

Adaptive numerische Quadratur und Kubatur mit automatischer Ergebnisverifikation

DISSERTATION

von

Chin-Yun Chen

Karlsruhe 1998

Adaptive numerische Quadratur und Kubatur mit automatischer Ergebnisverifikation

Zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
von der Fakultät für Mathematik
der Universität Karlsruhe (Technische Hochschule)

genehmigte

Dissertation

von

Chin-Yun Chen

aus Taipeh, Taiwan

Tag der mündlichen Prüfung : 16. Dez. 1998
Referent : Prof. Dr. Ulrich Kulisch
Korreferent : H.-Doz. Dr. Rudolf Lohner

Im Anfang war das Wort, ...

So lautet es nicht nur in der Bibel, sondern auch Lao Tzu hat einige Jahrhunderte vor Christus das gleiche in Tauteking angedeutet.

Das Wort ist die Wahrheit aller Existenz, ist der Ursprung aller Lebenwesen, der bereits ausgestorbenen, der gerade lebenden, der noch nicht geborenen bzw. der noch nicht anerkannten.

Die normalen menschlichen Augen können nur die dreidimensionalen Sachen erkennen, dennoch interessieren sich die Mathematiker für die Verallgemeinerung ihrer Theorien auf höhere Dimensionen.

Bevor die Menschen aus der Ganzheit ein Stück Wahrheit entdecken, erkennen oder darüber diskutieren, existiert das "Wort" schon ewig da.

Worauf können die Menschen stolz sein, wenn sie sich von der Wahrheit trennen ?

Ich stehe am Eingang der wissenschaftlichen Arbeit, erinnere mich an die Ganzheit der Wahrheit, fühle mich so winzig, geschweige denn das kleine Buch, das vor ihren Augen liegt, da es nämlich eine kleine Episode aus meinem diesmaligen Leben ist.

Danksagung

Diese Arbeit entstand im Rahmen eines Promotionsstipendiums des Landes Baden-Württemberg. Für die finanzielle Unterstützung danke ich herzlichst der Landesregierung Baden-Württemberg und der Vergabekommission.

Bei Herrn Prof. Dr. Ulrich Kulisch möchte ich mich an dieser Stelle besonders herzlich bedanken, da erst durch seine stete Unterstützung mir die Möglichkeit gegeben wurde, diese Thematik zu bearbeiten und damit die vorliegende Arbeit in diesem Umfang durchzuführen.

Frau Dr. Ulrike Storck möchte ich auch für die Anregung dieser Arbeitsthematik, die stetige Diskussionsbereitschaft und die wertvollen Ratschläge in jeder Phase besonders herzlich danken.

Herrn H.-Doz. Dr. Rudolf Lohner danke ich herzlichst für die freundliche Übernahme des Korreferats sowie die fruchtbaren Diskussionen und Anregungen, die zur Entstehung dieser Arbeit einen wichtigen Beitrag geleistet haben.

Herrn Prof. Dr. Edgar Kaucher möchte ich ebenfalls herzlich danken für sein Interesse an meiner Arbeit und die aufschlußreichen Hinweise, die er mir zu der Forschungsarbeit gegeben hat.

Herrn Dipl.-Math. Stefan Dietrich danke ich aufrichtig für seine zeitaufwendige Unterstützung, mir die numerischen Grunddaten bereitzustellen, die für meine Untersuchungen wichtig waren.

Herrn Dipl.-Math. Stefan Wedner und Herrn Dr. habil. Dietmar Ratz danke ich ebenso herzlich für ihre Bereitschaft zur Diskussion und Bereitstellung unterschiedlicher Hilfsmittel, so daß ich meine Untersuchungen leicht durchführen konnte.

Herrn Dipl.-Phys. Lothar Gerhardt schulde ich meinen besonderen Dank für die ständige Unterstützung und den zeitraubenden Dienst beim Korrekturlesen der Manuskripte der Dissertation und der Arbeitsberichte, die ich für das Erlangen des Promotionsstipendiums schreiben mußte.

Herrn H.-Doz. Dr. Walter Krämmer und Herrn Priv.-Doz. Dr. Knut Petras gebührt ebenfalls mein herzlicher Dank für die hilfreichen Verbesserungsvorschläge zu dem Manuskript dieser Dissertation.

Bei meinen Kollegen und ehemaligen Kollegen am Institut für Angewandte Mathematik möchte ich mich ebenfalls für die technische und moralische Unterstützung während meines Aufenthalts an diesem Institut bedanken.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation und Übersicht	1
1.1.1	Numerische Integration	1
1.1.2	Adaptive Strategien	3
1.1.3	Automatische Ergebnisverifikation	3
1.2	Automatische und Verifizierte Berechnung	5
1.2.1	Selbstverifizierende Algorithmen	5
1.2.2	Die Programmiersprache PASCAL-XSC	5
1.3	Überblick über die vorliegende Arbeit	6
2	Grundlagen der Verifikationsnumerik	7
2.1	Rundungen und maximal genaue Verknüpfungen	7
2.2	Reelle Intervallarithmetik	9
2.2.1	Intervallrechnung über den endlichen Zahlen	11
2.2.2	Erweiterte Intervallrechnung	12
2.3	Wertebereichseinschließung reeller Funktionen	13
2.3.1	Die Konvergenz und die Mittelwertform	14
2.3.2	Einschließung durch Verfeinerung	16
2.4	Nullstellenbestimmung einer reellwertigen Funktion	16
2.4.1	Erweitertes Intervall-Newton-Verfahren	17
2.4.2	Die Methode von Hansen/Neumaier	17
2.5	Differentiationsarithmetik	20
2.5.1	Eindimensionale Taylorkoeffizienten	20
2.5.2	Zweidimensionale Taylorkoeffizienten	23
3	Eindimensionale numerische Integration	28
3.1	Integration durch Transformation	29
3.2	Interpolationsquadraturen	29
3.2.1	Symmetrische Quadraturformeln	30
3.2.2	Orthogonalpolynome	30

3.3	Spezielle Quadraturformeln	32
3.3.1	Gauß-Legendre-Quadratur	32
3.3.2	Clenshaw-Curtis-Quadratur	32
3.4	Quadraturfehler	33
3.4.1	Peanosche Kerndarstellung	34
3.4.2	Bestimmung der Fehlerschranken	37
3.4.3	Verifizierte Berechnung der Restgliedfaktoren	40
4	Mehrdimensionale numerische Integration	58
4.1	Integration durch Transformation	58
4.2	Kubaturformeln	59
4.2.1	Vollsymmetrische Kubaturformeln	60
4.2.2	Produktformeln	61
4.3	Kubaturfehler	62
4.3.1	Der Approximationsfehler von Produktformeln	62
4.3.2	Sard-Kerndarstellungen und Fehlereinschließungen	64
4.3.3	Eigenschaften der Sard-Kerne	70
4.3.4	Verifizierte Berechnung der Restgliedfaktoren	96
5	Verifizierte automatische Integration	105
5.1	Doppelte adaptive Strategien	106
5.1.1	Adaptive Verfeinerungsstrategien	107
5.1.2	Adaptive Exaktheitsgrade	110
5.2	Unterteilung eines Integrationsbereichs	111
5.2.1	Zerlegung in 2^n Teilbereiche	111
5.2.2	Binäre Unterteilung	111
5.3	Zuverlässige und effiziente Fehlerbehandlung	113
5.3.1	Reine absolute Fehlerbehandlung	114
5.3.2	Absolute und relative Fehlerbehandlung	115
5.4	Verifikation über dem Bereichsrand	116
5.4.1	Eindimensionaler Bereich	116
5.4.2	Mehrdimensionaler Bereich	116
5.5	Das Programm VANI	119
5.5.1	Benutzerschnittstellen	119
5.5.2	Der verwendete Baum	122
5.5.3	Programmaufbau	124
6	Numerische Beispiele	129
6.1	Numerische Quadratur	129

