal{I}}_m^{(d)}$ *und* $r \in \{1,\ldots,d\}$ *ist:*

$$\begin{aligned} X_{i_r}^{[r]} \equiv D_{i_r}^{[r]} \quad &\Leftrightarrow \quad r \in I^D(\vec{\imath}) \\ X_{i_r}^{[r]} \equiv \delta_{i_r}^{[r]} \quad &\Leftrightarrow \quad r \in I^\delta(\vec{\imath}), \end{aligned} \tag{27}$$

wobei gilt:

$$\begin{aligned} I^D(\vec{\imath}) \quad &= \quad \Big\{ r \;\Big|\; \begin{array}{l} i_r = \max(\vec{\imath}) \quad \textit{oder} \\ i_r = \max(\vec{\imath}) - 1 \quad \textit{und} \;\; \forall \xi, r < \xi \le d : i_r \ne \max(\vec{\imath}) \end{array} \Big\} \\ r \in I^\delta(\vec{\imath}) \quad &\Leftrightarrow \quad r \notin I^D(\vec{\imath}). \end{aligned}$$

Bevor der Satz bewiesen wird, sollen die Aussagen durch zwei Beispiele veranschaulicht werden. Es sei die Multiindexmenge $\mathcal{I}_{10}^{(5)}$ gegeben, dann gilt: $\tilde{\mathcal{I}}_{10}^{(5)} = \left\{\vec{\imath} \;\middle|\; |\vec{\imath}| = 11\right\}$. Die Restglieder für folgende Indizes aus $\tilde{\mathcal{I}}_{10}^{(5)}$ ergeben sich gemäß:

$$\begin{aligned} \vec{\imath} = (3,2,1,3,2) \Rightarrow I^D(\vec{\imath}) = \{1,4,5\} \Rightarrow I^\delta(\vec{\imath}) = \{2,3\} \Rightarrow X_{\vec{\imath}} \equiv D_3^{[1]}\delta_2^{[2]}\delta_1^{[3]}D_3^{[4]}D_2^{[5]} \\ \vec{\imath} = (3,2,2,3,1) \Rightarrow I^D(\vec{\imath}) = \{1,4\} \Rightarrow I^\delta(\vec{\imath}) = \{2,3,5\} \Rightarrow X_{\vec{\imath}} \equiv D_3^{[1]}\delta_2^{[2]}\delta_2^{[3]}D_3^{[4]}\delta_1^{[5]}. \end{aligned}$$

Beweis. Die Behauptungen (27) und (26) sollen induktiv über die *Metaexpansionsschritte* des Verfahrens 3.9 gezeigt werden. Unter einem Metaexpansionsschritt verstehen wir dabei die Zusammenfassung *aller* Expansionsschritte auf *einen* Summanden mit beliebigem Multiindex $\vec{\imath} \in \mathcal{I}_m^{(d)}$. Nach einem solchen Schritt gilt also: $X_{\vec{\imath}} \equiv \delta_{\vec{\imath}}$.

Wir zeigen zunächst, daß bei ausschließlicher Verwendung von Metaexpansionsschritten Eigenschaft (27) sogar für alle Summanden der Form $X_{\vec{\imath}} \not\equiv \delta_{\vec{\imath}}$ während der Formelumformung gilt. Daraus folgt insbesondere Behauptung (27) für $X_{\vec{\imath}}, \vec{\imath} \in \tilde{\mathcal{I}}_m^{(d)}$, denn zum einen werden die Summanden der Form $X_{\vec{\imath}} \equiv \delta_{\vec{\imath}}$ durch Verfahren 3.9 nur für $\vec{\imath} \in \mathcal{I}_m^{(d)}$ erzeugt, und zum anderen gilt $\mathcal{I}_m^{(d)} \cap \tilde{\mathcal{I}}_m^{(d)} = \emptyset$.

Vereinfachend schreiben wir $\alpha := \max(\vec{\imath})$, außerdem bezeichne δ^* in einem Ausdruck alle nicht explizit aufgeführten Operatoren der Form $\delta_{i_k}^{[k]}$.

Induktionsanfang: Nach Schritt 0 von Verfahren 3.9 besitzt die Summenformel folgende Gestalt:

$$T\left(\vec{h}\right) \;=\; D_0^{[1]}D_0^{[2]}\cdots D_0^{[d]} \equiv X_{\vec{0}}.$$

Dies ergibt sich ebenfalls für Behauptung (27), denn wegen $I^D\left(\vec{0}\right) = \{1,2,\ldots,d\}$ gilt: $X_{\vec{0}} \equiv D_0^{[1]}D_0^{[2]}\cdots D_0^{[d]}$.

Induktionsschritt: Nach Induktionsvoraussetzung sind die Expansionsschritte für alle $\vec{\imath}$ aus einer Indexmenge $\mathcal{J} \subset \mathcal{I}$ durchgeführt. Für alle Summanden $X_{\vec{\imath}}, \vec{\imath} \in \mathcal{J}$

gilt also: $X_{\vec{\imath}} \equiv \delta_{\vec{\imath}}$, und alle übrigen $X_{\vec{\imath}}, \vec{\imath} \in \tilde{\mathcal{J}}$ erfüllen (27). Wir haben zu zeigen, daß nach Ausführung eines Metaexpansionsschritts auf einen Summanden $X'_{\vec{\imath}}, \vec{\imath} \in \tilde{\mathcal{J}} \cap \mathcal{I}$ Eigenschaft (27) für alle neu hinzugekommenen Summanden der Formel gilt. Wir betrachten alle $n = \left|I^D(\vec{\imath})\right|$ Koordinatenrichtungen $r_1 < r_2 < \ldots < r_n$ mit $r_\xi \in I^D(\vec{\imath}), \xi = 1, \ldots, n$. Da $X'_{\vec{\imath}}$ Bedingung (27) erfüllt (Induktionsvoraussetzung), gibt es ein $k \in \{1, \ldots, n\}$, so daß

$$i_{r_\xi} = \begin{cases} \alpha & \text{für } 1 \leq \xi \leq k \\ \alpha - 1 & \text{für } k < \xi \leq n \end{cases} \tag{28}$$

gilt. $X'_{\vec{\imath}}$ besitzt also vor dem Metaexpansionsschritt die Form

$$X'_{\vec{\imath}} \equiv D_\alpha^{[r_1]} D_\alpha^{[r_2]} \cdots D_\alpha^{[r_k]} D_{\alpha-1}^{[r_{k+1}]} D_{\alpha-1}^{[r_{k+2}]} \cdots D_{\alpha-1}^{[r_n]} \delta^*.$$

Nach Voraussetzung gilt

$$\forall s \in I^\delta(\vec{\imath}): \qquad i_s \leq \alpha - 1,$$

denn die gegenteilige Annahme führt zum Widerspruch:

$$i_s > \alpha - 1 \quad \Rightarrow \quad i_s = \alpha \quad \Rightarrow \quad s \in I^D(\vec{\imath}) \quad \Rightarrow \quad s \notin I^\delta(\vec{\imath}).$$

Nach (25) wird der Summand $X'_{\vec{\imath}}$ nacheinander in den Richtungen r_{k+1}, $r_{k+2}, \ldots, r_n, r_1$, $r_2 \ldots r_k$ expandiert. Es ergeben sich folgende Umformungsschritte von $X'_{\vec{\imath}}$:

$$\begin{array}{cccccccccc}
 & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & D_{\alpha-1}^{[r_{k+1}]} & D_{\alpha-1}^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
\overset{r_{k+1}}{\equiv} & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & D_{\alpha-1}^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
+ & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & D_\alpha^{[r_{k+1}]} & D_{\alpha-1}^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
\overset{r_{k+2}}{\equiv} & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & \delta_{\alpha-1}^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
+ & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & D_\alpha^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
+ & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & D_\alpha^{[r_{k+1}]} & D_{\alpha-1}^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
= & & & & & & & & & \\
\vdots & & & & & & & & & \\
\overset{r_k}{\equiv} & \delta_\alpha^{[r_1]} & \delta_\alpha^{[r_2]} & \cdots & \delta_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & \delta_{\alpha-1}^{[r_{k+2}]} & \cdots & \delta_{\alpha-1}^{[r_n]} & \delta^* \\
+ & \delta_\alpha^{[r_1]} & \delta_\alpha^{[r_2]} & \cdots & D_{\alpha+1}^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & \delta_{\alpha-1}^{[r_{k+2}]} & \cdots & \delta_{\alpha-1}^{[r_n]} & \delta^* \\
\vdots & & & & & & & & & \\
+ & \delta_\alpha^{[r_1]} & D_{\alpha+1}^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & \delta_{\alpha-1}^{[r_{k+2}]} & \cdots & \delta_{\alpha-1}^{[r_n]} & \delta^* \\
+ & D_{\alpha+1}^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & \delta_{\alpha-1}^{[r_{k+2}]} & \cdots & \delta_{\alpha-1}^{[r_n]} & \delta^* \\
+ & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & \delta_{\alpha-1}^{[r_{k+2}]} & \cdots & D_\alpha^{[r_n]} & \delta^* \\
\vdots & & & & & & & & & \\
+ & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & \delta_{\alpha-1}^{[r_{k+1}]} & D_\alpha^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^* \\
+ & D_\alpha^{[r_1]} & D_\alpha^{[r_2]} & \cdots & D_\alpha^{[r_k]} & D_\alpha^{[r_{k+1}]} & D_{\alpha-1}^{[r_{k+2}]} & \cdots & D_{\alpha-1}^{[r_n]} & \delta^*.
\end{array} \tag{29}$$

Wie man sieht, erfüllen nach der letzten Umformung alle Summanden, in denen der Operator D enthalten ist, Behauptung (27).

Behauptung (26) zeigen wir durch Induktion über m.

Induktionsanfang ($m = 0$): Verfahren 3.9 expandiert den Summanden $D_{\vec{0}}$ gemäß (25) nacheinander in den Koordinatenrichtungen $1, 2, \ldots, d$, so daß sich ergibt:

$$\begin{aligned}
D_0^{[1]} D_0^{[2]} \cdots D_0^{[d]} &= \delta_0^{[1]} \delta_0^{[2]} \cdots \delta_0^{[d]} \\
&+ \delta_0^{[1]} \delta_0^{[2]} \cdots \delta_0^{[d-1]} D_1^{[d]} \\
&+ \delta_0^{[1]} \delta_0^{[2]} \cdots \delta_0^{[d-2]} D_1^{[d-1]} D_0^{[d]} \\
&\vdots \\
&+ \delta_0^{[1]} D_1^{[2]} D_0^{[3]} \cdots D_0^{[d]} \\
&+ D_1^{[1]} D_0^{[2]} \cdots D_0^{[d]} \\
&= \delta_{\vec{0}} + \sum_{\vec{\imath} \in \tilde{\mathcal{I}}_1^{(d)}} X_{\vec{\imath}},
\end{aligned}$$

wodurch für $m = 0$ die Gültigkeit von Behauptung (26) gezeigt ist.

Induktionsschritt ($m \to m+1$): Nach Voraussetzung gilt also

$$\tilde{\mathcal{I}}_m^{(d)} = \left\{ \vec{\imath} \;\middle|\; |\vec{\imath}| = m + 1 \right\},$$

womit die entsprechende Summenformel lautet:

$$D_{\vec{0}} = \sum_{|\vec{\imath}| \leq m} \delta_{\vec{\imath}} + \sum_{|\vec{\imath}| = m+1} X_{\vec{\imath}}. \tag{30}$$

Um die Formel für $\mathcal{I} = \mathcal{I}_{m+1}^{(d)}$ zu erhalten, sind noch die Metaexpansionsschritte auf die Summanden $X_{\vec{\imath}}, |\vec{\imath}| = m + 1$ durchzuführen. Aus (29) ist ersichtlich, daß dadurch für beliebiges $\vec{\imath}$ mit $|\vec{\imath}| = m$ der Summand $\delta_{\vec{\imath}}$ und einige Summanden $X_{\vec{\jmath}_1}$, $X_{\vec{\jmath}_2}$, $X_{\vec{\jmath}_3} \ldots$ mit $|\vec{\jmath}_1| = |\vec{\jmath}_2| = |\vec{\jmath}_3| = \ldots = m + 2$ erzeugt werden. Wir geben nun für einen beliebigen Summanden $X_{\vec{\jmath}}, |\vec{\jmath}| = m + 2$ den entsprechenden Summanden $X_{\vec{\imath}}, |\vec{\imath}| = m + 1$ an, aus dem dieser durch Metaexpansion hergeleitet wurde. Daraus folgt schließlich, daß die Summenformel (30) übergeht zu

$$D_{\vec{0}} = \sum_{|\vec{\imath}| \leq m+1} \delta_{\vec{\imath}} + \sum_{|\vec{\imath}| = m+2} X_{\vec{\imath}},$$

womit die Behauptung gezeigt ist.

Es seien alle Richtungen $r_\xi \in I^D(\vec{\jmath})$ aufsteigend sortiert: $r_1 < r_2 < \ldots < r_n$ ($n = |I^D(\vec{\jmath})|$). Dann gibt es genau ein $k \in \{1, \ldots, n\}$, das (28) erfüllt. Wir unterscheiden zwei Fälle:

Fall 1 ($k \neq 1$): Es seien $s_1 < s_2 < \ldots < s_{n'}$ alle Koordinatenrichtungen $s_\xi, \xi \in \{1, \ldots, n'\}$ mit:

$$s_\xi \in I^\delta(\vec{\jmath}) \quad \wedge \quad r_{k-1} < s_\xi < r_k \quad \wedge \quad j_{s_\xi} = \alpha - 1.$$

$X_{\vec{\jmath}}$ besitzt also die Form:

$$X_{\vec{\jmath}} \equiv D_{\alpha}^{[r_1]} D_{\alpha}^{[r_2]} \cdots D_{\alpha}^{[r_{k-1}]} \delta_{\alpha-1}^{[s_1]} \delta_{\alpha-1}^{[s_2]} \cdots \delta_{\alpha-1}^{[s_{n'}]} D_{\alpha}^{[r_k]} D_{\alpha-1}^{[r_{k+1}]} \cdots D_{\alpha-1}^{[r_n]} \delta^{*}.$$

Dann ist $X_{\vec{\jmath}}$ aus

$$X_{\vec{\imath}} \equiv D_{\alpha}^{[r_1]} D_{\alpha}^{[r_2]} \cdots D_{\alpha}^{[r_{k-1}]} D_{\alpha-1}^{[s_1]} D_{\alpha-1}^{[s_2]} \cdots D_{\alpha-1}^{[s_{n'}]} D_{\alpha-1}^{[r_k]} D_{\alpha-1}^{[r_{k+1}]} \cdots D_{\alpha-1}^{[r_n]} \delta^{*}$$

entstanden, nachdem in Richtung $s_1, s_2, \ldots, s_{n'}, r_k$ expandiert worden ist.
Fall 2 ($k = 1$): Es seien $s_1 < s_2 < \ldots < s_{n'}$ und $t_1 < t_2 < \ldots t_{n''}$ alle Koordinatenrichtungen $s_\xi, 1 \leq \xi \leq n'$ und $t_\eta, 1 \leq \eta \leq n''$ mit

$$\begin{array}{ccccc} s_\xi \in I^{\delta}(\vec{\jmath}) & \wedge & 1 \leq s_\xi < r_1 & \wedge & j_{s_\xi} = \alpha - 1 \\ t_\eta \in I^{\delta}(\vec{\jmath}) & \wedge & r_n < t_\eta \leq d & \wedge & j_{t_\eta} = \alpha - 2. \end{array}$$

$X_{\vec{\imath}}$ besitzt also die Form:

$$X_{\vec{\imath}} \equiv \delta_{\alpha-1}^{[s_1]} \delta_{\alpha-1}^{[s_2]} \cdots \delta_{\alpha-1}^{[s_{n'}]} D_{\alpha}^{[r_1]} D_{\alpha-1}^{[r_2]} \cdots D_{\alpha-1}^{[r_n]} \delta_{\alpha-2}^{[t_1]} \delta_{\alpha-2}^{[t_2]} \cdots \delta_{\alpha-2}^{[t_{n''}]} \delta^{*}.$$

Dann ist $X_{\vec{\imath}}$ aus

$$X_{\vec{\jmath}} \equiv D_{\alpha-1}^{[s_1]} D_{\alpha-1}^{[s_2]} \cdots D_{\alpha-1}^{[s_{n'}]} D_{\alpha-1}^{[r_1]} D_{\alpha-1}^{[r_2]} \cdots D_{\alpha-1}^{[r_n]} D_{\alpha-2}^{[t_1]} D_{\alpha-2}^{[t_2]} \cdots D_{\alpha-2}^{[t_{n''}]} \delta^{*}$$

entstanden, nachdem in Richtung $t_1, t_2, \ldots t_{n''}, s_1, s_2, \ldots, s_{n'}, r_1$ expandiert worden ist.
$\square$

Wir legen nun für $k = 1, \ldots, d$ und $w \in \mathbb{N}_0$ fest:

$$\begin{array}{rcl} D_0^{[k]}(h_k) & = & D_0^{[k]} \\ \dot{\delta}_w^{[k]} \cdot h_k^{2w} & = & \delta_w^{[k]} \\ \dot{D}_w^{[k]}(h_k) \cdot h_k^{2w} & = & D_w^{[k]}. \end{array}$$

Entsprechend gilt entweder $\dot{X}_{i_k}^{[k]} \equiv \dot{\delta}_{i_k}^{[k]}$ oder $\dot{X}_{i_k}^{[k]} \equiv \dot{D}_{i_k}^{[k]}$. Unter Verwendung dieser Schreibweisen fassen wir das Ergebnis dieses Abschnitts zusammen:

KOROLLAR 3.11. *Es sei eine Multiindexmenge $\mathcal{I}$, eine Funktion $f \in C^{\mathcal{I}}\left[H^d\right]$ und eine Schrittweite $\vec{h} = (h_1, h_2, \ldots, h_d)$ gegeben. Dann gilt:*

$$D_{\vec{0}}\left(\vec{h}\right) = \sum_{\vec{\imath} \in \mathcal{I}} \dot{\delta}_{\vec{\imath}} \cdot h^{2\vec{\imath}} + \sum_{\vec{\imath} \in \tilde{\mathcal{I}}} \dot{X}_{\vec{\imath}}\left(\vec{h}\right) \cdot h^{2\vec{\imath}} \quad \text{mit} \quad \mathcal{I} \cap \tilde{\mathcal{I}} = \emptyset. \tag{31}$$

Für den Fall $\mathcal{I} = \mathcal{I}_m^{(d)}$ gilt:

$$
\begin{aligned}
D_{\vec{0}}\left(\vec{h}\right) &= \sum_{|\vec{\imath}|\leq m} \dot{\delta}_{\vec{\imath}} \cdot h^{2\vec{\imath}} + \sum_{|\vec{\imath}|=m+1} \dot{X}_{\vec{\imath}}\left(\overline{h}\right) \cdot h^{2\vec{\imath}} \\
\textit{mit}\ \ \dot{X}_{\vec{\imath}}\left(\overline{h}\right) &\equiv \dot{D}_{i_{r_1}}^{[r_1]}\left(h_{r_1}\right) \dot{D}_{i_{r_2}}^{[r_2]}\left(h_{r_2}\right) \cdots \dot{D}_{i_{r_n}}^{[r_n]}\left(h_{r_n}\right) \dot{\delta}_{i_{s_1}}^{[s_1]} \dot{\delta}_{i_{s_2}}^{[s_2]} \cdots \dot{\delta}_{i_{s_{n'}}}^{[s_{n'}]} \\
\overline{h} &= \left(h_{r_1}, h_{r_2}, \ldots, h_{r_n}\right) \\
r_\xi &\in I^D\left(\vec{\imath}\right) \quad \textit{mit} \quad \xi = 1, \ldots, n \quad \textit{und} \quad n = \left|I^D\left(\vec{\imath}\right)\right| \\
s_\xi &\in I^\delta\left(\vec{\imath}\right) \quad \textit{mit} \quad \xi = 1, \ldots, n' \quad \textit{und} \quad n' = \left|I^\delta\left(\vec{\imath}\right)\right|.
\end{aligned} \tag{32}
$$

Zwei Eigenschaften von (31) und (32) sind für die anschließenden Fehleruntersuchungen von Bedeutung. Zum einen sind die Fehlergliedkoeffizienten $\dot{\delta}_{\vec{\imath}}$ für $\vec{\imath} \in \mathcal{I}$ unabhängig von der multivariaten Schrittweite $\vec{h}$. Zum anderen läßt sich $\dot{X}_{\vec{\imath}}\left(\overline{h}\right)$ mit $\vec{\imath} \in \tilde{\mathcal{I}}_m^{(d)}$ durch Anwendung des Mittelwertsatzes in alle Koordinatenrichtungen $k = 1, 2, \ldots, d$ durch eine von $\vec{h}$ unabhängige Konstante $A_{\vec{\imath}}$ abschätzen, wenn $f \in C^{\mathcal{I}_m^{(d)}}\left[H^d\right]$ gilt:

$$
\left|\dot{X}_{\vec{\imath}}\left(\overline{h}\right)\right| \leq A_{\vec{\imath}}.
$$

3.3.3. Auslöschung von Fehlergliedern. Wir zeigen nun, daß die Fehlerglieder der Ordnungen $h^{2\vec{\imath}}$, $\vec{\imath} \in \mathcal{I} \setminus \left\{\vec{0}\right\}$ der Entwicklung (31) durch das vorgestellte Quadraturverfahren eliminiert werden. Es können beliebige, streng monoton nach Null fallende Schrittweitenfolgen $H^{[k]} = \{h_{k,0}, h_{k,1}, \ldots\}, k = 1, \ldots, d$ und beliebige Multiindexmengen verwendet werden.

Im folgenden werden Rückwärtsdifferenzen durch einen Querstrich kenntlich gemacht. Für eine Folge $P = \{p_0, p_1, \ldots\}$ gilt dann:

$$
\bar{p}_k := \begin{cases} p_0 & \text{falls } k = 0 \\ p_k - p_{k-1} & \text{falls } k > 0. \end{cases}
$$

Daran anlehnend sei festgelegt:

$$
\begin{aligned}
\bar{1}_k &:= \begin{cases} 1 & \text{falls } k = 0 \\ 0 & \text{falls } k > 0 \end{cases} \\
\bar{1}_{\vec{l}} &:= \bar{1}_{l_1} \cdot \bar{1}_{l_2} \cdot \ldots \cdot \bar{1}_{l_d}.
\end{aligned}
$$

Zunächst werden folgende Eigenschaften gezeigt:

Lemma 3.12. *Es seien die Schrittweitenfolge $H^{[k]} = \{h_{k,0}, h_{k,1}, \ldots\}, k \in \{1, \ldots, d\}$ gegeben, und es gelte:*

$$
p_{l_k, i_k}^{[k]} := \sum_{j=0}^{i_k} c_{i_k, j}^{[k]} \cdot h_{k,j}^{2 l_k} \tag{33}
$$

$$
\begin{aligned}
& mit \;\; c_{i_k,j}^{[k]} &=& \prod_{\substack{\varrho=0\\ \varrho\neq j}}^{i_k} \frac{h_{k,\varrho}^2}{h_{k,\varrho}^2 - h_{k,j}^2} \\
(34) \qquad & \bar{p}_{l_k,i_k}^{[k]} &:=& \begin{cases} p_{l_k,i_k}^{[k]} & falls \;\; i_k = 0 \\ p_{l_k,i_k}^{[k]} - p_{l_k,i_k-1}^{[k]} & falls \;\; i_k > 0 \end{cases} .
\end{aligned}
$$

Dann gilt:

$$
\begin{aligned}
(35) \qquad & p_{l_k,i_k}^{[k]} &=& \bar{1}_{l_k} \quad f\ddot{u}r \;\; i_k \geq l_k \geq 0 \\
(36) \qquad & \bar{p}_{l_k,i_k}^{[k]} &=& 0 \quad f\ddot{u}r \;\; i_k > l_k \geq 0 \\
(37) \qquad & \sum_{j=0}^{l_k} \bar{p}_{l_k,j}^{[k]} &=& \bar{1}_{l_k}.
\end{aligned}
$$

Beweis. Behauptung (35): Es sei $g(x)$ ein beliebiges Polynom der Form

$$
g(x) \;\; = \;\; \sum_{j=0}^{i_k} a_j \cdot x^{2j}
$$

mit festem $i_k \in \mathbb{N}_0$. Dann gilt nach der Lagrangeschen Interpolationsformel und der Eindeutigkeit der Polynominterpolation:

$$
\sum_{j=0}^{i_k} c_{i_k,j}^{[k]} \cdot g(h_{k,j}) \;\; = \;\; g(0).
$$

Wählen wir $g(x) = x^{2l_k}$ mit $l_k \in \{0, 1, \ldots, i_k\}$, so folgt Behauptung (35) wegen:

$$
\begin{aligned}
p_{l_k,i_k}^{[k]} &= \sum_{j=0}^{i_k} c_{i_k,j}^{[k]} \cdot h_{k,j}^{2l_k} \\
&= \sum_{j=0}^{i_k} c_{i_k,j}^{[k]} \cdot g(h_{k,j}) \\
&= g(0) = \bar{1}_{l_k}.
\end{aligned}
$$

Behauptung (36) ergibt sich aus:

$$
\begin{aligned}
\bar{p}_{l_k,i_k}^{[k]} &\overset{i_k \geq 0}{=} p_{l_k,i_k}^{[k]} - p_{l_k,i_k-1}^{[k]} \\
&\overset{(35), i_k > l_k}{=} \bar{1}_{l_k} - \bar{1}_{l_k} = 0.
\end{aligned}
$$

Behauptung (37) läßt sich wie folgt beweisen:

$$
\sum_{j=0}^{l_k} \bar{p}_{l_k,j}^{[k]} \overset{(34)}{=} p_{l_k,0}^{[k]} + \sum_{j=1}^{l_k} \left(p_{l_k,j}^{[k]} - p_{l_k,j-1}^{[k]} \right) = p_{l_k,l_k}^{[k]} \overset{(35)}{=} \bar{1}_{l_k}.
$$

□

Hiermit läßt sich folgendes Lemma zeigen:

LEMMA 3.13. *Wir betrachten den d-dimensionalen Fall. Gegeben seien eine beliebige Multiindexmenge $\mathcal{I}$, für jede Koordinatenrichtung $k = 1, \ldots, d$ eine Folge von Schrittweiten $H^{[k]} = \{h_{k,0}, h_{k,1}, \ldots\}$ und ein Multiindex $\vec{l} \in \mathcal{I}$. Mit $h_{\vec{\imath}}^{2\vec{l}} = h_{1,i_1}^{2l_1} \cdot h_{2,i_2}^{2l_2} \cdots h_{d,i_d}^{2l_d}$ gilt dann:*

$$\Phi\left(h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}} = \bar{1}_{\vec{l}}.$$

Beweis. Unter Ausnutzung der Definitionen 3.2 und 3.4 gilt:

$$\begin{aligned}
& \Phi\left(h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}} \\
= \quad & \Phi\left(h_{1,i_1}^{2l_1} \cdot h_{2,i_2}^{2l_2} \cdot \ldots \cdot h_{d,i_d}^{2l_d}\right)_{\vec{\imath}\in\mathcal{I}} \\
= \quad & \Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\Lambda^{[d]}\Lambda^{[d-1]}\cdots\Lambda^{[1]}\left(h_{1,i_1}^{2l_1} \cdot h_{2,i_2}^{2l_2} \cdot \ldots \cdot h_{d,i_d}^{2l_d}\right)_{\vec{\imath}\in\mathcal{I}} \\
= \quad & \Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\Lambda^{[d]}\Lambda^{[d-1]}\cdots\Lambda^{[2]}\left(\sum_{j=0}^{i_1}\left(c_{i_1,j}^{[1]} \cdot h_{1,j}^{2l_1}\right) \cdot h_{2,i_2}^{2l_2} \cdot \ldots \cdot h_{d,i_d}^{2l_d}\right)_{\vec{\imath}\in\mathcal{I}} \\
\overset{(33)}{=} \quad & \Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\Lambda^{[k]}\Lambda^{[k-1]}\cdots\Lambda^{[3]} \\
& \qquad \left(p_{l_1,i_1}^{[1]} \cdot \sum_{j=0}^{i_2}\left(c_{i_2,j}^{[2]} \cdot h_{2,j}^{2l_2}\right) \cdot h_{3,i_3}^{2l_3} \cdot \ldots \cdot h_{d,i_d}^{2l_d}\right)_{\vec{\imath}\in\mathcal{I}} \\
\vdots \quad & \\
\overset{(33)}{=} \quad & \Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\left(p_{l_1,i_1}^{[1]} \cdot p_{l_2,i_2}^{[2]} \cdot \ldots \cdot p_{l_d,i_d}^{[d]}\right)_{\vec{\imath}\in\mathcal{I}} \\
\overset{(34)}{=} \quad & \Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[2]}\left(\bar{p}_{l_1,i_1}^{[1]} \cdot p_{l_2,i_2}^{[2]} \cdot \ldots \cdot p_{l_d,i_d}^{[d]}\right)_{\vec{\imath}\in\mathcal{I}} \\
\vdots \quad & \\
\overset{(34)}{=} \quad & \Sigma\left(\bar{p}_{l_1,i_1}^{[1]} \cdot \bar{p}_{l_2,i_2}^{[2]} \cdot \ldots \cdot \bar{p}_{l_d,i_d}^{[d]}\right)_{\vec{\imath}\in\mathcal{I}} \\
\overset{(\vec{l}\in\mathcal{I}),(36)}{=} \quad & \sum_{\vec{\imath}\leq\vec{l}} \bar{p}_{l_1,i_1}^{[1]} \cdot \bar{p}_{l_2,i_2}^{[2]} \cdot \ldots \cdot \bar{p}_{l_d,i_d}^{[d]} \\
= \quad & \sum_{i_1=0}^{l_1} \bar{p}_{l_1,i_1}^{[1]} \cdot \sum_{i_2=0}^{l_2} \bar{p}_{l_2,i_2}^{[2]} \cdot \ldots \cdot \sum_{i_d=0}^{l_d} \bar{p}_{l_d,i_d}^{[d]} \\
\overset{(37)}{=} \quad & \bar{1}_{l_1} \cdot \bar{1}_{l_2} \cdot \ldots \cdot \bar{1}_{l_d} = \bar{1}_{\vec{l}}.
\end{aligned}$$

□

Hiermit ergibt sich schließlich die Auslöschung von Fehlergliedern:

Satz 3.14. *Es seien* $\mathcal{I}, f \in C^{\mathcal{I}}\left[H^d\right]$ *sowie die Folgen von Schrittweiten* $H^{[k]} = \{h_{k,0}, h_{k,1}, \ldots\}$ *für* $k = 1, 2, \ldots, d$ *gegeben. Außerdem existiere für* $D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)$ *eine Entwicklung nach (31). Dann gilt:*

$$\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}} = \delta_{\vec{0}} + \sum_{\vec{l}\in\tilde{\mathcal{I}}} \Phi\left(\dot{X}_{\vec{l}}\left(\vec{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}}.$$

Beweis. Es gilt:

$$\begin{aligned}
\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}} &\overset{3.11}{=} \Phi\left(\sum_{\vec{l}\in\mathcal{I}} \dot{\delta}_{\vec{l}} \cdot h_{\vec{\imath}}^{2\vec{l}} + \sum_{\vec{l}\in\tilde{\mathcal{I}}} \dot{X}_{\vec{l}}\left(\vec{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}} \\
&\overset{3.5}{=} \sum_{\vec{l}\in\mathcal{I}} \dot{\delta}_{\vec{l}} \cdot \Phi\left(h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}} + \sum_{\vec{l}\in\tilde{\mathcal{I}}} \Phi\left(\dot{X}_{\vec{l}}\left(\vec{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}} \\
&\overset{3.13}{=} \sum_{\vec{l}\in\mathcal{I}} \dot{\delta}_{\vec{l}} \cdot \bar{1}_{\vec{l}} + \sum_{\vec{l}\in\tilde{\mathcal{I}}} \Phi\left(\dot{X}_{\vec{l}}\left(\vec{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}} \\
&= \delta_{\vec{0}} + \sum_{\vec{l}\in\tilde{\mathcal{I}}} \Phi\left(\dot{X}_{\vec{l}}\left(\vec{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}}.
\end{aligned}$$

□

Das Verfahren eliminiert also die Fehlerglieder $\dot{\delta}_{\vec{l}} \cdot h_{\vec{\imath}}^{2\vec{l}}$ für $\vec{l} \in \mathcal{I} \setminus \left\{\vec{0}\right\}$. Der exakte Wert $\delta_{\vec{0}}$ bleibt erhalten. Da die Multiindexmenge $\mathcal{I}$ beliebig gewählt werden kann, ist eine *gezielte* Auslöschung von Fehlergliedern der Euler-Maclaurinschen Summenformel möglich.

Diese Flexibilität in der Wahl von $\mathcal{I}$ kann für eine adaptive Variante des vorgestellten Quadraturverfahrens genutzt werden. Das Schema

$$\left(u_{\vec{\imath}}^*\right)_{\vec{\imath}\in\mathcal{I}} = \Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\Lambda^{[d]}\Lambda^{[d-1]}\cdots\Lambda^{[1]}\left(Q_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}}$$

enthält die Beiträge, aus denen sich das Endergebnis

$$\Phi\left(Q_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}} = \sum_{\vec{\imath}\in\mathcal{I}} u_{\vec{\imath}}^*$$

zusammensetzt. Durch iterative Ausführung der Schritte: Analyse eines Schemas $\left(u_{\vec{\imath}}^*\right)_{\vec{\imath}\in\mathcal{I}(j)}$, Erweiterung von $\mathcal{I}(j)$ zu einer Multiindexmenge $\mathcal{I}(j+1)$ und Berechnung von $\Phi(Q_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}(j+1)}$ erhält man ein Verfahren mit adaptiver Wahl einer geeigneten Multiindexmenge. Dieser Ansatz soll hier nicht weiter verfolgt werden.

3.3.4. Fehlerabschätzung des Restgliedes. Aus Satz 3.14 geht hervor, daß das vom Quadraturverfahren gelieferte Resultat mit dem Fehler $\sum_{\vec{l}\in\tilde{\mathcal{I}}} \Phi\left(\dot{X}_{\vec{l}}\left(\vec{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}}$

behaftet ist. Es soll nun gezeigt werden, daß der Verfahrensfehler beschränkt ist, wenn eine Multiindexmenge

$$\mathcal{I}_m^{(d)} \;=\; \left\{\vec{\imath} \;\middle|\; |\vec{\imath}| \le m\right\} \quad \text{für} \quad m \ge 0,$$

eine Funktion $f \in C^{\mathcal{I}_m^{(d)}}\left[H^d\right]$ und eine für jede Koordinatenrichtung $k = 1, 2, \ldots, d$ geometrisch mit dem Faktor $b, 0 < b < 1$ fallende Schrittweitenfolge

$$H^{[k]} \;=\; \left\{h_\xi \;\middle|\; h_\xi = h_0 \cdot b^\xi, \xi \in \mathbb{N}_0\right\}$$

vorgegeben ist.

Eine zentrale Rolle nimmt hierbei ein mehrdimensionales Polynom ein, das wir mit

$$p_m^{(d)}\left(z_1, z_2, \ldots, z_d\right)$$

bezeichnen wollen. Mit dessen Hilfe kann ein Restgliedsummand

$$\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath} \in \mathcal{I}_m^{(d)}}$$

für ein beliebiges $\vec{l} \in \tilde{\mathcal{I}}_m^{(d)}$ betragsmäßig abgeschätzt werden. Eine obere Schranke für den Verfahrensfehler erhält man, indem die Abschätzungen für alle $\vec{l} \in \tilde{\mathcal{I}}_m^{(d)}$ summiert werden.

Da die Operatoren Σ, $\Delta^{[k]}$ und $\Lambda^{[k]}$, aus denen Φ zusammengesetzt ist, linear sind, kann das Quadraturverfahren auch als gewichtete Summe über die zugrunde gelegten Integralnäherungen aufgefaßt werden:

$$\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath} \in \mathcal{I}_m^{(d)}} \;=\; \sum_{\vec{\imath} \in \mathcal{I}_m^{(d)}} g_{\vec{\imath}} \cdot D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right), \tag{38}$$

wobei die Gewichte $g_{\vec{\imath}}, \vec{\imath} \in \mathcal{I}_m^{(d)}$ durch d, m und θ eindeutig festgelegt sind. Wir betrachten nun das gemischte Polynom

$$p_m^{(d)}\left(z_1, z_2, \ldots, z_d\right) \;:=\; \sum_{\vec{\imath} \in \mathcal{I}_m^{(d)}} g_{\vec{\imath}} \cdot z_1^{i_1} \cdot z_2^{i_2} \cdot \ldots \cdot z_d^{i_d}. \tag{39}$$

Mit $h_{i_k}^{2l_k} = h_0^{2l_k} \cdot \theta^{i_k l_k}$ gilt für beliebiges $\vec{l} \in \mathcal{I}_m^{(d)}$:

$$\begin{aligned} p_m^{(d)}\left(\theta^{l_1}, \theta^{l_2}, \ldots, \theta^{l_d}\right) &\overset{(39)}{=} \sum_{\vec{\imath} \in \mathcal{I}_m^{(d)}} g_{\vec{\imath}} \cdot \left(\theta^{l_1}\right)^{i_1} \cdot \left(\theta^{l_2}\right)^{i_2} \cdot \ldots \cdot \left(\theta^{l_d}\right)^{i_d} \\ &= \sum_{\vec{\imath} \in \mathcal{I}_m^{(d)}} g_{\vec{\imath}} \cdot \theta^{i_1 l_1} \cdot \theta^{i_2 l_2} \cdot \ldots \cdot \theta^{i_d l_d} \\ &= h_0^{-2\left|\vec{l}\right|} \cdot \sum_{\vec{\imath} \in \mathcal{I}_m^{(d)}} g_{\vec{\imath}} \cdot h_{i_1}^{2l_1} \cdot h_{i_2}^{2l_2} \cdot \ldots \cdot h_{i_d}^{2l_d} \\ &\overset{(38)}{=} h_0^{-2\left|\vec{l}\right|} \cdot \Phi\left(h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath} \in \mathcal{I}_m^{(d)}} \\ &\overset{3.13}{=} \bar{1}_{\vec{l}}. \end{aligned} \tag{40}$$

(39) definiert $p_m^{(d)}$ als ein gemischtes Polynom in $\vec{z} = (z_1, z_2, \ldots, z_d)$ vom Grad kleiner gleich m, (40) wiederum besagt, daß $p_m^{(d)}$ für $\vec{z} = \left(\theta^{l_1}, \theta^{l_2}, \cdots, \theta^{l_d}\right)$ und $\vec{l} \in \mathcal{I}_m^{(d)} \setminus \left\{\vec{0}\right\}$ den Wert 0, für $\vec{l} = \vec{0}$ den Wert 1 annimmt. Damit ist $p_m^{(d)}$ auch durch (40) eindeutig bestimmt.

$p_m^{(d)}$ soll nun unsere weitere Aufmerksamkeit gehören. Bevor wir eine explizite Darstellung dieses Polynoms angeben, werden folgende Eigenschaften gezeigt:

Lemma 3.15. *Es sei* $0 < \theta < 1$ *und* $i, l \in \mathbb{N}_0$ *sowie*

$$q(i,z) := \prod_{\varrho=1}^{i} \frac{z - \theta^\varrho}{1 - \theta^\varrho} \tag{41}$$

$$\bar{q}(i,z) := \begin{cases} q(0,z) & \textit{falls} \quad i = 0 \\ q(i,z) - q(i-1,z) & \textit{falls} \quad i > 0 \end{cases} \tag{42}$$

$$\sigma(i) := \sum_{\varrho=1}^{i} \varrho \tag{43}$$

$$\sigma(i,j) := \sum_{\varrho=1}^{i} \min(\varrho, j). \tag{44}$$

Dann gilt:

$$q(i,z) = \sum_{\xi=0}^{i} (-1)^{i-\xi} \cdot \alpha_\xi \cdot z^\xi \quad \textit{mit} \quad \alpha_\xi > 0,\ \xi = 0, \ldots, i \tag{45}$$

$$\bar{q}(i,z) = q(i,z) \cdot \frac{z-1}{z - \theta^i} \qquad \textit{falls} \quad z \neq \theta^i \tag{46}$$

$$q\left(i, \theta^l\right) = \begin{cases} \bar{1}_l & \textit{falls} \quad i \geq l \\ (-1)^i \cdot \alpha'_l & \textit{mit} \quad \alpha'_l \geq 0 \quad \textit{falls} \quad l > i \geq 0 \end{cases} \tag{47}$$

$$q\left(i, -\theta^l\right) = (-1)^i \cdot \theta^{\sigma(i,l)} \cdot c_1(i,l) \tag{48}$$

$$\bar{q}\left(i, -\theta^l\right) = (-1)^i \cdot \theta^{\sigma(i-1,l)} \cdot c_2(i,l) \tag{49}$$

$$\bar{q}\left(i, \theta^l\right) = \begin{cases} 0 & \textit{falls} \quad i > l \geq 0 \\ (-1)^i \cdot \theta^{\sigma(i-1)} \cdot c_3(i,l) & \textit{falls} \quad l \geq i \geq 0 \end{cases} \tag{50}$$

und $0 < c_k(i,l) < c_k(\theta) < \infty$ *für* $k = 1, 2, 3$.

Beweis. Behauptung (45): Wegen $0 < \theta < 1$ bildet (41) ein Polynom in z mit wechselndem Vorzeichen.

Behauptung (46): Es ist $z \neq \theta^i$ vorausgesetzt. Wir unterscheiden zwei Fälle:

Fall 1 $(i = 0)$*:* Linke und rechte Seite von (46) ergeben sich zu:

$$\begin{aligned} \bar{q}(0,z) &\overset{(42)}{=} q(0,z) \\ q(0,z) \cdot \frac{z-1}{z-\theta^0} &\overset{(z \neq \theta^0)}{=} q(0,z). \end{aligned}$$

Fall 2 ($i > 0$):

$$
\begin{aligned}
\bar{q}(i,z) &\overset{(42)}{=} q(i,z) - q(i-1,z) \\
&= \prod_{\varrho=1}^{i} \frac{z-\theta^{\varrho}}{1-\theta^{\varrho}} - \prod_{\varrho=1}^{i-1} \frac{z-\theta^{\varrho}}{1-\theta^{\varrho}} \\
&\overset{(z\neq\theta^{i})}{=} \prod_{\varrho=1}^{i} \frac{z-\theta^{\varrho}}{1-\theta^{\varrho}} - \prod_{\varrho=1}^{i} \frac{z-\theta^{\varrho}}{1-\theta^{\varrho}} \cdot \frac{1-\theta^{i}}{z-\theta^{i}} \\
&= \prod_{\varrho=1}^{i} \frac{z-\theta^{\varrho}}{1-\theta^{\varrho}} \cdot \left(1 - \frac{1-\theta^{i}}{z-\theta^{i}}\right) \\
&= q(i,z) \cdot \frac{z-1}{z-\theta^{i}},
\end{aligned}
$$

womit die Behauptung gezeigt ist.
Behauptung (47): Für den Fall $i \geq l$ kann die Gültigkeit unmittelbar durch Einsetzen in (41) gezeigt werden. Für $l > i \geq 1$ gilt zusammen mit der Voraussetzung $0 < \theta < 1$: $\frac{\theta^{l}-\theta^{\varrho}}{1-\theta^{\varrho}} < 0$ für $1 \leq \varrho < l$. Aus (41) folgt für $l > i > 0$ unmittelbar die Behauptung. Für $i = 0$ folgt die Behauptung durch Einsetzen.
Behauptung (48): Durch Ausklammern von $-\theta^{\varrho}$ beziehungsweise $-\theta^{l}$ ergibt sich
Fall 1 ($i \leq l$):

$$
\begin{aligned}
q\left(i,-\theta^{l}\right) &= \prod_{\varrho=1}^{i} \frac{-\theta^{l}-\theta^{\varrho}}{1-\theta^{\varrho}} \\
&\overset{(i\leq l)}{=} (-1)^{i} \cdot \theta^{\sigma(i)} \cdot c_1(i,l) \\
&= (-1)^{i} \cdot \theta^{\sigma(i,l)} \cdot c_1(i,l) \\
\text{mit} \quad c_1(i,l) &= \prod_{\varrho=1}^{i} \frac{1+\theta^{l-\varrho}}{1-\theta^{\varrho}} > 0,
\end{aligned}
$$

Fall 2 ($i > l$):

$$
\begin{aligned}
q\left(i,-\theta^{l}\right) &= \prod_{\varrho=1}^{i} \frac{-\theta^{l}-\theta^{\varrho}}{1-\theta^{\varrho}} \\
&= (-1)^{i} \cdot \theta^{\sigma(i,l)} \cdot c_1(i,l) \\
\text{mit} \quad c_1(i,l) &= \prod_{\varrho=1}^{l} \frac{1+\theta^{l-\varrho}}{1-\theta^{\varrho}} \cdot \prod_{\varrho=l+1}^{i} \frac{1+\theta^{\varrho-l}}{1-\theta^{\varrho}} > 0.
\end{aligned}
$$

Wegen $0 < \theta < 1$ gilt $0 < c_1(i,l) < c_1(\theta) < \infty$.
Behauptung (49): Es gilt

$$
\bar{q}\left(i,-\theta^{l}\right) \overset{(46)}{=} q\left(i,-\theta^{l}\right) \cdot \frac{-\theta^{l}-1}{-\theta^{l}-\theta^{i}}
$$

$$
\begin{aligned}
&\overset{(48)}{=} && (-1)^i \cdot \theta^{\sigma(i,l)} \cdot c_2(i,l) \\
\text{mit} \quad c_2(i,l) &= && c_1(i,l) \cdot \frac{\theta^l + 1}{\theta^l + \theta^i},
\end{aligned}
$$

wobei $c_2(i,l)$ abgeschätzt werden kann zu: $0 < c_2(i,l) < c_2(\theta) < \infty$.
Behauptung (50): Wir unterscheiden
Fall 1 ($i > l \geq 0$):

$$
\bar{q}\left(i, \theta^l\right) \overset{(l \neq i),(46)}{=} q\left(i, \theta^l\right) \cdot \frac{\theta^l - 1}{\theta^l - \theta^i} = \underbrace{\prod_{\varrho=1}^{i} \frac{\theta^l - \theta^\varrho}{1 - \theta^\varrho}}_{=0 \text{ für } i \geq l \geq 1} \cdot \underbrace{\frac{\theta^l - 1}{\theta^l - \theta^i}}_{=0 \text{ für } l=0} = 0,
$$

Fall 2 ($i = l = 0$):

$$
\bar{q}\left(0, \theta^0\right) = q(0,1) = 1 = (-1)^0 \cdot \theta^{\sigma(-1)} \cdot c_3(0,0) \quad \text{mit} \quad c_3(0,0) = 1 > 0,
$$

Fall 3 ($i = l \neq 0$):

$$
\begin{aligned}
\bar{q}\left(i, \theta^i\right) &\overset{(42)}{=} && q\left(i, \theta^i\right) - q\left(i-1, \theta^i\right) \\
&\overset{(47),(i \neq 0)}{=} && -q\left(i-1, \theta^i\right) \\
&= && -\prod_{\varrho=1}^{i-1} \frac{\theta^i - \theta^\varrho}{1 - \theta^\varrho} \\
&= && (-1)^i \cdot \theta^{\sigma(i-1)} \cdot c_3(i,i) \\
\text{mit} \quad c_3(i,i) &= && \prod_{\varrho=1}^{i-1} \frac{1 - \theta^{i-\varrho}}{1 - \theta^\varrho} > 0,
\end{aligned}
$$

Fall 4 ($l > i \geq 0$):

$$
\begin{aligned}
\bar{q}\left(i, \theta^l\right) &\overset{(l \neq i),(46)}{=} && q\left(i, \theta^l\right) \cdot \frac{\theta^l - 1}{\theta^l - \theta^i} \\
&= && \prod_{\varrho=1}^{i} \frac{\theta^l - \theta^\varrho}{1 - \theta^\varrho} \cdot \frac{\theta^l - 1}{\theta^l - \theta^i} \\
&= && -\prod_{\varrho=1}^{i-1} \frac{\theta^l - \theta^\varrho}{1 - \theta^\varrho} \cdot \frac{1 - \theta^l}{1 - \theta^i} \\
&= && (-1)^i \cdot \theta^{\sigma(i-1)} \cdot c_3(i,l) \\
\text{mit} \quad c_3(i,l) &= && \prod_{\varrho=1}^{i-1} \frac{1 - \theta^{l-\varrho}}{1 - \theta^\varrho} \cdot \frac{1 - \theta^l}{1 - \theta^i} > 0.
\end{aligned}
$$

Auch hier gilt wieder: $0 < c_3(i,l) < c_3(\theta) < \infty$. $\Box$

Mit dem folgenden Lemma erhalten wir eine explizite Darstellung von $p_m^{(d)}$.

Lemma 3.16. *Es seien $d > 1, m \geq 0$ sowie $0 < \theta < 1$ gegeben. Dann gilt:*

$$p_m^{(d)}(z_1, z_2, \ldots, z_d) = \sum_{r=0}^{d-1} \left((-1)^r \cdot \binom{d-1}{r} \cdot \sum_{|\vec{\imath}|=m-r} q(i_1, z_1) \cdot q(i_2, z_2) \cdot \ldots \cdot q(i_d, z_d) \right). \tag{51}$$

Beweis. Wir zeigen, daß die rechte Seite von Gleichung (51) für $(z_1, z_2, \ldots, z_d) = \left(\theta^{l_1}, \theta^{l_2}, \ldots, \theta^{l_d}\right)$ äquivalent zu $\bar{1}_{\vec{l}}$ mit $\vec{l} \in \mathcal{I}_m^{(d)}$ ist. Aus (40) und der Eindeutigkeit der Polynomdarstellung folgt dann die Behauptung.

Wir nutzen die allgemein bekannten Rechenregeln für Binomialkoeffizienten:

$$\binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k} \quad \text{für} \quad n, k \geq 0 \tag{52}$$

$$\binom{n}{k} = 0 \quad \text{falls} \quad k < 0 \quad \text{oder} \quad n < k, \tag{53}$$

und formen um:

$$\begin{aligned}
& \sum_{r=0}^{d-1} \left((-1)^r \cdot \binom{d-1}{r} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
\overset{(52)}{=} & \sum_{r=0}^{d-1} \left((-1)^r \cdot \binom{d-2}{r} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
+ & \sum_{r=0}^{d-1} \left((-1)^r \cdot \binom{d-2}{r-1} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
\overset{(53)}{=} & \sum_{r=0}^{d-2} \left((-1)^r \cdot \binom{d-2}{r} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
+ & \sum_{r=1}^{d-1} \left((-1)^r \cdot \binom{d-2}{r-1} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
= & \sum_{r=0}^{d-2} \left((-1)^r \cdot \binom{d-2}{r} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
- & \sum_{r=0}^{d-2} \left((-1)^r \cdot \binom{d-2}{r} \cdot \sum_{|\vec{\imath}|=m-r-1} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \\
= & \sum_{r=0}^{d-2} \left((-1)^r \cdot \binom{d-2}{r} \cdot \left(\sum_{|\vec{\imath}|=m-r} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right.\right. \\
& \qquad \left.\left. - \sum_{|\vec{\imath}|=m-r-1} q\left(i_1, \theta^{l_1}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot q\left(i_d, \theta^{l_d}\right) \right) \right)
\end{aligned}$$

$$
\begin{aligned}
&\stackrel{(42)}{=} && \sum_{r=0}^{d-2}\left((-1)^r \cdot \binom{d-2}{r} \cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1,\theta^{l_1}\right)\cdot \ldots \cdot q\left(i_{d-1},\theta^{l_{d-1}}\right)\cdot \bar{q}\left(i_d,\theta^{l_d}\right)\right)\\
&\vdots\\
&= && \sum_{r=0}^{d-3}\left((-1)^r \cdot \binom{d-3}{r}\right.\\
& && \left.\cdot \sum_{|\vec{\imath}|=m-r} q\left(i_1,\theta^{l_1}\right)\cdot \ldots \cdot q\left(i_{d-2},\theta^{l_{d-2}}\right)\cdot \bar{q}\left(i_{d-1},\theta^{l_{d-1}}\right)\cdot \bar{q}\left(i_d,\theta^{l_d}\right)\right)\\
&\vdots\\
&= && \sum_{|\vec{\imath}|=m} q\left(i_1,\theta^{l_1}\right)\cdot \bar{q}\left(i_2,\theta^{l_2}\right)\cdot \bar{q}\left(i_3,\theta^{l_3}\right)\cdot \ldots \cdot \bar{q}\left(i_d,\theta^{l_d}\right)\\
&= && \sum_{|\vec{\imath}|\le m} q\left(i_1,\theta^{l_1}\right)\cdot \bar{q}\left(i_2,\theta^{l_2}\right)\cdot \bar{q}\left(i_3,\theta^{l_3}\right)\cdot \ldots \cdot \bar{q}\left(i_d,\theta^{l_d}\right)\\
&- && \sum_{|\vec{\imath}|\le m-1} q\left(i_1,\theta^{l_1}\right)\cdot \bar{q}\left(i_2,\theta^{l_2}\right)\cdot \bar{q}\left(i_3,\theta^{l_3}\right)\cdot \ldots \cdot \bar{q}\left(i_d,\theta^{l_d}\right)\\
&\stackrel{(42)}{=} && \sum_{|\vec{\imath}|\le m} \bar{q}\left(i_1,\theta^{l_1}\right)\cdot \bar{q}\left(i_2,\theta^{l_2}\right)\cdot \bar{q}\left(i_3,\theta^{l_3}\right)\cdot \ldots \cdot \bar{q}\left(i_d,\theta^{l_d}\right)\\
&\stackrel{(\vec{l}\in\mathcal{I}_m^{(d)}),(50)}{=} && \sum_{|\vec{\imath}|\le \vec{l}} \bar{q}\left(i_1,\theta^{l_1}\right)\cdot \bar{q}\left(i_3,\theta^{l_3}\right)\cdot \bar{q}\left(i_2,\theta^{l_2}\right)\cdot \ldots \cdot \bar{q}\left(i_d,\theta^{l_d}\right)\\
&= && \sum_{i_1=0}^{l_1} \bar{q}\left(i_1,\theta^{l_1}\right)\cdot \sum_{i_2=0}^{l_2} \bar{q}\left(i_2,\theta^{l_2}\right)\cdot \ldots \cdot \sum_{i_d=0}^{l_d} \bar{q}\left(i_d,\theta^{l_d}\right)\\
&\stackrel{(42)}{=} && q\left(l_1,\theta^{l_1}\right)\cdot q\left(l_2,\theta^{l_2}\right)\cdot \ldots \cdot q\left(l_d,\theta^{l_d}\right)\\
&\stackrel{(47)}{=} && \bar{1}_{l_1}\cdot \bar{1}_{l_2}\cdot \ldots \cdot \bar{1}_{l_d} = \bar{1}_{\vec{l}}.
\end{aligned}
$$

□

Das nächste Lemma zeigt, daß ein Restgliedsummand $\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ betragsmäßig mit Hilfe von $p_m^{(d)}$ abgeschätzt werden kann.

Lemma 3.17. *Es seien $m \ge 0, d > 0, h_\xi = h_0 \cdot b^\xi$ mit $0 < b < 1$, $\theta = b^2$ und $f \in C^{\mathcal{I}_m^{(d)}}\left[H^d\right]$ gegeben. Außerdem sei ein Multiindex $\vec{l} \in \tilde{\mathcal{I}}_m^{(d)}$ fest vorgegeben, und der Restgliedkoeffizient $\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)$ der Euler-Maclaurinschen Summenformel nach (32) mit*

$$
\begin{aligned}
\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right) &\equiv \dot{D}_{l_{r_1}}^{[r_1]}\left(h_{i_{r_1}}\right)\dot{D}_{l_{r_2}}^{[r_2]}\left(h_{i_{r_2}}\right)\cdots \dot{D}_{l_{r_n}}^{[r_n]}\left(h_{i_{r_n}}\right)\dot{\delta}_{l_{s_1}}^{[s_1]}\dot{\delta}_{l_{s_2}}^{[s_2]}\cdots\dot{\delta}_{l_{s_{n'}}}^{[s_{n'}]}\\
r_\xi &\in I^D\left(\vec{l}\right) \quad \text{für} \quad \xi = 1,\ldots,n \quad \text{mit} \quad n = \left|I^D\left(\vec{l}\right)\right|\\
s_\xi &\in I^\delta\left(\vec{l}\right) \quad \text{für} \quad \xi = 1,\ldots,n' \quad \text{mit} \quad n' = \left|I^\delta\left(\vec{l}\right)\right|
\end{aligned}
$$

werde durch eine von den Schrittweiten unabhängige Konstante $A_{\vec{l}}$ abgeschätzt zu:

$$\left|\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\right| \leq A_{\vec{l}} \qquad \forall\, \overline{h}_{\vec{\imath}} = \left(h_{i_{r_1}}, h_{i_{r_2}}, \ldots, h_{i_{r_d}}\right).$$

Dann gilt:

$$\begin{aligned}\left|\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right) \cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}\right| &\leq h_0^{2m+2} \cdot A_{\vec{l}} \cdot \left|p_m^{(d)}\left(\underline{\theta}^{l_1}, \underline{\theta}^{l_2}, \ldots, \underline{\theta}^{l_d}\right)\right| \\ \textit{mit}\ \ \underline{\theta}^{l_\xi} &= \begin{cases} -\theta^{l_\xi} & \textit{falls}\ \ \xi \in I^D\left(\vec{l}\right) \\ \ \ \theta^{l_\xi} & \textit{falls}\ \ \xi \in I^\delta\left(\vec{l}\right). \end{cases}\end{aligned}$$

Beweis. O.B.d.A. führen wir eine vom vorgegebenen Multiindex $\vec{l}$ abhängige Permutation $\mathcal{P}_{\vec{l}}$ der Koordinatenrichtungen durch. $\mathcal{P}_{\vec{l}}$ ist so gewählt, daß zuerst alle Richtungen aus $I^D\left(\vec{l}\right)$, dann die aus $I^\delta\left(\vec{l}\right)$ aufsteigend durchnumeriert werden. Darüber hinaus werden durch $\mathcal{P}_{\vec{l}}$ die Komponenten von $\vec{l}$ abfallend sortiert, was wegen Satz 3.10 mit der ersten Forderung vereinbar ist. Die Permutation

$$\begin{aligned} \mathcal{P}_{\vec{l}} &= \begin{pmatrix} 1 & 2 & \ldots & n & n+1 & n+2 & \ldots & d \\ r_1 & r_2 & \ldots & r_n & s_1 & s_2 & \ldots & s_{n'} \end{pmatrix} \\ \text{mit} &\quad l_{r_1} \geq l_{r_2} \geq \ldots \geq l_{r_n} \geq l_{s_1} \geq l_{s_2} \geq \ldots \geq l_{s_{n'}} \\ \text{und} &\quad r_\xi \in I^D\left(\vec{l}\right), s_\xi \in I^\delta\left(\vec{l}\right) \end{aligned} \tag{54}$$

bildet den Multiindex $\vec{\imath} = (i_1, i_2, \ldots, i_d)$ ab auf

$$\mathcal{P}_{\vec{l}}\left(\vec{\imath}\right) = \left(i_{r_1}, i_{r_2}, \ldots, i_{r_n}, i_{s_1}, i_{s_2}, \ldots, i_{s_{n'}}\right).$$

So ergibt sich beispielsweise für $\vec{l} = (3,4,5,4,5,3,4)$ nach Satz 3.10:

$$\begin{aligned} X_{\vec{l}} &\equiv \delta_3^{[1]}\delta_4^{[2]}D_5^{[3]}\delta_4^{[4]}D_5^{[5]}\delta_3^{[6]}D_4^{[7]} \\ I^D\left(\vec{l}\right) &= \{3,5,7\} \\ I^\delta\left(\vec{l}\right) &= \{1,2,4,6\}. \end{aligned}$$

Eine mögliche Permutation wäre

$$\mathcal{P}_{\vec{l}} = \begin{pmatrix} 1\,2\,3\,4\,5\,6\,7 \\ 5\,3\,7\,4\,2\,1\,6 \end{pmatrix}.$$

Damit ergibt sich:

$$\mathcal{P}_{\vec{l}}\left(X_{\vec{l}}\right) = D_5^{[1]}D_5^{[2]}D_4^{[3]}\delta_4^{[4]}\delta_4^{[5]}\delta_3^{[6]}\delta_3^{[7]}.$$

Wir führen folgende Notation für einen permutierten Multiindex $\vec{\imath}$ ein:

$$\begin{aligned} \bar{\imath} &:= (i_1, i_2, \ldots, i_n) \\ \bar{\bar{\imath}} &:= (i_{n+1}, i_{n+2}, \ldots, i_d). \end{aligned}$$

Nach der Permutation der Koordinatenrichtungen ist also zu zeigen:

$$\begin{aligned}
\left|\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}\right| &\leq h_0^{2m+2}\cdot A_{\vec{l}}\cdot\left|p_m^{(d)}\left(\underline{\theta}^{l_1},\underline{\theta}^{l_2},\ldots,\underline{\theta}^{l_d}\right)\right| \\
\text{mit}\quad \dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right) &\equiv \dot{D}_{l_1}^{[1]}(h_{i_1})\,\dot{D}_{l_2}^{[2]}(h_{i_2})\cdots\dot{D}_{l_n}^{[n]}(h_{i_n})\,\dot{\delta}_{l_{n+1}}^{[n+1]}\dot{\delta}_{l_{n+2}}^{[n+2]}\cdots\dot{\delta}_{l_d}^{[d]} \\
\overline{h}_{\vec{\imath}} &= (h_{i_1},h_{i_2},\ldots,h_{i_n}) \\
\underline{\theta}^{l_\xi} &= \begin{cases} -\theta^{l_\xi} & \text{für}\quad \xi=1,\ldots,n \\ \theta^{l_\xi} & \text{für}\quad \xi=n+1,\ldots,d. \end{cases}
\end{aligned}$$

Mit $h_{i_k}^{2l_k}=h_0^{2l_k}\cdot\theta^{i_kl_k}$ und $|\vec{l}\,|=m+1$ gilt:

$$\begin{aligned}
&\left|\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}\right| \\
=\;&\left|\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_{i_1}^{2l_1}\cdot h_{i_2}^{2l_2}\cdot\ldots\cdot h_{i_d}^{2l_d}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}\right| \\
\overset{(38)}{=}\;&\left|\sum_{|\vec{\imath}|\leq m}\left(g_{\vec{\imath}}\cdot\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_{i_1}^{2l_1}\cdot h_{i_2}^{2l_2}\cdot\ldots\cdot h_{i_d}^{2l_d}\right)\right| \\
=\;&\left|\sum_{|\vec{\imath}|\leq m}\left(g_{\vec{\imath}}\cdot\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_0^{2m+2}\cdot\theta^{l_1i_1}\cdot\theta^{l_2i_2}\cdot\ldots\cdot\theta^{l_di_d}\right)\right| \\
=\;&\left|h_0^{2m+2}\cdot\sum_{|\overline{\imath}|\leq m}\left(\theta^{l_1i_1}\cdot\theta^{l_2i_2}\cdot\ldots\cdot\theta^{l_ni_n}\cdot\dot{X}_{\vec{l}}\left(\overline{h}_{\overline{\imath}}\right)\right.\right. \\
&\qquad\left.\left.\cdot\sum_{|\overline{\overline{\imath}}|\leq m-|\overline{\imath}|}g_{\vec{\imath}}\cdot\theta^{l_{n+1}i_{n+1}}\cdot\theta^{l_{n+2}i_{n+2}}\cdot\ldots\cdot\theta^{l_di_d}\right)\right| \\
\leq\;& h_0^{2m+2}\cdot\sum_{|\overline{\imath}|\leq m}\left(\theta^{l_1i_1}\cdot\theta^{l_2i_2}\cdot\ldots\cdot\theta^{l_ni_n}\cdot\left|\dot{X}_{\vec{l}}\left(\overline{h}_{\overline{\imath}}\right)\right|\right. \\
&\qquad\left.\cdot\left|\sum_{|\overline{\overline{\imath}}|\leq m-|\overline{\imath}|}g_{\vec{\imath}}\cdot\theta^{l_{n+1}i_{n+1}}\cdot\theta^{l_{n+2}i_{n+2}}\cdot\ldots\cdot\theta^{l_di_d}\right|\right) \\
\leq\;& h_0^{2m+2}\cdot A_{\vec{l}}\cdot\sum_{|\overline{\imath}|\leq m}\left(\theta^{l_1i_1}\cdot\theta^{l_2i_2}\cdot\ldots\cdot\theta^{l_ni_n}\right. \\
&\qquad\left.\cdot\underbrace{\left|\sum_{|\overline{\overline{\imath}}|\leq m-|\overline{\imath}|}g_{\vec{\imath}}\cdot\theta^{l_{n+1}i_{n+1}}\cdot\theta^{l_{n+2}i_{n+2}}\cdot\ldots\cdot\theta^{l_di_d}\right|}_{(*)\quad\text{signum}(\ldots)=(-1)^{m-|\overline{\imath}|}}\right) \\
=\;& h_0^{2m+2}\cdot A_{\vec{l}}\cdot\sum_{|\overline{\imath}|\leq m}\left(\theta^{l_1i_1}\cdot\theta^{l_2i_2}\cdot\ldots\cdot\theta^{l_ni_n}\cdot(-1)^{m-|\overline{\imath}|}\right. \\
&\qquad\left.\cdot\sum_{|\overline{\overline{\imath}}|\leq m-|\overline{\imath}|}g_{\vec{\imath}}\cdot\theta^{l_{n+1}i_{n+1}}\cdot\theta^{l_{n+2}i_{n+2}}\cdot\ldots\cdot\theta^{l_di_d}\right) \\
=\;& h_0^{2m+2}\cdot A_{\vec{l}}\cdot\sum_{|\overline{\imath}|\leq m}\left(\left(-\theta^{l_1}\right)^{i_1}\cdot\left(-\theta^{l_2}\right)^{i_2}\cdot\ldots\cdot\left(-\theta^{l_n}\right)^{i_n}\cdot(-1)^m\right.
\end{aligned}$$

$$
\begin{aligned}
& \cdot \sum_{|\bar{\bar{\imath}}|\leq m-|\bar{\imath}|} g_{\vec{\imath}} \cdot \theta^{l_{n+1} i_{n+1}} \cdot \theta^{l_{n+2} i_{n+2}} \cdot \ldots \cdot \theta^{l_d i_d} \Big) \\
= \quad & h_0^{2m+2} \cdot A_{\vec{l}} \cdot (-1)^m \cdot \sum_{|\vec{\imath}|\leq m} \Big(g_{\vec{\imath}} \cdot \left(-\theta^{l_1}\right)^{i_1} \cdot \left(-\theta^{l_2}\right)^{i_2} \cdot \ldots \cdot \left(-\theta^{l_n}\right)^{i_n} \\
& \cdot \theta^{l_{n+1} i_{n+1}} \cdot \theta^{l_{n+2} i_{n+2}} \cdot \ldots \cdot \theta^{l_d i_d} \Big) \\
\overset{(39)}{=} \quad & h_0^{2m+2} \cdot A_{\vec{l}} \cdot (-1)^m \cdot p_m^{(d)} \left(-\theta^{l_1}, -\theta^{l_2}, \ldots, -\theta^{l_n}, \theta^{l_{n+1}}, \theta^{l_{n+2}}, \ldots, \theta^{l_d}\right) \\
= \quad & h_0^{2m+2} \cdot A_{\vec{l}} \cdot \left| p_m^{(d)} \left(\underline{\theta}^{l_1}, \underline{\theta}^{l_2}, \ldots, \underline{\theta}^{l_d}\right) \right| .
\end{aligned}
$$

Es bleibt noch Behauptung $(*)$ zu zeigen:

$$
\operatorname{signum} \left(\sum_{|\bar{\bar{\imath}}|\leq m-|\bar{\imath}|} g_{\vec{\imath}} \cdot \theta^{l_{n+1} i_{n+1}} \cdot \theta^{l_{n+2} i_{n+2}} \cdot \ldots \cdot \theta^{l_d i_d} \right) \quad = \quad (-1)^{m-|\bar{\imath}|} .
$$

Dazu benutzen wir:

$$
(55) \quad q\left(i_1, z_1\right) \cdot q\left(i_2, z_2\right) \cdot \ldots \cdot q\left(i_n, z_n\right) \quad \overset{(45)}{=} \quad \sum_{\bar{\jmath}\leq\bar{\imath}} (-1)^{|\bar{\imath}|-|\bar{\jmath}|} \cdot \alpha''_{\bar{\imath}\bar{\jmath}} \cdot z_1^{j_1} \cdot z_2^{j_2} \cdot \ldots \cdot z_n^{j_n}
$$

mit $\alpha''_{\bar{\imath}\bar{\jmath}} > 0$ und

$$
(56) \qquad \bar{q}\left(i_{n+1}, \theta^{l_{n+1}}\right) \cdot \bar{q}\left(i_{n+2}, \theta^{l_{n+2}}\right) \cdot \ldots \cdot \bar{q}\left(i_d, \theta^{l_d}\right) \quad \overset{(50)}{=} \quad (-1)^{|\bar{\bar{\imath}}|} \cdot \alpha'_{\bar{\bar{\imath}}}
$$

mit $\alpha'_{\bar{\bar{\imath}}} \geq 0$. Mit Hilfe der Umformungen für den Beweis von Lemma 3.16 $(**)$ erhalten wir für beliebiges $\overline{z} = (z_1, z_2, \ldots, z_n)$:

$$
\begin{aligned}
& \sum_{|\bar{\imath}|\leq m} z_1^{i_1} \cdot z_2^{i_2} \cdot \ldots \cdot z_n^{i_n} \cdot \sum_{|\bar{\bar{\imath}}|\leq m-|\bar{\imath}|} g_{\vec{\imath}} \cdot \theta^{l_{n+1} i_{n+1}} \cdot \theta^{l_{n+2} i_{n+2}} \cdot \ldots \cdot \theta^{l_d i_d} \\
= \quad & \sum_{|\vec{\imath}|\leq m} g_{\vec{\imath}} \cdot z_1^{i_1} \cdot z_2^{i_2} \cdot \ldots \cdot z_n^{i_n} \cdot \theta^{l_{n+1} i_{n+1}} \cdot \theta^{l_{n+2} i_{n+2}} \cdot \ldots \cdot \theta^{l_d i_d} \\
\overset{(39)}{=} \quad & p_m^{(d)} \left(z_1, z_2, \ldots, z_n, \theta^{l_{n+1}}, \theta^{l_{n+2}}, \ldots, \theta^{l_d}\right) \\
\overset{3.16}{=} \quad & \sum_{r=0}^{d-1} \Bigg((-1)^r \cdot \binom{d-1}{r} \cdot \sum_{|\vec{\imath}|=m-r} \Big(q\left(i_1, z_1\right) \cdot q\left(i_2, z_2\right) \cdot \ldots \cdot q\left(i_n, z_n\right) \\
& \cdot q\left(i_{n+1}, \theta^{l_{n+1}}\right) \cdot q\left(i_{n+2}, \theta^{l_{n+2}}\right) \cdot \ldots \cdot q\left(i_d, \theta^{l_d}\right) \Big) \Bigg) \\
\overset{(**)}{=} \quad & \sum_{r=0}^{d-2} \Bigg((-1)^r \cdot \binom{d-2}{r} \cdot \sum_{|\vec{\imath}|=m-r} \Big(q\left(i_1, z_1\right) \cdot q\left(i_2, z_2\right) \cdot \ldots \cdot q\left(i_n, z_n\right) \\
& \cdot q\left(i_{n+1}, \theta^{l_{n+1}}\right) \cdot q\left(i_{n+2}, \theta^{l_{n+2}}\right) \cdot \ldots \cdot q\left(i_{d-1}, \theta^{l_{d-1}}\right) \cdot \bar{q}\left(i_d, \theta^{l_d}\right) \Big) \Bigg) \\
& \vdots
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{r=0}^{n-1} (-1)^r \cdot \binom{n-1}{r} \cdot \sum_{|\vec{\imath}|=m-r} \Bigg(\Big(q(i_1, z_1) \cdot q(i_2, z_2) \cdot \ldots \cdot q(i_n, z_n) \\
&\qquad\qquad \cdot \bar{q}\left(i_{n+1}, \theta^{l_{n+1}}\right) \cdot \bar{q}\left(i_{n+2}, \theta^{l_{n+2}}\right) \cdot \ldots \cdot \bar{q}\left(i_d, \theta^{l_d}\right) \Big) \Bigg) \\
&\overset{(55)(56)}{=} \sum_{r=0}^{n-1} \Bigg((-1)^r \cdot \binom{n-1}{r} \\
&\qquad\qquad \cdot \sum_{|\vec{\imath}|=m-r} \left(\sum_{|\mathrm{j}|\le|\mathrm{i}|} (-1)^{|\mathrm{i}|-|\mathrm{j}|} \cdot \alpha''_{\mathrm{ij}} \cdot z_1^{j_1} \cdot z_2^{j_2} \cdot \ldots \cdot z_n^{j_n} \cdot (-1)^{|\bar{\mathrm{i}}|} \cdot \alpha'_{\bar{\mathrm{i}}} \right) \Bigg) \\
&= \sum_{r=0}^{n-1} \left(\binom{n-1}{r} \cdot \sum_{|\vec{\imath}|=m-r} \left(\sum_{|\mathrm{j}|\le|\mathrm{i}|} (-1)^{m-|\mathrm{j}|} \cdot \alpha'_{\bar{\mathrm{i}}} \cdot \alpha''_{\mathrm{ij}} \cdot z_1^{j_1} \cdot z_2^{j_2} \cdot \ldots \cdot z_n^{j_n} \right) \right) \\
&= \sum_{|\mathrm{i}|\le m} z_1^{i_1} \cdot z_2^{i_2} \cdot \ldots \cdot z_n^{i_n} \cdot (-1)^{m-|\mathrm{i}|} \cdot \alpha'''_{\mathrm{i}}.
\end{aligned}
$$

Also:

$$
\sum_{|\bar{\mathrm{i}}|\le m-|\mathrm{i}|} g_{\vec{\imath}} \cdot \theta^{l_{n+1}i_{n+1}} \cdot \theta^{l_{n+2}i_{n+2}} \cdot \ldots \cdot \theta^{l_d i_d} \quad = \quad (-1)^{m-|\mathrm{i}|} \cdot \alpha'''_{\mathrm{i}}.
$$

Da $\alpha'''_{\mathrm{i}} \ge 0$ folgt die Behauptung $(*)$, womit das Lemma bewiesen ist. □

Wir geben nun eine Abschätzung von $\left|p_m^{(d)}\left(\underline{\theta}^{l_1}, \underline{\theta}^{l_2}, \ldots, \underline{\theta}^{l_d}\right)\right|$ in Abhängigkeit von θ und $\vec{l}$ an.