6.1.1	Glatte Funktionen	129
6.1.2	Funktionen mit Peak	131
6.1.3	Oszillierende Funktionen	133
6.2	Numerische Kubatur	134
6.2.1	Zum Vergleich der vorhandenen Kubaturverfahren	135
6.2.2	Glatte Funktionen	137
6.2.3	Funktionen mit Peak	139
6.2.4	Oszillierende Funktionen	141
7	Zusammenfassung und Ausblick	143
A	Programm zur Berechnung der Fehlerkonstante bzgl. der Peano-Kerne	146
B	Das Modul ditaylor2	157
	Literatur	170

Tabellenverzeichnis

3.1	<u>Restgliedfaktoren $E(x^{d+1})$ der Gauß-Legendre-Formeln</u>	40
3.2	<u>Restgliedfaktoren $r! f_{-1}^1 K_r^+$ und $r! f_{-1}^1 K_r^-$ der Gauß-Legendre-Formeln</u> . . .	44
3.3	<u>Vergleich der Restgliedfaktoren $r! f_{-1}^1 K_r^+$ der Gauß-Legendre-Formeln</u> . .	48
3.4	<u>Restgliedfaktoren $r! f_{-1}^1 K_r^+$ und $r! f_{-1}^1 K_r^-$ der Clenshaw-Curtis-Formeln</u> . .	50
3.5	<u>Vergleich der Restgliedfaktoren $r! f_{-1}^1 K_r$ der Clenshaw-Curtis-Formeln</u> .	57
4.1	<u>1-dim. Fehlerkonstanten : VSK_4 über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 3</u>	99
4.2	<u>2-dim. Fehlerkonstanten : VSK_4 über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 3</u>	99
4.3	<u>1-dim. Fehlerkonstanten von VSK_{12} über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 7</u> . .	100
4.4	<u>2-dim. Fehlerkonstanten von VSK_{12} über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 7</u> . .	101
4.5	<u>Vergleich 1-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 3$</u> . . .	102
4.6	<u>Vergleich 2-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 3$</u> . . .	102
4.7	<u>Vergleich 1-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 7$</u> . . .	103
4.8	<u>Vergleich 2-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 7$</u> . . .	104

Kapitel 1

Einleitung

Die vorliegende Arbeit beschäftigt sich mit der adaptiven numerischen Integration. An Stelle einer Näherung wird gemäß einer geforderten Genauigkeit ein Intervall berechnet, welches das exakte Integral garantiert einschließt. Alle Elemente aus dem berechneten Intervall sind Näherungen für das Integral. Im Gegensatz zu Näherungsverfahren liefern Einschließungsverfahren eine mathematisch gesicherte Aussage.

1.1 Motivation und Übersicht

1.1.1 Numerische Integration

Da einerseits selbst für einfache Funktionen $f : \Omega \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ das bestimmte Integral $I(f) := \int_{\Omega} f(\vec{x}) d\vec{x}$ nicht immer analytisch berechnet werden kann (vg. [47], S. 1), und andererseits die Stammfunktionen, wenn sie sich in geschlossenen Formen darstellen lassen, numerisch schwierig auswertbar sein können (vgl. [40], S. 2), ist die numerische Integration, die sich mit der näherungsweise Berechnung der Integrale von Funktionen beschäftigt, erforderlich. Die numerische Integration ist eines der ältesten mathematischen Probleme und wird in verschiedenen wissenschaftlichen Bereichen angewendet (vgl. [15], [152]).

Zur Durchführung der numerischen Integration sind zwei Teilaufgaben zu bewältigen. Eine beschäftigt sich mit der Konstruktion von Integrationsformeln und die andere mit der Abschätzung der Approximationsgüte der Integrationsformeln. Für die beiden Teilaufgaben wurden lange Zeit hauptsächlich Funktionen in einer Veränderlichen betrachtet (vgl. [146], [32], [104]).

Mit Hilfe von Integrationsformeln, die endliche Linearkombinationen von Funktionswerten darstellen, wird das gesuchte Integral näherungsweise berechnet, d. h. $I(f) \cong S(f) := \sum_{i=1}^M w_i f(\vec{x}_i)$. Im Eindimensionalen werden diese Formeln als Quadratur- und im Mehrdimensionalen als Kubaturformeln bezeichnet. Die interpolatorischen Quadraturformeln,

die mit den Orthogonalpolynomen zusammenhängen, bilden eine wichtige Klasse von Integrationsformeln. Im Mehrdimensionalen ist die numerische Integration wesentlich komplizierter als im Eindimensionalen. Dies liegt daran, daß es im Mehrdimensionalen unendlich viele Integrationsbereiche gibt, die bzgl. affiner Abbildungen nicht äquivalent sind (vgl. [146]). Außerdem lassen sich im Mehrdimensionalen nicht immer gute Kubaturformeln konstruieren, auch wenn mehrdimensionale Orthogonalpolynome herangezogen werden (vgl. [32]). Für die mehrdimensionale numerische Integration werden interpolatorische vollsymmetrische Kubaturformeln mit geringer Knotenzahl bevorzugt. Eine weitere wichtige Klasse von Kubaturformeln sind die Produktformeln, die sich nur für gewisse Integrationsbereiche durch Tensorproduktbildung aus den Integrationsformeln in niedrigeren Dimensionen konstruieren lassen.

Zur Beurteilung der Approximationsgüte der Integrationsformeln wird der Diskretisierungsfehler $E(f) := I(f) \Leftrightarrow S(f)$ betrachtet. Im konkreten Fall erweist sich die Bestimmung günstiger Fehlerschranken oft als schwierig und aufwendig. Für diese Aufgabe werden in der Literatur (z. B. [47], [40]) verschiedene Methoden, wie z. B. die Kernmethode für hinreichend oft differenzierbare Funktionen und die ableitungsfreie Methode (siehe auch [2], [135]) für Integranden, die in einem komplexen Gebiet holomorph fortsetzbar sind, angegeben. In dieser Arbeit wird die Kernmethode, die ursprünglich von Peano für den Quadraturfehler hergeleitet und später von Sard auf den Kubaturfehler verallgemeinert wurde, eingesetzt. Die Sard-Kerndarstellung gilt sowohl für Nicht- als auch für Produktformeln (vgl. [146]). Für Produktformeln existieren jedoch noch weitere Fehlerdarstellungen, unter denen die Nikolskii'sche Methode am günstigsten ist. Im allgemeinen läßt sich der Kubaturfehler einer Produktformel auf die Approximationsfehler der zugrundeliegenden Integrationsformeln zurückführen (vgl. [99]).

Unter der Bedingung, daß die Integranden hinreichend oft stetig differenzierbar sind, wird bei Anwendung der Kernmethode der Diskretisierungsfehler mit Hilfe von dualen Normen abgeschätzt. Üblicherweise wird die Supremum-Norm für die (partiellen) Ableitungen der Integranden und die L_1 -Norm für die Kernfunktionen verwendet. In der Praxis ist es allerdings schwierig, die Supremum-Norm der (partiellen) Ableitungen zu gewinnen. Mit Hilfe der Intervall- und der Differentiationsarithmetik lassen sich jedoch die Wertebereiche der Ableitungen effizient und zuverlässig abschätzen (vgl. [111]). Aber auch die Berechnung der L_1 -Norm der Kernfunktionen erweist sich als eine schwierige Aufgabe. Daher wurde meistens diese L_1 -Norm wiederum durch numerische Integration näherungsweise berechnet (vgl. [147], [146], [102]). Mit der Verifikationsnumerik kann jedoch diese Aufgabe durch Einsatz von Überdeckungsmethoden, bei denen die Nullstellen der Kernfunktionen explizit bzw. implizit berücksichtigt werden, zuverlässig gelöst werden.