Lemma 3.18. *Es seien $m \ge 0$, $d > 0$, $0 < \theta < 1$ sowie $\vec{l} \in \tilde{\mathcal{I}}_m^{(d)}$ gegeben. Dann gilt:*

$$
\begin{aligned}
\left|p_m^{(d)}\left(\underline{\theta}^{l_1}, \underline{\theta}^{l_2}, \ldots, \underline{\theta}^{l_d}\right)\right| &\le c_5(\theta, d) \cdot \theta^{\sum_{\xi=1}^{d} \sigma(l_\xi - 1)} \quad \textit{und} \quad 0 \le c_5(\theta, d) < \infty \\
\textit{mit} \ \underline{\theta}^{l_\xi} &= \begin{cases} -\theta^{l_\xi} & \textit{falls} \ \xi \in I^D\left(\vec{l}\right) \\ \theta^{l_\xi} & \textit{falls} \ \xi \in I^\delta\left(\vec{l}\right) \end{cases} \\
\sigma(i) &= \sum_{\varrho=1}^{i} \varrho.
\end{aligned}
$$

Beweis. O.B.d.A. führen wir, wie im Beweis des vorigen Lemmas, eine Permutation $\mathcal{P}_{\vec{l}}$ der Koordinatenrichtungen gemäß (54) durch. Es ist dann mit $n = \left|I^\delta\left(\vec{l}\right)\right|$ zu zeigen:

$$
\begin{aligned}
\left|p_m^{(d)}\left(\underline{\theta}^{l_1}, \underline{\theta}^{l_2}, \ldots, \underline{\theta}^{l_d}\right)\right| &\le c_5(\theta, d) \cdot \theta^{\sum_{\xi=1}^{d} \sigma(l_\xi - 1)} \\
\text{mit} \ \underline{\theta}^{l_\xi} &= \begin{cases} -\theta^{l_\xi} & \text{für} \ \xi = 1, \ldots, n \\ \theta^{l_\xi} & \text{für} \ \xi = n+1, \ldots, d. \end{cases}
\end{aligned}
$$

Mit Hilfe der Umformungen für den Beweis von Lemma 3.16 $(**)$ erhalten wir:

$$
\left|p_m^{(d)}\left(\underline{\theta}^{l_1}, \underline{\theta}^{l_2}, \ldots, \underline{\theta}^{l_d}\right)\right|
$$

$$
\begin{aligned}
&\overset{(**)}{=} && \left|\sum_{|\vec{\imath}|=m} q\left(i_1,\underline{\theta}^{l_1}\right)\cdot\bar{q}\left(i_2,\underline{\theta}^{l_2}\right)\cdot\ldots\cdot\bar{q}\left(i_d,\underline{\theta}^{l_d}\right)\right| \\
&= && \left|\sum_{|\vec{\imath}|=m}\left(q\left(i_1,-\theta^{l_1}\right)\cdot\prod_{\xi=2}^{n}\bar{q}\left(i_\xi,-\theta^{l_\xi}\right)\cdot\prod_{\xi=n+1}^{d}\bar{q}\left(i_\xi,\theta^{l_\xi}\right)\right)\right| \\
&\overset{(48)(49)(50)}{=} && \left|\sum_{\substack{|\vec{\imath}|=m\\ \vec{\imath}\leq\vec{l}}}\left((-1)^{i_1}\cdot\theta^{\sigma(i_1,l_1)}\cdot c_1(i_1,l_1)\cdot\prod_{\xi=2}^{n}\left((-1)^{i_\xi}\cdot\theta^{\sigma\left(i_\xi-1,l_\xi\right)}\cdot c_2(i_\xi,l_\xi)\right)\right.\right. \\
& && \left.\left.\cdot\prod_{\xi=n+1}^{d}\left((-1)^{i_\xi}\cdot\theta^{\sigma\left(i_\xi-1\right)}\cdot c_3(i_\xi,l_\xi)\right)\right)\right| \\
&\overset{c_1,c_2,c_3>0}{=} && |(-1)^m|\cdot\sum_{\substack{|\vec{\imath}|=m\\ \vec{\imath}\leq\vec{l}}}\left(\theta^{\sigma(i_1,l_1)}\cdot c_1(i_1,l_1)\cdot\prod_{\xi=2}^{n}\left(\theta^{\sigma\left(i_\xi-1,l_\xi\right)}\cdot c_2(i_\xi,l_\xi)\right)\right. \\
& && \left.\cdot\prod_{\xi=n+1}^{d}\left(\theta^{\sigma\left(i_\xi-1\right)}\cdot c_3(i_\xi,l_\xi)\right)\right) \\
&\overset{(48)(49)}{\leq} && c_1(\theta)\cdot c_2^{n-1}(\theta)\cdot c_3^{d-n}(\theta)\cdot\sum_{\substack{|\vec{\imath}|=m\\ \vec{\imath}\leq\vec{l}}}\theta^{\sigma(i_1,l_1)+\sum_{\xi=2}^{n}\sigma\left(i_\xi-1,l_\xi\right)+\sum_{\xi=n+1}^{d}\sigma\left(i_\xi-1\right)} \\
&\overset{(*)}{\leq} && c_1(\theta)\cdot c_2^{n-1}(\theta)\cdot c_3^{d-n}(\theta)\cdot c_4(\theta,d)\cdot\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)} \\
&\leq && c_5(\theta,d)\cdot\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)} \\
&\text{mit} && c_5(\theta,d)=c_1(\theta)\cdot c_2^{n-1}(\theta)\cdot c_3^{d-n}(\theta)\cdot c_4(\theta,d).
\end{aligned}
$$

Um die für $(*)$ notwendige Ungleichung

$$
\sum_{\substack{|\vec{\imath}|=m\\ \vec{\imath}\leq\vec{l}}}\theta^{\sigma(i_1,l_1)+\sum_{\xi=2}^{n}\sigma\left(i_\xi-1,l_\xi\right)+\sum_{\xi=n+1}^{d}\sigma\left(i_\xi-1\right)} \quad\leq\quad c_4(\theta,d)\cdot\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)}
$$

zu zeigen, benutzen wir mit $\sigma(i,l)=\sum_{\varrho=1}^{i}\min(\varrho,j)$ die Beziehungen

(57)
$$
\begin{aligned}
\sigma(i,l) &= \begin{cases}\sigma(l-1)+(i-l+1)\cdot l & \text{für } i\geq l\\ \sigma(l-1)-\sum\limits_{\varrho=1}^{l-i-1}(l-\varrho) & \text{für } i<l\end{cases}\\
\sigma(i-1,l) &= \begin{cases}\sigma(l-1)+(i-l)\cdot l & \text{für } i\geq l\\ \sigma(l-1)-\sum\limits_{\varrho=1}^{l-i}(l-\varrho) & \text{für } i<l\end{cases}\\
\sigma(i-1) &= \sigma(l-1)-\sum_{\varrho=1}^{l-i}(l-\varrho) \quad \text{für } i\leq l.
\end{aligned}
$$

Zum anderen machen wir Gebrauch von

$$
\begin{aligned}
\max\left(\vec{l}\right) &= l_1 \\
l_\xi &\geq l_1 - 1 \quad \text{für} \quad \xi = 1, \ldots, n \qquad (58)\\
l_\xi &\leq l_1 - 1 \quad \text{für} \quad \xi = n+1, \ldots, d,
\end{aligned}
$$

wobei sich die erste Eigenschaft aus der Permutation ergab. Die zweite und dritte sind unmittelbare Folgen von Satz 3.10. Nach Voraussetzung gilt $|\vec{\imath}| = m$, $\left|\vec{l}\right| = m+1$ und $\bar{\bar{\imath}} \leq \bar{\bar{l}}$, somit folgt ($\langle \ldots \rangle$ gilt für $\xi = 1$):

$$
\begin{aligned}
&\sigma(i_1, l_1) + \sum_{\xi=2}^{n} \sigma(i_\xi - 1, l_\xi) + \sum_{\xi=n+1}^{d} \sigma(i_\xi - 1) \\
\overset{(57)}{=}\; & \sum_{\substack{\xi=1 \\ i_\xi \geq l_\xi}}^{n} \left(\sigma(l_\xi - 1) + (i_\xi - l_\xi\langle +1 \rangle) \cdot l_\xi\right) + \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{n} \left(\sigma(l_\xi - 1) - \sum_{\varrho=1}^{l_\xi - i_\xi \langle -1 \rangle} (l_\xi - \varrho)\right) \\
& \qquad + \sum_{\substack{\xi=n+1 \\ i_\xi \leq j_\xi}}^{d} \left(\sigma(l_\xi - 1) - \sum_{\varrho=1}^{l_\xi - i_\xi} (l_\xi - \varrho)\right) \\
=\; & \sum_{\xi=1}^{d} \sigma(l_\xi - 1) + \sum_{\substack{\xi=1 \\ i_\xi \geq l_\xi}}^{n} (i_\xi - l_\xi\langle +1 \rangle) \cdot l_\xi - \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{d} \sum_{\varrho=1}^{l_\xi - i_\xi \langle -1 \rangle} (l_\xi - \varrho) \\
\overset{(58)}{\geq}\; & \sum_{\xi=1}^{d} \sigma(l_\xi - 1) + \sum_{\substack{\xi=1 \\ i_\xi \geq l_\xi}}^{n} (i_\xi - l_\xi\langle +1 \rangle) \cdot (l_1 - 1) - \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{d} \sum_{\varrho=1}^{l_\xi - i_\xi \langle -1 \rangle} (l_1 - 1 - \varrho + 1) \\
=\; & \sum_{\xi=1}^{d} \sigma(l_\xi - 1) + \sum_{\substack{\xi=1 \\ i_\xi \geq l_\xi}}^{n} (i_\xi - l_\xi\langle +1 \rangle) \cdot (l_1 - 1) - \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{d} (l_\xi - i_\xi\langle -1 \rangle) \cdot (l_1 - 1) \\
& \qquad + \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{d} \sum_{\varrho=1}^{l_\xi - i_\xi \langle -1 \rangle} (\varrho - 1) \\
=\; & \sum_{\xi=1}^{d} \sigma(l_\xi - 1) + \left(\sum_{\substack{\xi=1 \\ i_\xi \geq l_\xi}}^{n} (i_\xi - l_\xi\langle +1 \rangle) + \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{d} (i_\xi - l_\xi\langle +1 \rangle) + \underbrace{\sum_{\substack{\xi=n+1 \\ i_\xi = l_\xi}}^{d} (i_\xi - l_\xi)}_{=0}\right) \\
& \qquad \cdot (l_1 - 1) + \sum_{\substack{\xi=1 \\ i_\xi < l_\xi}}^{d} \sum_{\varrho=1}^{l_\xi - i_\xi - 1 \langle -1 \rangle} \varrho \\
\overset{\bar{\bar{\imath}} \leq \bar{\bar{l}}}{=}\; & \sum_{\xi=1}^{d} \sigma(l_\xi - 1) + \sum_{\xi=1}^{d} (i_\xi - l_\xi\langle +1 \rangle) \cdot (l_1 - 1) + \sum_{\xi=1}^{d} \sigma(l_\xi - i_\xi - 1\langle -1 \rangle)
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{\xi=1}^{d} \sigma\left(l_\xi - 1\right) + \left(m - (m+1) + 1\right) \cdot \left(l_1 - 1\right) + \sum_{\xi=1}^{d} \sigma\left(l_\xi - i_\xi - 1\langle -1\rangle\right) \\
&= \sum_{\xi=1}^{d} \sigma\left(l_\xi - 1\right) + \sum_{\xi=1}^{d} \sigma\left(l_\xi - i_\xi - 1\langle -1\rangle\right).
\end{aligned}
$$

Hieraus folgt:

$$
\begin{aligned}
&\sum_{\substack{|\vec{\imath}|=m \\ \vec{\imath}\leq\vec{l}}} \theta^{\sigma(i_1,l_1)+\sum_{\xi=2}^{n}\sigma\left(i_\xi-1,l_\xi\right)+\sum_{\xi=n+1}^{d}\sigma\left(i_\xi-1\right)} \\
&\quad\leq \quad \theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)} \cdot \sum_{\substack{|\vec{\imath}|=m \\ \vec{\imath}\leq\vec{l}}} \theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-i_\xi-1\langle -1\rangle\right)} \\
&\quad\leq \quad c_4\left(\theta,d\right) \cdot \theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)} \\
&\quad\text{mit} \quad c_4\left(\theta,d\right) = \sum_{\substack{|\vec{\imath}|=m \\ \vec{\imath}\leq\vec{l}}} \theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-i_\xi-1\langle -1\rangle\right)}.
\end{aligned}
$$

Es bleibt noch zu zeigen, daß $c_4\left(\theta,d\right)$ für beliebig großes m beschränkt ist. Dazu schwächen wir ab:

$$
c_4\left(\theta,d\right) \leq \sum_{|\vec{\imath}|=m} \theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-i_\xi-1\langle -1\rangle\right)}.
$$

Wir betrachten die rechte Seite dieser Ungleichung und fassen Summanden mit gleichem Exponenten zur Basis θ zusammen, wobei zu berücksichtigen ist, daß gilt:

$$
\sigma\left(a\right) = \sigma\left(0\right) = 0 \quad \text{falls} \quad a < 0.
$$

Damit ist der Summand $\theta^{\sigma(0)+\sigma(0)+\ldots+\sigma(0)}$ beispielsweise äquivalent zu:

$$
\begin{aligned}
\theta^{\sigma(0)+\sigma(0)+\sigma(0)+\ldots+\sigma(0)} &= \theta^{\sigma(-1)+\sigma(0)+\sigma(0)+\ldots+\sigma(0)} \\
&= \theta^{\sigma(0)+\sigma(-1)+\sigma(0)+\ldots+\sigma(0)} \\
&\vdots \\
&= \theta^{\sigma(0)+\sigma(0)+\sigma(0)+\ldots+\sigma(-1)} \\
&= \theta^{\sigma(-2)+\sigma(0)+\sigma(0)+\ldots+\sigma(0)} \\
&= \theta^{\sigma(0)+\sigma(-2)+\sigma(0)+\ldots+\sigma(0)} \\
&\vdots \\
&= \theta^{\sigma(0)+\sigma(0)+\sigma(0)+\ldots+\sigma(-2)} \\
&= \theta^{\sigma(-1)+\sigma(-1)+\sigma(0)+\ldots+\sigma(0)} \\
&= \theta^{\sigma(-1)+\sigma(0)+\sigma(-1)+\ldots+\sigma(0)} \\
&\vdots
\end{aligned}
$$

Da

$$\begin{aligned}\sum_{\xi=1}^{d}\left(l_\xi - i_\xi - 1\langle -1\rangle\right) &= \sum_{\xi=1}^{d} l_\xi - \sum_{\xi=1}^{d} i_\xi - d - 1\\ &= m+1-m-d-1\\ &= -d\end{aligned}$$

gilt, gibt es gemäß den Gesetzen der Kombinatorik, insgesamt $\binom{d+d-1}{d-1}$ Möglichkeiten $\theta^{\sigma(0)+\sigma(0)+\ldots+\sigma(0)}$ darzustellen (d nicht unterscheidbare Marken auf d numerierte Stellen verteilen), wenn man $m > d$ und $l_\xi > 1\langle +1\rangle$ voraussetzt. Analog gibt es für

$$\theta^{\sigma(1)+\sigma(0)+\ldots+\sigma(0)},\ \theta^{\sigma(0)+\sigma(1)+\sigma(0)+\ldots+\sigma(0)},\ \ldots,\ \theta^{\sigma(0)+\sigma(0)+\ldots+\sigma(0)+\sigma(1)}$$

jeweils $\binom{d+1+d-1-1}{d-1-1}$ Möglichkeiten ($d+1$ Marken auf $d-1$ Stellen), für

$$\theta^{\sigma(2)+\sigma(0)+\ldots+\sigma(0)},\ \theta^{\sigma(0)+\sigma(2)+\sigma(0)+\ldots+\sigma(0)},\ \ldots,\ \theta^{\sigma(0)+\sigma(0)+\ldots+\sigma(0)+\sigma(2)}$$

jeweils $\binom{d+2+d-1-1}{d-1-1}$ ($d+2$ Marken auf $d-1$ Stellen), für

$$\theta^{\sigma(1)+\sigma(1)+\sigma(0)+\ldots+\sigma(0)},\ \theta^{\sigma(1)+\sigma(0)+\sigma(1)+\sigma(0)+\ldots+\sigma(0)},\ \ldots,\ \theta^{\sigma(0)+\ldots+\sigma(0)+\sigma(1)+\sigma(1)}$$

jeweils $\binom{d+2+d-2-1}{d-2-1}$ ($d+2$ Marken auf $d-2$ Stellen), ...

Hat ein Multiindex $\vec{\jmath}$ insgesamt $d_0(\vec{\jmath}) := |\{\xi \mid j_\xi = 0\}|$ Stellen gleich Null, dann existieren für $\theta^{\sigma(j_1)+\sigma(j_2)+\ldots+\sigma(j_d)}$ also $\binom{d+|\vec{\jmath}|+d_0(\vec{\jmath})-1}{d_0(\vec{\jmath})-1}$ Darstellungsmöglichkeiten.

Wir können nun zeigen:

$$\begin{aligned}c_4(\theta,d) &\leq \sum_{|\vec{\imath}|=m} \theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi - i_\xi - 1\langle -1\rangle\right)}\\ &\leq \lim_{m\to\infty}\sum_{\mu=0}^{m}\sum_{|\vec{\jmath}|=\mu}\theta^{\sum_{\xi=1}^{d}\sigma(j_\xi)}\cdot\binom{d+\mu+d_0(\vec{\jmath})-1}{d_0(\vec{\jmath})-1}\\ &\leq \lim_{m\to\infty}\sum_{\mu=0}^{m}\sum_{|\vec{\jmath}|=\mu}\theta^{\sum_{\xi=1}^{d}j_\xi}\cdot\binom{d+\mu+d_0(\vec{\jmath})-1}{d_0(\vec{\jmath})-1}\\ &= \lim_{m\to\infty}\sum_{\mu=0}^{m}\theta^{\mu}\cdot\sum_{|\vec{\jmath}|=\mu}\binom{d+\mu+d_0(\vec{\jmath})-1}{d_0(\vec{\jmath})-1}\\ &\overset{d_0(\vec{\jmath})\leq d}{\leq} \lim_{m\to\infty}\sum_{\mu=0}^{m}\theta^{\mu}\cdot\binom{d+\mu+d-1}{d-1}\cdot\sum_{|\vec{\jmath}|=\mu}1\\ &= \lim_{m\to\infty}\sum_{\mu=0}^{m}\theta^{\mu}\cdot\binom{2d+\mu-1}{d-1}\cdot\binom{\mu+d-1}{d-1}\\ &\leq \lim_{m\to\infty}\sum_{\mu=0}^{m}\theta^{\mu}\cdot(2d+\mu-1)^{d-1}\cdot(\mu+d-1)^{d-1}\\ &< \infty,\end{aligned}$$

falls $0<\theta<1$ und d fest. Aus der Beschränktheit von c_4 folgt die Beschränktheit von c_5, wodurch die letzte Lücke des Beweises geschlossen ist. □

Die letzten beiden Lemmata gestatten uns den Fehleranteil *eines* Restglieds abzuschätzen:

$$\left|\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{1}}\right)\cdot h_{\vec{1}}^{2\vec{l}}\right)_{\vec{1}\in\mathcal{I}_m^{(d)}}\right| \quad\leq\quad h_0^{2m+2}\cdot A_{\vec{l}}\cdot c_5\left(\theta,d\right)\cdot\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)}. \tag{59}$$

Das folgende Lemma benötigen wir, um die Summe *aller* Fehleranteile bilden zu können.

Lemma 3.19. *Es seien $m\geq 0, d>0$ und $0<\theta<1$ gegeben. Dann gilt:*

$$\sum_{\left|\vec{l}\right|=m}\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)} \quad\leq\quad c_6\left(\theta,d\right)\cdot\theta^{\frac{1}{2}\cdot\frac{(m-d)\cdot m}{d}} \quad \textit{und} \quad 0<c_6\left(\theta,d\right)<\infty.$$

Beweis. Mit $\sigma\left(n\right)=\frac{n^2+n}{2}$ für $n\geq -1$ gilt die Beziehung:

$$\begin{aligned}
&\sigma\left(a\right)+\left(b-a\right)\cdot a+\sigma\left(b-a\right)=\\
&\quad\tfrac{1}{2}\cdot a\cdot\left(a+1\right)+\left(b-a\right)\cdot a+\tfrac{1}{2}\cdot\left(b-a\right)\cdot\left(b-a+1\right)\\
&=\tfrac{1}{2}\left(a^2+a+2ab-2a^2+b^2-2ab+a^2+b-a\right)\\
&=\tfrac{1}{2}\left(b^2+b\right)\\
&=\sigma\left(b\right),
\end{aligned}\tag{60}$$

falls $a,b,b-1\geq -1$. Mit $b=l_\xi-1$ und $a=\frac{m}{d}-1$ und unter Ausnutzung von $\sum_{\xi=1}^{d}l_\xi=m$ ergibt sich:

$$\begin{aligned}
\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right) &\overset{(60)}{=} \sum_{\xi=1}^{d}\left(\sigma\left(\frac{m}{d}-1\right)+\left(l_\xi-1-\frac{m}{d}+1\right)\cdot\left(\frac{m}{d}-1\right)\right.\\
&\qquad\left.+\sigma\left(l_\xi-1-\frac{m}{d}+1\right)\right)\\
&= d\cdot\sigma\left(\frac{m}{d}-1\right)+\underbrace{\left(\sum_{\xi=1}^{d}l_\xi-m\right)}_{=0}\cdot\left(\frac{m}{d}-1\right)+\sum_{\xi=1}^{d}\sigma\left(l_\xi-\frac{m}{d}\right)\\
&= \frac{d}{2}\cdot\frac{m-d}{d}\cdot\frac{m}{d}+\frac{1}{2}\cdot\sum_{\xi=1}^{d}\left(l_\xi^2-2\cdot l_\xi\cdot\frac{m}{d}+\left(\frac{m}{d}\right)^2+l_\xi-\frac{m}{d}\right)\\
&= \frac{1}{2}\cdot\frac{(m-d)\cdot m}{d}+\frac{1}{2}\cdot\left(\sum_{\xi=1}^{d}l_\xi^2-2\cdot m\cdot\frac{m}{d}+d\cdot\left(\frac{m}{d}\right)^2+m-d\cdot\frac{m}{d}\right)\\
&= \frac{1}{2}\cdot\frac{(m-d)\cdot m}{d}+\frac{1}{2}\cdot\left(\sum_{\xi=1}^{d}l_\xi^2-d\cdot\left(\frac{m}{d}\right)^2\right)
\end{aligned}$$

$$= \quad \frac{1}{2} \cdot \frac{(m-d) \cdot m}{d} + \frac{1}{2} \cdot \sum_{\xi=1}^{d} \left(l_\xi^2 - \left(\frac{m}{d} \right)^2 \right).$$

Daraus folgt:

$$\begin{aligned}
\sum_{|\vec{l}|=m} \theta^{\sum_{\xi=1}^{d} \sigma\left(l_\xi - 1\right)} &= \theta^{\frac{1}{2} \cdot \frac{(m-d) \cdot m}{d}} \cdot \sum_{|\vec{l}|=m} \theta^{\frac{1}{2} \cdot \sum_{\xi=1}^{d} \left(l_\xi^2 - \left(\frac{m}{d} \right)^2 \right)} \\
&= \theta^{\frac{1}{2} \cdot \frac{(m-d) \cdot m}{d}} \cdot c_6(\theta, d, m) \\
\text{mit} \quad & c_6(\theta, d, m) = \sum_{|\vec{l}|=m} \theta^{\frac{1}{2} \cdot \sum_{\xi=1}^{d} \left(l_\xi^2 - \left(\frac{m}{d} \right)^2 \right)}.
\end{aligned}$$

Es bleibt noch zu zeigen, daß $\lim\limits_{m \to \infty} c_6(\theta, d, m) = c_6(\theta, d) < \infty$ gilt. Wir führen den Beweis mittels Induktion über d.

Induktionsanfang $(d = 1)$ *:*

$$c_6(\theta, 1, m) = \sum_{l_1 = m} \theta^{\frac{1}{2}\left(l_1^2 - l_1^2\right)} = \theta^0 = 1 < \infty.$$

Induktionsschritt $(d \to d+1)$ *:* Unter Ausnutzung von

$$\begin{aligned}
& l_{d+1}^2 - \frac{m^2}{d+1} + \frac{\left(m - l_{d+1}\right)^2}{d} \\
&= \frac{l_{d+1}^2 \cdot d \cdot (d+1) - d \cdot m^2 + (d+1) \cdot \left(m^2 - 2 \cdot m \cdot l_{d+1} + l_{d+1}^2 \right)}{d \cdot (d+1)} \\
&= \frac{l_{d+1}^2 \cdot (d+1)^2 - d \cdot m^2 + d \cdot m^2 + m^2 - 2 \cdot (d+1) \cdot m \cdot l_{d+1}}{d \cdot (d+1)} \qquad (61) \\
&= \frac{(d+1)}{d} \cdot l_{d+1}^2 - \frac{2 \cdot (d+1) \cdot m \cdot l_{d+1}}{d \cdot (d+1)} + \frac{m^2}{d \cdot (d+1)} \\
&= \frac{d+1}{d} \cdot \left(l_{d+1} - \frac{m}{d+1} \right)^2
\end{aligned}$$

folgt

$$\begin{aligned}
& \sum_{\xi=1}^{d+1} \left(l_\xi^2 - \left(\frac{m}{d+1} \right)^2 \right) \\
&= l_{d+1}^2 - (d+1) \cdot \left(\frac{m}{d+1} \right)^2 + \sum_{\xi=1}^{d} l_\xi^2 \\
&= l_{d+1}^2 - \frac{m^2}{d+1} + d \cdot \left(\frac{m - l_{d+1}}{d} \right)^2 - d \cdot \left(\frac{m - l_{d+1}}{d} \right)^2 + \sum_{\xi=1}^{d} l_\xi^2 \\
&= l_{d+1}^2 - \frac{m^2}{d+1} + \frac{\left(m - l_{d+1}\right)^2}{d} + \sum_{\xi=1}^{d} \left(l_\xi^2 - \left(\frac{m - l_{d+1}}{d} \right)^2 \right)
\end{aligned}$$

$$\overset{(61)}{=} \frac{d+1}{d}\cdot\left(l_{d+1}-\frac{m}{d+1}\right)^2+\sum_{\xi=1}^{d}\left(l_\xi^2-\left(\frac{m-l_{d+1}}{d}\right)^2\right)$$
$$> \left(l_{d+1}-\frac{m}{d+1}\right)^2+\sum_{\xi=1}^{d}\left(l_\xi^2-\left(\frac{m-l_{d+1}}{d}\right)^2\right),$$

womit

$$\sum_{\xi=1}^{d+1}\left(l_\xi^2-\left(\frac{m}{d+1}\right)^2\right) > \left(l_{d+1}-\frac{m}{d+1}\right)^2+\sum_{\xi=1}^{d}\left(l_\xi^2-\left(\frac{m-l_{d+1}}{d}\right)^2\right) \tag{62}$$

gilt. Für einen Multiindex $\vec{l}=(l_1,l_2,\ldots,l_d,l_{d+1})$ lautet die Induktionsvoraussetzung:

$$\sum_{l_1+l_2+\ldots+l_d=m-l_{d+1}}\theta^{\frac{1}{2}\cdot\sum_{\xi=1}^{d}\left(l_\xi^2-\left(\frac{m-l_{d+1}}{d}\right)^2\right)} < c_6(\theta,d) < \infty.$$

Wir erhalten somit:

$$\begin{aligned}
c_6(\theta,d+1,m) &= \sum_{l_{d+1}=0}^{m}\ \sum_{l_1+l_2+\ldots+l_d=m-l_{d+1}}\theta^{\frac{1}{2}\cdot\sum_{\xi=1}^{d+1}\left(l_\xi^2-\left(\frac{m}{d+1}\right)^2\right)}\\
&\overset{(62)}{\leq} \sum_{l_{d+1}=0}^{m}\theta^{\frac{1}{2}\cdot\left(l_{d+1}-\frac{m}{d+1}\right)^2}\sum_{l_1+l_2+\ldots+l_d=m-l_{d+1}}\theta^{\frac{1}{2}\cdot\sum_{\xi=1}^{d}\left(l_\xi^2-\left(\frac{m-l_{d+1}}{d}\right)^2\right)}\\
&\overset{\text{(Ind.vor.)}}{<} \sum_{l_{d+1}=0}^{m}\theta^{\frac{1}{2}\cdot\left(l_{d+1}-\frac{m}{d+1}\right)^2}\cdot c_6(\theta,d)\\
&= c_6(\theta,d)\cdot\sum_{l_{d+1}=-\frac{m}{d+1}}^{m-\frac{m}{d+1}}\theta^{\frac{1}{2}l_{d+1}^2}\\
&< \infty,
\end{aligned}$$

wenn $0<\theta<1$ und d fest ist, womit das Lemma bewiesen ist. □

Wir sind jetzt in der Lage, eine Fehlerabschätzung für das Quadraturverfahren anzugeben:

Satz 3.20. *Es seien $m\geq 0, d>0$, eine Multiindexmenge $\mathcal{I}_m^{(d)}$ sowie die Schrittweitenfolge*

$$H^{[k]} = \left\{h_\xi \;\middle|\; h_\xi=h_0\cdot b^\xi,\xi\in\mathbb{N}_0\right\}$$

für jede Koordinatenrichtung $k=1,2,\ldots,d$ gegeben mit $0<b<1$ und $\theta=b^2$. Die Funktion $f\in C^{\mathcal{I}_m^{(d)}}\left[H^d\right]$ erfülle für alle $\vec{l}\in\tilde{\mathcal{I}}_m^{(d)}$:

$$\left|\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\right|\leq A_{\vec{l}}\leq A_{m+1}\qquad\forall\,\overline{h}_{\vec{\imath}}=\left(h_{i_{r_1}},h_{i_{r_2}},\ldots,h_{i_{r_d}}\right)$$
$$\textit{mit}\quad r_\xi\in I^D\left(\vec{l}\right)\quad\textit{und}\quad\vec{\imath}\in\mathcal{I}_m^{(d)}.$$

Dann gilt:

$$\begin{aligned}\left|\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}-\delta_{\vec{0}}\right| &\leq h_0^{2m+2}\cdot A_{m+1}\cdot c(\theta,d)\cdot\theta^{\frac{1}{2}\cdot\frac{(m+1-d)\cdot(m+1)}{d}}\\ \textit{und}\quad & 0<c(\theta,d)<\infty.\end{aligned}$$

Beweis. Es gilt:

$$\begin{aligned}
&\left|\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}-\delta_{\vec{0}}\right|\\
\overset{3.14}{=}\quad &\left|\sum_{\vec{l}\in\tilde{\mathcal{I}}_m^{(d)}}\Phi\left(\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{\imath}}\right)\cdot h_{\vec{\imath}}^{2\vec{l}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}\right|\\
\overset{3.17}{\leq}\quad &\sum_{\vec{l}\in\tilde{\mathcal{I}}_m^{(d)}} h_0^{2m+2}\cdot A_{\vec{l}}\cdot\left|p_m^{(d)}\left(\underline{\theta}^{l_1},\underline{\theta}^{l_2},\ldots,\underline{\theta}^{l_d}\right)\right|\\
\overset{3.18}{\leq}\quad & h_0^{2m+2}\cdot\sum_{\vec{l}\in\tilde{\mathcal{I}}_m^{(d)}}A_{\vec{l}}\cdot c_5(\theta,d)\cdot\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)}\\
\leq\quad & h_0^{2m+2}\cdot A_{m+1}\cdot c_5(\theta,d)\cdot\sum_{\vec{l}\in\tilde{\mathcal{I}}_m^{(d)}}\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)}\\
=\quad & h_0^{2m+2}\cdot A_{m+1}\cdot c_5(\theta,d)\cdot\sum_{\left|\vec{l}\right|=m+1}\theta^{\sum_{\xi=1}^{d}\sigma\left(l_\xi-1\right)}\\
\overset{3.19}{\leq}\quad & h_0^{2m+2}\cdot A_{m+1}\cdot c_5(\theta,d)\cdot c_6(\theta,d)\cdot\theta^{\frac{1}{2}\frac{(m+1-d)\cdot(m+1)}{d}}\\
=\quad & h_0^{2m+2}\cdot A_{m+1}\cdot c(\theta,d)\cdot\theta^{\frac{1}{2}\frac{(m+1-d)\cdot(m+1)}{d}}\\
\text{mit}\quad & c(\theta,d)=c_5(\theta,d)\cdot c_6(\theta,d).
\end{aligned}$$

Da $0<c_5(\theta,d)<\infty$ und $0<c_6(\theta,d)<\infty$, folgt $0<c(\theta,d)<\infty$. □

Hiermit ist nun gezeigt, daß $\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ ein Quadraturverfahren der Ordnung $O\left(h^{2m+2}\right)$ ist, wenn für die zugrundeliegende Integrationsregel $D_{\vec{0}}\left(\vec{h}\right)$ eine Entwicklung existiert, wie sie in Korollar 3.11 beschrieben ist. Dies trifft auf kartesische Produkte sowohl der Trapez-, als auch der Mittelpunktregel zu, sofern der Integrand hinreichend glatt ist.

Wir hatten bereits im eindimensionalen Fall gesehen, daß $\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ in die Romberg-Quadratur übergeht. Ebenso kann die Fehlerabschätzung nach (19) für den Fall $H=\left\{h_j\;\middle|\;h_j=h_0\cdot b^j, j\in\mathbb{N}_0\right\}$ aus Satz 3.20 hergeleitet werden:

$$\begin{aligned}\left|\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(1)}}-\delta_{\vec{0}}\right| &\leq h_0^{2m+2}\cdot A_{m+1}\cdot c(\theta,1)\cdot\theta^{\frac{1}{2}m\cdot(m+1)}\\ &= h_0^{2m+2}\cdot A_{m+1}\cdot c(\theta,1)\cdot\theta^{0+1+2+\ldots+m}\\ &= h_0^2\cdot h_1^2\cdot\ldots\cdot h_m^2\cdot A_{m+1}\cdot c(\theta,1).\end{aligned}$$

Im Rahmen der numerischen Beispiele wird die Abschätzung nach Satz 3.20 genauer diskutiert. An dieser Stelle soll lediglich noch auf die Koeffizienten $A_{\vec{l}}$ für $\left|\vec{l}\right| = m+1$ eingegangen werden. Sie ergeben sich aus $\dot{X}_{\vec{l}}\left(\overline{h}_{\vec{l}}\right)$ durch Anwendung des Mittelwertsatzes auf alle Koordinatenrichtungen. Man erhält:

$$A_{\vec{l}} \;\leq\; \prod_{\xi=1}^{d} \left|\frac{B_{2l_\xi}}{(2l_\xi)!}\right| \cdot \max_{\vec{x}\in H^d} \left|f^{\left(2\vec{l}\right)}(\vec{x}\,)\right|$$

(B_{2l_ξ} sind die Bernoulli-Zahlen). Da $\left|\frac{B_{2w}}{(2w)!}\right| \approx \frac{2}{(2\pi)^{2w}}$ gilt, folgt:

$$A_{\vec{l}} \;\approx\; \frac{2}{(2\pi)^{2m+2}} \cdot \max_{\vec{x}\in H^d} \left|f^{\left(2\vec{l}\right)}(\vec{x}\,)\right|. \tag{63}$$

Liegt im mehrdimensionalen Fall $A_{\vec{l}}$ für alle $\vec{l}$ mit $\left|\vec{l}\right| = m+1$ etwa in der gleichen Größenordnung, also $A_{\vec{l}} \approx A_{m+1}$, so wird der Verfahrensfehler durch das „mittlere“ Glied $\Phi\left(\dot{X}_{\vec{\lambda}}\left(\overline{h}_{\vec{1}}\right)\cdot h_{\vec{1}}^{2\vec{\lambda}}\right)_{\vec{1}\in\mathcal{I}_m^{(d)}}$ bestimmt, wobei für den Multiindex $\vec{\lambda} = (\lambda,\lambda,\ldots,\lambda)$ bei geeignetem m gilt: $m+1 = \lambda\cdot d = \left|\vec{\lambda}\right|$. Mit

$$\begin{aligned}
\sum_{\xi=1}^{d} \sigma(\lambda-1) &= d\cdot\sigma\left(\frac{m+1}{d}-1\right)\\
&= d\cdot\frac{1}{2}\cdot\left(\frac{m+1}{d}\right)\cdot\left(\frac{m+1}{d}-1\right)\\
&= \frac{1}{2}\cdot\frac{(m+1)\cdot(m+1-d)}{d}
\end{aligned}$$

folgt nämlich aufgrund (59):

$$\begin{aligned}
\left|\Phi\left(\dot{X}_{\vec{\lambda}}\left(\overline{h}_{\vec{1}}\right)\cdot h_{\vec{1}}^{2\vec{\lambda}}\right)_{\vec{1}\in\mathcal{I}_m^{(d)}}\right| &\overset{(59)}{\leq} h_0^{2m+2}\cdot A_{\vec{\lambda}}\cdot c_5(\theta,d)\cdot\theta^{\sum_{\xi=1}^{d}\sigma(\lambda-1)}\\
&= h_0^{2m+2}\cdot A_{\vec{\lambda}}\cdot c_5(\theta,d)\cdot\theta^{\frac{1}{2}\cdot\frac{(m+1)\cdot(m+1-d)}{d}}\\
&\approx h_0^{2m+2}\cdot A_{m+1}\cdot c(\theta,d)\cdot\theta^{\frac{1}{2}\cdot\frac{(m+1)\cdot(m+1-d)}{d}}.
\end{aligned}$$

3.4. Implementation des Verfahrens. Für den Fall $\mathcal{I} = \mathcal{I}_m^{(d)}$ soll nun eine Implementation des Quadraturverfahrens vorgestellt werden, die wir mit `qho` (Quadratur höherer Ordnung) bezeichnen wollen. Als Randbedingung soll festgelegt werden, daß die Parameter d und m erst zur Laufzeit bekannt sind. Damit kann `qho` als universelle Quadraturroutine verwendet werden, bei der die Dimension des Integranden und die Verfahrensordnung in der Parameterliste beim Aufruf übergeben werden.

Wir beschreiben im folgenden die grundlegende Datenstruktur, gehen dann auf Realisierungsdetails ein und stellen zum Schluß einige weitere Varianten des Verfahrens vor.

3.4.1. Datenstruktur. Definition 3.2, die die Elementaroperatoren $\Lambda^{[k]}, \Delta^{[k]}$ und Σ von Φ beschreibt, legt es nahe, ein Schema $(u_{\vec{\imath}})_{\vec{\imath} \in \mathcal{I}_m^{(d)}}$ als grundlegende Datenstruktur zu verwenden, so daß die Operatoren „am Platz“ arbeiten können.

Eine Datenstruktur, deren Komponenten durch einen Multiindex $\vec{\imath} = (i_1, i_2, \ldots, i_d)$ mit $|\vec{\imath}| \leq m$ adressiert werden, soll im folgenden als *Simplex* der Tiefe m bezeichnet werden. Unter der m-ten *Hyperebene* eines Simplizes verstehen wir die Menge aller Komponenten $u_{\vec{\imath}}$ mit $|\vec{\imath}| = m$.

Simplizes, die von gebräuchlichen Programmiersprachen nicht als Basisdatentyp zur Verfügung gestellt werden, können ersatzweise durch mehrdimensionale Felder realisiert werden. Bei dieser Lösung muß allerdings ein Speicherverschnitt hingenommen werden. So wird für $d = 2$ ein Feld mit $(m+1)^2$ Einträgen reserviert, von denen aber nur $\frac{(m+1)\cdot(m+2)}{2}$, also etwa die Hälfte, benötigt werden. Mit zunehmender Dimension verschlechtert sich der Speichernutzungsgrad drastisch zu $\frac{1}{d!}$. Da außerdem die Dimension des Feldes gemäß der eingangs aufgestellten Forderungen erst zur Laufzeit bekannt ist und derartige Datenstrukturen in den meisten Programmiersprachen nicht unterstützt werden, soll diese Implementationsmöglichkeit nicht weiter verfolgt werden.

Als Alternative bietet es sich an, ein lineares Feld der Größe $\left|\mathcal{I}_m^{(d)}\right|$ dynamisch bereitzustellen und darauf eine eineindeutige Speicherabbildungsfunktion

$$I_{\mathcal{I}_m^{(d)}} : \mathcal{I}_m^{(d)} \quad \rightarrow \quad \left\{0, 1, \ldots, \left|\mathcal{I}_m^{(d)}\right| - 1\right\}$$

zur Verfügung zu stellen. In Abbildung 14 sind zwei mögliche Beispiele einer Zuordnung zwischen $\mathcal{I}_3^{(2)}$ und $\{0, 1, \ldots, 9\}$ wiedergegeben. In der linken Darstellung wird das Schema zeilenweise, in der rechten diagonal (hyperebenenweise) durchnumeriert. Die zweite Speicherabbildungsfunktion weist für unsere Zwecke einige Vorteile gegenüber der ersten auf. Zum einen ist sie unabhängig von m. Möchte man beispielsweise die Tiefe des Simplizes erhöhen, was notwendig ist, wenn die Ordnung des Verfahrens dynamisch erhöht wird, so ist keine Neuorganisation der bisherigen Einträge nötig. Des weiteren vereinfachen sich einige Operatoren, die auf Hyperebenen arbeiten. So kann $\sum_{|\vec{\imath}|=m} u_{\vec{\imath}}$ effizient implementiert werden, indem Anfangs- und Endindex der m-ten Hyperebene berechnet und alle Einträge in diesem Indexbereich summiert werden. Eine direkte Umsetzung dieses Operators würde zuerst systematisch alle Multiindizes $\vec{\imath}$ mit $|\vec{\imath}| = m$ erzeugen und diese dann auf Feldindizes abbilden. Hierfür müßte eine starke Einbuße in der Laufzeiteffizienz hingenommen werden.

Es soll nun die hyperebenenweise Speicherabbildungsfunktion für eine beliebige Dimension d spezifiziert werden. Mit den Hilfsmitteln der Kombinatorik läßt sich zeigen, daß die j-te Hyperebene eines d-dimensionalen Simplizes

$$\left|\left\{\vec{\imath} \;\middle|\; |\vec{\imath}| = j\right\}\right| = \binom{d-1+j}{j}$$

Speicherplätze benötigt. Insgesamt müssen also für einen Simplex $(u_{\vec{\imath}})_{\vec{\imath} \in \mathcal{I}_m^{(d)}}$

$$\left|\mathcal{I}_m^{(d)}\right| = \sum_{j=0}^{m} \left|\left\{\vec{\imath} \;\middle|\; |\vec{\imath}| = j\right\}\right| = \sum_{j=0}^{m} \binom{d-1+j}{j} = \binom{d+m}{d}$$

$i_2 \downarrow$ / $i_1 \rightarrow$	0	1	2	3
0	0	1	2	3
1	4	5	6	
2	7	8		
3	9			

$i_2 \downarrow$ / $i_1 \rightarrow$	0	1	2	3
0	0	1	3	6
1	2	4	7	
2	5	8		
3	9			

ABB. 14. *Zwei mögliche Abbildungen von* $(i_1, i_2) \in \mathcal{I}_3^{(2)}$ *auf* $\{0, 1, \ldots, 9\}$.

Speicherplätze vorgesehen werden. Der Abbildungswert $I_{\mathcal{I}_m^{(d)}}(\vec{\imath})$ für einen beliebigen Multiindex $\vec{\imath} = (i_1, i_2, \ldots, i_d)$ mit $|\vec{\imath}| \leq m$ ergibt sich aus der Anfangsadresse der Hyperebene mit der Nummer $|\vec{\imath}|$, nämlich

$$\sum_{k=0}^{|\vec{\imath}|-1} \binom{d-1+j}{j} = \binom{d+|\vec{\imath}|-1}{d},$$

und dem Abbildungwert für den $(d-1)$-dimensionalen Multiindex $(i_1, i_2, \ldots, i_{d-1})$. Damit erhalten wir folgende rekursive Definition der Speicherabbildungsfunktion:

$$\begin{aligned} I_{\mathcal{I}_m^{(1)}}(i_1) &= i_1 && \text{falls } d = 1 \\ I_{\mathcal{I}_m^{(d)}}(i_1, i_2, \ldots, i_{d-1}, i_d) &= \tbinom{d+|\vec{\imath}|-1}{d} + I_{\mathcal{I}_m^{(d-1)}}(i_1, i_2, \ldots, i_{d-1}) && \text{falls } d > 1. \end{aligned}$$

3.4.2. Realisierung. Nach Definition 3.4 bietet sich folgende Gliederung des Algorithmus an:

- Berechnung des Anfangsschemas $\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath} \in \mathcal{I}_m^{(d)}}$
- Extrapolation ($\Lambda^{[k]}$ für $k = 1, \ldots, d$)
- Berechnung der Rückwärtsdifferenzen mit anschließender Summation der Verbesserungen ($\Sigma\Delta^{[d]}\Delta^{[d-1]} \cdots \Delta^{[1]}$).