Bis jetzt wurden mit den Hilfsmitteln der Verifikationsnumerik nur die Fehlerkonstanten bzgl. der eindimensionalen Peano-Kerne berechnet ([144], vgl. auch [69]). Verifizierte Berechnung der Fehlerkonstanten bzgl. der Sard-Kerne wird in dieser Arbeit erstmals durchgeführt. Für diese Aufgabe waren die Untersuchungen viel umfangreicher als für die Peano-Kerne, da es im Mehrdimensionalen mehrere Lösungsansätze und mehrere Teilaspekte zu berücksichtigen gibt.

1.1.2 Adaptive Strategien

Um für die Integralnäherung die erwünschte Genauigkeit effektiv zu erreichen, erweist es sich als zweckmäßig, den Integrationsbereich, abhängig vom Funktionsverhalten des Integranden und von der geforderten Genauigkeit, dynamisch in verschiedene Teilbereiche zu zerlegen. Für jeden Teilbereich wird dann eine Integrationsformel verwendet. Die dynamische Zerlegung des Integrationsbereiches wird auch adaptive Verfeinerung genannt. Falls zusätzlich die lokalen Integrationsformeln unterschiedliche Exaktheitsgrade besitzen, spricht man von adaptiven Exaktheitsgraden. Häufig bezieht sich die adaptive numerische Integration auf diese doppelte Adaption.

Für die Bereichsverfeinerung unterteilen sich die verwendeten Methoden in zwei Klassen, die als die lokale bzw. die globale Strategie bezeichnet werden. Bei der globalen Strategie wird die gesamte Genauigkeit geprüft. Erfüllt die erzielte Integralnäherung bzw. Integraleinschließung die geforderte Genauigkeit nicht, dann wird der Teilbereich mit dem maximalen lokalen Fehler weiter unterteilt. Hingegen muß bei der lokalen Strategie jede Teilintegralnäherung bzw. jede lokale Fehlereinschließung jeweils eine lokale Genauigkeit erfüllen. Alle Teilbereiche, für welche die lokalen Genauigkeiten nicht erfüllt werden können, werden weiter unterteilt. Im Extremfall kann dadurch ein unendlicher Programmablauf verursacht werden. Es bestehen zwar Möglichkeiten, den unendlichen Programmablauf zu unterbrechen, jedoch muß man eine sehr grobe Integraleinschließung bzw. eine ziemlich lange Rechenzeit in Kauf nehmen (s. Kapitel 5). Diese Problematik läßt sich bei Anwendung der globalen Strategie vermeiden, da unerreichbare Genauigkeiten rechtzeitig entdeckt werden können. Dadurch werden unnötige Berechnungen erspart. Die gewöhnlichen automatischen Integratoren sind im allgemeinen nicht in der Lage, unerreichbare Genauigkeiten zu entdecken (s. numerische Beispiele in Kapitel 6). Mit den verifizierten automatischen Integratoren, in denen die lokale Verfeinerungsstrategie verwendet wird, lassen sich unerreichbare Genauigkeiten nur passiv entdecken. Dies bedeutet im Extremfall eine Fülle von unnötigem Rechenaufwand.

Um einen stabilen automatischen Integrator zu implementieren, wird in dieser Arbeit die globale Strategie eingesetzt. Bei Anwendung der globalen Strategie müssen Informationen zu sämtlichen Teilbereichen abgespeichert werden. Für einen effizienten Zugriff auf den Teilbereich mit dem maximalen lokalen Fehler spielt die Datenstruktur zur Registrierung der lokalen Informationen eine wichtige Rolle. Aus Effizienzgründen wird in dieser Arbeit ein balancierter partiell geordneter Baum verwendet, da für jeden neuen Knoten der Sortierungsaufwand logarithmisch ist (s. Kapitel 5).

1.1.3 Automatische Ergebnisverifikation

Das Zahlensystem in einer Rechenanlage ist nur eine endliche Teilmenge der reellen Zahlen. Im einfachsten Fall spricht man von einem einfach genauen Gleitkommasystem $R(b, l, e_{\min}, e_{\max})$ (single precision floating-point numbers). Für höhere Genauigkeitsanforderungen wird ein doppelt genaues Gleitkommasystem $D(b, 2l, \tilde{e}_{\min}, \tilde{e}_{\max})$ verwendet,

dessen Raster feiner als R ist. Hierbei bezeichnet b die Basis des Zahlensystems, l die maximale Mantissenlänge, e_{\min} den kleinsten Exponenten und e_{\max} den größten Exponenten. Das numerische Rechnen vollzieht sich nicht nur in diesen Zahlenmengen sondern auch in den davon abgeleiteten Produkträumen.

$$\begin{array}{lcl} \mathbb{R} & \supset & D \quad \supset \quad R \\ \mathbb{R}^n & \supset & D^n \quad \supset \quad R^n \\ \mathbb{R}^{m \times n} & \supset & D^{m \times n} \quad \supset \quad R^{m \times n} \end{array}$$

Bei vielen Anwendungen treten noch komplexe Räume auf. Davon wird in dieser Arbeit kein Gebrauch gemacht.

Da die Gleitkommasysteme echte Teilmengen der reellen Zahlen sind, folgt, daß nicht alle reellen Zahlen in R bzw. D exakt dargestellt werden können. Viele Eingangsgrößen und Zwischenergebnisse eines numerischen Algorithmus werden deshalb gerundet abgespeichert. Numerische Lösungen, die auf dieser Basis berechnet sind, weichen wegen der Rundungsfehler von den exakten Lösungen ab. In manchen Fällen können die akkumulierten Rundungsfehler sogar die numerischen Ergebnisse total verfälschen. Aussagen über die Genauigkeit bzw. Statistiken über den Rechenaufwand sind besonders wertvoll, wenn man die numerischen Ergebnisse verifizieren kann. Eine genaue Analyse der Fortpflanzung von Rundungsfehlern ist wegen der Komplexität der numerischen Algorithmen häufig nicht möglich. Um die Rundungsfehler zu minimieren, ist es zweckmäßig, eine Rechnerarithmetik zu verwenden, welche die Verknüpfungsergebnisse maximal genau berechnet. Dies bedeutet, daß alle Grundoperationen, insbesondere die, die in den Produkträumen definiert sind, so durchgeführt werden, daß das Rechenergebnis nur mit einer Rundung aus der exakten Größe bestimmt wird und zwischen dem gerundeten und dem exakten Ergebnis keine weitere Gleitkommazahl (bzw. Gleitkommamatrix) liegt. In [89] wird eine solche Rechnerarithmetik vorgestellt, die nach dem Prinzip des Semimorphismus definiert ist.