Zu Beginn muß der Simplex mit den Werten $D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)$ für $\vec{\imath} \in \mathcal{I}_m^{(d)}$ aufgefüllt werden, wobei die Schrittweite $\vec{h}_{\vec{\imath}} = (h_{1,i_1}, h_{2,i_2}, \ldots, h_{d,i_d})$ in Abhängigkeit der verwendeten Schrittweitenfolgen $H^{[k]}, k = 1, \ldots, d$ zu wählen ist. Für $D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)$ kann wahlweise die Trapez- oder Mittelpunktsumme verwendet werden. Abbildung 15 stellt ein C-Programmstück dar, welches die Mittelpunktsumme `ms` mit der Schrittweite `h[0]`,`h[1]`,... `h[d]` über der Funktion `f` berechnet. Diese intuitiv naheliegende Realisierung ist allerdings ungeeignet, da die Verschachtelungstiefe der `for`-Schleifen von `d` abhängt, welches zur Übersetzungszeit noch nicht bekannt ist. Abhilfe schafft hier Rekursion. Abbildung 16 zeigt ein äquivalentes Programmstück. Hier wird mit jeder Inkarnation von `loop` dynamisch eine weitere Verschachtelungsebene der `for`-Schleife erzeugt, wobei die Rekursionstiefe durch den formalen Parameter `j` gesteuert wird.

```
ms = 0.;
for(x[d-1]=h[d-1]/2.;x[d-1]<1.;x[d-1]+=h[d-1])
  for(x[d-2]=h[d-2]/2.;x[d-2]<1.;x[d-2]+=h[d-2])
    ⋮
      for(x[0]=h[0]/2.;x[0]<1.;x[0]+=h[0])
          ms += f(x);
```

ABB. 15. *Statisch d-fach verschachtelte* `for`*-Schleife.*

```
ms = loop(d-1,x);
  ⋮
double loop(j,x);
int j; double *x;
{
  double m1;

  if(j<0) return f(x);
  else {
    m1 = 0.;
    for(x[j]=h[j]/2.;x[j]<1.;x[j]+=h[j])
      m1 += loop(j-1,x);
    return m1;
  }
}
```

ABB. 16. *Dynamisch d-fach verschachtelte* `for`*-Schleife.*

Es hat sich gezeigt, daß die Laufzeiteffizienz dieses Programmkonstrukts bei hoher Dimension stark abnimmt. Dies rührt nicht, wie man auf den ersten Blick vermutet, aus dem Aufwand für die Unterprogrammsprünge her, sondern aus ungünstigen Verteilungen der Schleifen. Wird beispielsweise bei einer doppelt verschachtelten Schleife die innere einmal, die äußere n-mal $(n > 1)$ durchlaufen, so müssen insgesamt $(n + 1)$ Schleifeninitialisierungen durchgeführt werden. Vertauscht man die Reihenfolge der Schleifen, was aufgrund der Kommutativität der Addition in diesem Fall möglich ist, so sind nur noch zwei Schleifeninitialisierungen durchzuführen. Der Initialisierungsaufwand der

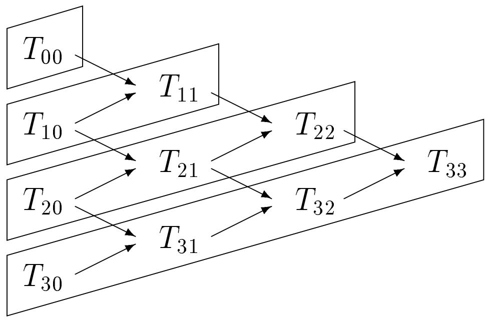

ABB. 17. *Abbildung von Zwischenwerten auf Speicherstellen.*

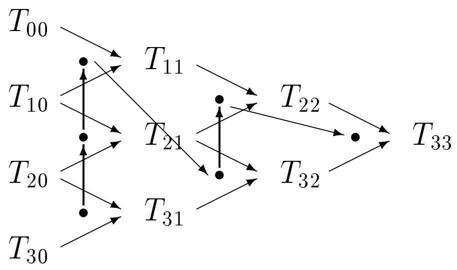

ABB. 18. *Die Berechnungsreihenfolge der Werte T_{ij}.*

Schleifen minimiert sich also, wenn die Zahl der Durchläufe mit der Verschachtelungstiefe zunimmt. Dieses Problem wurde in [BR92] aufgegriffen und näher untersucht. Die dort vorgenommenen Optimierungen beschleunigten dieses Programmsegment für die angeführten Beispiele etwa um den Faktor 10.

Die Extrapolation $(\hat{u}_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}_m^{(d)}} = \Lambda^{[k]}\,(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ für $k = 1, 2, \ldots, d$ kann nach dem Verfahren von Aitken und Neville durchgeführt werden. Bei vorgegebenem $\vec{\imath}$ mit $|\vec{\imath}| = m$ ergeben sich die Ein- und Ausgabewerte $T_{j,0}$ und $T_{j,j}$ des Verfahrens, wie es in (16) beschrieben ist, zu:

$$\begin{aligned} T_{j,0} &= u_{\vec{\imath}[i_k=j]} \\ T_{j,j} &= \hat{u}_{\vec{\imath}[i_k=j]} \end{aligned}$$

für $j = 0, 1, \ldots, i_k$. Da $\Lambda^{[k]}$ am Platz, also ohne zusätzliche Datenstrukturen arbeiten soll, ist die in Abbildung 17 angedeutete Zusammenfassung der Zwischenergebnisse notwendig. Die Programmschleifen sind so anzuordnen, daß die in Abbildung 18 wiedergegebene Berechnungsreihenfolge der $T_{i,j}$ eingehalten wird.

Die Bildung der Rückwärtsdifferenzen mit anschließender Summation vereinfacht sich für den Fall $\mathcal{I} = \mathcal{I}_m^{(d)}$ zu:

$$\Sigma\Delta^{[d]}\Delta^{[d-1]}\cdots\Delta^{[1]}\,(u_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}_m^{(d)}} = \sum_{r=0}^{d-1}\left((-1)^r \cdot \binom{d-1}{r} \cdot \sum_{|\vec{\imath}|=m-r} u_{\vec{\imath}}\right).$$

Die Gültigkeit dieser Äquivalenz läßt sich mit den Beweisschritten zu Lemma 3.16 zeigen. Es ist bereits darauf hingewiesen worden, daß der Teilausdruck $\sum_{|\vec{\imath}|=m-r} u_{\vec{\imath}}$ aufgrund der hyperebenenweisen Speicherabbildung besonders effizient implementiert werden kann.

3.4.3. Verfahrensvarianten. Sowohl Trapez- als auch Mittelpunktsummen als Anfangsapproximationen können für die Extrapolation verwendet werden. Implementationstechnisch gesehen muß dazu der erste Schritt des Verfahrens (Berechnung des Schemas $\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$) angepaßt werden. Für eine effiziente Implementation der Trapezsummen ist darauf zu achten, daß der Integrand nicht mehrfach an der gleichen Stelle ausgewertet wird. Bei Verwendung der Romberg-Schrittweitenfolge beispielsweise kann man hierzu die Beziehung $T\left(\frac{h}{2}\right) = \frac{1}{2}\cdot\left(M\left(h\right) + T\left(h\right)\right)$ ausnutzen, um die Folge $T\left(h\right), T\left(\frac{h}{2}\right), T\left(\frac{h}{4}\right), \ldots$ aus $T(h), M(h), M(\frac{h}{2}), \ldots$ zu berechnen.

Weitere Varianten von `qho` ergeben sich durch die Verwendung von beliebigen streng monoton nach Null fallenden Schrittweitenfolgen $H^{[k]}$, die für jede Koordinatenrichtung $k = 1, \ldots, d$ unabhängig gewählt werden können. Dabei ist darauf zu achten, daß dem Operator $\Lambda^{[k]}$ die richtige Schrittweitenfolge $H^{[k]}$ zur Verfügung gestellt wird.

Numerische Experimente im eindimensionalen Fall [BS64] haben gezeigt, daß die Extrapolation oftmals bessere Ergebnisse liefert, wenn rationale Funktionen (Nennergrad $\approx$ Zählergrad) anstelle von Polynomen verwendet werden. Da die Extrapolationsoperatoren $\Lambda^{[k]}, k = 1, \ldots, d$ auf die eindimensionale Extrapolation zurückgeführt sind, läßt sich diese Variante ebenfalls vorsehen. Programmtechnisch weist dieses Extrapolationsverfahren große Ähnlichkeit zu Verfahren von Aitken und Neville auf, weshalb es leicht in `qho` eingebaut werden kann.

Schließlich sei noch auf ein verwandtes Quadraturverfahren hingewiesen, das sich durch Auslassung des Extrapolationsschrittes ergibt:

$$\Psi\left(Q_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}} \quad := \quad \Sigma\Delta^{[d]}\Delta^{[d-1]}\ldots\Delta^{[1]}\left(Q_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}.$$

Hierbei handelt es sich um die Anwendung der *Kombinationstechnik* [GSZ90] auf die Integralnäherungen $Q_{\vec{\imath}}$, $\vec{\imath} \in \mathcal{I}_m^{(d)}$. Das Verfahren steht in enger Verwandtschaft zu `arch`. Legt man nämlich in allen Richtungen die Romberg-Schrittweitenfolgen zugrunde und wählt multivariate Trapezsummen $T_{\vec{\imath}} = T\left(h_{1,i_1}, h_{2,i_2}, \ldots, h_{d,i_d}\right)$ als Integrationsregeln, so stellt $\Psi\left(T_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ die nicht-adaptive Variante von `arch` mit maximaler Randfeinheit $h_{k,0}\cdot 2^{-m}$ dar. Dieses Verfahren ist somit von der Ordnung $O\left(h^2\cdot\left(\mathrm{ld}\left(h^{-1}\right)\right)^{d-1}\right)$.

Für $d = 2, 3$ wurde das Verfahren $\Psi\left(T_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ bereits in [Del89] und [DN89] näher untersucht. Hier wurde gezeigt, daß der Verfahrensfehler bei periodischen Integranden vergleichbar mit dem zahlentheoretischer Verfahren ist.

Extrapolationstechniken für die mehrdimensionale Quadratur wurden bereits in [Gen74] behandelt. Hier ist ein Verfahren vorgestellt worden, das numerisch die gleichen Resultate liefert wie `qho` mit Mittelpunktsummen und Romberg-Schrittweitenfolge. Die Extrapolation wird mit Hilfe eines verallgemeinerten rekursiven multivariaten Interpolationsverfahrens nach [McK72] durchgeführt. Es bietet allerdings nicht die Flexibilität, wie sie bei `qho` vorhanden ist. So ist es weder möglich, unterschiedliche Schrittweitenfolgen in den Koordinatenrichtungen zu wählen, noch eine andere Extrapolationsstrategie oder ein anderes Schema als $\mathcal{I}_m^{(d)}$ zu verwenden.

3.5. Numerische Experimente. Wir wählen als Integranden

Beispiel 6: $$f(x_1,\ldots,x_d) = e^{x_1+\ldots+x_d} \quad \text{mit} \quad (x_1,\ldots,x_d) \in [0,1]^d,$$

für $d = 2,\ldots,20$ und wenden darauf `qho` an, wobei wir die Romberg-Schrittweitenfolge ($\theta = \frac{1}{4}$) und Mittelpunktsummen zugrunde legen. Dabei wird der Parameter m, der die Ordnung des Verfahrens bestimmt, schrittweise erhöht. Die ermittelten Ergebnisse (Anzahl der Integrandenauswertungen und Genauigkeit in Dezimalstellen) sind in Abbildung 19 (links) wiedergegeben, wobei die Meßpunkte für festes d durch einen Polygonzug verbunden sind. Die Kennlinien sind nur bis zu den Ordnungen aufgetragen, an denen das Verfahren noch vernünftige Resultate lieferte. Bei höheren Ordnungen war entweder die Rechengenauigkeit von 15 Dezimalstellen erreicht, oder Rundungsfehler verfälschten die Resultate zu sehr. Im folgenden sollen die Kennlinienverläufe erklärt werden.

`qho` wertet ein reguläres dünnes Gitter aus. Für die Anzahl der Gitterpunkte gilt:

$$n \;=\; O\left(2^m \cdot (\mathrm{ld}\,(2^m))^{d-1}\right) = O\left(2^m \cdot m^{d-1}\right).$$

Somit sind die Meßwerte in der Horizontalen bei logarithmischer Achsenskalierung annähernd äquidistant aufgetragen.

Es soll nun die Fehlerabschätzung nach Satz 3.20 angewendet werden. Aufgrund der Eigenschaft

$$f^{\left(2\vec{l}\right)}(\vec{x}) = f(\vec{x}) = e^{x_1+x_2+\ldots+x_d}$$

des Integranden für beliebiges $\vec{l} \in \tilde{\mathcal{I}}_m^{(d)}$ und mit

$$\max_{\vec{x} \in H^d} |f(\vec{x})| = e^d$$

ergibt sich für den Koeffizienten A_{m+1} nach (63)

$$\begin{aligned} A_{m+1} \;&\approx\; \frac{2}{(2\pi)^{2m+2}} \cdot \max_{\vec{x} \in H^d} |f(\vec{x})| \\ &=\; \frac{2}{(2\pi)^{2m+2}} \cdot e^d. \end{aligned}$$

Wegen $h_0^{2m+2} = 1$ folgt aus Satz 3.20 der Verfahrensfehler

$$\begin{aligned} \left| \Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath} \in \mathcal{I}_m^{(d)}} - \delta_{\vec{0}} \right| \;&\approx\; \frac{2}{(2\pi)^{2m+2}} \cdot e^d \cdot c\left(\tfrac{1}{4}, d\right) \cdot \left(\tfrac{1}{4}\right)^{\frac{1}{2} \cdot \frac{(m+1-d)\cdot(m+1)}{d}} \\ &=\; O\left(\tfrac{1}{4}^{\frac{1}{2} \cdot \frac{m^2}{d}}\right). \end{aligned}$$

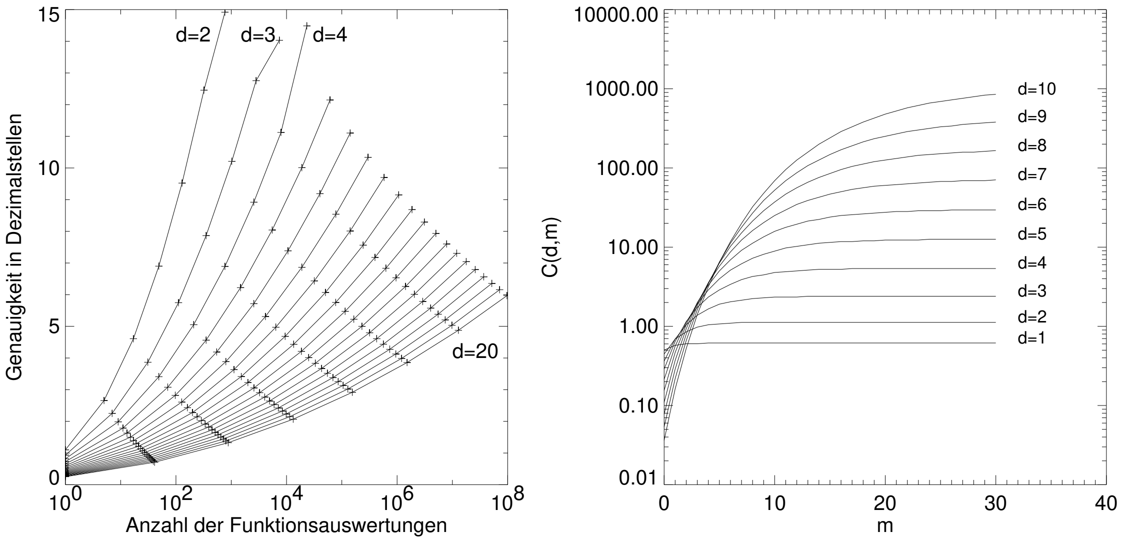

ABB. 19. BEISPIEL 6: *Ergebnisse für $d = 2, \ldots, 20$ (links) und $C(d,m)$ in Abhängigkeit von d und m (rechts).*

Die Kennlinien haben also annähernd einen parabelförmigen, mit der Dimension abflachenden Verlauf.

Nun soll gezeigt werden, daß mit der obigen Abschätzung der Verfahrensfehler im wesentlichen beschrieben ist. Wir bilden daher den Quotienten aller bekannten Größen der linken und rechten Seite und definieren:

$$C(d,m) \;:=\; \frac{\left|\Phi\left(D_{\vec{0}}\left(\vec{h}_{\vec{1}}\right)\right)_{\vec{1}\in\mathcal{I}_m^{(d)}} - \delta_{\vec{0}}\right|}{\frac{2}{(2\pi)^{2m+2}} \cdot e^d \cdot \left(\frac{1}{4}\right)^{\frac{1}{2}\cdot\frac{(m+1-d)\cdot(m+1)}{d}}}.$$

Mit $C(d,m)$ sind die aus den Termen A_{m+1} und $c\left(\frac{1}{4}, d\right)$ herrührenden Unschärfen der Verfahrensfehlerabschätzung zusammengefaßt. In Abbildung 19 (rechts) ist $C(d,m)$ in Abhängigkeit von $m = 0, \ldots, 30$ für $d = 1, \ldots, 10$ aufgetragen. Man erkennt, daß die nicht genauer untersuchten Einflußgrößen sehr schnell gegen eine obere Schranke für $m \to \infty$ streben.

Die Kurvenschar wurde mit Hilfe des Computer-Algebra-Programms MAPLE erstellt, wobei eine geschlossene Form der Mittelpunktsummen über den Integranden verwendet wurde. Für $d = 10$ und $m = 30$ wären andernfalls etwa $3.69 \cdot 10^{17}$ Funktionsauswertungen nötig gewesen. Um die Verfahrensfehler zu ermitteln, mußte teilweise bis auf 300 Dezimalstellen Genauigkeit gerechnet werden.

Das zweite Beispiel zeigt, daß Extrapolation in besonders ungünstigen Fällen zu schlechten Ergebnissen führen kann. Wir wählen als Integrand:

BEISPIEL 7: $\quad f(x_1, \ldots, x_d) = e^{-12.5\cdot\left(x_1^2+\ldots+x_d^2\right)} \quad$ für $\quad (x_1, \ldots, x_d) \in [\frac{1}{2}, \frac{1}{2}]^d,$

mit $d = 2, \ldots, 6$. Die Ergebnisse sind in Abbildung 20 (links) dargestellt. Der parabolische Verlauf der Kennlinien ist auch hier festzustellen. Die Resultate sind jedoch für kleines m völlig unbrauchbar. So wird etwa im Fall $d = 6$ und $m = 12$, das entspricht 38862849 Integrandenauswertungen, eine Genauigkeit von einer gültigen Dezimalstelle erreicht. Für kleineres m weichen die Resultate über zwei Zehnerpotenzen vom exakten Wert ab. Für $d > 6$ ist zu erwarten, daß auf diese Weise keine akzeptablen Ergebnisse mit realistischem Rechenaufwand zu erreichen sind.

Bessere Resultate in der Anfangsphase erhält man, wenn der Extrapolationsschritt ausgelassen wird, wenn also das Verfahren $\Psi\left(M_{\vec{\imath}}\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ verwendet wird. Abbildung 20 (rechts) gibt die Ergebnisse für $d = 2, \ldots, 6$ wieder. Das Verfahren entspricht nun der nicht-adaptiven Archimedischen Quadratur, weshalb sich der folgende Verfahrensfehler wegen $h = 2^{-m}$ ergibt zu

$$\begin{aligned}\left|\Psi\left(D_{\vec{0}}\left(\vec{h}_{\vec{\imath}}\right)\right)_{\vec{\imath}\in\mathcal{I}_m^{(d)}} - \delta_{\vec{0}}\right| &= O\left(2^m \cdot \left(\operatorname{ld}\left(\left(2^{-m}\right)^{-1}\right)\right)^{d-1}\right) \\ &= O\left(2^{-2m} \cdot m^{d-1}\right).\end{aligned}$$

Trotz der asymptotischen Überlegenheit des Extrapolationsverfahrens wird hier in der Anfangsphase höhere Genauigkeit erzielt. In Tabelle 1 sind nochmals die numerischen Ergebnisse beider Verfahren aufgeführt, in denen Φ gerade noch unterlegen ist, beziehungsweise Ψ gerade überholt hat.

Es handelt sich hier um ein besonders ungünstiges Beispiel, da das Maximum des Integranden als einzige Stützstelle für die erste Integralnäherung herangezogen wird. Diese besonders schlechte Näherung fließt beim Extrapolationsverfahren bei beliebig großem m in das Resultat ein, weshalb erst bei höheren Ordnungen eine Verbesserung des Ergebnisses eintritt. Beim anderen Verfahren geht diese Näherung bei $m \geq d$ nicht mehr ins Endresultat ein.

Die schlechte Anfangsnäherung wird vermieden, wenn die verwendete Schrittweite $H_1 = \left\{h_i \;\middle|\; h_i = 2^{-i}, i \in \mathbb{N}_0\right\}$ ersetzt wird durch $H_2 = \left\{h_i \;\middle|\; h_i = 2^{-i-1}, i \in \mathbb{N}_0\right\}$. Abbildung 21 gibt die entsprechenden Ergebnisse wieder. Wie man sieht, erzielen beide Verfahren in der Anfangsphase nun eine deutlich höhere Genauigkeit.

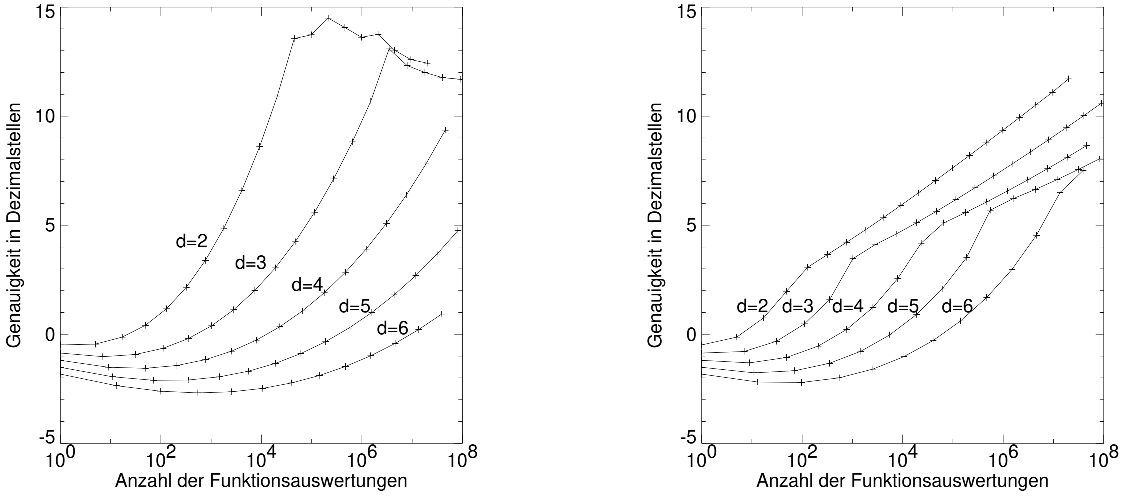

ABB. 20. BEISPIEL 7: *links mit Extrapolation, rechts ohne Extrapolation.*

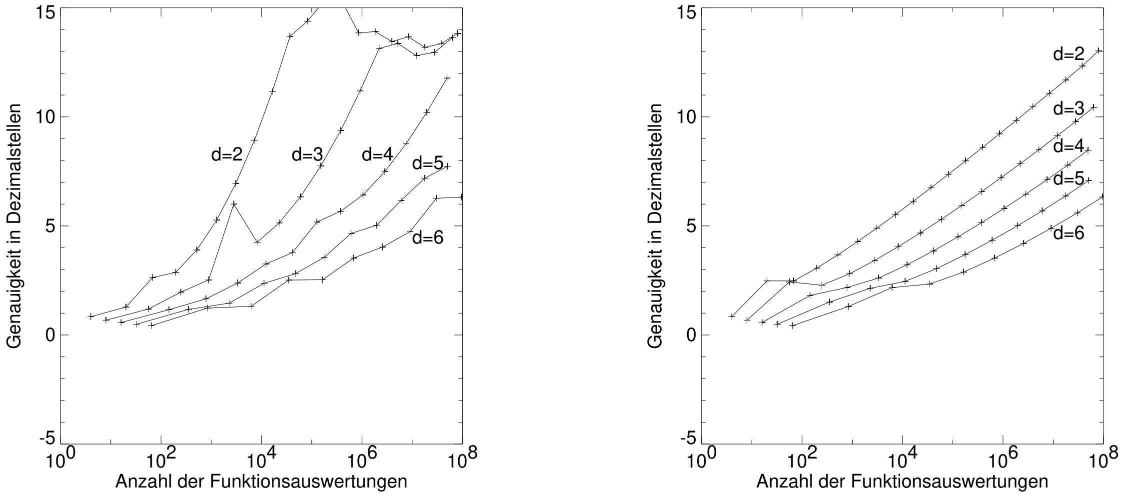

ABB. 21. BEISPIEL 7 *mit Schrittweitenfolge H_2: links mit Extrapolation, rechts ohne Extrapolation.*

d	m	$\left\|\mathcal{I}_m^{(d)}\right\|$	Genauigkeit in Dezimalstellen	
			ohne Extrapolation	mit Extrapolation
2	6	769	4.2193	3.3854
2	7	1793	4.7803	4.8638
3	10	114687	6.1746	5.5981
3	11	274431	6.7143	7.1234
4	14	18808833	8.1166	7.8081
4	15	45547521	8.6449	9.3589

TABELLE 1
BEISPIEL 7: *Vergleich der Resultate mit und ohne Extrapolation.*

4. Adaptive Quadratur höherer Ordnung. In den vorangegangenen Kapiteln haben wir zwei Quadraturverfahren mit unterschiedlichen Eigenschaften kennengelernt. Bei der Archimedischen Quadratur handelte es sich um ein verfeinerungsadaptives Verfahren fester Ordnung. Das Verfahren `qho` hingegen besitzt keine Adaptivität, dafür ist die Verfahrensordnung frei wählbar. Trotz ihrer Verschiedenheit haben sie gemeinsam, daß sie auf dünnen Gittern arbeiten.

In diesem Kapitel soll ein drittes Verfahren vorgestellt werden, das die Vorzüge der ersten beiden, also Verfeinerungsadaptivität und Ordnungsadaptivität, in sich vereinigt. Wir wollen es mit `aqho` (Adaptive Quadratur höherer Ordnung) bezeichnen.

Die Grundidee dieses Verfahrens bildet die Rekursivität regulärer dünner Gitter. Alle Stützstellen, die man durch die Auswertung eines solchen Gitters erhält, werden hier in einer Baumdatenstruktur organisiert, in der jeder Teilbaum für sich wiederum ein dünnes Gitter beschreibt. Werden die Auswertungen an die Vaterknoten weitergereicht, so läßt sich in jedem Knoten für das Gitter, das durch seinen Teilbaum repräsentiert wird, ein Schema von Mittelpunktsummen aufstellen, auf das das im vorigen Kapitel beschriebene Extrapolationsverfahren angewendet werden kann. Der Baum wächst breadth-first und paßt sich adaptiv dem Integranden an.

Im folgenden beschreiben wir zunächst die Baumdatenstruktur und deren dynamischen Aufbau. Des weiteren wird für die Extrapolation eine effiziente Realisierung vorgeführt sowie eine Adaptivitätsstrategie angegeben. Dabei muß auch die Frage geklärt werden, wie die Extrapolation durchzuführen ist, wenn, bedingt durch die Adaptivität, gewisse Stützstellen fehlen. An drei Stellen des Algorithmus werden wir Realisierungsvarianten vorstellen. Diese Modifikationen können beliebig kombiniert werden und ergeben somit acht Varianten von `aqho`. Für den Speicheraufwand der Datenstruktur wird gezeigt, daß die Kosten linear mit der Anzahl der Gitterpunkte ansteigt, obwohl für jeden Gitterknoten eine Datenstruktur bereitgestellt wird, auf der die Extrapolation durchgeführt wird. Zum Abschluß werden die Eigenschaften der Verfahrensvarianten anhand numerischer Beispiele demonstriert.

4.1. Datenstruktur. Die von `aqho` verwendete grundlegende Datenstruktur, ein Baum, läßt sich am besten mit Hilfe der rekursiven Definition für reguläre dünne Gitter erklären. Wir wollen hierfür eine geeignete Beschreibung angeben und betrachten zunächst den zweidimensionalen Fall.

Wir bezeichnen mit $G_m^{(2)}(a_1, a_2, b_1, b_2)$ die Menge aller Punkte eines regulären dünnen Gitters auf dem Grundgebiet $[a_1, b_1] \times [a_2, b_2]$ und der Feinheitsstufe m, das heißt mit $2^{m+1} - 1$ Gitterpunkten auf den Mittellinien. Im nicht-trivialen Fall $(m > 0)$ setzt sich dieses Gitter zusammen aus den Punkten auf der Mittellinie und denen der beiden Hälften, wobei letztere wiederum reguläre dünne Gitter darstellen, nun aber der Feinheitsstufe $(m - 1)$. Wir erhalten für $G_m^{(2)}(a_1, a_2, b_1, b_2)$ folgende rekursive Beschreibung:

$$\begin{aligned} G_m^{(2)}(a_1, a_2, b_1, b_2) &= g_m^{(2)}(a_1, a_2, b_1, b_2) \\ g_0^{(2)}(a_1, a_2, b_1, b_2) &= \left\{ \left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2} \right) \right\} \end{aligned}$$

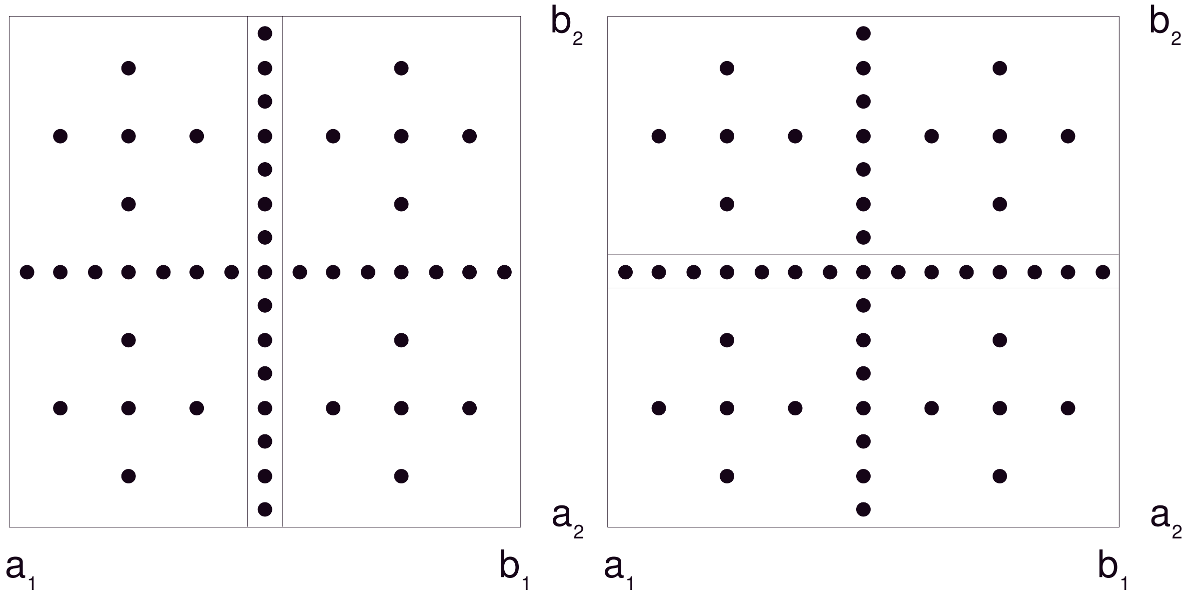

ABB. 22. *Zwei Möglichkeiten das dünne Gitter* $G_3^{(2)}(a_1, a_2, b_1, b_2)$ *zu zerlegen.*

$$
\begin{aligned}
g_{m'}^{(2)}(a_1, a_2, b_1, b_2) &= g_{m'}^{(1)}(a_1, a_2, b_1, b_2) \\
&\cup \; g_{m'-1}^{(2)}\left(a_1, a_2, \frac{a_1+b_1}{2}, b_2\right) \\
&\cup \; g_{m'-1}^{(2)}\left(\frac{a_1+b_1}{2}, a_2, b_1, b_2\right) \quad \text{für} \quad m' > 0 \\
g_0^{(1)}(a_1, a_2, b_1, b_2) &= \left\{\left(\frac{a_1+b_1}{2}, \frac{a_2+b_2}{2}\right)\right\} \\
g_{m'}^{(1)}(a_1, a_2, b_1, b_2) &= g_{m'}^{(0)}(a_1, a_2, b_1, b_2) \\
&\cup \; g_{m'-1}^{(1)}\left(a_1, a_2, b_1, \frac{a_2+b_2}{2}\right) \\
&\cup \; g_{m'-1}^{(1)}\left(a_1, \frac{a_2+b_2}{2}, b_1, b_2\right) \quad \text{für} \quad m' > 0 \\
g_{m'}^{(0)}(a_1, a_2, b_1, b_2) &= \left\{\left(\frac{a_1+b_1}{2}, \frac{a_2+b_2}{2}\right)\right\}.
\end{aligned}
$$

In Abbildung 22 (links) sind die Gitterpunkte für $G_3^{(2)}(a_1, a_2, b_1, b_2)$ dargestellt. Die aus der Definition herrührende Zerlegung

$$
\begin{aligned}
G_3^{(2)}(a_1, a_2, b_1, b_2) &= g_3^{(2)}(a_1, a_2, b_1, b_2) \\
&= g_3^{(1)}(a_1, a_2, b_1, b_2) \\
&\cup \; g_2^{(2)}\left(a_1, a_2, \frac{a_1+b_1}{2}, b_2\right)
\end{aligned}
$$

$$\cup \quad g_2^{(2)}\left(\frac{a_1+b_1}{2}, a_2, b_1, b_2\right) \quad \text{für} \quad m > 0$$

ist in der Abbildung durch Trennlinien angedeutet: $g_3^{(1)}(a_1, a_2, b_1, b_2)$ umfaßt die Gitterpunkte auf der vertikalen Mittellinie, $g_2^{(2)}\left(a_1, a_2, \frac{a_1+b_1}{2}, b_2\right)$ die der linken und $g_2^{(2)}\left(\frac{a_1+b_1}{2}, a_2, b_1, b_2\right)$ die der rechten Hälfte.

Man sieht also, daß $G_m^{(2)}(a_1, a_2, b_1, b_2)$ über Gebietshalbierungen in Richtung 1 definiert ist. Man hätte auch Richtung 2 dafür vorsehen können, wie dies in Abbildung 22 (rechts) angedeutet ist. Dazu sind in obigen Definitionsgleichungen lediglich die Variablen a_1 und a_2 beziehungsweise b_1 und b_2 paarweise zu vertauschen. Dieser Aspekt der beliebigen Wahl der Halbierungsrichtung soll bei den weiteren Betrachtungen mit berücksichtigt werden.

Wir wollen nun eine Definition für mehrdimensionale reguläre dünne Gitter auf einem beliebigen Gebiet

$$\left[\vec{a}, \vec{b}\right] \quad := \quad [a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_d, b_d]$$

angeben. Dabei fordern wir, daß die Halbierungsrichtung $k \in \{1, \ldots, d\}$ bei jeder Anwendung einer Definitionsgleichung frei gewählt werden kann. Zu diesem Zweck wird ein zusätzlicher Parameter mitgeführt:

$$\vec{v} \quad = \quad (v_1, v_2, \ldots, v_d) \quad \text{mit} \quad v_\xi \in \{0, 1\} \quad \text{für} \quad \xi = 1, \ldots, d,$$

der die aktuelle Gebundenheit aller Koordinatenrichtungen wiedergibt. Wir legen fest:

$$(64) \qquad v_\xi \quad \equiv \quad \begin{cases} 0 & \text{falls die Koordinatenrichtung } \xi \text{ nicht gebunden ist} \\ 1 & \text{falls die Koordinatenrichtung } \xi \text{ gebunden ist} \end{cases}$$

für $\xi = 1, \ldots, d$.

Für die Menge $G_m^{(d)}\left(\vec{a}, \vec{b}\right)$ der Punkte des regulären d-dimensionalen dünnen Gitters der Feinheitsstufe m auf dem Grundgebiet $\left[\vec{a}, \vec{b}\right]$ ergibt sich somit

$$(65) \qquad \begin{aligned} G_m^{(d)}\left(\vec{a}, \vec{b}\right) \quad &= \quad g_m^{(d)}\left(\vec{0}, \vec{a}, \vec{b}\right) \\ g_0^{(d')}\left(\vec{v}, \vec{a}, \vec{b}\right) \quad &= \quad \left\{\left(\frac{a_1+b_1}{2}, \frac{a_2+b_2}{2}, \ldots, \frac{a_d+b_d}{2}\right)\right\} \quad \text{für} \quad d' > 0 \\ g_{m'}^{(0)}\left(\vec{v}, \vec{a}, \vec{b}\right) \quad &= \quad \left\{\left(\frac{a_1+b_1}{2}, \frac{a_2+b_2}{2}, \ldots, \frac{a_d+b_d}{2}\right)\right\} \quad \text{für} \quad m' > 0 \\ g_{m'}^{(d')}\left(\vec{v}, \vec{a}, \vec{b}\right) \quad &= \quad g_{m'}^{(d'-1)}\left(\vec{v}\,[v_k = 1], \vec{a}, \vec{b}\right) \\ &\cup \quad g_{m'-1}^{(d')}\left(\vec{v}, \vec{a}, \vec{b}\left[b_k = \frac{a_k+b_k}{2}\right]\right) \\ &\cup \quad g_{m'-1}^{(d')}\left(\vec{v}, \vec{a}\left[a_k = \frac{a_k+b_k}{2}\right], \vec{b}\right), \end{aligned}$$

wobei k für jede Anwendung einer Definitionsgleichung aus der Menge der ungebundenen Richtungen zu wählen ist. Es gilt also $v_k = 0$. In jeder Gleichung gilt die Integritätsbedingung:

$$d' = \left|\left\{\xi \;\middle|\; v_\xi = 0 \quad \text{mit} \quad \xi = 1, \ldots, d\right\}\right| .$$

Definition (65) soll nun in eine andere Form gebracht werden. Durch fortgesetzte Substitution des Ausdrucks

$$g_{m'}^{(d'-1)}\left(\vec{v}\,[v_k = 1]\,, \vec{a}\,, \vec{b}\right)$$

erhält man:

$$\begin{aligned}
&(66) \qquad & G_m^{(d)}\left(\vec{a}\,, \vec{b}\right) &= g_m^{(d)}\left(\vec{0}\,, \vec{a}\,, \vec{b}\right) \\
&(67) \qquad & g_0^{(d')}\left(\vec{v}\,, \vec{a}\,, \vec{b}\right) &= \left\{\left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}, \ldots, \frac{a_d + b_d}{2}\right)\right\} \\
&(68) \qquad & g_{m'}^{(d')}\left(\vec{v}\,, \vec{a}\,, \vec{b}\right) &= \left\{\left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}, \ldots, \frac{a_d + b_d}{2}\right)\right\} \\
&& &\cup \bigcup_{j=1}^{d'} g_{m'-1}^{(j)}\left(\vec{v}_j, \vec{a}\,, \vec{b}\left[b_{k_j} = \frac{a_{k_j} + b_{k_j}}{2}\right]\right) \\
&& &\cup \bigcup_{j=1}^{d'} g_{m'-1}^{(j)}\left(\vec{v}_j, \vec{a}\left[a_{k_j} = \frac{a_{k_j} + b_{k_j}}{2}\right], \vec{b}\right) \quad \text{für} \quad m' > 0.
\end{aligned}$$

Für Gleichung (68) muß dabei lokal eine beliebige Folge aller ungebundenen Koordinatenrichtungen

$$\mathcal{F} = \{k_1, k_2, \ldots, k_{d'}\} \quad \text{mit} \quad v_{k_\xi} = 0 \quad \text{für} \quad \xi = 1, \ldots, d'$$

(k_ξ paarweise verschieden) angegeben werden, aus der die Koordinatenbindungen $\vec{v}_j$ für $j = 1, \ldots, d'$ wie folgt ermittelt werden:

$$(69) \qquad \vec{v}_j = (v_{j,1}, v_{j,2}, \ldots, v_{j,d}) : \quad v_{j,\xi} = 0 \quad \Leftrightarrow \quad \xi \in \{k_1, k_2, \ldots, k_j\} .$$

Die Gleichungen (67) und (68) bilden den Ausgangspunkt für die Beschreibung der Baumdatenstruktur.

Es sei nun auf einige Eigenschaften der Definitionsgleichungen hingewiesen. Zum einen bezeichnet $g_{m'}^{(d')}\left(\vec{v}\,, \vec{a}\,, \vec{b}\right)$ ein d'-dimensionales reguläres dünnes Gitter mit $2^{m+1} - 1$ Gitterpunkten auf der Mittelachse. Jedes Teilgitter $g_{m'}^{(d')}\left(\vec{v}\,, \vec{a}\,, \vec{b}\right)$ ist eindeutig durch seinen Mittelpunkt

$$\left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}, \ldots, \frac{a_d + b_d}{2}\right)$$

bestimmt. Zum anderen setzt sich jedes nicht-triviale Teilgitter rekursiv aus einer in der Dimension aufsteigenden Folge von Paaren von Teilgittern zusammen.

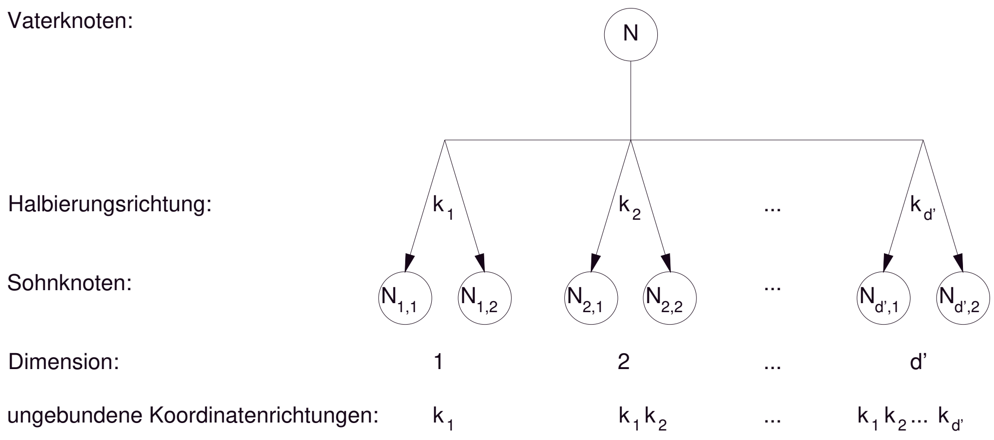

ABB. 23. *d'-dimensionaler Knoten mit seinen Sohnknoten.*

Wir leiten nun aus (67) und (68) eine Datenstruktur ab, die wir als *Dünn-Gitter-Baum* bezeichnen wollen. Jedem Knoten des Baumes ist dabei ein Teilgitter sowie der zugehörige Mittelpunkt zugeordnet. Die Verweisstruktur der Knoten untereinander ergibt sich aus den Kompositionsgleichungen von Teilgittern gemäß (67) und (68). Jedem Knoten sind die aktuellen Parameter $\vec{a}$, $\vec{b}$ und $\vec{v}$ verfügbar. Außerdem ist jeder Knoten selber Wurzel eines Teilbaumes, der ein d'-dimensionales dünnes Teilgitter beschreibt. Gilt $m' = 0$ für einen beliebigen Knoten N, so ist N ein Blatt. Gilt hingegen $m' > 0$, so besitzt N nach (68) insgesamt $2 \cdot d'$ Sohnknoten, die wir mit

$$N_{j,\psi} \quad \text{mit} \quad j = 1, \ldots, d' \quad \text{und} \quad \psi = 1, 2$$

bezeichnen wollen. Wir legen fest, daß ein Sohnknoten $N_{j,\psi}$ insgesamt j ungebundene Koordinatenrichtungen besitzt. In Abbildung 23 sind der Zusammenhang zwischen Vater- und Sohnknoten dargestellt sowie die jeweiligen Bindungen der Koordinatenrichtungen angegeben.

Für das dünne Gitter $G_3^{(2)}$ ist in Abbildung 24 ein möglicher Dünn-Gitter-Baum dargestellt. Man beachte dabei, daß die Folge $\mathcal{F}$ für jeden Knoten beliebig festgelegt werden kann. Im Knoten A erfolgte die Halbierung in Richtung 1, es gilt: $\mathcal{F}_A = \{2, 1\}$. Im Knoten B wurde in Richtung 2 halbiert ($\mathcal{F}_B = \{1, 2\}$), im Knoten C wieder in Richtung 1 ($\mathcal{F}_C = \{2, 1\}$).

4.2. Kompilation. Wir werfen einen Blick zurück auf unser Ausgangsproblem, der näherungsweisen Berechnung von

$$I \;=\; \int_{a_1}^{b_1} \int_{a_2}^{b_2} \ldots \int_{a_d}^{b_d} f\left(x_1, x_2, \ldots, x_d\right) dx_d \,\ldots\, dx_2 \, dx_1,$$

und legen über das Integrationsgebiet das dünne Gitter $G_m^{(d)}\left(\vec{a}, \vec{b}\right)$, zu dem wir einen entsprechenden Dünn-Gitter-Baum erzeugen. In jedem Knoten führen wir eine Auswertung des Integranden am zugeordneten Gitterpunkt durch und reichen diese Werte an

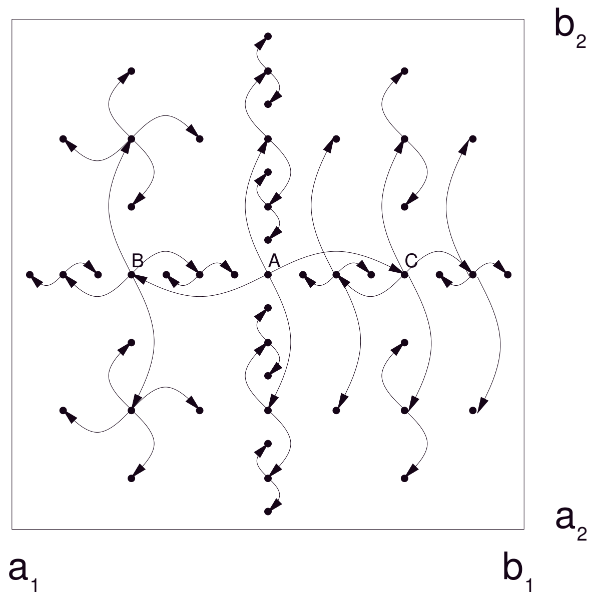

ABB. 24. *Baumstruktur der regulären dünnen Gitters* $G_3^{(2)}$.

alle Vaterknoten weiter. Im Wurzelknoten des Dünn-Gitter-Baumes kann hiermit ein Schema von (multivariaten) Mittelpunktsummen $(M_{\vec{\imath}})_{\vec{\imath}\in\mathcal{I}_m^{(d)}}$ gebildet werden, wie es bereits vom Verfahren `qho` bekannt ist. Aufgrund der Rekursivität regulärer dünner Gitter lassen sich in allen Knoten für die jeweiligen Teilgitter derartige Schemata aufstellen.

Wir wollen die in einem beliebigen Knoten N verfügbaren Mittelpunktsummen mit $M_{\vec{\imath}}\,(N)$ bezeichnen, wobei die Menge der Multiindizes $\vec{\imath}$, bedingt durch die ungebundenen Richtungen des Knotens, auf

$$\mathcal{I}_{m'}^{\langle N\rangle} \;:=\; \left\{\vec{\imath}\;\middle|\;\vec{\imath}=(i_1,i_2,\ldots,i_d)\in\mathcal{I}_{m'}^{(d)}\;\text{ mit }\; i_\xi=0\;\text{ falls }\; v_\xi=1\;\text{ für }\;\xi=1,\ldots,d\right\}$$

zu beschränken ist. Dabei bezeichnet $(m'+1)$ die Tiefe des Teilbaumes mit Wurzel N.

Die Schemawerte eines Knotens können aus denen seiner Sohnknoten berechnet werden. Diesen Vorgang bezeichnen wir als ,,Kompilation". Wir betrachten einen Knoten N, dem das Gitter $g_{m'}^{(d')}\left(\vec{v}\,,\vec{a}\,,\vec{b}\,\right)$ zugeordnet ist, sowie die zugehörige Folge $\mathcal{F}_N=\{k_1,k_2,\ldots,k_{d'}\}$ ungebundener Richtungen. Es seien $N_{\xi,1}$ und $N_{\xi,2}$ die beiden ξ-dimensionalen Sohnknoten zu N, dann ergibt sich die Kompilation für $\vec{\imath}\neq\vec{0}$ zu

$$M_{\vec{\imath}}\,(N) \;=\; M_{\vec{\imath}-\vec{e}_k}\,(N_{k,1})+M_{\vec{\imath}-\vec{e}_k}\,(N_{k,2})\,, \tag{70}$$

wobei zwischen k, $\vec{\imath}$ und der Folge $\mathcal{F}_N=\{k_1,k_2,\ldots,k_{d'}\}$ folgende Bedingungen einzuhalten sind:

$$\begin{aligned} k \;&\equiv\; k_\xi\in\mathcal{F}_N \\ \text{mit}\quad i_{k_\xi}\neq 0 \quad\wedge\quad & i_{k_{\xi+1}}=i_{k_{\xi+2}}=\ldots=i_{k_{d'}}=0. \end{aligned}$$

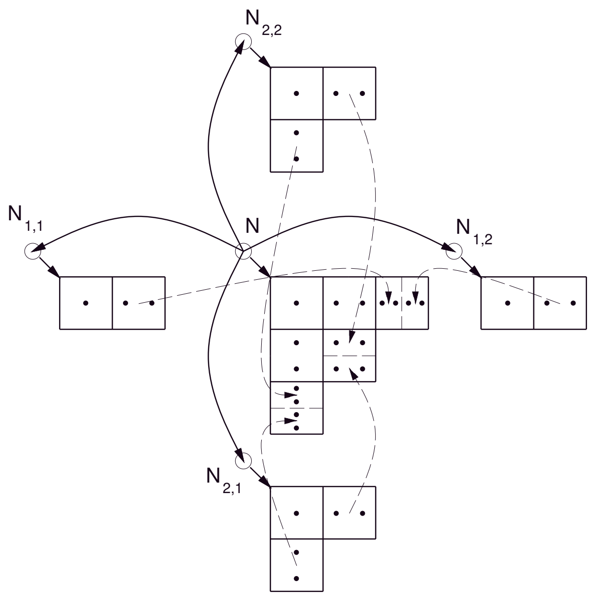

ABB. 25. *Kompilation.*

Für $\vec{\imath} = \vec{0}$ gilt:

$$M_{\vec{0}}(N) = (b_1 - a_1) \cdot (b_2 - a_2) \cdot \ldots \cdot (b_d - a_d) \cdot f\left(\frac{a_1 + b_1}{2}, \frac{a_2 + b_2}{2}, \ldots, \frac{a_d + b_d}{2}\right). \tag{71}$$

Abbildung 25 gibt den Kompilationsvorgang für $M_{\vec{\imath}}(N)$ mit $\vec{\imath} \in \mathcal{I}_2^{(2)}$ $(\mathcal{F}_N = \{1, 2\})$ wieder.

Auf das Schema $(M_{\vec{\imath}}(N))_{\vec{\imath} \in \mathcal{I}_{m'}^{\langle N \rangle}}$ jedes beliebigen Knotens kann das multivariate Extrapolationsverfahren des vorausgegangenen Kapitels angewendet werden. Man erhält eine Näherung

$$\begin{aligned} \hat{I}_{m'}(N) &:= \Phi\left(M_{\vec{\imath}}(N)\right)_{\vec{\imath} \in \mathcal{I}_{m'}^{\langle N \rangle}} \\ &= \Sigma \Delta^{[1]} \Delta^{[2]} \ldots \Delta^{[d]} \Lambda^{[1]} \Lambda^{[2]} \ldots \Lambda^{[d]} \left(M_{\vec{\imath}}(N)\right)_{\vec{\imath} \in \mathcal{I}_{m'}^{\langle N \rangle}}. \end{aligned} \tag{72}$$

der Ordnung $O\left(h^{2m'+2}\right)$ für das Integral

$$\begin{aligned} I(N) &:= \int_{a_1}^{b_1} \int_{a_2}^{b_2} \ldots \int_{a_d}^{b_d} f(\varphi_1, \varphi_2, \ldots, \varphi_d)\, dx_d \ldots dx_2\, dx_1 \\ \text{mit} \quad \varphi_\xi &\equiv \begin{cases} x_\xi & \text{falls} \quad v_\xi = 0 \\ \frac{a_\xi + b_\xi}{2} & \text{falls} \quad v_\xi = 1 \end{cases} \end{aligned}$$

$(\xi = 1, \ldots, d)$, sofern der Integrand hinreichend glatt ist. Gleichung (72) vereinfacht sich

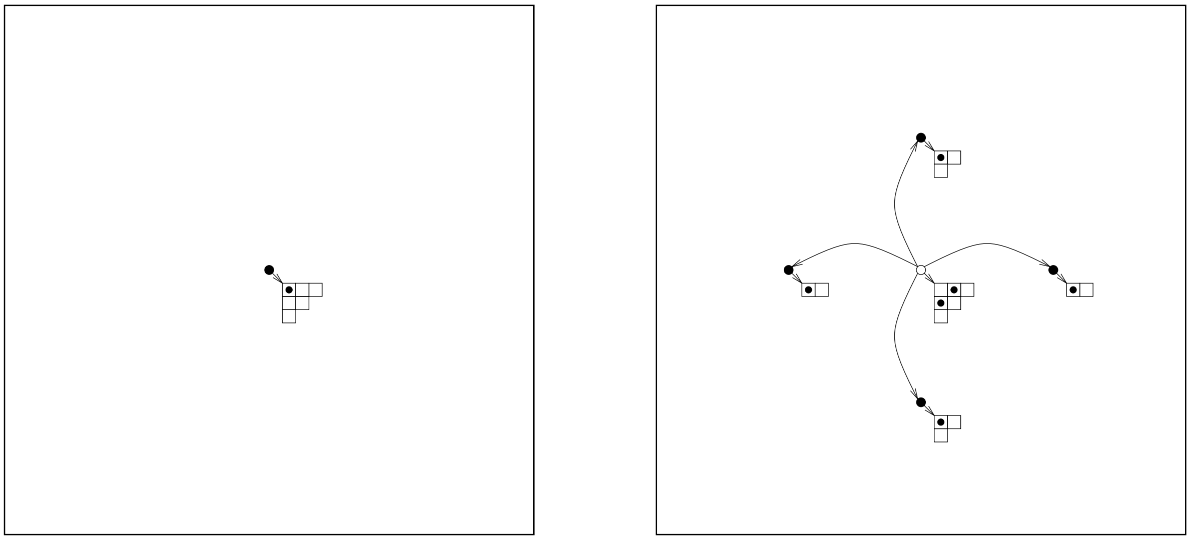

ABB. 26. *Aufbauschritt 0 (links) und 1 (rechts).*

dabei zu

$$\hat{I}_{m'}(N) \;=\; \Sigma\Delta^{[k_1]}\Delta^{[k_2]}\ldots\Delta^{[k_{d'}]}\Lambda^{[k_1]}\Lambda^{[k_2]}\ldots\Lambda^{[k_{d'}]}\left(M_{\vec{\imath}}(N)\right)_{\vec{\imath}\in\mathcal{I}_{m'}^{\langle N\rangle}},$$

da für alle Multiindizes $\vec{\imath} = (i_1, i_2, \ldots, i_d) \in \mathcal{I}_{m'}^{\langle N\rangle}$

$$i_\xi = 0 \quad \text{falls} \quad v_\xi = 1, 1 \le \xi \le d$$

gilt und die Operatoren $\Lambda^{[\xi]}$ und $\Delta^{[\xi]}$ für alle ξ mit $v_\xi = 1$ nach Definition in die Identität übergehen.

4.3. Aufbau der Datenstruktur. Das Verfahren erzeugt sukzessive die Gitter $G_0^{(d)}(\vec{a}, \vec{b})$, $G_1^{(d)}(\vec{a}, \vec{b})$, $G_2^{(d)}(\vec{a}, \vec{b})$, ... und baut den entsprechenden Dünn-Gitter-Baum auf. Dabei wird in einem beliebigen Schritt $m > 0$ die Datenstruktur für $G_{m-1}^{(d)}(\vec{a}, \vec{b})$ um die fehlenden Blattknoten ergänzt. Dazu wird der Baum depth-first durchlaufen. Trifft man auf einen Blattknoten, so wird dieser um seine Sohnknoten erweitert. Für jeden neu erzeugten Blattknoten wird der Integrand am zugeordneten Gitterpunkt ausgewertet und gemäß (71) im lokalen Schema gespeichert. Die Werte werden durch den Kompilationsvorgang in die Vaterknoten transportiert, wodurch schließlich alle Schemata um eine Hyperebene erweitert werden können. Nach erfolgter Extrapolation ist in allen Knoten eine neue Näherung für das lokal betrachtete Integral verfügbar, deren Ordnung höher als alle bisher errechneten Näherungen ist. Die Verfahrensordnung steigt also mit der Gesamttiefe des Baumes.

In den Abbildungen 26 und 27 sind die Aufbauschritte 0, 1 und 2 für den Fall $d = 2$ dargestellt. Die im jeweiligen Schritt neu errechneten Schemaeinträge sind durch Punkte markiert.

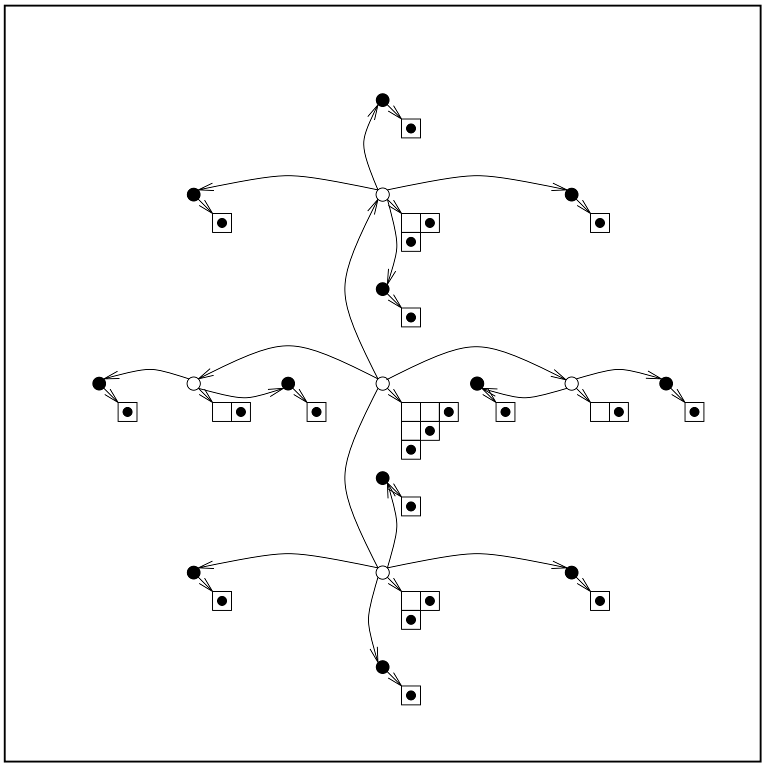

ABB. 27. *Aufbauschritt 2.*

Wird nun in einem Knoten N festgestellt, daß die zuletzt errechnete Näherung $\hat{I}_{m'}(N)$ hinreichend genau ist, so wird in N eine Markierung (Stop-Flag) gesetzt, die angibt, daß der zugehörige Teilbaum in folgenden Aufbauschritten nicht mehr durchlaufen wird. Die Markierung wird ebenfalls gesetzt, wenn *alle* Sohnknoten mit dem Stop-Flag gekennzeichnet sind. Auf diese Weise kann die noch zu spezifizierende Adaptivitätssteuerung das Wachstum des Baumes beeinflussen.

Der Baumaufbau ist abgeschlossen, wenn das Stop-Flag des Wurzelknotens gesetzt ist. Ist dies nach dem m-ten Aufbauschritt der Fall, so ergibt sich das Resultat aus dem Wurzelknoten W zu

$$I \quad \approx \quad \hat{I}_m(W) \,.$$

Wird im Laufe der Berechnung ein regulärer Dünn-Gitter-Baum aufgebaut, findet also keine adaptive Verfeinerung statt, so liefert `aqho` mit $\hat{I}_m(W)$ das gleiche Resultat wie das nicht-adaptive Verfahren `qho` bei entsprechender Ordnung.

4.4. Extrapolation. Wir wollen nun eine, im Hinblick auf einen schrittweisen Baumaufbau effiziente Realisierung der Extrapolation vorstellen. Hierzu betrachten wir zunächst den eindimensionalen Fall.

$$
\begin{tikzcd}[row sep=tiny, column sep=small]
M_0\,(N) = M_{0,0} \arrow[dr] & & & \\
& M_{1,1} \arrow[dr] & & \\
M_1\,(N) = M_{1,0} \arrow[ur] \arrow[dr] & & M_{2,2} \arrow[dr] & \\
& M_{2,1} \arrow[ur] \arrow[dr] & & M_{3,3} = \hat{I}_3\,(N) \\
M_2\,(N) = M_{2,0} \arrow[ur] \arrow[dr] & & M_{3,2} \arrow[ur] & \\
& M_{3,1} \arrow[ur] & & \\
M_3\,(N) = M_{3,0} \arrow[ur] & & &
\end{tikzcd}
$$

ABB. 28. *Das Extrapolationstableau nach dem Verfahren von Neville und Aitken.*

Es seien N, die Wurzel eines Baumes der Tiefe $(m'+1)$, sowie seine beiden Sohnknoten N_1 und N_2 gegeben. Des weiteren stehen die Mittelpunktsummen

$$
\begin{array}{lll}
M_i\,(N) & \text{für} & i = 0, \ldots, m', \\
M_i\,(N_1) & \text{für} & i = 0, \ldots, m'-1, \\
M_i\,(N_2) & \text{für} & i = 0, \ldots, m'-1
\end{array}
$$

zur Verfügung, wobei nach (70) gilt:

$$M_i\,(N) = M_{i-1}\,(N_1) + M_{i-1}\,(N_2) \quad \text{für} \quad i = 1, \ldots, m'.$$

Der Wert $\hat{I}_{m'}\,(N)$ ergibt sich aus der Extrapolation über alle $M_i\,(N)\,, i = 0, \ldots, m'$. Das Verfahren von Neville und Aitken lautet hierzu:

$$
\text{(73)} \qquad
\begin{array}{rcl}
a_k & = & \dfrac{-1}{4^k - 1} \\
M_{i,0} & = & M_i\,(N) \\
M_{i,k} & = & (1 - a_k) \cdot M_{i,k-1} + a_k \cdot M_{i-1,k-1} \quad \text{für} \quad 1 \le k \le i \le m' \\
\hat{I}_{m'}\,(N) & = & M_{m',m'}.
\end{array}
$$

In Abbildung 28 sind die Datenabhängigkeiten der Tableauwerte $M_{i,k}, 0 \le k \le i \le 3$ angegeben.

Mit dem nächsten Baumaufbauschritt ist der neue Startwert

$$M_{m'+1}\,(N) = M_{m'}\,(N_1) + M_{m'}\,(N_2)$$

verfügbar, womit nun nach (73) die Näherung $\hat{I}_{m'+1}\,(N)$ extrapoliert werden kann. Hierzu werden *alle* Tableauwerte $M_{i,k}, 1 \le k \le i \le m'+1$ ermittelt, obwohl die Werte $M_{i,k}, 1 \le k \le i \le m'$ in der vorangegangenen Extrapolation für $\hat{I}_{m'}\,(N)$ schon einmal berechnet wurden. Es stellt sich also die Frage, ob sich die Extrapolation nicht vereinfachen läßt.

Wir wollen nun zeigen, daß dieser Nachteil behoben wird, wenn in allen Knoten N die Rückwärtsdifferenzen der Näherungswerte gehalten werden, also

$$
\text{(74)} \qquad M_i^*\,(N) \;:=\; \begin{cases} \hat{I}_0\,(N) & \text{für} \quad i = 0 \\ \hat{I}_i\,(N) - \hat{I}_{i-1}\,(N) & \text{für} \quad i > 0. \end{cases}
$$

$\hat{I}_{m'}(N)$ ergibt sich dann zu:

$$\hat{I}_{m'}(N) \;=\; \sum_{j=0}^{m'} M_i^*(N)\,.$$

Der Zusammenhang zwischen den Werten M_i^* von N, N_1 und N_2 zu den Zwischenergebnissen der Extrapolation lautet:

$$
\begin{aligned}
M_0^*(N) &= M_{0,0} \\
M_i^*(N) &= M_{i,i} - M_{i-1,i-1} \quad \text{für} \quad i = 1,\ldots,m' \\
M_0^*(N_1) + M_0^*(N_2) &= M_{1,0} \\
M_i^*(N_1) + M_i^*(N_2) &= M_{i+1,i} - M_{i,i-1} \quad \text{für} \quad i = 1,\ldots,m'-1
\end{aligned}
\tag{75}
$$

Es gilt nun folgende Beziehung:

Satz 4.1. *Es sei* $i = 0,\ldots,m'$ *und* $a_i \in \mathbb{R} \setminus \{0,1\}$ *sowie*

$$M_{i,i} = (1-a_i)\cdot M_{i,i-1} + a_i \cdot M_{i-1,i-1} \quad \textit{mit} \quad i = 0,\ldots,m, \tag{76}$$

dann gilt:

$$M_{1,1} - M_{0,0} \;=\; (1-a_1)\cdot(M_{1,0} - M_{0,0}) \tag{77}$$

$$
\begin{aligned}
M_{i,i} - M_{i-1,i-1} \;=\;& \frac{a_{i-1}\cdot(1-a_i)}{1-a_{i-1}}\cdot(M_{i-1,i-1} - M_{i-2,i-2}) \\
+\;& (1-a_i)\cdot(M_{i,i-1} - M_{i-1,i-2})\,.
\end{aligned}
\tag{78}
$$

Beweis. Behauptung (77) ergibt sich sofort durch Umformen:

$$
\begin{aligned}
M_{1,1} - M_{0,0} &\overset{(76)}{=} (1-a_1)\cdot M_{1,0} + a_1 \cdot M_{0,0} - M_{0,0} \\
&= (1-a_1)\cdot(M_{1,0} - M_{0,0})\,.
\end{aligned}
$$

Behauptung (78) ergibt sich nach mehrmaligem Einsetzen von (76) und Umformen:

$$
\begin{aligned}
& M_{i,i} - M_{i-1,i-1} \\
\overset{(76)}{=}\;& (1-a_i)\cdot M_{i,i-1} + a_i\cdot M_{i-1,i-1} - M_{i-1,i-1} \\
=\;& (1-a_i)\cdot M_{i,i-1} - (1-a_i)\cdot M_{i-1,i-1} \\
\overset{(76)}{=}\;& (1-a_i)\cdot M_{i,i-1} - (1-a_i)\cdot(1-a_{i-1})\cdot M_{i-1,i-2} - (1-a_i)\cdot a_{i-1}\cdot M_{i-2,i-2} \\
=\;& -\frac{a_{i-1}\cdot(1-a_i)}{1-a_{i-1}}\cdot(1-a_{i-1})\cdot M_{i-2,i-2} \\
& + \left(-(1-a_i) + \frac{a_{i-1}\cdot(1-a_i)}{1-a_{i-1}}\cdot(1-a_{i-1})\right)\cdot M_{i-1,i-2} + (1-a_i)\cdot M_{i,i-1}
\end{aligned}
$$

$$
\begin{tikzcd}[column sep=small]
M_{0,0} \arrow[dr, "0"] & & & \\
& M_{1,1} - M_{0,0} \arrow[dr, "\frac{a_1 \cdot (1-a_2)}{1-a_1}"] & & \\
M_{1,0} - M_{0,0} \arrow[ur, "1-a_1"'] & & M_{2,2} - M_{1,1} \arrow[dr, "\frac{a_2 \cdot (1-a_3)}{1-a_2}"] & \\
& M_{2,1} - M_{1,0} \arrow[ur, "1-a_2"'] & & M_{3,3} - M_{2,2} \\
& & M_{3,2} - M_{2,1} \arrow[ur, "1-a_3"'] &
\end{tikzcd}
$$

ABB. 29. *Das Berechnungsschema nach Satz 4.1.*

$$
\begin{aligned}
&= -\frac{a_{i-1} \cdot (1-a_i)}{1-a_{i-1}} \cdot \left(M_{i-2,i-2} - (1-a_{i-1}) \cdot M_{i-1,i-2} - a_{i-1} \cdot M_{i-2,i-2}\right) \\
&\quad + (1-a_i) \cdot \left(M_{i,i-1} - M_{i-1,i-2}\right) \\
&\overset{(76)}{=} \frac{a_{i-1} \cdot (1-a_i)}{1-a_{i-1}} \cdot \left(M_{i-1,i-1} - M_{i-2,i-2}\right) + (1-a_i) \cdot \left(M_{i,i-1} - M_{i-1,i-2}\right).
\end{aligned}
$$

□

Das sich aus (77) und (78) ergebende Berechnungsschema ist in Abbildung 29 wiedergegeben.

Mit Hilfe von $a_i = \frac{-1}{4^i-1}$, den Beziehungen (75) und Satz 4.1 ergibt sich eine andere Realisierung der Extrapolation, wie sie in `aqho` verwendet wird:

$$
\begin{aligned}
M_0^*(N) &= M_0(N) \\
M_1^*(N) &= \tfrac{4}{3} \cdot \left(M_0^*(N_1) + M_0^*(N_2) - M_0^*(N)\right) \\
M_i^*(N) &= -\frac{4}{4^i-1} \cdot M_{i-1}^*(N) \\
&\quad + \frac{4^i}{4^i-1} \cdot \Big(M_{i-1}^*(N_1) + M_{i-1}^*(N_2)\Big) \quad \text{für} \quad i > 1.
\end{aligned} \tag{79}
$$

In Abbildung 30 ist das sich daraus ergebende Berechnungsschema wiedergegeben.

(79) stellt nun eine effizientere Realisierung der Extrapolation dar als das Verfahren von Neville und Aitken. Anstatt ein komplettes Tableau zu errechnen, braucht jetzt nur noch *eine* der Gleichungen (79) angewendet zu werden. Der sich daraus ergebende Datenfluß zwischen Sohn- und Vaterknoten ist in Abbildung 31 (links) angedeutet.

Darüber hinaus bietet dieser Extrapolations-Algorithmus zwei weitere Vorteile: Zum einen gibt M_i^* den Beitrag wieder, um den die Integralnäherung durch Erhöhung der Ordnung verändert wurde. M_i^* kann somit als Fehlerschätzer für Adaptivitätsstrategien verwendet werden.

Zum anderen erleichtert sich die Extrapolation bei Adaptivität. Wir betrachten dazu den Sohnknoten N_1, dessen Stop-Flag gesetzt wurde, nachdem $M_{m'-1}^*(N_1)$ berechnet

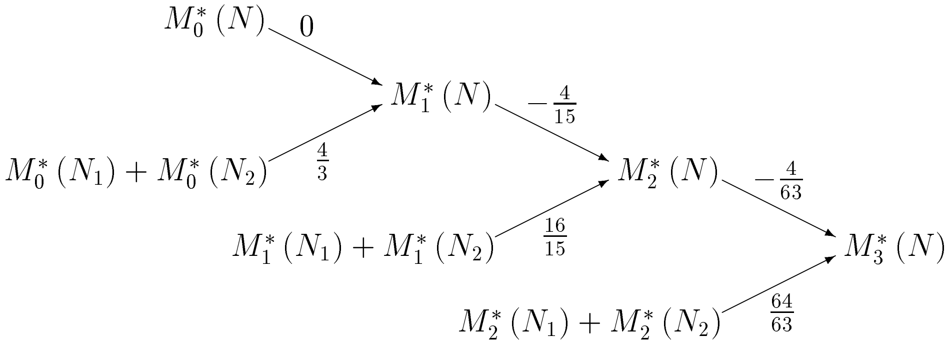

ABB. 30. *Extrapolation mit Hilfe der Rückwärtsdifferenzen.*

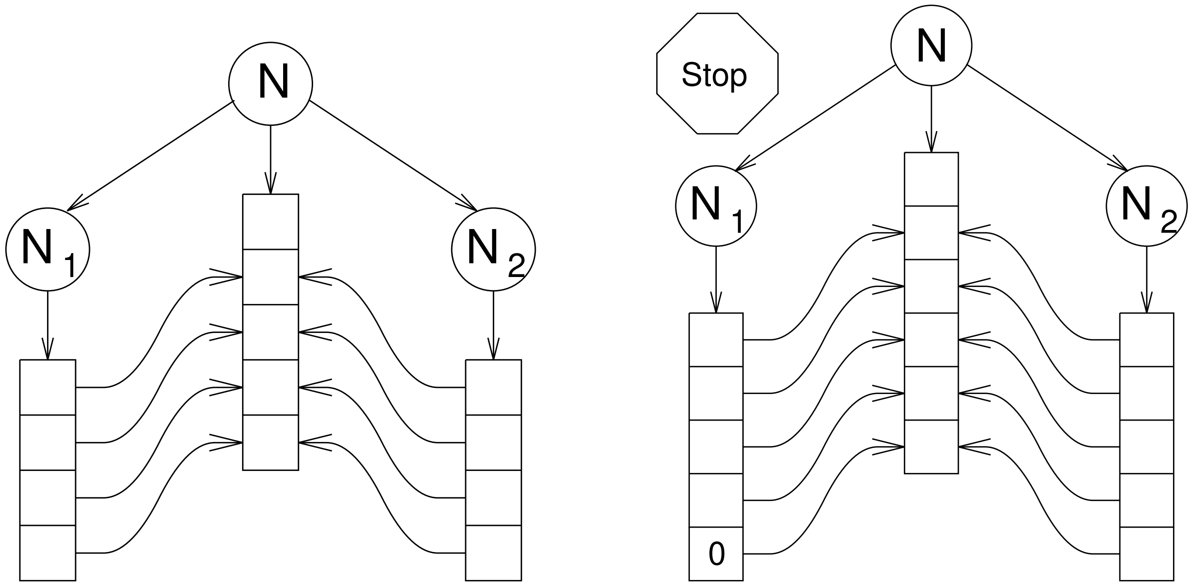

ABB. 31. *Adaptive Quadratur höherer Ordnung im eindimensionalen Fall.*

wurde. Es stehen also die Werte

$$M_0^*(N_1), M_1^*(N_1), \ldots, M_{m'-1}^*(N_1) \tag{80}$$

zur Verfügung. Es tritt nun das Problem auf, daß die für $M_{i+1}^*(N)$, $i \geq m'$ benötigten Werte $M_i^*(N_1)$ nicht mehr errechnet werden. Eine Lösung ergibt sich, wenn man die fehlenden $M_i^*(N_1)$ zu Null annimmt (siehe Abbildung 31 rechts). Dies ist sinnvoll, da das Stop-Flag von N_1 gesetzt wurde, weil durch die Erhöhung der Ordnung keine wesentliche Verbesserung der Genauigkeit für das Gesamtresultat zu erwarten ist und somit die fehlenden Wert sehr nahe bei Null liegen.

Würde man hingegen die Mittelpunktsummen

$$M_0(N_1), M_1(N_1), \ldots, M_{m'-1}(N_1) \tag{81}$$

in den Knoten halten, so müßten erst die fehlenden Werte $M_i(N_1), i \geq m'$ nach dem Verfahren von Aitken-Neville extrapoliert werden, was einen zusätzlichen Rechenaufwand bedeutet.

Wir wollen uns nun der mehrdimensionalen Verallgemeinerung des beschriebenen Extrapolationsverfahrens (79) zuwenden. Für den d-dimensionalen Fall gilt:

$$(M^*_{\vec{\imath}}(N))_{\vec{\imath}\in\mathcal{I}^{\langle N\rangle}_{m'}} \;:=\; \Delta^{[k_1]}\Delta^{[k_2]}\ldots\Delta^{[k_{d'}]}\Lambda^{[k_1]}\Lambda^{[k_2]}\ldots\Lambda^{[k_{d'}]}(M_{\vec{\imath}}(N))_{\vec{\imath}\in\mathcal{I}^{\langle N\rangle}_{m'}}.$$

Wir betrachten einen Knoten N, für den $\vec{v}$ und $\mathcal{F}_N = \{k_1, k_2, \ldots, k_{d'}\}$ gegeben sind, sowie seine Sohnknoten $N_{j,\psi}, j = 1, \ldots, d'$ und $\psi = 1, 2$. Außerdem befindet sich der Dünn-Gitter-Baum mit Wurzel N im Aufbauschritt m', wobei die Werte $M^*_{\vec{\imath}}(N_{j,\psi})$ mit $\vec{\imath} \in \mathcal{I}^{\langle N_{j,\psi}\rangle}_{m'-1}$ mit $j = 1, \ldots, d'$ und $\psi = 1, 2$ bereits vorliegen und mit denen des Vaterknotens verrechnet werden müssen. Für N muß nun $M^*_{\vec{\imath}}(N)$ für $\vec{\imath} \in \tilde{\mathcal{I}}^{\langle N\rangle}_{m'}$ mit

$$\tilde{\mathcal{I}}^{\langle N\rangle}_{m'} \;:=\; \left\{\vec{\imath} \;\middle|\; \vec{\imath} \in \mathcal{I}^{\langle N\rangle}_{m'} \quad\wedge\quad |\vec{\imath}| = m'\right\}$$

berechnet werden. Wir betrachten einen beliebigen Multiindex $\vec{\imath} \in \tilde{\mathcal{I}}^{\langle N\rangle}_{m'}$ und ermitteln diejenige ungebundene Koordinatenrichtung $k \equiv k_j \in \mathcal{F}_N = \{k_1, k_2, \ldots, k_{d'}\}$, für die gilt:

$$i_{k_j} \neq 0 \quad\wedge\quad i_{k_{j+1}} = i_{k_{j+2}} = \ldots = i_{k_{d'}} = 0.$$

Die Berechnung des zugehörigen $M^*_{\vec{\imath}}(N)$ für $\vec{\imath} \in \tilde{\mathcal{I}}^{\langle N\rangle}_{m}$ erfolgt in Analogie zur Kompilation der Mittelpunktsummen aus den Sohnknoten. Es gilt somit:

$$\begin{aligned}
M^*_{\vec{0}}(N) &= M_{\vec{0}}(N)\\
M^*_{\vec{\imath}}(N) &= \tfrac{4}{3}\cdot\Big(M^*_{\vec{\imath}-\vec{e}_k}(N_{j,1}) + M^*_{\vec{\imath}-\vec{e}_k}(N_{j,2}) - M^*_{\vec{\imath}-\vec{e}_k}(N)\Big) \quad \text{falls} \quad i_k = 1\\
M^*_{\vec{\imath}}(N) &= -\frac{4}{4^{i_k}-1}\cdot M^*_{i-1}(N)\\
&\quad+\frac{4^{i_k}}{4^{i_k}-1}\cdot\Big(M^*_{\vec{\imath}-\vec{e}_k}(N_{j,1}) + M^*_{\vec{\imath}-\vec{e}_k}(N_{j,2})\Big) \quad \text{falls} \quad i_k > 1.
\end{aligned} \tag{82}$$

4.5. Festlegung der Halbierungsrichtungen. Die Gebietszerlegungsstrategie des Verfahrens wird durch die Wahl der Folge $\mathcal{F}_N$ von ungebundenen Koordinatenrichtungen in einem beliebigen Knoten N bestimmt. Wir haben bereits bei der Archimedischen Quadratur gesehen, daß bei zyklischem Tausch der Halbierungsrichtung eine gleichmäßigere Abtastung des Integranden möglich ist, die zu leicht verbesserten Ergebnissen führt. Diese Strategie soll im folgenden formalisiert werden. Alternativ dazu stellen wir eine weitere Strategie vor, die die Glattheit des Integranden in den jeweiligen Koordinatenrichtungen berücksichtigt.

Es sei N ein beliebiger Knoten mit $\mathcal{F}_N = \{k_1, k_2, \ldots, k_{d'}\}$, dann besitzt ein j-dimensionaler Sohnknoten $N_{j,\psi}, \psi \in \{1, 2\}$ gemäß (69) die ungebundenen Koordinatenrichtungen $\{k_1, k_2, \ldots, k_j\}, 1 \leq j \leq d'$. Nach zyklischem Tausch ergibt sich somit:

$$\mathcal{F}_{N_{j,\psi}} \;=\; \{k_j, k_1, k_2, \ldots, k_{j-1}\}.$$

Es muß noch $\mathcal{F}_W$ für den Wurzelknoten W festgelegt werden. Wir wählen:

$$\mathcal{F}_W \quad = \quad \{d, d-1, \ldots, 2, 1\},$$

womit die erste Strategie vollständig beschrieben ist.