Damit auch die Rundungsfehler berücksichtigt werden können, muß für die automatische Ergebnisverifikation die Intervallarithmetik eingesetzt werden. Neben den Skalar- und Produkträumen sind also zusätzlich Intervallräume zu betrachten :

$$\begin{array}{lcl} I\mathbb{R} & \supset & ID \quad \supset \quad IR \\ I\mathbb{R}^n & \supset & ID^n \quad \supset \quad IR^n \\ I\mathbb{R}^{m \times n} & \supset & ID^{m \times n} \quad \supset \quad IR^{m \times n} \end{array}$$

Die arithmetischen Operationen auf $I\mathbb{R}$ werden durch die Operationen auf \mathbb{R} definiert :

$$[a] \circ [b] := \{ a \circ b \mid a \in [a], b \in [b] \}, \quad [a], [b] \in I\mathbb{R}.$$

Es wird hier angenommen, daß außer den maximal genauen Grundoperationen in den Skalarräumen auch sämtliche inneren und äußeren Verknüpfungen in den Produkt- und Intervallräumen maximal genau definiert und standardmäßig im Computersystem als Grundoperationen bereitgestellt sind, so daß die automatische Ergebnisverifikation zuverlässig und einfach durchgeführt werden kann. In den XSC-Programmiersprachen, die für die automatische Ergebnisverifikation entwickelt wurden, ist eine solche maximal genaue Rechnerarithmetik realisiert.

1.2 Automatische und Verifizierte Berechnung

Ein wichtiges Ziel der automatischen Ergebnisverifikation ist die Zuverlässigkeit und i. a. eine hohe Genauigkeit des berechneten Ergebnisses. Um dieses Ziel zu erreichen, wird als Basis eine maximal genaue Rechnerarithmetik benötigt. Für Skalar- und Produkt Räume müssen die Rundungen \square , \triangle bzw. ∇ bereitgestellt werden, welche das exakte Ergebnis jeweils auf die nächstgelegene, die nächstgrößere bzw. die nächstkleinere Gleitkommazahl abbilden. Speziell für die Intervallrechnung wird noch die Intervallrundung \diamond benötigt, die ein reelles Intervall auf das kleinste einschließende Gleitkommaintervall abbildet.

1.2.1 Selbstverifizierende Algorithmen

Außer einer maximal genauen Rechnerarithmetik müssen für die automatische Ergebnisverifikation Algorithmen entwickelt werden, welche sowohl die Rundungsfehler als auch die Diskretisierungsfehler automatisch und zuverlässig erfassen. Um all diese Fehler sicher erfassen zu können, muß Intervallararithmetik verwendet werden. Damit berechnet man an Stelle einer Näherung ein (enges) Intervall, welches die exakte Lösung garantiert einschließt. Derartige Algorithmen werden als selbstverifizierende Algorithmen bezeichnet. Enge Wertebereichseinschließungen von Funktionen können erhalten werden, indem man eine Verfeinerungsstrategie auf den Definitionsbereich anwendet.

Durch Einsatz selbstverifizierender Algorithmen läßt sich nicht nur die absolute sondern häufig auch die relative Genauigkeit zuverlässig erfassen. Ferner können mit selbstverifizierenden Algorithmen unter Verwendung von Fixpunktsätzen häufig auch die Existenz und die Eindeutigkeit der exakten Lösung in dem Suchgebiet überprüft werden. Im allgemeinen sind solche Algorithmen komplizierter als entsprechende Algorithmen aus der gewöhnlichen Numerik und deshalb auch aufwendiger zu implementieren. Zahlreiche Anwendungen selbstverifizierender Algorithmen findet man in [61] und [87].

1.2.2 Die Programmiersprache PASCAL-XSC

Die Softwareprogramme, die im Rahmen dieser Arbeit für die numerischen Untersuchungen implementiert wurden, sind in der Programmiersprache PASCAL-XSC (PASCAL-eXtension for Scientific Computation) geschrieben. PASCAL-XSC wurde zum Zweck der automatischen Ergebnisverifikation entwickelt und in der ist eine maximal genaue Rechnerarithmetik implementiert (vgl. [84]).

Die vier Grundoperationen $\{+, \Leftrightarrow, \cdot, /\}$ stehen im PASCAL-XSC nicht nur für ganze Zahlen und Gleitkommazahlen zur Verfügung, sondern auch für Vektoren, Matrizen und Intervalle, die ebenfalls als arithmetische Standardtypen definiert sind. Bei der Multiplikation von Matrizen, wird das optimale Skalarprodukt für jede Komponente automatisch durchgeführt, d. h. jede Komponente der Ergebnismatrix wird nur mit einer einzigen

Rundung berechnet. Ferner sind alle in PASCAL-XSC verfügbaren elementaren Funktionen auf 1 Ulp (Unit in the last place) genau, d.h., daß zwischen dem exakten und dem berechneten Ergebnis keine weitere Gleitkommazahl liegt. Für intervallmäßige elementare Funktionen ist die Abweichung höchstens 2 Ulp.

Darüber hinaus können die monotonen Rundungen $\{\square, \triangle, \nabla, \diamond\}$ sowohl für arithmetische Standarddatentypen als auch für arithmetische Ausdrücke explizit verwendet werden. Dies ermöglicht eine flexible Kontrolle der Rundungen.

Um für die selbstverifizierenden Algorithmen eine hohe Genauigkeit zu erlangen, wurde in PASCAL-XSC der neue Datentyp *dotprecision* eingeführt. Anhand des Festkommaformates (langer Akkumulator vom Bereich $R(b, 2l, 2e_{\min}, 2e_{\max})$ mit $R = R(2, 53, \Leftrightarrow 1021, 1024)$) erlaubt *dotprecision* eine exakte Summation von beliebigen Produkten zweier Gleitkommazahlen. Skalarprodukte, die man ohne Rundung berechnen möchte, können durch das Lattenkreuzsymbol # gekennzeichnet werden.

Neben den maximal genauen Rundungen und Operatoren unterstützt PASCAL-XSC auch noch Konzepte moderner Programmiersprachen und moderner Programmiermethoden. U. a. sind dies z. B. das universelle Operatorkonzept (benutzerdefinierte Operatoren), Funktionen und Operatoren mit beliebigem Ergebnistyp, Überladen von Prozeduren, Funktionen und Operatoren, das Modulkonzept und dynamische Felder. Diese Eigenschaften verstärken die Benutzerfreundlichkeit (Schreibbarkeit), erhöhen die Lesbarkeit der Programme und erleichtern die Programmwartung. Für die Implementierung selbstverifizierender Algorithmen, welche im allgemeinen komplizierter als die entsprechenden Algorithmen aus der gewöhnlichen Numerik sind, sind diese Konzepte besonders förderlich und erforderlich.

1.3 Überblick über die vorliegende Arbeit

Im nächsten Kapitel wird zuerst eine kurze Einführung in die Intervallrechnung gegeben. Danach werden verschiedene Hilfsmitteln, die für die Abschätzung der Diskretisierungsfehler von Integrationsformeln notwendig sind, diskutiert.

Die Grundlagen der numerischen Quadratur werden in Kapitel 3 beschrieben und die Grundlagen der numerischen Kubatur in Kapitel 4. In den beiden Kapiteln wird der Schwerpunkt auf die Betrachtung des Diskretisierungsfehlers gelegt. Insbesondere werden die Eigenschaften der Kernfunktionen untersucht.

Mit Hilfe der Werkzeuge der Verifikationsnumerik wurde im Rahmen dieser Arbeit ein automatischer Integrator implementiert. Die verschiedenen Teilaspekte zur adaptiven Durchführung der numerischen Quadratur und Kubatur mit automatischer Ergebnisverifikation werden in Kapitel 5 behandelt.