Für die zweite Strategie benötigen wir eine Abschätzung der Glattheit des Integranden in Abhängigkeit von einer Richtung k. Wir definieren für einen beliebigen Knoten N:

$$\lambda\left(\xi\right) \quad := \quad \left|M_{\vec{0}}\left(N\right) - M_{\vec{e}_\xi}\left(N\right)\right|$$

für $\xi = 1, \ldots, d$ mit $v_\xi = 0$. $\mathcal{F}_N$ wird dann wie folgt festgelegt:

$$\mathcal{F}_N = \{k_1, k_2, \ldots, k_{d'}\}: \quad \lambda\left(k_1\right) \leq \lambda\left(k_2\right) \leq \ldots \leq \lambda\left(k_{d'}\right).$$

Die Strategie bewirkt, daß beispielsweise bei der Integration von

$$f\left(x_1, x_2\right) = \tilde{f}\left(x_1\right) \quad \text{mit} \quad \left[x_1, x_2\right] \in [0,1]^2$$

nur Gebietszerlegungen in Richtung x_1 durchgeführt werden, was sinnvoll ist, da f entlang den anderen Richtung konstant ist. Bezogen auf den d-dimensionalen Fall bedeutet diese Strategie, daß die Zahl der ungebundenen Variablen in einem Sohnknoten von der Glattheit des Integranden im Vaterknoten in der jeweiligen Richtung abhängt. Je höher diese ist, um so mehr Freiheitsgrade sind dem Sohnknoten zugeordnet.

4.6. Adaptivität. Das Wachstum des Dünn-Gitter-Baumes wird durch die Stop-Flags gesteuert. Die Entscheidung über das Setzen eines Stop-Flags in einem Knoten N fällt, sobald der zu N gehörige Teilbaum durchlaufen ist und eine neue Näherung $\hat{I}_{m'}\left(N\right)$ errechnet wurde, also unmittelbar bevor der m'-te Aufbauschritt des zu N gehörigen Dünn-Gitter-Baumes abgeschlossen ist.

Im folgenden sollen zwei Adaptivitätsstrategien vorgestellt werden, von denen die erste als fehlerorientiert, die zweite als kostenorientiert bezeichnet werden kann.

4.6.1. Die fehlerorientierte Adaptivitätssteuerung. Ziel dieser Strategie ist es, den Baumaufbau so zu steuern, daß der relative Fehler des Gesamtintegrals

$$\tilde{\varepsilon} \quad = \quad \left|\frac{I\left(W\right) - \hat{I}_m\left(W\right)}{I\left(W\right)}\right|$$

nach einem beliebigen Aufbauschritt m kleiner als eine vom Anwender vorgegebene Schranke ε_{rel} ist. Teilbäume, in denen das lokal betrachtete Integral hinreichend genau berechnet wurde, sollen nicht weiter wachsen. Hierbei wird in jedem Knoten N eine Abschätzung $E_{m'}\left(N\right)$ für den (absoluten) Fehler der betrachteten Näherung $\hat{I}_{m'}\left(N\right)$ ermittelt und mit einem maximal zulässigen Fehler $\tilde{E}_{m'}\left(N\right)$ verglichen. Das Stop-Flag wird im Aufbauschritt m' des betrachteten Dünn-Gitter-Baumes gesetzt, sobald die Bedingung

$$E_{m'}(N) \quad < \quad \tilde{E}_{m'}(N)$$

erstmals erfüllt ist.

Der tatsächliche Fehler $\left|I(N) - \hat{I}_{m'}(N)\right|$ der Näherung $\hat{I}_{m'}(N), m' > 0$ kann als Differenz zur Näherung der nächstniedrigeren Ordnung abgeschätzt werden:

$$E_{m'}(N) \quad := \quad \left|\hat{I}_{m'}(N) - \hat{I}_{m'-1}(N)\right| \approx \left|I(N) - \hat{I}_{m'}(N)\right|.$$

Aufgrund der Beziehung $\hat{I}_{\tilde{m}}(N) = \Sigma\left(M^*_{\vec{\imath}}(N)\right)_{\vec{\imath}\in\mathcal{I}^{\langle N\rangle}_{\tilde{m}}}$ für beliebiges $\tilde{m} \in \mathbb{N}_0$ gilt:

$$E_{m'}(N) \quad = \quad \left|\sum_{\vec{\imath}\in\tilde{\mathcal{I}}^{\langle N\rangle}_{m'}} M^*_{\vec{\imath}}(N)\right|. \tag{83}$$

In den Knoten werden Integrale unterschiedlicher Dimensionen berechnet, auf die nicht das gleiche Fehlerkriterium angewendet werden kann. Es stellt sich daher die Frage, wie die Berechnung der einzelnen Integrale durch den maximal zulässigen Fehler ε_{rel} gesteuert wird. Eine Lösung ergibt sich, wenn ε_{rel} auf die Anzahl $\left|\tilde{\mathcal{I}}^{\langle W\rangle}_{m'}\right|$ der Summanden des Fehlerschätzers $E_{m'}(W)$ im Wurzelknoten W bezogen wird.

Wir definieren die Hilfsgröße

$$\bar{E}_m(W) \quad := \quad \frac{\varepsilon_{\text{rel}}}{\left|\tilde{\mathcal{I}}^{\langle W\rangle}_m\right|} \cdot \hat{I}_{m-1}(W)$$

mit $\hat{I}_{-1}(W) := 0$. Aus der Extrapolation nach (82) folgt, daß in jedes $M^*_{\vec{\imath}}(N)$ ($\vec{\imath} \neq \vec{0}$) zwei Sohnknotenwerte $M^*_{\vec{\imath}-\vec{e}_k}(N_{j,1})$ und $M_{\vec{\imath}-\vec{e}_k}(N_{j,2})$ eingehen. Daher gilt für die Vererbung von $\bar{E}_m(N)$ eines beliebigen Knotens N an seinen Sohnknoten $N_{j,\psi}$, $j = 1, \ldots, d'$, $\psi = 1, 2$:

$$\bar{E}_{m'-1}(N_{j,\psi}) \quad = \quad \tfrac{1}{2} \cdot \bar{E}_{m'}(N). \tag{84}$$

Infolge dessen wird das Stop-Flag eines Knotens N im Aufbauschritt m' gesetzt, wenn die Bedingung

$$E_{m'}(N) \quad < \quad \left|\tilde{\mathcal{I}}^{\langle N\rangle}_{m'}\right| \cdot \bar{E}_{m'}(N)$$

erfüllt ist, womit die Adaptivitätsstrategie spezifiziert ist.

Gleichung (84) erzwingt, daß der maximal zulässige Fehler ε_{rel} über alle Integralnäherungen im Baum gleichmäßig verteilt wird. Es kann aber vorkommen, daß das Erzielen der gleichen Genauigkeit auf verschiedenen Gebieten mit unterschiedlich hohen Kosten, gemessen in Funktionsauswertungen, verbunden ist. Diesem Aspekt wird in der folgenden Adaptivitätsstrategie Rechnung getragen.

4.6.2. Die kostenorientierte Adaptivitätssteuerung. Diese Strategie verfolgt das Ziel, nur solche Mengen von Gitterpunkten zur Näherung des Gesamtintegrals heranzuziehen, die einen vorgegebenen Mindestbeitrag δ_{rel} zur Gesamtnäherung pro verwendetem Gitterpunkt leisten. Diese Strategie kann auch als kostenorientiert bezeichnet werden, da sie den ,,Nutzen" (Beitrag zur Gesamtnäherung) in ein Verhältnis zum ,,Aufwand" (gemessen in der Anzahl der benötigten Integrandenstützstellen) setzt. Gitterpunktmengen, in denen dieses Verhältnis nicht gegeben ist, sind als ,,zu teuer" zu betrachten und sollen nicht in die Berechnung eingehen.

Wir bezeichnen mit $n_{m'}(N)$ die Anzahl der im Aufbauschritt m' des Dünn-Gitter-Teilbaumes mit Wurzel N ausgewerteten Integrandenstützstellen. Durch die Auswertung des Integranden an diesen $n_{m'}(N)$ Stützstellen wird der Beitrag

$$\left|\hat{I}_{m'}(N) - \hat{I}_{m'-1}(N)\right| = \left|\sum_{\vec{\imath}\in\tilde{\mathcal{I}}_{m'}^{\langle N\rangle}} M_{\vec{\imath}}^{*}(N)\right|$$

zur Näherung von $I(N)$ geleistet. Wir definieren daher das Nutzen-Kosten-Verhältnis im Knoten N des m'-ten Aufbauschrittes:

$$\delta_{m'}(N) = \frac{\left|\sum_{\vec{\imath}\in\tilde{\mathcal{I}}_{m'}^{\langle N\rangle}} M_{\vec{\imath}}^{*}(N)\right|}{n_{m'}(N)}.$$

$n_{m'}(N)$ bezeichnet hierbei die *tatsächliche* Anzahl der benötigten Punkte. Es muß darin also auch berücksichtigt werden, daß in gesperrten Teilbäumen keine Knoten erzeugt wurden. $n_{m'}(N)$ errechnet sich somit rekursiv aus seinen Sohnknoten $N_{j,\psi}, j = 1,\ldots,d', \psi = 1,2$ gemäß:

$$n_{m'}(N) = \begin{cases} 0 & \text{falls } N \text{ gesperrt ist} \\ 1 & \text{falls } m' = 0 \\ \sum\limits_{j=1}^{d'} \left(n_{m'-1}(N_{j,1}) + n_{m'-1}(N_{j,2})\right) & \text{sonst} \end{cases} .$$

Das Stop-Flag des Knotens N wird im Schritt m' gesetzt, wenn folgende Behauptung erstmals wahr ist:

$$\delta_{m'}(N) < \delta_{\mathrm{rel}} \cdot \hat{I}_{m-1}(W),$$

wobei m den Aufbauschritt des Gesamtbaumes bezeichnet.

Sowohl die fehlerorientierte wie auch die kostenorientierte Adaptivitätsstrategie gehen von der Annahme aus, daß die Folgen $\{\varepsilon_0(N), \varepsilon_1(N), \varepsilon_2(N), \ldots\}$ beziehungsweise $\{\delta_0(N), \delta_1(N), \delta_2(N), \ldots\}$ deren Glieder sie schrittweise errechnen, streng monoton nach Null konvergieren. Diese Voraussetzungen an die Folgen müssen nicht gegeben sein. Vielmehr kann es vorkommen, daß ein Folgenglied ,,zufälligerweise" unterhalb der vorgegebenen Schranke liegt. In diesem Fall würde das Verfahren in die Irre geführt werden. Solche Effekte lassen sich ohne weitere Kenntnis über die Eigenschaften des

Integranden nicht vermeiden. Die Robustheit des Verfahrens kann aber gesteigert werden, wenn in das Adaptivitätskriterium nicht nur das gerade errechnete Folgenglied eingeht, sondern auch alle bisher errechneten Glieder einbezogen werden. So läßt sich dann durch Vergleich feststellen, ob das gerade berechnete Glied ein „Ausreißer“ ist oder ob es sich in die durch die Vorgänger angedeutete Asymptotik einreiht. Das Schema $M^*_{\vec{\imath}}(N)$, $\vec{\imath} \in \mathcal{I}^{\langle N \rangle}_{m'}$ ermöglicht derartige Maßnahmen zur Steigerung der Robustheit. Sie sollen aber hier nicht weiter verfolgt werden.

In Abbildung 32 ist das Verfahren in C-artiger Syntax zusammengefaßt. Die Routine `main` stößt die Baumaufbauschritte an, bis im Wurzelknoten das Stop-Flag gesetzt ist. Das Ergebnis ergibt sich als Summe der Rückwärtsdifferenzen im Wurzelknoten. Die Routine `traverse` durchläuft rekursiv die Baumdatenstruktur. Sie sorgt für das breadth-first-Wachstum des Baumes, führt die Extrapolation durch und entscheidet über das Setzen des Stop-Flags.

4.7. Richtungsabhängige Begrenzung der Maximalverfeinerung. Unterschiedlich ausgeprägte Glattheit des Integranden in den Koordinatenrichtungen drückt sich im Schema $(M^*_{\vec{\imath}}(N))_{\vec{\imath} \in \mathcal{I}^{\langle N \rangle}_{m'}}$ darin aus, daß die Einträge in Richtungen großer Glattheit stärker nach Null konvergieren als in Richtungen, in denen die Glattheit schwächer ausgeprägt ist. Im folgenden wollen wir eine Variante von `aqho` vorstellen, die diesem Umstand Rechnung trägt und die Gitterverfeinerungen unter Berücksichtigung der Glattheitseigenschaften des Integranden richtungsabhängig beschränkt.

Obwohl es anhand der in `aqho` verwendeten Datenstruktur möglich ist, richtungsabhängige Glattheit zu erkennen, kann dies durch das bisher vorgestellte Verfahren aus zwei Gründen noch nicht umgesetzt werden. Zum einen erlauben Stop-Flags keine richtungsabhängige Begrenzung der Maximalverfeinerung, zum anderen beruhen die Adaptivitätssteuerungen auf dem Fehlerschätzer

$$E_{m'}(N) \;=\; \left| \sum_{\vec{\imath} \in \tilde{\mathcal{I}}^{\langle N \rangle}_{m'}} M^*_{\vec{\imath}}(N) \right|, \tag{85}$$

durch den die Glattheit nicht richtungsabhängig aufgeschlüsselt wird. Die vorzustellende Variante von `aqho` setzt an diesen beiden Punkten an.

Der erste Problempunkt, die richtungsabhängige Begrenzung der Maximalverfeinerung, wird gelöst, indem für jeden Knoten N ein Begrenzungsmultiindex

$$\vec{L}_{m'}(N) \;=\; (L_{m',1}(N), L_{m',2}(N), \ldots, L_{m',d}(N))$$

vorgesehen wird, der die Menge der neu zu berechnenden Werte in N des nächsten Aufbauschritts von

$$M^*_{\vec{\imath}}(N) \quad \text{mit} \quad \vec{\imath} \in \tilde{\mathcal{I}}^{\langle N \rangle}_{m'+1}$$

mindestens auf

$$M^*_{\vec{\imath}}(N) \quad \text{mit} \quad \vec{\imath} \in \tilde{\mathcal{I}}^{\langle N \rangle}_{m'+1} \quad \wedge \quad \vec{\imath} \le \vec{L}_{m'}(N) \tag{86}$$

```
main()
{
  erzeuge Wurzelknoten W;
  do{
    traverse(W);
  }while(Stop-Flag in W nicht gesetzt);
  Resultat = Σ (M*_ı⃗ (W))_{ı⃗∈I_m^(d)};
}

traverse(N)
{ /* d' und m' sind lokal gegeben */
  if( Stop-Flag in N gesetzt ){
    ∀ı⃗ ∈ Ĩ_{m'}^⟨N⟩:     M*_ı⃗ (N) = 0;
    return;
  }
  if( m' == 1) { /* N ist ein Blattknoten */
    lege F_N fest;
    ∀j = 1,...,d' und  ∀ψ = 1,2:
      erzeuge Sohnknoten N_{j,ψ};
  }
  else{/* durchlaufe rekursiv alle Unterbäume */
    ∀j = 1,...,d' und ∀ψ = 1,2:
      traverse(N_{j,ψ});
  }
  ∀ı⃗ ∈ Ĩ_{m'}^⟨N⟩: berechne M*_ı⃗ (N) nach (82);
  entscheide, ob in N das Stop-Flag zu setzen ist;
}
```

ABB. 32. *Adaptive mehrdimensionale Quadratur höherer Ordnung.*

einschränkt. Die Komponenten von $\vec{L}_{m'}(N)$ können die folgenden Werte annehmen:

$$L_{m',\xi} \;\in\; \{\mathbb{N}_0 \cup \{\square\}\} \quad \text{für} \;\; \xi = 1, \ldots, d,$$

wobei im Zusammenhang mit dem Symbol $\square$, das das Vorhandensein keiner Begrenzung angibt, folgende Rechenregeln gelten:

$$\begin{aligned} \square - 1 &= \square \\ \square &> x \quad \forall x \in \mathbb{N}_0 \\ \min(x, \square) &= x \quad \forall x \in \{\mathbb{N}_0 \cup \{\square\}\}. \end{aligned}$$

Den zweiten Problempunkt, die Erfassung von richtungsabhängiger Glattheit, lösen wir, indem wir die Werte $M^*_{\vec{\imath}}\,(N)$ *einzeln* mit einer Testfunktion $B_{\vec{\imath}}\,(N) \in \{\mathtt{T},\mathtt{F}\}$ ($\mathtt{T} \mathrel{\hat{=}}$ true, $\mathtt{F} \mathrel{\hat{=}}$ false) untersuchen. Für die Definition von $B_{\vec{\imath}}\,(N)$ unterscheiden wir wieder zwischen fehler- und kostenorientierter Adaptivität.

Im Falle der fehlerorientierten Adaptivitätssteuerung ist eine vernünftige Definition von $B_{\vec{\imath}}\,(N)$ einfach, denn mit $\bar{E}_{m'}\,(N)$, das mit dem m'-ten Aufbauschritt eines Knotens N zur Verfügung steht, ist der Maximalbetrag für ein $M^*_{\vec{\imath}}\,(N)$ mit $|\vec{\imath}| = m'$ schon gegeben. Wir definieren also:

$$B_{\vec{\imath}}\,(N) = \mathtt{T} \quad \Leftrightarrow \quad |M^*_{\vec{\imath}}\,(N)| < \bar{E}_{|\vec{\imath}|}\,(N)\,.$$

Bei kostenorientierter Adaptivitätssteuerung ist in jedem Knoten der minimale Verbesserungsbeitrag δ_{rel} pro Integrandenauswertung gegeben. Die Funktion $B_{\vec{\imath}}(N)$ läßt sich für diesen Fall wie folgt festlegen:

$$(87) \qquad B_{\vec{\imath}}\,(N) = \mathtt{T} \quad \Leftrightarrow \quad \begin{cases} \frac{\left|M^*_{\vec{\imath}}(N)\right|}{n_{\vec{\imath}}(N)} < \delta_{\text{rel}} \cdot I_{|\vec{\imath}|-1}\,(W) & \text{falls} \;\; n_{\vec{\imath}}\,(N) \neq 0 \\ \qquad\quad \mathtt{T} & \text{falls} \;\; n_{\vec{\imath}}\,(N) = 0 \end{cases},$$

dabei bezeichnet $n_{\vec{\imath}}\,(N)$ die Anzahl der Integrandenauswertungen, die im aktuellen Aufbauschritt in die Berechnung von $M^*_{\vec{\imath}}\,(N)$ eingegangen sind. Sie liegt je nach Verteilung der Stop-Flags in den Unterbäumen zwischen 0 und $2^{|\vec{\imath}|}$. Der Fall $n_{\vec{\imath}}\,(N) = 0$ ist in (87) gesondert behandelt.

In Abbildung 33 ist eine entsprechend modifizierte Variante des Verfahrens `aqho` dargestellt. Die Funktion `traverse` ist hier um den Parameter $\vec{g}$ erweitert worden, der den aktuell gültigen Begrenzungsmultiindex an die Sohnknoten übergibt und dafür sorgt, daß nur die notwendigen Gitterpunkte ausgewertet werden. Aus Sicht eines beliebigen Knotens N faßt $\vec{g}$ die Begrenzungsmultiindizes der Vaterknoten zusammen. Knoten N kann den Begrenzungsmultiindex $\vec{g}$ aufgrund seiner lokalen Beschränkung $\vec{L}_{m'-1}\,(N)$ weiter einschränken und an seine Sohnknoten weiterreichen. Bei der Vererbung von $\vec{g}$ an die Sohnknoten ist eine Indexverschiebung zu berücksichtigen, die aus der Extrapolation nach (82) folgt. Der Wert $M^*_{\vec{\imath}}\,(N)$ ergibt sich nämlich aus den Sohnknotenwerten $M^*_{\vec{\imath}'}\,(N_{j,\psi})\,, \psi \in \{1,2\}$ mit $\vec{\imath}' = \vec{\imath} - \vec{e}_{k_j}$.

Für einen Blattknoten N, also $m' = 0$, wird der Begrenzungsmultiindex mit

$$\vec{L}_0\,(N) = (\Box, \Box, \ldots, \Box)$$

initialisiert. Die Berechnung von $\vec{L}_{m'}\,(N)$ für $m' > 0$ erfolgt nach der Extrapolation unter Verwendung der Testfunktion $B_{\vec{\imath}}\,(N)$:

$$(88) \qquad \begin{aligned} & \mathcal{M} := \left\{ \vec{\imath} \;\middle|\; \vec{\imath} \in \tilde{\mathcal{I}}^{\langle N \rangle}_{m'} \;\wedge\; \vec{\imath} \leq \vec{g} \;\wedge\; B_{\vec{\imath}}\,(N) = \mathtt{F} \right\} \\ & \text{falls} \quad \mathcal{M} \neq \emptyset : \\ & \qquad\qquad \vec{\jmath} := \max\nolimits_{\text{comp}}\,(\mathcal{M}) \\ & \qquad\qquad L_{m',\xi}\,(N) = \begin{cases} j_\xi + 1 & \text{falls} \;\; \min\,(m', g_\xi) > j_\xi \\ L_{m'-1,\xi} & \text{sonst} \end{cases}, \end{aligned}$$

```
main()
{
  g⃗ = (□, □, ..., □);
  erzeuge Wurzelknoten W;
  L⃗_0(W) = (□, □, ..., □);
  do{
    traverse(W, g⃗);
  }while(Stop-Flag in W nicht gesetzt);
  Resultat = Σ (M*_ı⃗ (W))_{ı⃗ ∈ I_m^(d)};
}

traverse(N, g⃗)
{ /* d' ,m' und L⃗_m'(N) sind lokal gegeben */
  g⃗ = min_comp(g⃗, L_{m'-1}(N));
  if( Stop-Flag gesetzt || { ı⃗ | ı⃗ ∈ Ĩ_m'^⟨N⟩ ∧ ı⃗ ≤ g⃗ } == ∅){
    Stop-Flag setzen;
    ∀ı⃗ ∈ Ĩ_m'^⟨N⟩ :   M*_ı⃗ (N) = 0;
    return;
  }
  if( m' == 1 ) { /* N ist ein Blattknoten */
    lege F_N fest;
    ∀j = 1, ..., d' und ∀ψ = 1, 2:
      if(g_{k_j} > 0){
        erzeuge Sohnknoten N_{j,ψ};
        L⃗_0(N_{j,ψ}) = {□, □, ..., □};
      }
  }
  else { /* durchlaufe rekursiv alle Unterbäume */
    ∀j = 1, ..., d' und ∀ψ = 1, 2 :
      if(g_{k_j} > 0) traverse(N_{j,ψ}, g⃗ − e⃗_{k_j});
  }
  ∀ı⃗ ∈ Ĩ_m'^⟨N⟩  ∧  ı⃗ ≤ g⃗ : berechne M*_ı⃗ (N) nach (82);
  berechne L⃗_m'(N) nach (88);
  entscheide, ob in N das Stop-Flag zu setzen ist;
}
```

ABB. 33. *Adaptive mehrdimensionale Quadratur höherer Ordnung mit richtungsabhängiger Begrenzung der Maximalverfeinerung.*

i_2 \ i_1	0	1	2	3	4	5	6	7	8
0	F	F	F	F	F	F	F	F	T
1	F	F	F	F	F	F	F	F	T
2	F	F	F	F	F	F	F	F	T
3	F	F	F	F	T	T	T	T	T
4	F	F	F	T	T	T	T	T	T
5	T	T	T	T	T	T	T	T	T

ABB. 34. *Beispiel für* $B_{(i_1,i_2)}(W) : \mathbb{N}_0^2 \to \{\mathtt{T},\mathtt{F}\}$.

wobei $\max_{\text{comp}}(\mathcal{M})$ das komponentenweise Maximum über alle Multiindizes der Menge $\mathcal{M}$ bildet. Gilt $\mathcal{M} = \emptyset$, so wird für N das Stop-Flag gesetzt.

Die Arbeitsweise des Verfahrens soll anhand eines zweidimensionalen Beispiels verdeutlicht werden. Wir betrachten hierfür nur den Wurzelknoten W des aufzubauenden Dünn-Gitter-Baumes und setzen voraus, daß die zur Adaptivitätssteuerung verwendete Funktion $B_{(i_1,i_2)}(W)$ für verschiedene Multiindizes (i_1, i_2) die in der Tabelle von Abbildung 34 angegebenen Werte `T` und `F` annimmt. Abbildung 35 veranschaulicht die Schritte $m = 0, \ldots, 10$. Jeder Schritt ist durch das Schema symbolisiert, in dem $B_{(i_1,i_2)}$ den Wert `F` liefert. Ein Buchstabe `F` oder `T` an einer Indexposition (i_1, i_2) zeigt an, daß in diesem Schritt $M^*_{(i_1,i_2)}(W)$ berechnet wurde und daß $B_{(i_1,i_2)}(W)$ den angegebenen Wert lieferte. Punkte markieren die Indexpositionen, die in vorangegangenen Aufbauschritten behandelt wurden. Die Begrenzungsmultiindizes $L_{m,1}(W)$ und $L_{m,2}(W)$ sind durch horizontale beziehungsweise vertikale Pfeile angedeutet. Fehlt ein Pfeil, so liegt noch keine Begrenzung vor. In den Schritten $m = 0, \ldots, 5$ wird das Schema hyperebenenweise aufgebaut, wie wir es bereits kennen. In Schritt $m = 5$ gilt: $B_{(0,5)}(W) = \mathtt{T}$. Infolge dessen ergibt sich nach (88) die Begrenzung $\vec{L}_5(W) = (\square, 5)$, die in der Abbildung durch einen Pfeil bei $i_2 = 5$ symbolisiert wird. Im nächsten Schritt $(m = 6)$ wird $M^*_{(0,6)}(W)$ aufgrund dieser Begrenzung nicht berechnet. Da $\vec{L}_5(W)$ durch $\vec{g}$ in die Sohnknoten weitergeleitet wird, erfolgt auch keine Auswertung des Integranden an den zugehörigen Gitterpunkten. In den folgenden Schritten werden die Begrenzungen weiter verschärft. Die Berechnung schließt mit Schritt $m = 10$ ab, da die Begrenzungen im weiteren Verlauf keine Auswertungen zulassen.

4.8. Kosten der Datenstruktur. Im folgenden soll der Speicheraufwand für die rechnerinterne Repräsentation des Dünn-Gitter-Baumes abgeschätzt werden. Hier soll insbesondere die Frage geklärt werden, ob die Speicherkomplexität der Baumdatenstruktur durch die Simplizes, die den Knoten zugeordnet sind, beziehungsweise durch die Zeiger auf die Sohnknoten in der Ordnung erhöht werden. Wir betrachten zuerst

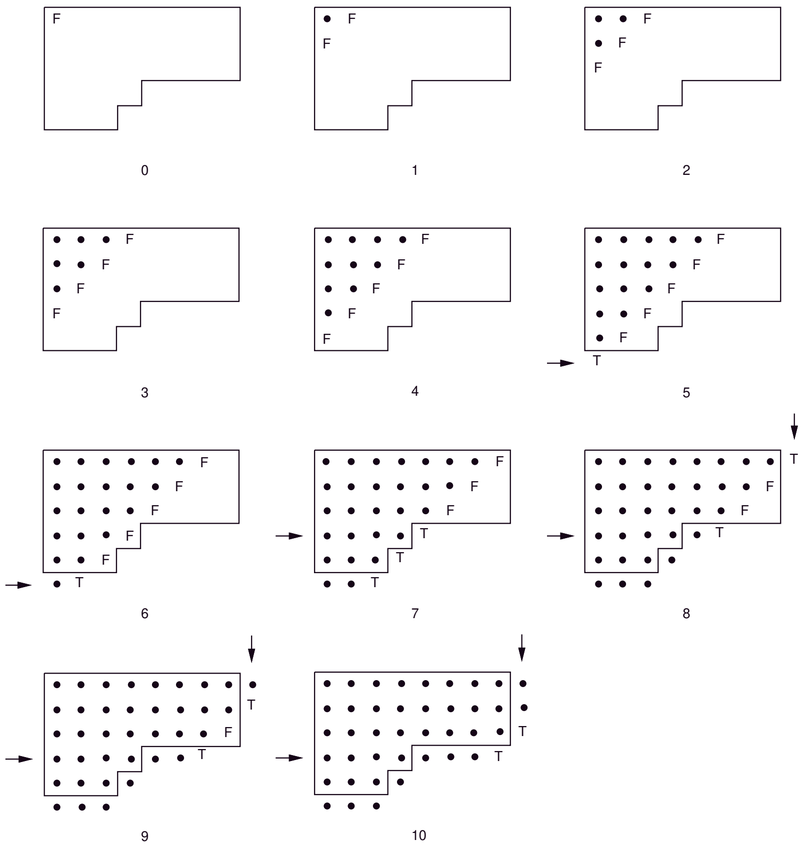

ABB. 35. *Arbeitsweise von* `aqho` *bei richtungsabhängiger Begrenzung der Maximalverfeinerung.*

den nicht-adaptiven Fall, das heißt, die den Knoten zugehörigen Punkte ergeben ein reguläres dünnes Gitter $G_m^{(d)}\left(\vec{a}\,,\vec{b}\,\right)$.

Wir untersuchen den Aufwand getrennt für die Knoten, die Zeiger auf die Sohnknoten und die Simplizes und bezeichnen die Gesamtkosten jeweils mit $C_{K_m^{(d)}}, C_{Z_m^{(d)}}$ und $C_{S_m^{(d)}}$. Die von *einem* Knoten verursachten Kosten bezeichnen wir entsprechend mit $c_{K_m^{(d)}}, c_{Z_m^{(d)}}$ und $c_{S_m^{(d)}}$. Wir legen fest:

$$\begin{aligned}
c_{K_m^{(d)}} &:= 1 \\
c_{Z_m^{(d)}} &:= d \\
c_{S_m^{(d)}} &:= \left|\mathcal{I}_m^{(d)}\right| = \binom{d+m}{d}.
\end{aligned}$$

$C_{X_m^{(d)}}$ mit $X \in \{K, Z, S\}$ ergibt sich nach (67) und (68) zu

$$C_{X_0^{(d)}} = c_{X_0^{(d)}} \tag{89}$$

$$C_{X_m^{(d)}} = c_{X_m^{(d)}} + 2 \cdot \sum_{j=1}^{d} C_{X_{m-1}^{(j)}} \quad \text{für} \quad m > 0. \tag{90}$$

Wir wollen nun $C_{X_m}^{(d)}, X \in \{K, Z, S\}$ abschätzen, wobei wir voraussetzen, daß m groß und d beliebig aber fest gewählt wird.

Besitzen zwei algebraische Ausdrücke $\mathcal{A}$ und $\mathcal{B}$ bezüglich m dasselbe führende Glied, so schreiben wir hierfür

$$\mathcal{A} \doteq \mathcal{B}.$$

Zunächst zeigen wir:

LEMMA 4.2. *Es seien $d > 0$, $m \geq 0$, dann gilt mit (89) und (90):*

$$C_{X_m}^{(d)} = \sum_{i=0}^{m} 2^i \cdot c_{X_{m-i}}^{(d)} + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d-1} C_{X_{m-i}}^{(j)} \right) \tag{91}$$

Beweis. Die Behauptung ergibt sich durch Anwenden von (90) und (89) auf $C_{X_m}^{(d)}, C_{X_{m-1}}^{(d)}, \ldots, C_{X_0}^{(d)}$ wie folgt:

$$\begin{aligned}
C_{X_m}^{(d)} &\overset{(90)}{=} c_{X_m}^{(d)} + 2 \cdot \sum_{j=1}^{d} C_{X_{m-1}}^{(j)} \\
&= c_{X_m}^{(d)} + 2 \cdot C_{X_{m-1}}^{(d)} + 2 \cdot \sum_{j=1}^{d-1} C_{X_{m-1}}^{(j)} \\
&\overset{(90)}{=} c_{X_m}^{(d)} + 2 \cdot c_{X_{m-1}}^{(d)} + 4 \cdot \sum_{j=1}^{d} C_{X_{m-2}}^{(j)} + 2 \cdot \sum_{j=1}^{d-1} C_{X_{m-1}}^{(d)} \\
&\vdots \\
&= \sum_{i=0}^{m} 2^i \cdot c_{X_{m-i}}^{(d)} + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d-1} C_{X_{m-i}}^{(j)} \right).
\end{aligned}$$

□

Für den eigentlichen Beweis benötigen wir folgende Beziehungen:

LEMMA 4.3. *Es seien $d > 0$ und $m \geq 0$, dann gilt:*

$$\sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} \binom{m-i+j-1}{j-1} \cdot 2^{m-i+1} \right) \doteq \binom{m+d}{d} \cdot 2^{m+1} \tag{92}$$

$$\sum_{i=0}^{m} 2^i \cdot \binom{d+1+m-i}{d+1} \doteq 2^{d+2} \cdot 2^m. \tag{93}$$

Beweis. Mit Hilfe des Additionstheorems für Binomialkoeffizienten

$$\binom{n+k+1}{k} = \sum_{j=0}^{k} \binom{m+j}{j} \tag{94}$$

ergibt sich:

$$\begin{aligned}
\sum_{i=1}^{m}\left(2^i\cdot\sum_{j=1}^{d}\binom{m-i+j-1}{j-1}\cdot 2^{m-i+1}\right) &= 2^{m+1}\cdot\sum_{i=1}^{m}\sum_{j=1}^{d}\binom{m-i+j-1}{j-1}\\
&\overset{(94)}{=} 2^{m+1}\cdot\sum_{i=1}^{m}\binom{m-i+d}{d-1}\\
&= 2^{m+1}\cdot\sum_{i=1}^{m}\binom{i-1+d}{d-1}\\
&= 2^{m+1}\cdot\left(\sum_{i=0}^{m}\binom{i-1+d}{i}-1\right)\\
&\overset{(94)}{=} 2^{m+1}\cdot\left(\binom{d-1+m+1}{m}-1\right)\\
&\doteq \binom{m+d}{d}\cdot 2^{m+1},
\end{aligned}$$

womit (92) gezeigt ist.

Wir zeigen die Gültigkeit von (93) durch Induktion über d.

Induktionsanfang ($d=1$):

$$\sum_{i=0}^{m}2^i\cdot\binom{2+m-i}{2} = 8\cdot 2^m-7-\tfrac{7}{2}\cdot m-\tfrac{1}{2}\cdot m^2\doteq 2^3\cdot 2^m.$$

Induktionsschritt ($d\to d+1$): Es gilt:

$$\begin{aligned}
\sum_{i=0}^{m}2^i\cdot\binom{d+2+m-i}{d+2} &= \sum_{i=0}^{m}\left(2^i\cdot\binom{d+2+m-i}{m-i}\right)\\
&\overset{(94)}{=} \sum_{i=0}^{m}\left(2^i\cdot\sum_{j=0}^{m-i}\binom{d+1+j}{d+1}\right)\\
&= \sum_{i=0}^{m}\left(2^i\cdot\sum_{j=0}^{m-i}\binom{d+1+(m-i)-j}{d+1}\right)\\
&= \sum_{j=0}^{m}\left(\sum_{i=0}^{m-j}2^i\cdot\binom{d+1+(m-j)-i}{d+1}\right)\\
&\overset{\text{(Ind.vor.)}}{\doteq} \sum_{j=0}^{m}2^{d+2}\cdot 2^{m-j}\\
&\doteq 2^{(d+1)+2}\cdot 2^m.
\end{aligned}$$

□

Satz 4.4. *Es seien $d > 0$ und $m \geq 0$ sowie*

$$c_{K_m^{(d)}} = 1 \tag{95}$$

$$c_{Z_m^{(d)}} = d \tag{96}$$

$$c_{S_m^{(d)}} = \binom{d+m}{d}. \tag{97}$$

Mit (89) und (90) gilt:

$$C_{K_m^{(d)}} \doteq \binom{m+d-1}{d-1} \cdot 2^{m+1} \tag{98}$$

$$C_{Z_m^{(d)}} \doteq \binom{m+d-1}{d-1} \cdot 2^{m+1} \tag{99}$$

$$C_{S_m^{(d)}} \doteq 2 \cdot \binom{m+d-1}{d-1} \cdot 2^{m+1}. \tag{100}$$

Beweis. Wir zeigen die Gültigkeit aller Behauptungen durch Induktion über d. Behauptung (98) ergibt sich zu:
Induktionsanfang ($d = 1$):

$$\begin{aligned} C_{K_m^{(1)}} &\overset{(91)}{=} \sum_{i=0}^{m} 2^i \cdot c_{K_{m-i}^{(1)}} \\ &\overset{(95)}{=} \sum_{i=0}^{m} 2^i \\ &= 2^{m+1} - 1 \\ &\doteq \binom{m+1-1}{1-1} \cdot 2^{m+1}. \end{aligned}$$

Induktionsschritt ($d \to d+1$):

$$\begin{aligned} C_{K_m^{(d+1)}} &\overset{(91)}{=} \sum_{i=0}^{m} 2^i \cdot c_{K_{m-i}^{(d+1)}} + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} C_{K_{m-i}^{(j)}}\right) \\ &\overset{(95),(\text{Ind.vor.})}{\doteq} \sum_{i=0}^{m} 2^i + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} \binom{m-i+j-1}{j-1} \cdot 2^{m-i+1}\right) \\ &\overset{(92)}{\doteq} 2^{m+1} + \binom{m+d}{d} \cdot 2^{m+1} \\ &\doteq \binom{m+d}{d} \cdot 2^{m+1}. \end{aligned}$$

Für Behauptung (99) ist der Induktionsanfang identisch mit dem für (98).
Induktionsschritt ($d \to d+1$):

$$C_{Z_m^{(d+1)}} \overset{(91)}{=} \sum_{i=0}^{m} 2^i \cdot c_{Z_{m-i}^{(d+1)}} + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} C_{Z_{m-i}^{(j)}}\right)$$

$$
\begin{aligned}
&\overset{(96),(\text{Ind.vor.})}{\doteq} && \sum_{i=0}^{m} 2^i \cdot (d+1) + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} \binom{m-i+j-1}{j-1} \cdot 2^{m-i+1} \right) \\
&\overset{(92)}{\doteq} && (d+1) \cdot 2^{m+1} + \binom{m+d}{d} \cdot 2^{m+1} \\
&\doteq && \binom{m+d}{d} \cdot 2^{m+1}.
\end{aligned}
$$

Behauptung (100) ergibt sich durch:
Induktionsanfang ($d = 1$):

$$
\begin{aligned}
C_{S_m^{(1)}} &\overset{(91)}{=} \sum_{i=0}^{m} 2^i \cdot c_{S_{m-i}^{(1)}} \\
&\overset{(97)}{=} \sum_{i=0}^{m} 2^i \cdot \binom{1+m-i}{1} \\
&= \sum_{i=0}^{m} 2^i \cdot (1+m-i) \\
&= 4 \cdot 2^m - 3 - m \\
&\doteq 2 \cdot \binom{m+1-1}{1-1} \cdot 2^{m+1}.
\end{aligned}
$$

Induktionsschritt ($d \to d+1$):

$$
\begin{aligned}
C_{S_m^{(d+1)}} &\overset{(91)}{=} && \sum_{i=0}^{m} 2^i \cdot c_{S_{m-i}^{(d+1)}} + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} C_{S_{m-i}^{(j)}} \right) \\
&\overset{(97),(\text{Ind.vor.})}{\doteq} && \sum_{i=0}^{m} 2^i \cdot \binom{d+1+m-i}{d+1} \\
& && \qquad + \sum_{i=1}^{m} \left(2^i \cdot \sum_{j=1}^{d} 2 \cdot \binom{m-i+j-1}{j-1} \cdot 2^{m-i+1} \right) \\
&\overset{(92)(93)}{\doteq} && 2^{d+2} \cdot 2^m + 2 \cdot \binom{m+d}{d} \cdot 2^{m+1} \\
&\doteq && 2 \cdot \binom{m+d}{d} \cdot 2^{m+1}.
\end{aligned}
$$

□

Das Ergebnis von Satz 4.4 ist sehr erstaunlich. Es besagt nämlich, daß bei hinreichend großem m je Knoten durchschnittlich zwei Simplexeinträge und ein Zeigerpaar für Sohnknoten benötigt werden. Es wäre eher zu erwarten gewesen, daß die Kosten der Gesamtdatenstruktur durch den Speicheraufwand für die Simplexelemente bestimmt werden, da deren Zahl polynomial mit der Tiefe m des Baumes steigt. Es überwiegt hier aber das exponentielle Ansteigen der Knotenanzahl mit m.

Bedenkt man, daß in jedem Knotenelement mindestens eine Auswertung des Integranden gespeichert werden muß, so folgt aus Satz 4.4, daß die Komplexität eines (d, m)-Dünn-Gitter-Baumes bei festem d durch *keine* Optimierungsmaßnahme in der Ordnung reduziert werden kann.

Für einen beliebigen Dünn-Gitter-Baum definieren wir die Kostenverhältnisse ϱ_Z und ϱ_S wie folgt:

$$\begin{aligned} \varrho_Z &:= \frac{\text{Aufwand für Zeiger}}{\text{Aufwand für Knoten}} \\ \varrho_S &:= \frac{\text{Aufwand für Simplizes}}{\text{Aufwand für Knoten}}. \end{aligned}$$

Speziell für den nicht-adaptiven Fall ergibt sich:

$$\begin{aligned} \varrho_{Z_m^{(d)}} &= \frac{C_{Z_m^{(d)}}}{C_{K_m^{(d)}}} \\ \varrho_{S_m^{(d)}} &= \frac{C_{S_m^{(d)}}}{C_{K_m^{(d)}}}. \end{aligned}$$

Wir wollen durch Nachrechnen über (89) und (90) untersuchen, wie schnell die aus Satz 4.4 folgenden Asymptotiken

$$\begin{aligned} \varrho_{Z_m^{(d)}} &\doteq 1 \\ \text{und}\quad \varrho_{S_m^{(d)}} &\doteq 2 \end{aligned}$$

erreicht werden. Abbildung 36 zeigt hierzu $\varrho_{Z_m^{(d)}}$ (links) und $\varrho_{S_m^{(d)}}$ (rechts) in Abhängigkeit von $d = 1, \ldots, 30$ und $m = 0, \ldots, 30$, wobei für $\varrho_{Z_m^{(d)}}$ auch die Höhenlinien für 2,3,4,5 und 10 eingezeichnet sind. Man sieht, daß $\varrho_{Z_m^{(d)}}$ insbesondere für kleine Dimensionen schnell gegen den Wert 1 konvergiert. Die errechneten Werte für $\varrho_{S_m^{(d)}}$ liegen alle zwischen 1 und 2, wobei der Grenzwert 2 mit steigendem m sehr schnell approximiert wird.

Es soll nun untersucht werden, inwiefern die asymptotischen Ergebnisse auf den adaptiven Fall zutreffen. Dazu betrachten wir die Funktion

$$f(\vec{x}) = \left(1 + \sum_{i=1}^{d} a_i \cdot x_i\right)^{(-d-1)} \quad \text{mit} \quad \vec{x} \in [0,1]^d$$

und $a_i = g_i(p) > 0$ für $i = 1, \ldots, d$, die im Ursprung ein Randmaximum, bedingt durch eine Singularität außerhalb des Integrationsgebiets, besitzt. Da die Funktion in Beispiel 14 nochmals aufgegriffen wird, sei an dieser Stelle nur erwähnt, daß die Ausprägung des Randmaximums durch Erhöhung des Parameters p gesteuert werden kann. `aqho` baut für diesen Integranden einen stark unausgeglichenen Baum auf. In Tabelle 2 sind die Ergebnisse für verschiedene Kombination von $d = 2, \ldots, 6$, $p = 256, \ldots, 32768$ und $\delta_{\text{rel}} = 10^{-9}, \ldots 10^{-15}$ aufgetragen. Die Spalte m gibt den Aufbauschritt an, nach dem das Verfahren terminiert, C_K die Anzahl der Knoten des Baumes und $C_{K_m^{(d)}}$ die

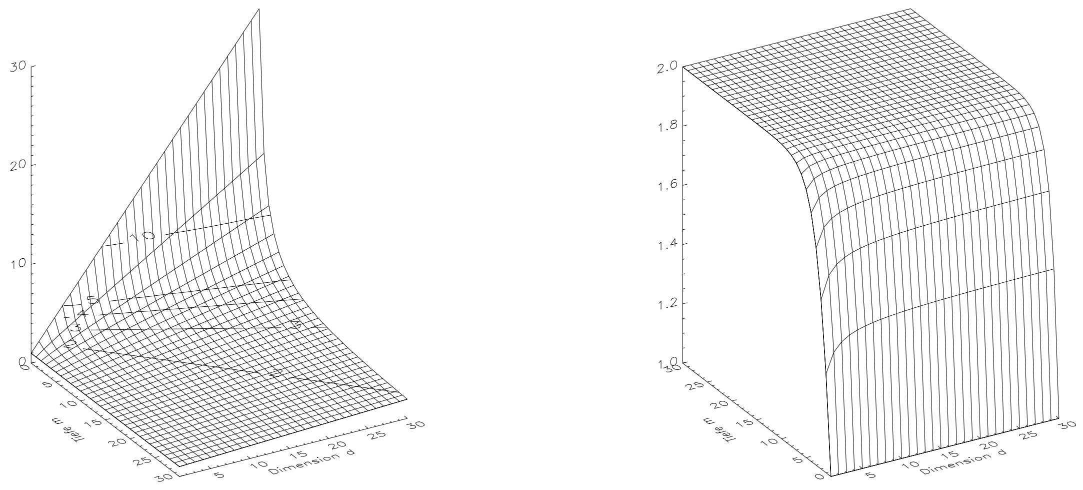

ABB. 36. ${\varrho_Z}_m^{(d)}$ *(links) und* ${\varrho_S}_m^{(d)}$ *(rechts) für* $d = 1..30$ *und* $m = 0..30$.

d	p	δ_{rel}	m	C_K	${C_K}_m^{(d)}$	ϱ_Z	ϱ_S
2	4096	10^{-10}	26	27665	$3.49 \cdot 10^{09}$	1.169	2.116
2	32768	10^{-15}	34	205793	$1.17 \cdot 10^{12}$	1.148	2.039
3	4096	10^{-10}	30	325751	$1.00 \cdot 10^{13}$	1.299	2.143
3	32768	10^{-11}	40	808407	$1.81 \cdot 10^{15}$	1.288	2.172
4	4096	10^{-10}	32	4467417	$5.17 \cdot 10^{13}$	1.401	2.141
5	1024	10^{-09}	23	4558975	$2.55 \cdot 10^{11}$	1.529	2.143
6	256	10^{-10}	14	5256873	$3.00 \cdot 10^{08}$	1.620	2.023

TABELLE 2
Verschiedene Experimente, für die ϱ_Z *und* ϱ_S *ermittelt wurden.*

Anzahl der Knoten, wenn nicht-adaptiv gerechnet werden würde. Das Verhältnis von C_K zu ${C_K}_m^{(d)}$ kann als Maß für die Unausgeglichenheit des Baumes angesehen werden. In den beiden rechten Spalten sind die ermittelten Werte für ϱ_Z und ϱ_S angegeben. Man sieht, daß in allen Beispielen ϱ_Z noch deutlich unter 2 liegt. Trotz der starken Entartung der Bäume liegt der Wert für ϱ_S nicht mehr als 10 Prozent oberhalb des Grenzwertes von 2.

4.9. Numerische Experimente. Bei der Beschreibung von `aqho` haben wir an drei Stellen alternative Realisierungen vorgestellt, denen wir jetzt Bezeichnungen geben wollen. Mit `aqho` bezeichnen wir von nun an die Variante mit fehlerorientierter Adaptivität, zyklischem Tausch der Halbierungsrichtung sowie ohne Verwendung von Begrenzungsmultiindizes. Für die Bezeichnung der übrigen Varianten verwenden wir die vom Betriebssystem UNIX her bekannte Schreibweise mit Optionsschaltern. `aqho` mit der Option `-K` bedeutet kostenorientierte Adaptivität, `-D` richtungsabhängige Wahl der Halbierungsrichtung und `-A` richtungsabhängige Begrenzung der Maximalverfeinerung. Alle Optionen können unabhängig voneinander verwendet werden. Somit stehen acht Varianten des Verfahrens zur Verfügung.

Anhand einiger numerischer Experimente sollen nun die Eigenschaften der `aqho`-Varianten demonstriert sowie ein Vergleich zu den bisher beschriebenen Verfahren `arch_wr` und `qho` gezogen werden.

Wir betrachten den folgenden zweidimensionalen Integranden, der im Punkt $(x_1, x_2) = (0, 0)$ ein Randmaximum besitzt:

Beispiel 8: $$f(x_1, x_2) = (1 + a_1 \cdot x_1 + a_2 \cdot x_2)^{-3} \quad \text{für} \quad (x_1, x_2) \in [0,1]^2,$$

mit $a_1 = 28.822\ldots$ und $a_2 = 121.178\ldots$ Der gesuchte Wert beträgt etwa $1.381 \cdot 10^{-4}$. Wir wenden zunächst das Verfahren `arch_wr` an und wählen $\varepsilon_{\text{abs}} = 5 \cdot 10^{-5}$, was einem relativen Fehler von etwa 0.362 entspricht. Hierfür liefert `arch_wr` bei 4895 Gitterpunkten eine Genauigkeit von 1.49 Dezimalstellen. Man beachte aber, daß aufgrund von Mehrfachauswertungen wesentlich mehr Integrandenauswertungen getätigt werden, nämlich 35839, also etwa das Siebenfache. In Abbildung 37 (links) sind die benötigten Gitterpunkte wiedergegeben.

Wendet man `aqho` mit $\varepsilon_{\text{rel}} = 10^{-3}$ auf obigen Integranden an, so erhöht sich die Genauigkeit bei fast gleichbleibender Anzahl von Stützstellen (4865) auf 3.76 Dezimalstellen. Hier macht sich die höhere Ordnung von `aqho` bemerkbar. Die verwendeten Integrandenstützstellen sind in Abbildung 37 (rechts) dargestellt. Man sieht, daß die Dünn-Gitter-Struktur bei `aqho` wesentlich stärker ausgeprägt ist als bei `arch_wr`. Dies ist wie folgt zu erklären: Durch den breadth-first-Aufbau des Dünn-Gitter-Baumes erzeugt `aqho` in der Anfangsphase ein reguläres dünnes Gitter, dessen Feinheit schrittweise zunimmt. Ist der in einem Teilbaum errechnete Fehler des zugehörigen Integranden aufgrund der erreichten Ordnung hinreichend klein, so wird das Stop-Flag gesetzt. In den folgenden Aufbauschritten wird dieses Teilgebiet ausgespart. Es verbleibt lokal ein reguläres dünnes Gitter. Abbildung 38 gibt die verwendeten Gitterpunkte nochmals wieder, zusätzlich sind aber Gitterpunkte, in denen für das betrachtete zweidimensionale Integral das Stop-Flag aufgrund der fehlerorientierten Adaptivitätssteuerung gesetzt wurde, durch Kreuze markiert. Außerdem ist das zugehörige Teilgebiet durch ein Rechteck umrandet.

`aqho` terminiert nach dem Aufbauschritt 15, das heißt, das gelieferte Resultat ist mit einem Fehler der Ordnung $O(h^{32})$ behaftet.

Im nächsten Beispiel betrachten wir das Verhalten von `aqho` in der Nähe von Singularitäten.

Beispiel 9: $$f(x_1, x_2, x_3) = (1 + a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3)^{-4}$$

für $(x_1, x_2, x_3) \in [0,1]^3$ mit $a_1 = 78.236\ldots$, $a_2 = 12.122\ldots$ und $a_3 = 20.754\ldots$ Wir wenden `qho` an und erhöhen schrittweise die Verfahrensordnung. In Abbildung 39 (links)

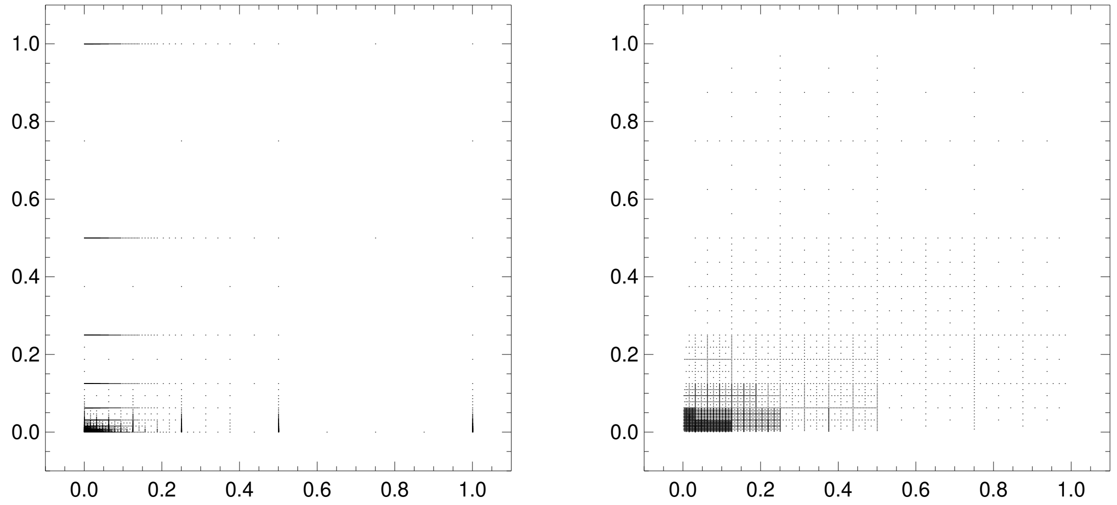

ABB. 37. BEISPIEL 8: *Adaptive Quadraturverfahren im Vergleich:* `arch_wr` *mit* $\varepsilon_{\text{abs}} = 0.5 \cdot 10^{-5}$ *(links) und* `aqho` *mit* $\varepsilon_{\text{rel}} = 10^{-3}$ *(rechts).*

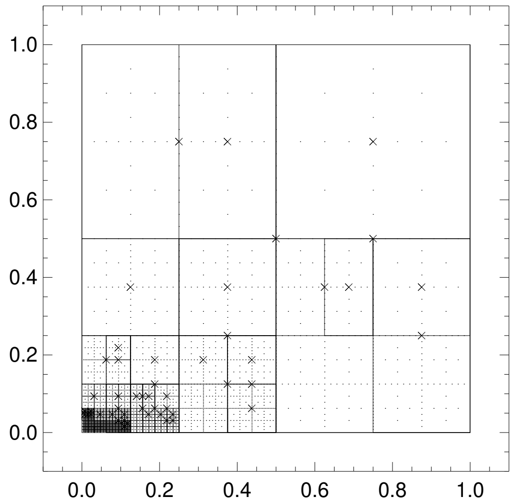

ABB. 38. BEISPIEL 8: `aqho` *mit* $\varepsilon_{\text{rel}} = 10^{-3}$. *Kreuze markieren Gitterpunkte, in denen das Stop-Flag gesetzt ist.*

ist die erzielte Genauigkeit in Abhängigkeit der benötigten Anzahl von Stützstellen aufgetragen. Die einzelnen Meßpunkte sind durch horizontale und vertikale Linien miteinander verbunden. Wie wir es bereits an anderen numerischen Beispielen gesehen haben, nehmen auch hier die Genauigkeitsgewinne mit Erhöhung der Ordnung zu. Im ganzen betrachtet steigt die erzielte Genauigkeit aber sehr schwach an, weil erst bei großen Verfeinerungsstufen das Gebiet nahe der Singularität durch das entsprechende reguläre dünne Gitter hinreichend gut abgetastet wird. Mit Ordnung $O(h^{40})$ erreicht man bei 90177535 Gitterpunkten eine Genauigkeit von 3.28 Dezimalstellen.

Ebenfalls in Abbildung 39 (links) ist das Ergebnis von `aqho` eingezeichnet. Hier wurde der Parameter ε_{rel} schrittweise verkleinert und die erzielte Genauigkeit in Abhängigkeit zur benötigten Anzahl der Gitterpunkte aufgetragen. Die diskreten Meßwerte wurden durch horizontale und vertikale Linien zu einer Kennlinie verbunden. Um gleiche Genauigkeit zu erreichen, benötigt `aqho` wesentlich weniger Gitterpunkte als `qho`. Für die oben erzielte Genauigkeit von 3.28 Dezimalstellen werden hier nur noch 78343 Gitterpunkte benötigt, also etwa drei Zehnerpotenzen weniger. Dies ist auf die Adaptivität des Verfahrens zurückzuführen, das, wie im Beispiel zuvor demonstriert, im Bereich des Randmaximums ein feineres Gitter verwendet als in den glatten Gebieten.

Die Kennlinie für `aqho` weist eine ähnliche Abstufung auf wie die von `qho`. Ein Vergleich der Resultate zeigt, daß die Stufen auf annähernd gleichem Genauigkeitsniveau liegen und daß beide Verfahren dazu die gleichen Ordnungen verwenden. Offensichtlich wird hier durch `aqho` das entsprechende reguläre dünne Gitter soweit ausgedünnt, daß noch keine Gitterpunkte verloren gehen, die einen wesentlichen Beitrag zum Gesamtergebnis leisten.

In Abbildung 39 (rechts) sind die Kennlinien von `aqho` und `aqho -K` (kostenorientierte Adaptivität) aufgetragen. Es fällt sofort auf, daß ab etwa 10^4 Funktionsauswertungen eine deutliche Verbesserung gegenüber der fehlerorientierten Adaptivität eintritt. `aqho -K` erzielt eine um ein bis zwei Dezimalstellen höhere Genauigkeit als `aqho`, beziehungsweise die gleiche Genauigkeit wird etwa mit einem fünftel der Gitterpunkte erreicht. Die in den Linien der linken Abbildung beobachtete Stufung tritt bei `aqho -K` nicht auf. Dieser Effekt soll im nächsten Beispiel untersucht werden.

Auf den Integranden

Beispiel 10: $$f(x_1) \quad = \quad (1 + 10000 \cdot x_1)^{-2} \quad \text{für} \quad x_1 \in [0,1]$$

wenden wir zunächst `aqho` mit $\varepsilon_{\text{rel}} = 10^{-7}$ an, sodann `aqho -K` mit $\delta_{\text{rel}} = 10^{-10}$. In beiden Fällen wird eine Genauigkeit von 11.29 Stellen erzielt. Im ersten werden dazu 1023, im zweiten 927 Stützstellen benötigt. Die von beiden Verfahren erzeugten Baumdatenstrukturen sind in Abbildung 40 dargestellt. Die Bäume weisen im Bereich des linken Randes, in dessen Nähe die Funktion eine Singularität besitzt, eine starke Verfeinerung auf, die eine Tiefe von 19 beziehungsweise 20 Knotenebenen erreicht. Knoten, in denen das Stop-Flag aufgrund des Adaptivitätskriteriums gesetzt wurde, sind in den Abbildungen durch Quadrate gekennzeichnet. Vergleicht man die Unterbäume der Knoten, in denen das Stop-Flag gesetzt ist, so stellt man fest, daß bei kostenorientierter Adaptivitätssteuerung kleinere Teilbäume erzeugt werden als bei fehlerorientierter. `aqho -K` kann aufgrund dessen mehr Zwischenformen von Dünn-Gitter-Bäumen erzeugen, so

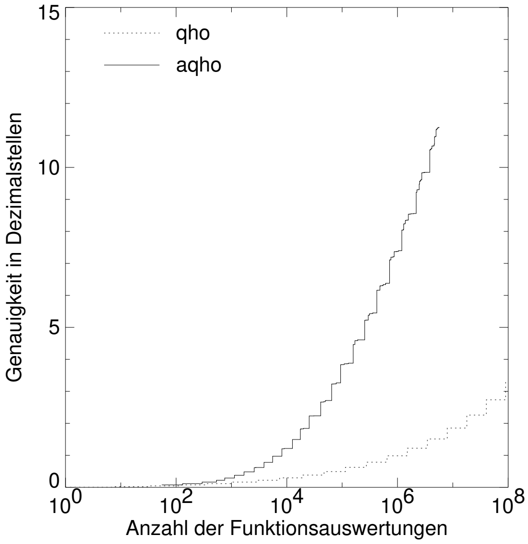

Abb. 39. Beispiel 9: *Vergleich:* `qho` - `aqho` - `aqho -K`.

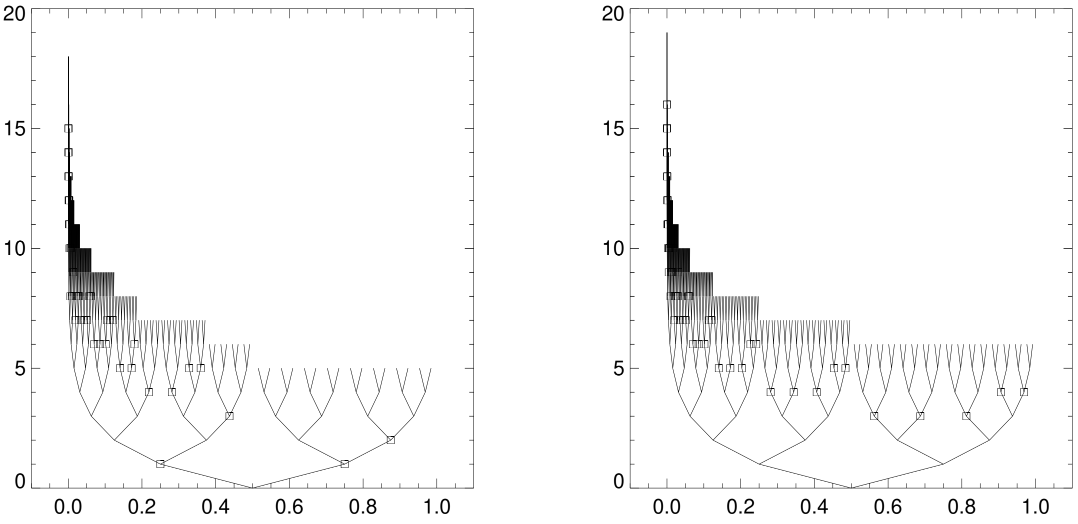

Abb. 40. Beispiel 10: *Baumdatenstruktur, erzeugt von* `aqho` *(links) und* `aqho -K` *(rechts).*

daß die ausgeprägte Abstufung der Kennlinie, wie sie bei `aqho` auftrat, nicht mehr zu beobachten ist.

Im nächsten Beispiel wollen wir untersuchen, wie sich die Varianten von `aqho` bei Integranden mit stark unterschiedlicher Glattheit in den Koordinatenrichtungen verhalten. Wir betrachten den folgenden, in Richtung x_1 sehr glatten Integranden:

Beispiel 11: $$f(x_1, x_2) \;=\; e^{0.05 \cdot x_1 + 99.95 \cdot x_2} \quad \text{für} \quad (x_1, x_2) \in [0,1]^2 .$$

Wir wenden die Varianten mit fehlerorientierter Adaptivität an und wählen hierzu $\varepsilon_{\text{rel}} = 10^{-9}$. Bei 3457 Stützstellen liefert `aqho` eine Genauigkeit von 10.86 Dezimalstellen. Abbildung 41 (links) zeigt das verwendete Gitter. Besonders auffällig ist die starke Verfeinerung im oberen Viertel des Integrationsgebiets. In Richtung x_1 trägt diese Verfeinerung aber nur unwesentlich zur Verbesserung des Gesamtergebnisses bei, da der Integrand in x_1 sehr glatt ist.

Ein besseres Resultat erhält man mit der Variante `aqho -A`, also mit richtungsabhängiger Begrenzung der Maximalverfeinerung. Hier wird annähernd gleiche Genauigkeit bereits mit 1745, also nur der Hälfte der Gitterpunkte, erreicht. Abbildung 41 (rechts) zeigt hierfür das Gitter. Man erkennt sofort die schwächere Verfeinerung in Richtung x_1, die aus dem Setzen des Begrenzungsmultiindizes im Wurzelknoten herrührt.

Ähnlich gute Ergebnisse werden mit `aqho -D` (adaptive Wahl der Halbierungsrichtung) erzielt. Bei nur 1445 Stützstellen beträgt die Genauigkeit sogar 11.05 Dezimalstellen. Das Gitter ist in Abbildung 42 (links) wiedergegeben. Auch hier erkennt man die starke Verfeinerung in Richtung x_1, allerdings nicht in der Ausprägung wie bei `aqho`.

Die Kombination beider Varianten, also `aqho -DA`, liefert das beste Ergebnis. Mit 1253 Punkten erreicht man eine Genauigkeit von 11.70 Stellen. Das Gitter (Abbildung 42 (rechts)) weist in x_1-Richtung analog zu `aqho -A` eine geringe Verfeinerung auf. Aber auch in Richtung x_2 erkennt man eine differenziertere Abstufung der Verfeinerung.

Dieses Beispiel zeigt, daß die in beiden Varianten `aqho -D` und `aqho -A` enthaltenen Verbesserungen an unterschiedlichen Stellen ansetzen und sich nicht gegenseitig implizieren.

In Abbildung 43 (links) sind die Kennlinien aller vier Varianten bei schrittweiser Verkleinerung von ε_{rel} wiedergegeben. `aqho` benötigt deutlich mehr Gitterpunkte, um ein zu den anderen Varianten qualitativ vergleichbares Ergebnis zu erzielen. `aqho -D` und `aqho -A` liefern annähernd gleich gute Ergebnisse. `aqho -D` ist hier geringfügig überlegen. `aqho -AD` liefert fast durchweg die besten Resultate. Im Vergleich zur Variante `aqho` werden hier für Ergebnisse gleicher Genauigkeit nur etwa 30 bis 50 Prozent der Gitterpunkte benötigt.

Wendet man die `aqho`-Varianten mit kostenorientierter Adaptivität an, wobei man $\delta_{\text{rel}} = 10^{-11}$ wählt, so erhält man Ergebnisse, deren Genauigkeit vergleichbar mit denen bei fehlerorientierter Adaptivitätssteuerung sind. Tabelle 3 gibt für alle acht Varianten die benötigte Anzahl von Gitterpunkten (n), die Genauigkeit (G) und die maximale Verfeinerung in Richtung x_1 ($i_{1,\max}$) an. Man erkennt, daß die kostenorientierte Adaptivität eine weitere Verbesserung des Verfahrens darstellt. Die entsprechenden Gitter unterscheiden sich nur wenig von ihren Gegenstücken mit fehlerorientierter Adaptivität, weshalb sie hier nicht wiedergegeben werden sollen. In Abbildung 43 (rechts) sind die

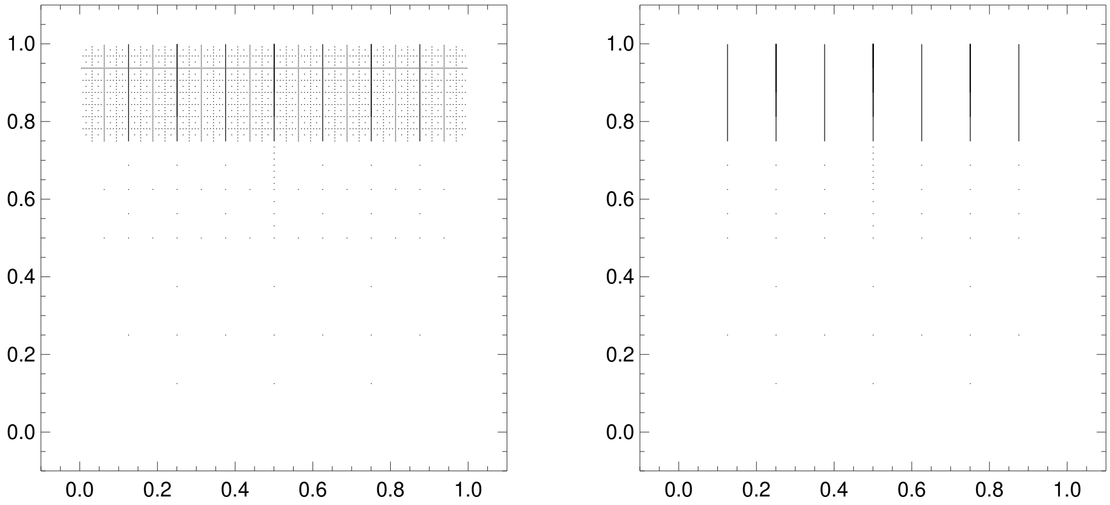

ABB. 41. BEISPIEL 11: *Gitter für* `aqho` *(links) und* `aqho -A` *(recht), jeweils mit* $\varepsilon_{\text{rel}} = 10^{-9}$.

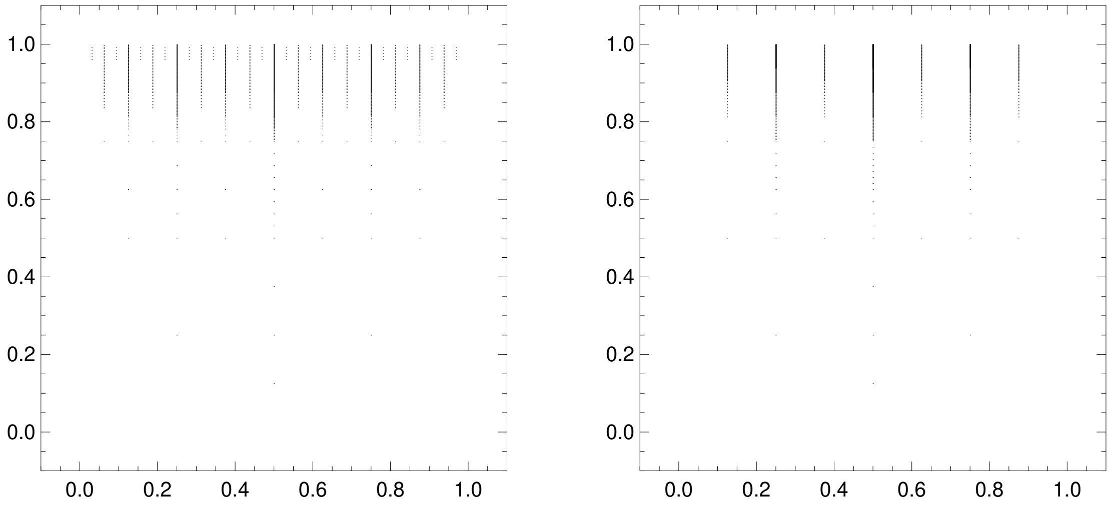

ABB. 42. BEISPIEL 11: *Gitter für* `aqho -D` *(links) und* `aqho -AD` *(recht), jeweils mit* $\delta_{\text{rel}} = 10^{-11}$.

Kennlinien der vier kostenorientierten Varianten dargestellt, die sich durch schrittweise Verkleinerung von δ_{rel} ergeben. Hier gilt das, was bereits zu den Kennlinien der ersten vier Varianten gesagt wurde.

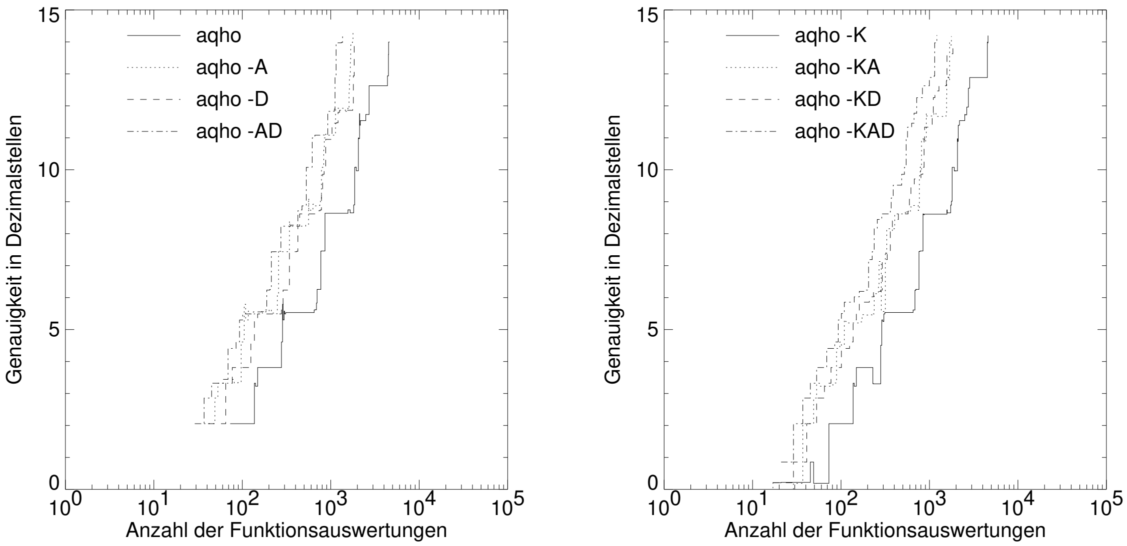

ABB. 43. BEISPIEL 11: *Kennlinien von* `aqho` *bei fehlerorientierter (links) und kostenorientierter (rechts) Adaptivitätssteuerung.*

Variante und Adaptivitätskriterium		n	G	$i_{1,\max}$
`aqho`	$\varepsilon_{\text{rel}} = 10^{-09}$	3457	10.86	7
`aqho -A`	$\varepsilon_{\text{rel}} = 10^{-09}$	1745	10.87	2
`aqho -D`	$\varepsilon_{\text{rel}} = 10^{-09}$	1445	11.05	4
`aqho -AD`	$\varepsilon_{\text{rel}} = 10^{-09}$	1253	11.70	2
`aqho -K`	$\delta_{\text{rel}} = 10^{-11}$	3177	10.60	7
`aqho -KA`	$\delta_{\text{rel}} = 10^{-11}$	1177	10.82	2
`aqho -KD`	$\delta_{\text{rel}} = 10^{-11}$	1113	11.25	4
`aqho -KAD`	$\delta_{\text{rel}} = 10^{-11}$	833	11.59	2

TABELLE 3
BEISPIEL 11: *Ergebnisse der acht Varianten von* `aqho`.

Wir betrachten den Integranden

BEISPIEL 12: $$f(x_1, x_2) = \frac{1}{2\pi} \cdot e^{-\frac{1}{2}\cdot\left(x_1^2 + x_2^2\cdot\left(1-x_1^2\right)\right)} \cdot \sqrt{1-x_1^2} \quad \text{für} \quad (x_1, x_2) \in [-1,1]^2,$$

der sich aus

$$f(y_1, y_2) = \frac{1}{2\pi} \cdot e^{-\frac{1}{2}\cdot\left(y_1^2+y_2^2\right)} \quad \text{für} \quad y_1^2 + y_2^2 \leq 1$$

nach Anwendung der Transformation

$$
\begin{aligned}
x_1 &= y_1 \\
x_2 &= \frac{y_2}{\sqrt{1-y_1^2}},
\end{aligned}
$$

ergibt. Bedingt durch die Dilatation in Richtung y_2, die das kreisförmige Integrationsgebiet auf ein Quadrat abbildet, weist der Integrand im Bereich des linken und rechten Randes starke Glattheit in Richtung x_2 auf.

Abbildung 44 (links) gibt die Gitterpunkte wieder, an denen der Integrand von `aqho` mit $\varepsilon_{\text{rel}} = 10^{-6}$ ausgewertet wird. Man erkennt wieder deutlich eine starke Häufung von Gitterpunkten im linken und rechten Randbereich, also gerade dort, wo der Integrand glatt ist. In Abbildung 44 (rechts) sind die Gitterpunkte für die Variante `aqho -A` (richtungsabhängige Begrenzung der Maximalverfeinerung) dargestellt. Wie im Beispiel zuvor wird durch diese Maßnahme auch hier eine starke Verfeinerung verhindert. Interessanter jedoch sind die Kennlinien der beiden Verfahren, die in Abbildung 45 (links) wiedergegeben sind. Bei beidseitig logarithmischer Skalierung steigt die Kennlinie für `aqho` annähernd linear, die für `aqho -A` quadratisch an. Die Verfeinerung bei `aqho` ist in der dargestellten Phase offensichtlich so stark, daß trotz Ordnungserhöhung die Resultate eines Verfahrens fester Ordnung geliefert werden.

In der Abbildung ist außerdem zu erkennen, daß durch richtungsabhängige Wahl der Halbierungsrichtung (`aqho -D`) ebenfalls die Ordnungsdämpfung aufgehoben werden kann. Allerdings sind hier die Ergebnisse nicht so gut wie bei `aqho -A`. Die Kombination beider Verfahrensvarianten (`aqho -AD`) bringt gegenüber `aqho -A` keine wesentliche Verbesserung. Die Beziehungen der vier Kennlinien untereinander ändern sich im wesentlichen nicht, wenn mit kostenorientierter Adaptivität gerechnet wird. Sie sind in Abbildung 45 (rechts) wiedergegeben. Im Vergleich zur fehlerorientierten Adaptivität erzielen die letzten vier Varianten gleiche Genauigkeit bei etwa der Hälfte der Stützstellen.

In Tabelle 4 sind die Ergebnisse der acht Varianten zusammengefaßt, wobei δ_{rel} und ε_{rel} so gewählt wurden, daß `aqho` und `aqho -K` gleich gute Resultate liefern.

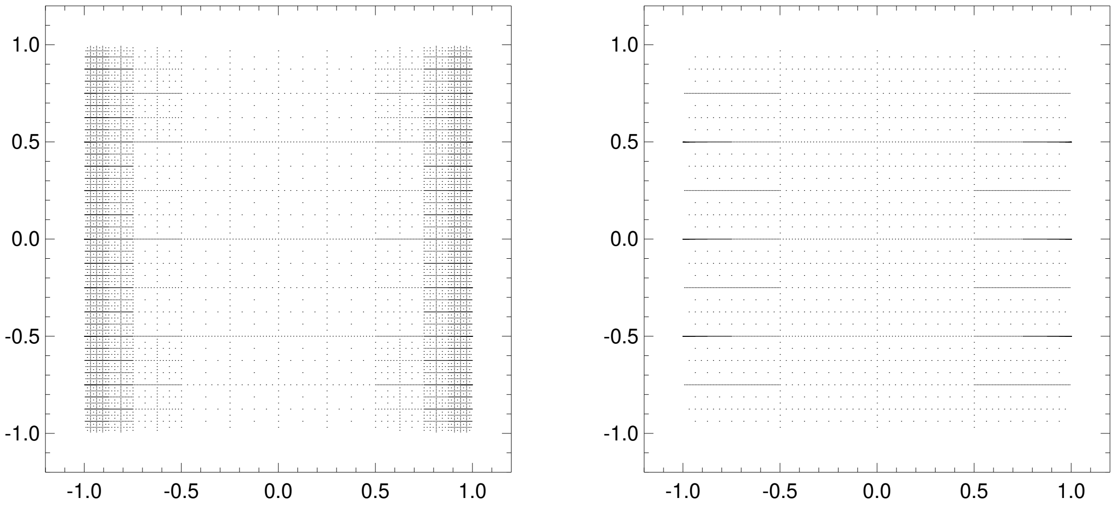

Abb. 44. Beispiel 12: *Gitter für* `aqho` *(links) und* `aqho -A` *(rechts) mit* $\varepsilon_{\mathrm{rel}} = 10^{-6}$.