In Kapitel 6 wird anhand von verschiedenen Beispielen die Effektivität des implementierten automatischen Integrators demonstriert. Abschließend werden in Kapitel 7 die durchgeführten Arbeiten zusammengefasst und weitere Untersuchungsmöglichkeiten vorgestellt.

Kapitel 2

Grundlagen der Verifikationsnumerik

2.1 Rundungen und maximal genaue Verknüpfungen

Die numerischen Räume $\mathbb{R} \supset D \supset R$, $\mathbb{R}^n \supset D^n \supset R^n$ und $\mathbb{R}^{m \times n} \supset D^{m \times n} \supset R^{m \times n}$, die mit Verknüpfungen und Ordnungsrelationen versehen sind, bilden geordnete algebraische Strukturen. Da die Gleitkommasysteme R und D endliche Teilmengen der reellen Zahlen \mathbb{R} sind, müssen für die Berechnungen auf elektronischen Rechenanlagen die reellen Zahlen zunächst durch Rundungen in R bzw. D abgebildet werden :

$$\mathbb{R} \xrightarrow{\circlearrowleft} R \quad \text{bzw.} \quad D, \quad \circlearrowleft \in \{\square, \triangle, \nabla\},$$

wobei \square eine reelle Zahl auf die nächstgelegene Gleitkommazahl abbildet, und \triangle bzw. ∇ jeweils die nächstgrößere bzw. die nächstkleinere Gleitkommazahl liefert.

Diese Abbildungen sind weder isomorph noch homomorph. Beispiele dafür findet man in [92]. Für $\circlearrowleft : M \rightarrow N$, M, N numerische Räume mit $M \supset N$, muß gelten ([92]) :

$$(R1) \quad \bigwedge_{a \in N} \circlearrowleft a = a \quad (\text{Rundung})$$

$$(R2) \quad \bigwedge_{a, b \in M} (a \leq b \Rightarrow \circlearrowleft a \leq \circlearrowleft b) \quad (\text{Monotonie})$$

$$(RG) \quad \bigwedge_{a, b \in N} a \circlearrowleft b := \circlearrowleft (a \circ b), \quad \circ \text{ Verknüpfung in } M$$

Zusätzlich gilt für \triangle bzw. ∇

$$(R3) \quad \bigwedge_{a \in M} a \leq \triangle a \quad (\text{nach oben gerichtet})$$

$$\bigwedge_{a \in M} \nabla a \leq a \quad (\text{nach unten gerichtet})$$

und für \square

$$(R4) \quad \bigwedge_{a \in M} \square(\Leftrightarrow a) = \Leftrightarrow \square a \quad (\text{Antisymmetrie})$$

Die Rundungen \square, Δ, ∇ bzw. die Verknüpfungen $\boxplus, \boxdot, \boxminus$ auf N , die gemäß (R1), (R2) und (RG) erklärt sind, sind *maximal genau*, d.h. zwischen a und $\square a$ bzw. zwischen $a \circ b$ und $\square(a \circ b)$, $\square \in \{\square, \Delta, \nabla\}$, existiert kein weiteres Element aus N .

Ferner impliziert die Antisymmetrie eine symmetrische Struktur von N :

$$0 \in N \quad \wedge \quad a \in N \quad \begin{array}{c} (R4) \\ \implies \\ (R1) \end{array} \quad \Leftrightarrow a \in N.$$

Die Antisymmetrie gilt nur für \square nicht jedoch für Δ und ∇ wegen

$$\Delta(\Leftrightarrow a) = \Leftrightarrow \nabla(a) \quad \text{und} \quad \nabla(\Leftrightarrow a) = \Leftrightarrow \Delta(a) \quad \text{für} \quad a \in M.$$

Eine antisymmetrische, monotone Rundung, deren Einschränkung auf die Bildmenge ein Homomorphismus ist, wird in der Literatur [92] als *Semimorphismus* bezeichnet.

Bei der Realisierung von (RG) im Rechner wird das Skalar $a \circ b$ durch eine darstellbare Zahl $a \tilde{\circ} b$ mit der Eigenschaft $\square(a \circ b) = \square(a \tilde{\circ} b)$ ersetzt, falls $a \circ b$ in einem Computer nicht exakt darstellbar ist. Ferner ist gemäß (RG) die Matrixmultiplikation durch das *optimale Skalarprodukt* definiert, d.h., daß jede Komponente der Ergebnismatrix nur mit *einer Rundung* berechnet wird :

$$a \odot b := \square(a \cdot b) = \square \left(\sum_{k=1}^n a_{ik} \cdot b_{kj} \right).$$

Entsprechend wird für die Intervallarithmetik (s. Abschnitt 2.2) auch eine *Intervallrundung* $\diamond : I\mathbb{R} \rightarrow I\mathbb{R}$ (bzw. *ID*) eingeführt. Die Intervallrundung \diamond erfüllt ebenfalls (R1), (R4) und (RG). Zusätzlich gilt für die Inklusionsrelation " \subseteq "

$$(R2) \quad \bigwedge_{[a], [b] \in I\mathbb{R}} ([a] \subseteq [b] \Rightarrow \diamond[a] \subseteq \diamond[b])$$

und

$$(R3) \quad \bigwedge_{[a] \in I\mathbb{R}} [a] \subseteq \diamond[a].$$

Nach dieser Definition wird ein reelles (Punkt-)Intervall auf das kleinste umfassende Gleitkommaintervall abgebildet, $\diamond[\underline{a}, \bar{a}] := [\nabla \underline{a}, \Delta \bar{a}]$.

Da die Operationen $+$, \cdot auf Vektorintervall- bzw. Matrixintervallräumen nicht durchführbar sind, werden bei der Implementierung dieser Verknüpfungen die Intervallvektor- und Intervallmatrixräume betrachtet. In [92] wird gezeigt, daß diese Räume isomorph sind.

$$\begin{aligned} I\mathbb{R}^{m \times n} &\longleftrightarrow (I\mathbb{R})^{m \times n} \\ ID^{m \times n} &\longleftrightarrow (ID)^{m \times n} \\ IR^{m \times n} &\longleftrightarrow (IR)^{m \times n} \end{aligned}$$

Es seien $A := ([a_{ij}])$, $B := ([b_{ij}])$ zwei Intervallmatrizen passender Dimension. Die Operationen $\{+, \cdot\}$ für die reellen Intervallmatrizen A , B , mit $[a_{ij}], [b_{ij}] \in I\mathbb{R}$, sind definiert durch

$$\begin{aligned} A + B &:= ([a_{ij}] + [b_{ij}]) \\ \text{und } A \cdot B &:= \left(\sum_{k=1}^n [a_{ik}] \cdot [b_{kj}] \right). \end{aligned}$$

Laut (RG) gilt also für $[a_{ij}], [b_{ij}]$ aus IR bzw. ID

$$\begin{aligned} A \diamond B &:= \diamond ([a_{ij}] + [b_{ij}]) = ([a_{ij}] \diamond [b_{ij}]), \\ A \diamond B &:= \diamond \left(\sum_{k=1}^n [a_{ik}] \cdot [b_{kj}] \right) = \left(\diamond \sum_{k=1}^n [a_{ik}] \cdot [b_{kj}] \right). \end{aligned}$$

Alle Komponenten werden nur einmal gerundet. Diese Operationen sind isomorph zu den maximal genauen Verknüpfungen, die auf $IR^{m \times n}$ bzw. $ID^{m \times n}$ gemäß (RG) definiert sind ([92]).

2.2 Reelle Intervallarithmetik

Die Intervallarithmetik ist ein unentbehrliches Werkzeug für die automatische Verifikation von numerischen Ergebnissen. Die Rundungsfehler, die die Begrenztheit des Computerzahlensystems verursacht, sowie die Diskretisierungsfehler, die durch die Anwendung numerischer Verfahren entstehen, lassen sich mit Hilfe der Intervallarithmetik u. U. zuverlässig abschätzen, d. h. der Gesamtfehler wird durch eine Unter- und eine Obergrenze sicher eingeschlossen. Man erhält so statt einer Näherung ein (enges) Intervall, das die exakte Lösung einschließt. Ausführliche Behandlungen und Anwendungen der Intervallarithmetik sind in [111], [66], [5], [112], [6], [124], [16], [115] enthalten. In diesem Abschnitt werden nur einige Grundbegriffe zusammengestellt.

Die Menge aller abgeschlossenen Intervalle $A := [a] := [\underline{a}, \bar{a}]$ über den reellen Zahlen \mathbb{R} wird mit $I\mathbb{R}$ bezeichnet :

$$I\mathbb{R} := \{ [\underline{a}, \bar{a}] \mid \underline{a}, \bar{a} \in \mathbb{R}, \underline{a} \leq \bar{a} \}.$$

Sind die untere und obere Intervallgrenze identisch, $\underline{a} = \bar{a}$, so heißt $[a]$ ein Punktintervall. Da Intervalle Mengen sind, gelten die Relationen $=, \in, \subset, \subseteq, \supset, \supseteq$ und die Verknüpfungen

\cap, \cup wie in der Mengenlehre erklärt. Zusätzlich wird die Relation $\overset{\circ}{\subset}$ und die Verknüpfung *Intervallhülle* $\overset{\circ}{\llcorner}$ eingeführt :

$$\begin{aligned} [a] \overset{\circ}{\subset} [b] &: \iff \underline{b} < \underline{a} \wedge \bar{a} < \bar{b}. \\ [a] \overset{\circ}{\llcorner} [b] &:= [\min\{\underline{a}, \underline{b}\}, \max\{\bar{a}, \bar{b}\}]. \end{aligned}$$

Zwischen zwei Intervallen $[a]$ und $[b]$ wird der Abstand folgendermaßen definiert :

$$q([a], [b]) := \max\{ |\underline{a} \leftrightarrow \underline{b}|, |\bar{a} \leftrightarrow \bar{b}| \}.$$

Nach dieser Vereinbarung gilt $q([a], [b]) = 0$, genau dann wenn $[a] = [b]$ ist.

Der *Durchmesser* eines abgeschlossenen Intervalls $[a]$ ist durch die Intervallgrenzen erklärt

$$d([a]) := \bar{a} \leftrightarrow \underline{a}.$$

Für die Berechnung des Durchmessers ist insbesondere zu bemerken :

$$d([a] \pm [b]) = d([a]) + d([b]).$$

Stellt das Intervall $[a]$ eine Einschließung des exakten Wertes a dar, so bezeichnet $d([a])$ den *maximalen absoluten Fehler* der Näherungen $\tilde{a} \in [a]$. Der *Mittelpunkt* von $[a]$

$$m([a]) := \frac{\underline{a} + \bar{a}}{2}$$

besitzt dann einen absoluten Fehler von höchstens $\frac{d([a])}{2}$.

Mit Hilfe des *Betrags* und des *Betragsminimums* eines Intervalls $[a]$

$$\begin{aligned} |[a]| &:= \max\{ |a| \mid a \in [a] \} = \max\{ |\underline{a}|, |\bar{a}| \}, \\ \langle [a] \rangle &:= \min\{ |a| \mid a \in [a] \}, \end{aligned}$$

wird der *relative Fehler* $\frac{|\tilde{a} \leftrightarrow a|}{|a|}$, $\tilde{a} \in [a]$, abgeschätzt durch $\frac{|\tilde{a} \leftrightarrow a|}{|a|} \leq \frac{d([a])}{\langle [a] \rangle}$.

Für Intervallvektoren $A := ([a_i]) \in (IR)^n$ und Intervallmatrizen $A := ([a_{ij}]) \in (IR)^{m \times n}$ sind die oben erwähnten Begriffe *komponentenweise* erklärt.

2.2.1 Intervallrechnung über den endlichen Zahlen

Die arithmetischen Grundoperationen $\circ \in \{+, \Leftrightarrow, \cdot, /\}$ auf $I\mathbb{R}$ sind definiert durch

$$[a] \circ [b] := \{ a \circ b \mid a \in [a], b \in [b] \}, \quad 0 \notin [b] \quad \text{für } \circ = /. \quad (2.1)$$

Da die Grundoperationen $+, \Leftrightarrow, \cdot, /$ stetige Operatoren sind, ist die Verknüpfungsmenge $[a] \circ [b]$, $\circ \in \{+, \Leftrightarrow, \cdot, /\}$, wieder ein Intervall ([16]), d.h. $\circ : I\mathbb{R} \times I\mathbb{R} \rightarrow I\mathbb{R}$.

Aufgrund der Monotonie läßt sich das Intervall $[a] \circ [b]$ wie folgt angeben :

$$\begin{aligned} [a] + [b] &= [\underline{a} + \underline{b}, \bar{a} + \bar{b}], \\ [a] \Leftrightarrow [b] &= [\underline{a} \Leftrightarrow \bar{b}, \bar{a} \Leftrightarrow \underline{b}], \\ [a] \cdot [b] &= [\min\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}, \max\{\underline{a}\underline{b}, \underline{a}\bar{b}, \bar{a}\underline{b}, \bar{a}\bar{b}\}], \\ [a] / [b] &= [a] \cdot [1/\bar{b}, 1/\underline{b}], \quad 0 \notin [b]. \end{aligned}$$

Wie bei reellen Zahlen, gelten die Assoziativität und die Kommutativität bzgl. $+$ und \cdot auch für die Intervallrechnung. Das Punktintervall $[0] := 0$ ist das neutrale Element bzgl. $+$ und das Punktintervall $[1] := 1$ ist das Einselement bzgl. \cdot . Die additiven und multiplikativen inversen Elemente existieren allerdings nur für Punktintervalle. Für Nicht-Punktintervalle $[a] = [\underline{a}, \bar{a}]$, mit $\underline{a} \neq \bar{a}$, gelten

$$\begin{aligned} 0 = [0] &\not\subseteq [a] \Leftrightarrow [a] \quad \text{und} \\ 1 = [1] &\not\subseteq [a] / [a]. \end{aligned}$$

Ferner kann die Distributivität auch nur unter bestimmten Voraussetzungen erfüllt werden (vgl. [139], [123]). Für die Intervallverknüpfungen $+$ und \cdot gilt speziell die *Subdistributivität*

$$[a] \cdot ([b] + [c]) \subseteq [a] \cdot [b] + [a] \cdot [c].$$

Eine weitere Eigenschaft, die direkt aus der Definition (2.1) folgt, ist die *Inklusionsisotonie*

$$[a] \subseteq [c], [b] \subseteq [d] \quad \Rightarrow \quad [a] \circ [b] \subseteq [c] \circ [d], \quad \circ \in \{+, \Leftrightarrow, \cdot, /\}.$$

Diese Eigenschaft gilt auch für die intervallmäßige Auswertung von Funktionen, die im Abschnitt 2.3 besprochen wird.