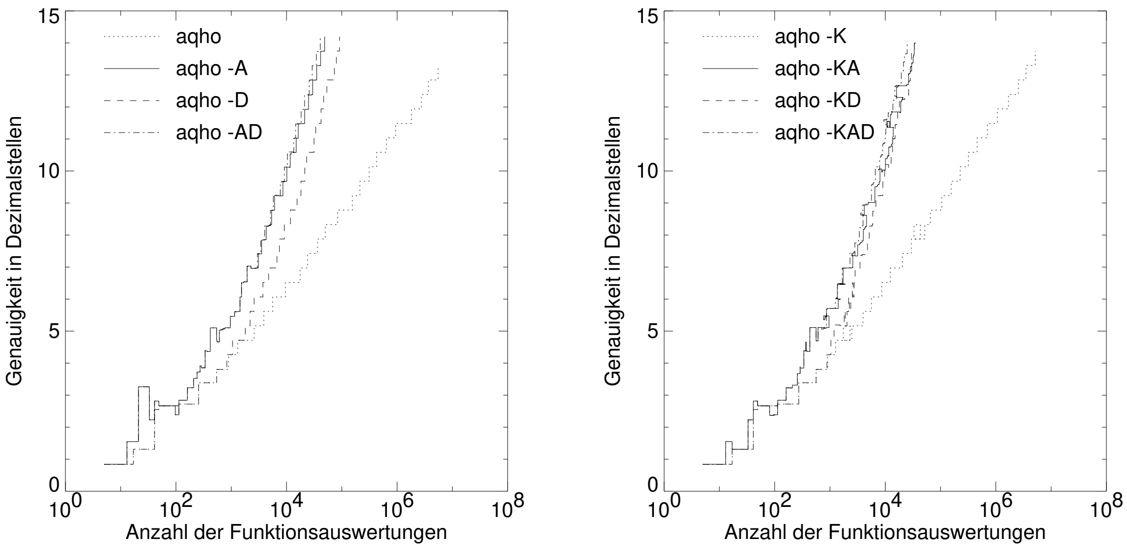

Abb. 45. Beispiel 12: *Kennlinien bei fehlerorientierter (links) und kostenorientierter Adaptivitätssteuerung.*

Variante und Adaptivitätskriterium		n	G	$i_{2,\max}$
`aqho`	$\varepsilon_{\text{rel}} = 10^{-6}$	10881	6.52	7
`aqho -A`	$\varepsilon_{\text{rel}} = 10^{-6}$	3585	7.84	5
`aqho -D`	$\varepsilon_{\text{rel}} = 10^{-6}$	3777	6.52	5
`aqho -AD`	$\varepsilon_{\text{rel}} = 10^{-6}$	3457	7.84	5
`aqho -K`	$\delta_{\text{rel}} = 10^{-9}$	8673	6.52	7
`aqho -KA`	$\delta_{\text{rel}} = 10^{-9}$	4049	8.17	5
`aqho -KD`	$\delta_{\text{rel}} = 10^{-9}$	3889	7.39	5
`aqho -KAD`	$\delta_{\text{rel}} = 10^{-9}$	3441	8.23	5

TABELLE 4
BEISPIEL 12: *Resultate der acht Varianten von* `aqho`.

5. Vergleich mit Konkurrenzverfahren. In der Einleitung dieser Arbeit wurde schon ein Überblick über allgemein verfügbare Quadraturroutinen gegeben. Aufgrund der numerischen Eigenschaften der einzelnen Verfahren kann man festhalten, daß `aqho` sowohl den Monte-Carlo- und zahlentheoretischen Verfahren wie auch den Transformationsmethoden aufgrund der höheren Ordnung asymptotisch überlegen ist, sofern der Integrand hinreichend glatt ist. Diese Verfahren sollen aus diesem Grund hier nicht weiter betrachtet werden.

Als stärkster Konkurrent zu den in dieser Arbeit entwickelten Verfahren erwies sich die Routine `d01fcf` der NAG-Programmbibliothek. Das Verfahren basiert auf einer adaptiven Gebietszerlegungsstrategie, die auf Teilgebieten eine Quadraturregel der Ordnung $O\left(h^8\right)$ anwendet. Das Verfahren wurde ursprünglich von van Dooren ([vDdR76]) vorgestellt. Eine von Genz ([GM80]) verbesserte Variante, die für $d = 2, \ldots, 9$ eine kostengünstigere Quadraturregel verwendet, findet sich in der Routine `d01fcf` wieder. Im folgenden soll dieses Verfahren kurz erklärt werden, soweit dies für einen Vergleich mit `aqho` nötig ist.

5.1. Die Routine `d01fcf`. Die grundlegende Quadraturregel, angewendet auf ein beliebiges Integrationsgebiet B, liefert eine Integralnäherung $I_7(B)$ der Ordnung $O\left(h^8\right)$, einen Fehlerschätzer

$$E(B) \quad = \quad \left|I_7(B) - I_5(B)\right|,$$

der ohne zusätzliche Gitterpunkte mit Hilfe einer zweiten Integralnäherung $I_5(B)$ der Ordnung $O\left(h^6\right)$ ermittelt wird, und die vierten dividierten Differenzen $\delta_\xi(B)$ in allen Richtungen $\xi = 1, \ldots, d$. Das Verfahren errechnet sukzessive eine Folge $\mathcal{S} = \{S_0, S_1, S_2, \ldots\}$ disjunkter vollständiger Zerlegungen des Integrationsgebiets $\{A\}$, wobei $S_0 = \{A\}$ gilt. Für eine gegebene Gebietszerlegung S_n ergibt sich die Näherung $I(S_n)$ für das Gesamtintegral sowie der dazugehörige Fehlerschätzer $E(S_n)$ wie folgt:

$$\begin{aligned} I(S_n) &= \sum_{X \in S_n} I_7(X) \\ E(S_n) &= \sum_{X \in S_n} E(X). \end{aligned}$$

Solange $E(S_n)$ größer als ein vorgegebener maximaler Fehler ε_{rel} ist, wird ein Teilgebiet $B \in S_n$ mit maximalem Fehler, also

$$E(B) \geq E(X) \quad \forall X \in S_n$$

in zwei Hälften B_1 und B_2 zerlegt, wobei die Halbierung in der Koordinatenrichtung $r \in \{1, \ldots, d\}$ mit maximaler vierter dividierter Differenz

$$\left|\delta_r(B)\right| \quad \geq \quad \left|\delta_\xi(B)\right| \quad \forall \xi = 1, \ldots, d$$

durchgeführt wird (Richtungsadaptivität). S_{n+1} ergibt sich dann zu

$$S_{n+1} \quad = \quad S_n \setminus \{B\} \cup \{B_1, B_2\}. \tag{101}$$

Wir wollen nun einige Unterschiede zwischen `d01fcf` und den `aqho`-Varianten aufzeigen. Zum einen ist festzuhalten, daß die Verfahrensordnung von `d01fcf` im Gegensatz zu der von `aqho` fest ist. Sie ist durch die zugrundeliegende Quadraturregel bestimmt. `aqho` hingegen beginnt mit der Verfahrensordnung $O\left(h^2\right)$, kann sie aber beliebig erhöhen. `aqho` wird daher asymptotisch überlegen sein, wenn der Integrand die notwendige Glattheit aufweist.

Nach (101) zerlegt `d01fcf` das gewählte Gebiet B in die Teilgebiete B_1 und B_2. Dabei gehen die Integrandenauswertungen, die mit B verbunden waren, nun nicht mehr in das Endergebnis ein. Dies hat zur Folge, daß über den gesamten Berechnungsverlauf gesehen etwa nur die Hälfte aller berechneten Stützstellen für das Endresultat verwendet werden. Die restlichen dienen lediglich zur Adaptivitätssteuerung. Bei `aqho` gehen alle Stützstellen in das Ergebnis mit ein.

Unterschiedliche Glattheit des Integranden in den Koordinatenrichtungen wird bei `d01fcf` durch die Richtungsadaptivität erkannt. Hierzu wird eine vierte dividierte Differenz herangezogen.

In Beispiel 11 hatten wir gesehen, daß bei starker Glattheit in einer Koordinatenrichtung Begrenzungsmultiindizes verwendet werden mußten, um eine zu starke Verfeinerung in Richtungen hoher Glattheit zu vermeiden. Da `d01fcf` stets eine feste Quadraturregel verwendet, sind solche Maßnahmen hier nicht notwendig.

Die von `d01fcf` berechnete Folge $\mathcal{S}$ ist unabhängig von benutzerwählbaren Parametern, nämlich maximal zulässiger relativer Fehler und maximale Anzahl von Integrandenauswertungen. Beide Parameter legen lediglich fest, nach welchem Folgenglied S_n die Berechnung zu beenden ist. Der Anwender besitzt die Möglichkeit, zu einem späteren Zeitpunkt auf diesem Berechnungsstatus aufzusetzen, um die nächsten Glieder von $\mathcal{S}$ zu berechnen. Bei den `aqho`-Varianten ist ein derartiges Wiederaufsetzen, bedingt durch die Eigenschaften des Algorithmus, nicht möglich.

Eine Unterscheidung zwischen fehler- und kostenorientierter Adaptivität, wie sie für `aqho` vorgestellt wurde, entfällt bei `d01fcf`, da die Anzahl der Funktionsauswertungen und damit die Kosten in jedem Schritt konstant sind.

Die in `d01fcf` verwendete Gauß-Quadraturregel kann durch das Verfahren `qho` der gleichen Ordnung ($m = 3$) ersetzt werden, denn für $I_5\left(B\right)$ und $\delta_\xi\left(B\right),\ \xi = 1, \ldots, d$, die für die Adaptivitätssteuerung von `d01fcf` benötigt werden, können von `qho` entsprechende Werte geliefert werden. Man kann nun die Kosten der Quadraturregeln miteinander vergleichen. Bezeichnen N_D, N_G und $N_{\mathtt{qho}}$ jeweils die Anzahl der Stützstellen für die Varianten von van Dooren, von Genz und mit `qho`, so gilt:

$$\begin{aligned} N_D &= \tfrac{1}{3} \cdot \left(4 \cdot d^3 - 6 \cdot d^2 + 20 \cdot d + 3\right) \quad \text{für} \quad d > 4 \\ N_G &= 2^d + 2 \cdot d^2 + 2 \cdot d + 1 \\ N_{\mathtt{qho}} &= \tfrac{1}{3} \cdot \left(4 \cdot d^3 + 18 \cdot d^2 + 20 \cdot d + 3\right). \end{aligned}$$

Es gilt $N_{\mathtt{qho}} > N_D$, beide sind aber von der Ordnung $O\left(\frac{4}{3} \cdot d^3\right)$. In Abbildung 46 sind die Werte für $d = 1, \ldots, 15$ aufgetragen. Man sieht, daß `qho` den anderen Quadraturregeln

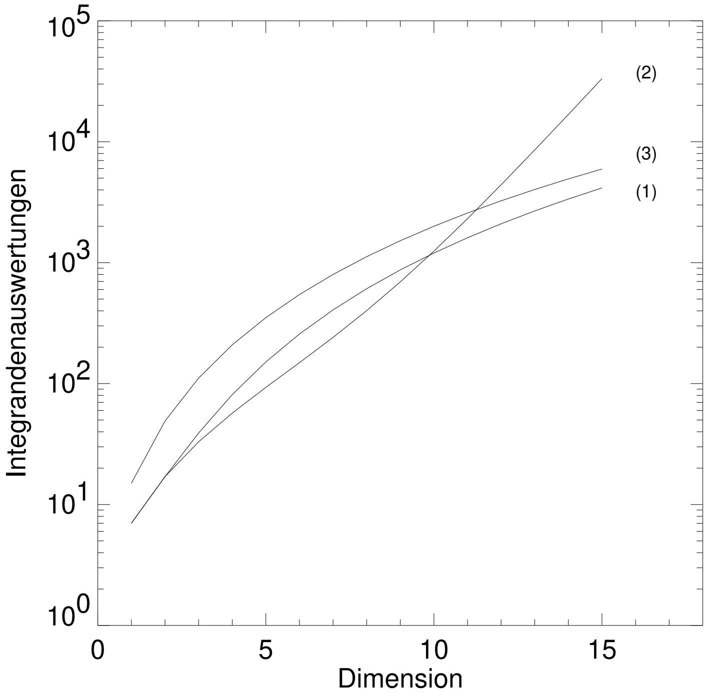

ABB. 46. *Kosten der verschiedenen Quadraturregeln:* N_D (1), N_G (2), N_{qho} (3).

unterlegen ist. Daran ändert sich auch nichts, wenn Mehrfachauswertungen verhindert werden. Es gilt dann:

$$N'_{\texttt{qho}} \;=\; \tfrac{1}{3} \cdot \left(4 \cdot d^3 + 12 \cdot d^2 + 8 \cdot d\right).$$

Trotzdem ist `qho` eine interessante Alternative für eine zugrundeliegende Quadraturregel für `d01fcf`, da leicht Regeln höherer Ordnung erzeugt werden können.

5.2. Numerische Experimente. Im folgenden wollen wir die Varianten von `aqho` mit `d01fcf` vergleichen. Wir betrachten dazu:

BEISPIEL 13: $\quad f\,(x_1, x_2, \ldots, x_d) \;=\; e^{x_1+x_2+\ldots+x_d} \;\text{ für }\; (x_1, x_2, \ldots, x_d) \in [-2, 2]^d.$

Für $d = 3, 4, 5, 6$ sind die Kennlinien von `d01fcf`, `aqho`, `aqho -A`, `aqho -K` und `aqho -AK` in Abbildung 47 einander gegenübergestellt. Die Varianten mit adaptiver Wahl der Halbierungsrichtung bringen aufgrund der Symmetrieeigenschaft des Integranden keine Verbesserung.

Die Kennlinie für `d01fcf` steigt im Gegensatz zu den `aqho`-Varianten nur linear an. Dies ergibt sich unmittelbar aus der festen Ordnung des Verfahrens. Die Steigung vermindert sich mit zunehmender Dimension. Dies läßt sich mit den eingangs vorgestellten Ergebnissen der Komplexitätstheorie zeigen, wonach zwischen der erzielten Genauigkeit

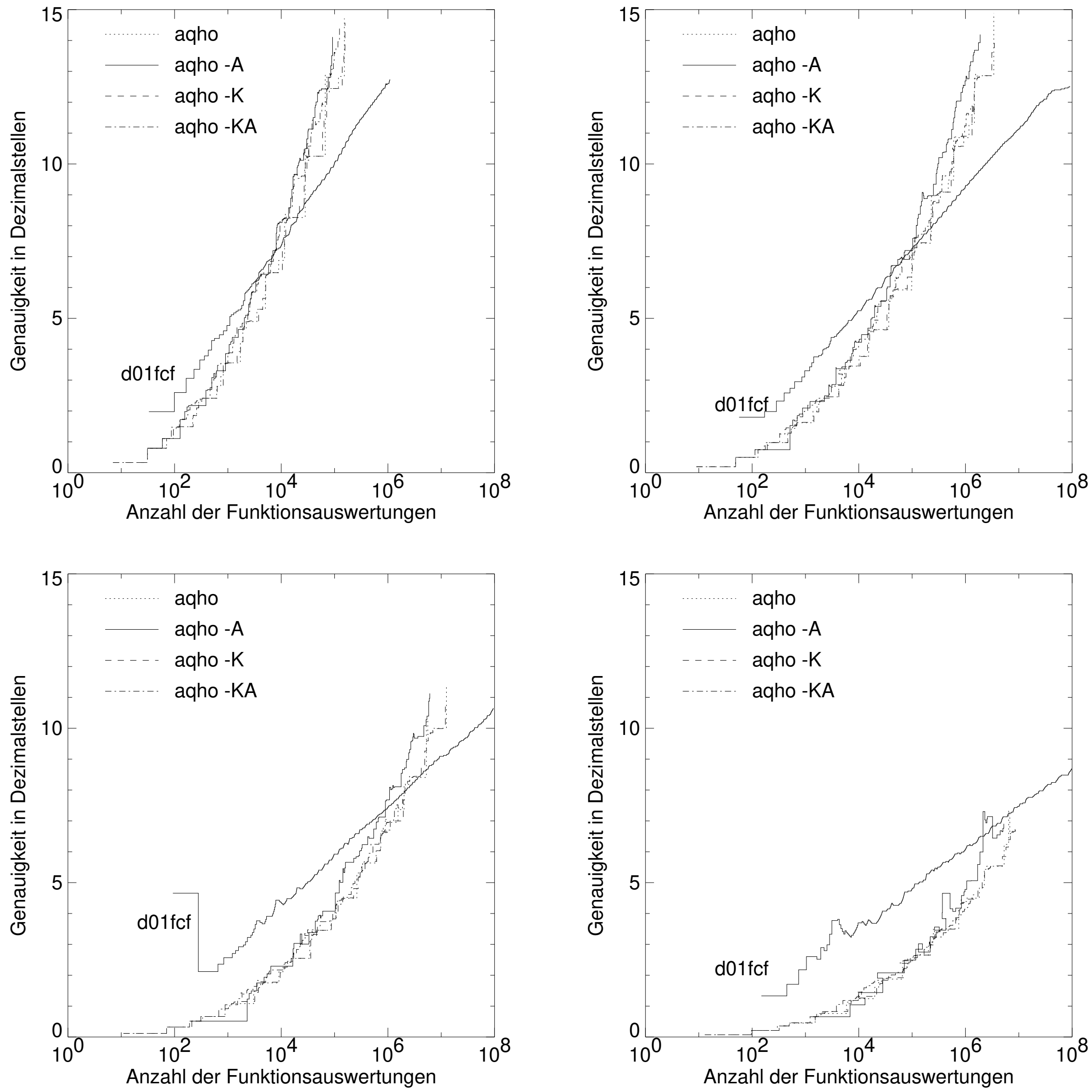

ABB. 47. BEISPIEL 13 *Kennlinien für* $d = 3, \ldots, 6$.

ε und der benötigten Anzahl von Stützstellen n der Zusammenhang

$$n \;=\; O\left(\varepsilon^{-\frac{d}{r}}\right)$$

gilt. Aufgrund der festen Ordnung von `d01fcf` gilt hier $r = 8$. Bei logarithmischer Achsenskalierung besitzt die Kennlinie eine Steigung von $\frac{8}{d}$, die durch Nachmessen in den Abbildungen bestätigt wird.

In der Anfangsphase zeigt sich bei `d01fcf` eine Überlegenheit, da hier die `aqho`-Varianten nicht die Ordnung von `d01fcf` erreichen. Mit zunehmender Verringerung der Adaptivitätssteuergrößen $\varepsilon_{\rm rel}$ beziehungsweise $\delta_{\rm rel}$ aber steigt die Ordnung der Verfahren an, so daß `d01fcf` überholt wird. In den Fällen $d = 3, 4, 5$ wird das Konkurrenzverfahren bei einer Genauigkeit von 7 bis 8 Dezimalstellen geschlagen. Im Falle $d = 6$ konnte der Verlauf der Kennlinien nicht mehr weiter berechnet werden, da der vom Betriebssystem zur Verfügung gestellte Datenspeicher von 256 MBytes erschöpft war. Der Verlauf der Kennlinie deutet aber darauf hin, daß `d01fcf` bei etwa 10^7 bis 10^8 Stützstellen ebenfalls bei 7 bis 8 Stellen Genauigkeit übertroffen wird.

Die aus der Ordnungserhöhung resultierende Stufung ist für die Variante `aqho` besonders deutlich zu erkennen. Die Kennlinien aller Varianten von `aqho` liegen sehr eng beieinander, was sich mit zunehmender Dimension noch verstärkt. Bei hohen Dimensionen neigen die `aqho`-Varianten dazu, in das nicht-adaptive Verfahren `qho` überzugehen.

Wir wollen nun untersuchen, inwiefern die Ergebnisse dieses Beispiels auch bei anderen Integranden zutreffen. Wir betrachten die von A. C. Genz ([Gen84]) vorgeschlagenen Testintegranden

Beispiel 14:

$$\begin{aligned} f_1(\vec{x}) &= \cos\left(2 \cdot \pi \cdot u_1 + \sum_{i=1}^{d} a_i \cdot x_i\right) \\ f_2(\vec{x}) &= \prod_{i=1}^{d} \left(a_i^{-2} + (x_i - u_i)^2\right)^{-1} \\ f_3(\vec{x}) &= \left(1 + \sum_{i=1}^{d} a_i \cdot x_i\right)^{-(d+1)} \\ f_4(\vec{x}) &= e^{-\left(\sum_{i=1}^{d} a_i^2 \cdot (x_i - u_i)^2\right)} \end{aligned}$$

mit $\vec{x} \in [0,1]^d$ und beliebigem $\vec{u} \in [0,1]^d$. Jede Integrandenklasse zeichnet sich durch eine charakteristische Eigenschaft aus. Bei $f_1(\vec{x})$ handelt es sich um oszillierende Funktionen, bei $f_2(\vec{x})$ um Funktionen mit einem Maximum im Inneren des Integrationsgebiets, bei $f_3(\vec{x})$ um Funktionen mit Randmaximum bei $\vec{x} = \vec{0}$ und bei $f_4(\vec{x})$ um Gauß-artige Funktionen. Alle diese Integranden besitzen den Vorteil, daß Funktionen beliebiger Dimension erzeugt werden können, daß sie relativ schnell ausgewertet werden können und daß das exakte Integral in hinreichender Genauigkeit analytisch berechenbar ist. Außerdem kann die Ausprägung der charakteristischen Eigenschaft durch den Parameter $\vec{a} = (a_1, a_2, \ldots, a_d)$ festgelegt werden. Für die Berechnung von $\vec{a}$ wird zunächst ein beliebiges $\vec{b} \in [0,1]^d$ gewählt, aus dem sich $\vec{a} = c \cdot \vec{b}$ ergibt, wobei c so zu wählen ist, daß gilt:

$$d^{e_i} \cdot |\vec{a}| = p.$$

Dabei ist e_i ein klassenspezifischer Parameter, mit dem die zunehmende Schwierigkeit des Quadraturproblems bei steigender Dimension kompensiert wird. Wir übernehmen die von Genz verwendeten Werte $(e_1, e_2, e_3, e_4) = (1.5, 2, 2, 1)$. Mit dem Parameter p kann die Glattheit des Integranden beeinflußt werden. Er bestimmt unmittelbar die Schwierigkeit des Quadraturproblems. Für die Festlegung von $\vec{u}$ und $\vec{a}$ wurde der Zufallszahlengenerator `drand48()` verwendet, der auf vielen UNIX-Systemen zur Verfügung steht, wobei die Initialisierung mit `srand48(1234567890)` erfolgte.

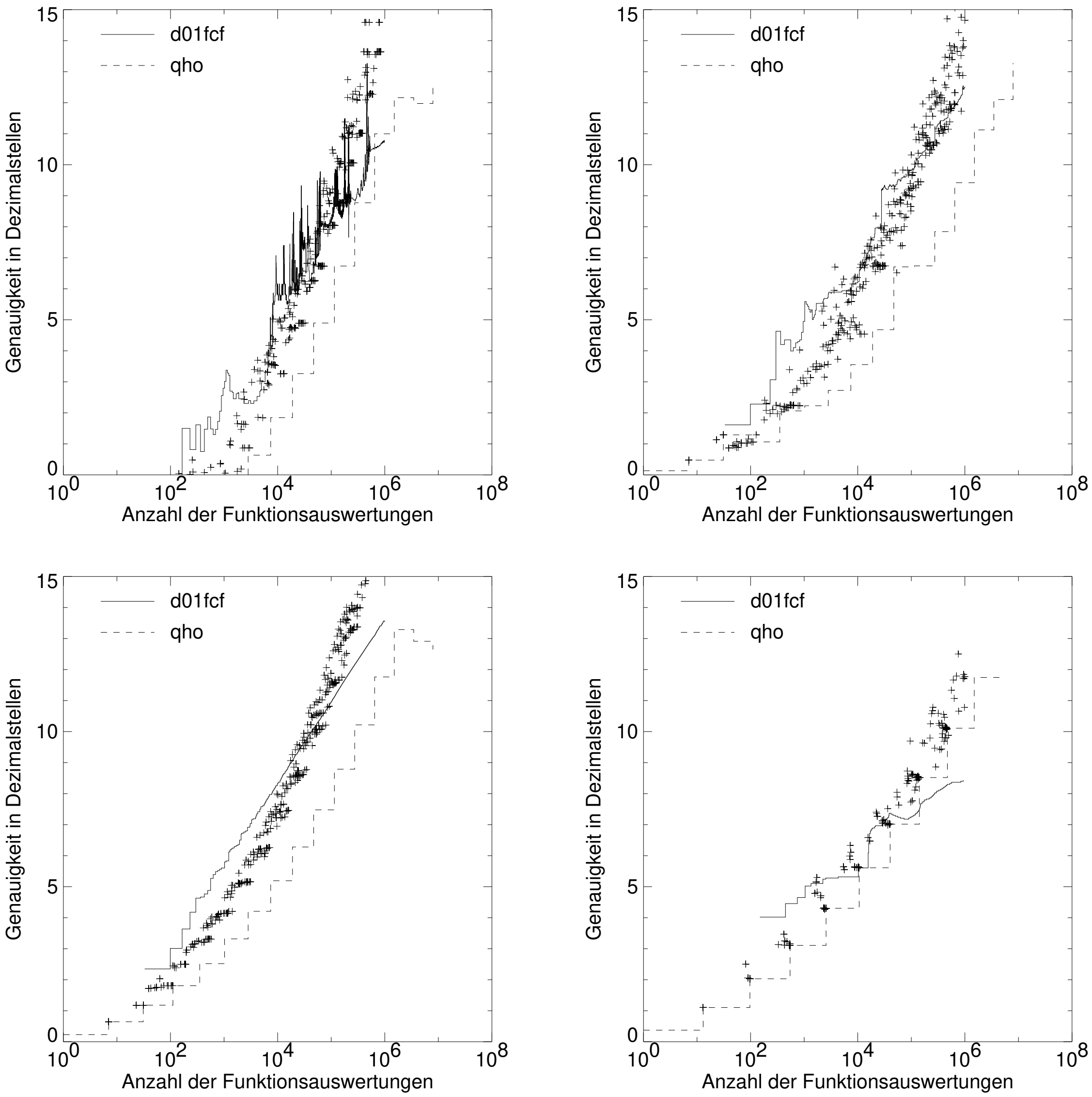

Abb. 48. Beispiel 14: *Ergebnisse für folgende Tupel von* (f, d, p)*:* $(f_1, 3, 128)$*,* $(f_2, 3, 128)$*,* $(f_3, 3, 32)$ *und* $(f_4, 6, 32)$*.*

Für alle acht `aqho`-Varianten sowie für `qho` und `d01fcf` wurden die Kennlinien der ersten Integranden der vier Testfamilien ermittelt, wobei $d \in \{2, 3, 4, 6, 8\}$ und $p \in \{8, 16, 32, \ldots, 256\}$ gewählt wurden. In allen Fällen wurden die Ergebnisse von Beispiel 13 bestätigt (lineare beziehungsweise parabelförmige Kennlinien, Unterlegenheit in der Anfangsphase). In Abbildung 48 sind die Ergebnisse von folgenden Tupeln von Testfamilie, d und p dargestellt (von links nach rechts und von oben nach unten): $(f_1, 3, 128), (f_2, 3, 128), (f_3, 3, 32)$ und $(f_4, 6, 32)$. Hier sind die Kennlinien für `d01fcf` und `qho` eingezeichnet. Für die acht `aqho`-Varianten sind die ermittelten Ergebnisse durch Punkte gekennzeichnet, die sich ergeben, wenn die Adaptivitätskenngröße schrittweise halbiert wird.

Generell stellte sich bei allen Experimenten heraus, daß die `aqho`-Varianten bei kleinem p annähernd die Ergebnisse des nicht-adaptiven Verfahrens `qho` lieferten. Dieser Effekt war ebenso bei zunehmender Dimension zu beobachten. Bei Integranden mit großem p, also geringerer Glattheit, erwies sich `d01fcf` oftmals robuster als die `aqho`-Verfahren. Dies ist darauf zurückzuführen, daß sich die Adaptivitätsentscheidungen von `aqho` auf eine zu geringe Anzahl von Gitterpunkten stützen. Abhilfe kann geschaffen werden, wenn `aqho` derart modifiziert wird, daß das Stop-Flag beziehungsweise die Begrenzungsmultiindizes erst in Knoten gesetzt werden, deren Teilbaum eine Mindesttiefe von $m_{\min} \geq 1$ erreicht hat.

Bei allen verwendeten Integranden wurde das Konkurrenzverfahren frühestens bei 7 bis 8 Dezimalstellen Genauigkeit überholt. In einigen Fällen kam es bei der Berechnung erst gar nicht so weit, da die `aqho`-Verfahren aus Mangel an Speicherplatz vorzeitig abbrachen.

Zum Schluß soll noch das Verhalten bei hohen Dimensionen vorgeführt werden. Wir betrachten den Integranden:

Beispiel 15: $$f(x_1, x_2, \ldots, x_d) = e^{-\sum_{i=1}^{n} a_i^2 \cdot (x_i - u_i)^2} \quad \text{für} \quad (x_1, x_2, \ldots, x_d) \in [0,1]^d,$$

mit $d \in \{11, 15\}$ und $p = 100$. Wie bereits zuvor erwähnt, gehen die `aqho`-Varianten in `qho` über, weshalb in Abbildung 49 nur noch die Ergebnisse von `d01fcf` und `qho` wiedergegeben sind. Das Konkurrenzverfahren wird mit $d = 11$ bei 7 beziehungsweise mit $d = 15$ bei 6 Dezimalstellen Genauigkeit geschlagen. Man beachte, daß `d01fcf` für die erste Näherung 2313 ($d = 11$) und 33249 ($d = 15$) Gitterpunkte benötigt. `qho` wertet 2575 ($d = 11$) beziehungsweise 5951 ($d = 15$) Stützstellen bei Ordnung $O(h^8)$ aus. Hier macht sich die mit der Dimension exponentielle Zunahme von Stützstellen der zugrundeliegenden Quadraturregel von `d01fcf` bemerkbar.

Abhilfe kann hier geschaffen werden, wenn die `d01fcf`-Variante von van Dooren verwendet wird, die im Fall $d = 15$ nur ein Zehntel des Aufwands erfordert.

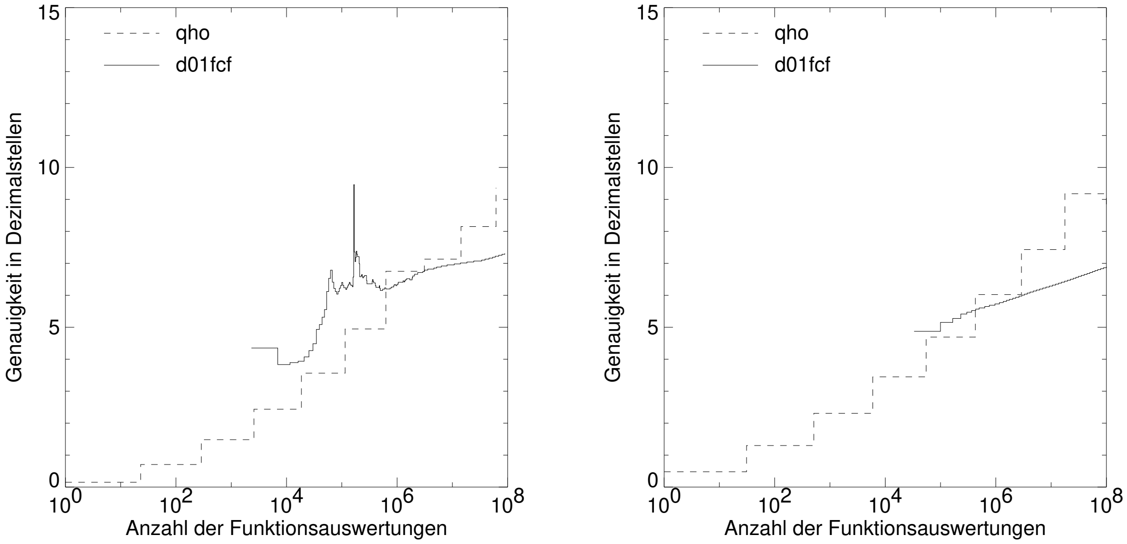

ABB. 49. BEISPIEL 15: *Kennlinien von* `d01fcf` *und* `qho` *für* $d = 11$ *(links) und* $d = 15$ *(rechts).*

6. Schlußbemerkungen. In dieser Arbeit wurden drei Verfahren zur numerischen Quadratur mehrdimensionaler Funktionen vorgestellt, die alle auf dünnen Gittern basieren. Es wurden zwei verschiedene Ansätze gewählt. Im ersten Verfahren sah man, daß sich dünne Gitter in natürlicher Weise aus der Kombination von Archimedischer Ausschöpfung und Cavalierischem Prinzip ergaben. Man erhält ein interpolierendes Verfahren basierend auf stückweise d-linearen Funktionen. Der zweite Ansatz nutzte die Eigenschaft, daß reguläre dünne Gitter als Überlagerung von vollen Gitter angesehen werden können, die entstehen, wenn Mittelpunktsummen unterschiedlicher Maschenweiten in den jeweiligen Koordinatenrichtungen berechnet werden. Es wurde hierzu ein Verfahren vorgestellt, das als eine multivariate Verallgemeinerung der Romberg-Quadratur angesehen werden kann. Das dritte Verfahren nutzte die Rekursivität von dünnen Gittern aus, um Verfeinerungsadaptivität und Ordnungsadaptivität zu vereinen. Es zeigte sich, daß die Resultate verbessert werden können, wenn die Verfeinerung richtungsabhängig begrenzt und die Halbierungsrichtung adaptiv gewählt wird. Außerdem ergab sich, daß eine Kosten-Nutzen-orientierte Adaptivitätssteuerung oftmals bessere Resultate lieferte. Anhand von zahlreichen Testintegranden mit $d = 2, \ldots, 8$ stellte sich heraus, daß die asymptotische Überlegenheit des Verfahrens gegenüber dem stärksten Konkurrenzverfahren, einem Verfahren fester Ordnung, frühestens bei 6 bis 7 Dezimalstellen Genauigkeit zum Zuge kommt. In höherdimensionalen Problemen ($d > 8$) erwiesen sich diese Verfahren wegen des nicht mit der Dimension exponentiell ansteigende Kostenaufwands, gepaart mit hoher Ordnung, als überlegen, sofern der Integrand hinreichend glatt ist.

Die zweidimensionale Archimedische Quadratur wurde als Prototyp verwendet, um zu untersuchen, inwiefern hierarchisch strukturierte numerische Algorithmen durch strombasierte Programmiersprachen beschrieben und effizient auf MIMD-Maschinen ausgeführt werden können ([Ebn94], [Rau92]). Des weiteren wurde für dieses Verfahren eine Fallstudie zur Entwicklung paralleler Programme mit Hilfe von formalen Entwurfs- und Verifikationsmethoden, basierend auf stromverarbeitenden Funktionen, durchgeführt ([Gor92],[Gor93]). Im Zuge der ständig zunehmenden Bedeutung von parallelen numerischen Algorithmen wurde das nicht-adaptive Quadraturverfahren höherer Ordnung auf einem iPSC-Hypercube-Parallelrechner implementiert ([Sin91]).

[Bon93] Th. Bonk. A New Algorithm for Multi-Dimensional Adaptive Numerical Quadrature. In W. Hackbusch, G. Wittum, editor, *Adaptive Methods: Algorithms, Theory and Applications, Notes on Numerical Fluid Mechanics*, volume 46. Vieweg Verlag, Braunschweig, 1993.

[BR92] Th. Bonk und U. Rüde. Performance Analysis and Optimization of Numerically Intensive Programs. SFB Bericht 342/26/92 A, Institut für Informatik, TU München, 1992.

[BS64] R. Bulirsch und J. Stoer. Fehlerabschätzungen und Extrapolation mit rationalen Funktionen bei Verfahren vom Richardson-Typus. *Numerische Mathematik*, 6:413–427, 1964.

[Bul64] R. Bulirsch. Bemerkungen zur Romberg-Integration. *Numerische Mathematik*, 6:6–16, 1964.

[Bun92] H.-J. Bungartz. Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung. Dissertation, TU München, Institut für Informatik, Juni 1992.

[Del89] F.-J. Delvos. R-th Order Blending Rectangle Rules. *International Series of Numerical Mathematics, Birkhäuser Verlag Basel*, 90:107–114, 1989.

[DN89] F.-J. Delvos und H. Nienhaus. A Trivariate Boolean Cubature Scheme. *International Series of Numerical Mathematics, Birkhäuser Verlag Basel*, 90:115–122, 1989.

[DR84] P. J. Davis und P. Rabinowitz. *Methods of Numerical Integration (2nd ed.)*. Academic Press, New York, 1984.

[Ebn94] R. Ebner. Neuimplementierung einer funktionalen Sprache zur Parallelisierung numerischer Algorithmen. Diplomarbeit, Technische Universität München, 1994.

[Eng80] H. Engels. *Numerical Quadrature and Cubature*. Academic Press, New York, 1980.

[Gen74] A. C. Genz. Some Extrapolation Methods for the Numerical Calculation of Multidimensional Integrals. *Software for Numerical Mathematics, Academic Press*, pages 159–172, 1974.

[Gen75] A. C. Genz. The Approximate Calculation of Multidimensional Integrals Using Extrapolation Methods. Dissertation, University of Kent at Canterbury, Juni 1975.

[Gen84] A. C. Genz. Testing Multidimensional Integration Routines. In B. Ford, J. C. Rault, F. Thomasset, editor, *Proceedings of INRIA Conference on Software Tools, Methods and Languages*. North-Holland, Amsterdam, 1984.

[GM80] A. C. Genz und A. A. Malik. Remarks on Algorithm 006: An Adaptive Algorithm for Numerical Integration over an N-dimensional Rectangular Region. *J. Comput. Appl. Math.*, 6(4):295–302, 1980.

[Gor92] S. Gorlatch. Parallel Program Development for a Recursiv Numerical Algorithm: a Case Study. SFB Bericht 342/7/92 A, Institut für Informatik, TU München, 1992.

[Gor93] S. Gorlatch. Deriving Efficient Parallel Programs by Systemating Coarsing Specification Parallelism. SFB Bericht 342/20/93 A, Institut für Informatik, TU München, 1993.

[GSZ90] M. Griebel, M. Schneider, und C. Zenger. A Combination Technique for the Solution of Sparse Grid Problems. SFB Bericht 342/19/90 A, Institut für Informatik, TU München, 1990.

[IMS91] IMSL Inc., Houston. *IMSL/MATH/LIBRARY: User's Manual (Version 2.0)*, 1991.

[Kor63] N. M. Korobov. *Number Theoretic Methods in Approximate Analysis*. Fizmatgiz, Moskau, 1963.

[Lau71] B. Lautrup. An Adaptive Multi-dimensional Integration Procedure. In *Proc. 2nd Coll. on Advanced Methods in Theoretical Physics*. Marseille, 1971.

[LM70] J. N. Lyness und B. J. J. McHugh. On the Remainder Term in the N-Dimensional Euler Maclaurin Expansion. *Numerische Mathematik*, 15:333–344, 1970.

[McK72] E. H. McKinney. Generalized Recursive Multivariate Interpolation. *Mathematics of Computation*, 26(119):723–735, 1972.

[NAG91] NAG Ltd., Oxford. *NAG Fortran Library Manual - Mark 15*, 1991.

[Rau92] H. Rauschmayer. Werkzeuge zur Bearbeitung funktionaler Programme auf Parallelrechnern. Diplomarbeit, Technische Universität München, 1992.

[RG27] C. Richardson und J. Gaunt. The Deferred Approach to the Limit. *Trans. Roy. Soc. Lond.*, 226:300–361, 1927.

[Sin91] C. Sing. Parallelisierung eines Algorithmus zur numerischen Quadratur im mehrdimensionalen Raum. Diplomarbeit, Technische Universität München, 1991.

[SS64] T. W. Sag und G. Szekeres. Numerical Evaluation of High-Dimensional Integrals. *Math. Comp.*, 18:245–253, 1964.

[Sto83] J. Stoer. *Einführung in die Numerische Mathematik I.* Band 1. Springer Verlag, Berlin, Heidelberg, New York, 1983.

[Str71] A. H. Stroud. *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Englewood Cliffs, New Jersey, 1971.

[TWW88] J. F. Traub, G. W. Wasilkowski, und H. Woźniakowski. *Information-Based Complexity.* Academic Press, New York, 1988.

[vDdR76] P. van Dooren und L. de Ridder. An Adaptive Algorithm for Numerical Integration over an N-dimensional Cube. *Journal of Computational and Applied Mathematics*, 2(3):207–217, 1976.

[Zen90] C. Zenger. Sparse Grids. SFB Bericht 342/18/90 A, Institut für Informatik, TU München, 1990.

[Zen93] C. Zenger. Funktionale Programmierung paralleler Algorithmen für numerische und technisch-wissenschaftliche Anwendungen. In H. G. Bock, W. Hackbusch, R. Rannacher, editor, *Numerische Algorithmen auf Transputer-Systemen.* Teubner Verlag, Stuttgart, 1993.