Für Intervallmatrizen gelten die Kommutativität und die Assoziativität jedoch nur in Bezug auf die Addition. Wegen der Subdistributivität der Intervallrechnung ist die Multiplikation von Intervallmatrizen im allgemeinen nicht assoziativ

$$A \cdot (B \cdot C) \stackrel{i.a.}{\neq} (A \cdot B) \cdot C, \quad A, B, C \text{ Intervallmatrizen.}$$

Die Gleichheit ergibt sich, wenn z. B. A und C Matrizen von Punktintervallen sind ([103]).

2.2.2 Erweiterte Intervallrechnung

Im Abschnitt 2.2.1 wurde für die Intervalldivision $[a]/[b]$ vorausgesetzt, daß $0 \notin [b]$ ist. Bei konkreten Anwendungen könnte jedoch auch $0 \in [b]$ vorkommen. In diesem Fall ist $[a]/[b]$ eine Vereinigungsmenge von unendlichen Intervallen $(\Leftrightarrow\infty, \rho]$, $[\lambda, \infty)$ bzw. $(\Leftrightarrow\infty, \infty)$ mit $\rho, \lambda \in \mathbb{R}$. Durch Hinzunahme von ∞ und $\Leftrightarrow\infty$ wird \mathbb{R} auf $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\} \cup \{\Leftrightarrow\infty\}$ erweitert. Die erweiterte Division $[a]/[b] \in I\overline{\mathbb{R}}$ von zwei endlichen Intervallen $[a], [b] \in I\mathbb{R}$ ist dann definiert durch (vgl. [128])

$$[a]/[b] := \begin{cases} [a] \cdot [1/\bar{b}, 1/\underline{b}] & \text{für } 0 \notin [b] \\ [\underline{a}/\bar{b}, \infty] & \text{für } 0 < \underline{a} \text{ und } 0 = \underline{b} < \bar{b} \\ [\Leftrightarrow\infty, \underline{a}/\underline{b}] \cup [\underline{a}/\bar{b}, \infty] & \text{für } 0 < \underline{a} \text{ und } \underline{b} < 0 < \bar{b} \\ [\Leftrightarrow\infty, \underline{a}/\underline{b}] & \text{für } 0 < \underline{a} \text{ und } \underline{b} < \bar{b} = 0 \\ [\Leftrightarrow\infty, \bar{a}/\bar{b}] & \text{für } \bar{a} < 0 \text{ und } 0 = \underline{b} < \bar{b} \\ [\Leftrightarrow\infty, \bar{a}/\bar{b}] \cup [\bar{a}/\underline{b}, \infty] & \text{für } \bar{a} < 0 \text{ und } \underline{b} < 0 < \bar{b} \\ [\bar{a}/\underline{b}, \infty] & \text{für } \bar{a} < 0 \text{ und } \underline{b} < \bar{b} = 0 \\ \emptyset & \text{für } 0 \notin [a] \text{ und } 0 = [b] \\ [\Leftrightarrow\infty, \infty] & \text{für } 0 \in [a] \text{ und } 0 \in [b]. \end{cases} \quad (2.2)$$

Diese erweiterte Intervalldivision tritt beispielsweise bei Anwendung des Intervall-Newton-Verfahrens (s. Abschnitt 2.4.1)

$$[x]^{\{k+1\}} := \left(m([x]^{\{k\}}) \Leftrightarrow \frac{f(m([x]^{\{k\}}))}{f'([x]^{\{k\}})} \right) \cap [x]^{\{k\}} \quad (2.3)$$

auf, welches zur verifizierten Bestimmung der Nullstellen 1-dimensionaler reellwertiger Funktionen eingesetzt wird.

Seien nun $m := m([x]^{\{k\}})$ und $[q] := \frac{f(m([x]^{\{k\}}))}{f'([x]^{\{k\}})}$, so läßt sich die Differenz $m \Leftrightarrow [q]$ wie folgt berechnen :

$$m \Leftrightarrow [q] := \begin{cases} [m \Leftrightarrow \bar{q}, \infty] & \text{für } [q] = [\Leftrightarrow\infty, \bar{q}] \\ [m \Leftrightarrow \bar{q}, m \Leftrightarrow \underline{q}] & \text{für } [q] = [\underline{q}, \bar{q}] \\ [\Leftrightarrow\infty, m \Leftrightarrow \underline{q}] & \text{für } [q] = [\underline{q}, \infty] \\ [\Leftrightarrow\infty, m \Leftrightarrow q_2] \cup [m \Leftrightarrow q_1, \infty] & \text{für } [q] = [\Leftrightarrow\infty, q_1] \cup [q_2, \infty] \\ [\Leftrightarrow\infty, \infty] & \text{für } [q] = [\Leftrightarrow\infty, \infty]. \\ \emptyset & \text{für } [q] = \emptyset. \end{cases} \quad (2.4)$$

Es ist leicht zu prüfen, daß die Verknüpfungen (2.2) und (2.4) auf $I\overline{\mathbb{R}}$ der Inklusionsisotonie genügen.

2.3 Wertebereichseinschließung reeller Funktionen

In der Verifikationsnumerik spielen die Wertebereiche von Funktionen eine große Rolle. Beispielsweise wird in dem Intervall-Newton-Schritt (2.3) der Wertebereich $f'([x]^{\{k\}})$ der Ableitung f' benötigt. Da der exakte Wertebereich einer beliebigen Funktion im allgemeinen unbekannt ist, werden Einschließungen des Wertebereichs herangezogen, die mit Hilfe der Intervallarithmetik berechnet werden.

Für $n \geq 1$ sei $f = f(x_1, \dots, x_n) : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ eine n -dimensionale reellwertige Funktion und $X = (X_i) \in (I\mathbb{R})^n$ sei ein n -dimensionaler Intervallvektor mit $X \subseteq D$. Es stellt sich die Frage, wie man den Wertebereich von f bzgl. X

$$f(X) := \{ f(x) \mid x \in X \}$$

möglichst gut einschließen kann.

Eine intuitive Methode, eine Einschließung des Wertebereichs $f(X)$ zu finden, ist die direkte Anwendung der *Intervallerweiterung* $F : X \mapsto F(X) \in I\mathbb{R}$. Dies erfolgt durch die Substitution sämtlicher Variablen x_i in der Darstellung von f durch die Intervalle X_i sowie die anschließende intervallmäßige Auswertung. Für eine gegebene Funktion f ist die Intervallauswertung $F(X)$ nicht eindeutig, denn je nach der angewendeten Darstellung von f kann $F(X)$ zu unterschiedlichen Ergebnissen führen. Ist X ein Punktintervallvektor, nämlich $X = x$, dann gilt $F(x) = f(x)$.

Die Intervallerweiterung F eines arithmetischen Ausdrucks f , dessen Operationen ausschließlich aus den Grundoperationen $+$, \Leftrightarrow , \cdot , $/$ bestehen, ist *inklusionsisoton*

$$Y \subseteq X \Rightarrow F(Y) \subseteq F(X),$$

da die Grundoperatoren selbst inklusionsisoton sind (Abschnitt 2.2.1). Ferner sind die Funktionen f , deren Intervallerweiterungen gemäß $F(X) := f(X)$ bestimmt werden, auf natürliche Weise inklusionsisoton. Diese sind z. B. die elementaren Funktionen

$$f \in \{ \sin, \cos, \tan, \cot, \exp, \text{sqr}, \dots \}.$$

In [112] und [124] wurde gezeigt, daß für die inklusionsisotone Intervallerweiterung F gilt:

$$f(X) \subseteq F(X). \tag{2.5}$$

Mathematische Ausdrücke, die sich aus den Grundoperationen $+$, \Leftrightarrow , \cdot , $/$ und Funktionen f , deren Intervallerweiterungen F inklusionsisoton sind, zusammensetzen, erfüllen also stets die Einschließungseigenschaft (2.5).

Die Gleichheit in (2.5) gilt, falls jede Variable x_i in der Darstellung von f höchstens einmal auftritt ([124]). Für Polynome $p(x)$ der Gestalt $p(x) = \sum_{k=0}^n a_k x^k$ kann insbesondere die Wertebereichseinschließung der Ableitung $p'(x)$ durch Anwendung des Hornerchemas verbessert werden ([6], S. 32).

2.3.1 Die Konvergenz und die Mittelwertform

Es sei $D \subseteq \mathbb{R}^n$ und $X = (X_i) \in (I\mathbb{R})^n$ mit $X \subseteq D$. Ferner bezeichne $I(D)$ die Menge aller n -dimensionaler Intervalle $X \subseteq D$. Eine Intervallfunktion $F : I(D) \rightarrow I\mathbb{R}$ heißt *Lipschitz*, wenn eine Konstante $K \geq 0$ existiert, so daß die folgende *Lipschitzbedingung* erfüllt wird (vgl. [124], S. 25) :

$$d(F(X)) \leq K \cdot \max_{1 \leq i \leq n} \{d(X_i)\}, \quad K \geq 0. \quad (2.6)$$

In [124], S. 28, wurde gezeigt, daß der Wertebereich $f(X)$ einer reellwertigen Funktion $f : D \rightarrow \mathbb{R}$ die Lipschitzbedingung (2.6) erfüllt, falls f auf X Lipschitz ist, d.h. $\exists K \geq 0 : |f(x) - f(y)| \leq K \cdot \max_{1 \leq i \leq n} |x_i - y_i|, \forall x, y \in X$.

Ferner erfüllt die Intervallerweiterung F einer reellwertigen Funktionen f ebenfalls die Lipschitzbedingung (2.6), wenn sich f aus den Grundoperationen $+$, \leftrightarrow , \cdot , $/$ sowie den Funktionen g_ν , deren Intervallerweiterungen stetig und Lipschitz sind, zusammensetzt (vgl. [124], S. 26). Hierbei ist nur die normale Intervalldivision zulässig.

Die Lipschitzbedingung (2.6) besagt eine lineare Konvergenz der Intervallerweiterung $F(X)$ gegen den Wertebereich $f(X)$ (Beweis vgl. [6]) :

$$q(F(X), f(X)) \leq \gamma \cdot \max_{1 \leq i \leq n} \{d(X_i)\}, \quad \gamma \geq 0. \quad (2.7)$$

Eine weitere Einschließungsmöglichkeit, die zu einer quadratischen Konvergenz gegen den Wertebereich $f(X)$ führt, läßt sich realisieren durch den Einsatz von *zentrierten Formen*. Die allgemeine Definition zentrierter Formen findet man z. B. in [124]. Häufig wird die Mittelwertform betrachtet, die auf dem Mittelwertsatz basiert.

Für $X \in I\mathbb{R}$ ergibt sich durch Anwendung des Mittelwertsatzes die folgende Gleichung

$$f(x) = f(m(X)) + f'(\xi) \cdot (x \leftrightarrow m(X)), \quad \xi \in X.$$

Daraus resultiert die eindimensionale *Mittelwertform*

$$F^m(X) := F(m(X)) + F'(X) \cdot (X \leftrightarrow m(X)), \quad (2.8)$$

wobei F' die Intervallerweiterung der Ableitung f' bezeichnet. Für die Anwendung der Formel (2.8) wird vorausgesetzt, daß f auf X differenzierbar ist und $f'(X) \subseteq F'(X)$ ist.

Die direkte Erweiterung der Formel (2.8) auf mehrere Dimensionen führt zu

$$F^m(X) := F(m(X)) + \nabla F(X) \cdot (X \leftrightarrow m(X)). \quad (2.9)$$

Hierbei stellt ∇F die Intervallerweiterung des Gradienten $\nabla f(x) := (f_{x_1}(x), \dots, f_{x_n}(x))$ dar, mit $f_{x_i} := \frac{\partial f}{\partial x_i}$.

Mittels der Idee von Hansen [65] kann die Formel (2.9) durch sequentielle Anwendungen des Mittelwertsatzes auf jede Komponenten x_i verbessert werden. Es gilt nämlich

$$\begin{aligned}
f(x_1, \dots, x_n) &= f(x_1, \dots, x_{n-1}, m(X_n)) \\
&+ f_{x_n}(x_1, \dots, x_{n-1}, \xi_n) \cdot (x_n \Leftrightarrow m(X_n)) \\
&= f(x_1, \dots, x_{n-2}, m(X_{n-1}), m(X_n)) \\
&+ f_{x_{n-1}}(x_1, \dots, x_{n-2}, \xi_{n-1}, m(X_n)) \cdot (x_{n-2} \Leftrightarrow m(X_{n-2})) \\
&+ f_{x_n}(x_1, \dots, x_{n-1}, \xi_n) \cdot (x_n \Leftrightarrow m(X_n)) \\
&= \dots \\
&\vdots \\
&= f(m(X_1), \dots, m(X_n)) \\
&+ f_{x_1}(\xi_1, m(X_2), \dots, m(X_n)) \cdot (x_1 \Leftrightarrow m(X_1)) \\
&\vdots \\
&+ f_{x_n}(x_1, \dots, x_{n-1}, \xi_n) \cdot (x_n \Leftrightarrow m(X_n)).
\end{aligned} \tag{2.10}$$

Daraus resultiert die *verbesserte Mittelwertform*

$$\hat{F}^m(X) := F(m(X)) + (F_{x_1}(X^{(1)}), \dots, F_{x_n}(X^{(n)})) \cdot (X \Leftrightarrow m(X)). \tag{2.11}$$

Hierbei bezeichnet $F_{x_i}(X^{(i)})$ die Intervallerweiterung der folgenden Funktion :

$$f_{x_i}(x^{(i)}) := \frac{\partial}{\partial x_i} f(x_1, \dots, x_i, m(X_{i+1}), \dots, m(X_n)). \tag{2.12}$$

Die Voraussetzungen für die Anwendung von (2.9) und (2.11) sind ebenfalls die Existenz von f_{x_i} und die Einschließungseigenschaft $f_{x_i}(X) \subseteq F_{x_i}(X)$. Unter diesen Voraussetzungen werden die folgenden Zusammenhänge sichergestellt :

$$f(X) \subseteq F^m(X) \quad \text{bzw.} \quad f(X) \subseteq \hat{F}^m(X).$$

Durch Einführung der Mittelpunkte $m(X_j)$ in den Argumenten der partiellen Ableitungen f_{x_i} stellt die Formel (2.11) eine Teilmenge von (2.9) dar.

Genügen die Intervallerweiterungen $F'(X)$, $F_{x_i}(X)$ bzw. $F_{x_i}(X^{(i)})$ der Lipschitzbedingung (2.6), dann konvergieren die Mittelwertformen (2.8), (2.9) bzw. (2.11) quadratisch gegen den Wertebereich $f(X)$ ([124], S. 66, 71) :

$$q(\hat{F}^m(X), f(X)) \leq \beta \cdot \max_{1 \leq i \leq n} \{d(X_i)^2\}, \quad \beta \geq 0. \tag{2.13}$$

Für Intervallvektoren X mit kleinem Durchmesser, z. B. $d(X_j) < 1, \forall j$, sind die Mittelwertformen $F^m(X)$ und $\hat{F}^m(X)$ vorteilhafter als die Intervallerweiterung $F(X)$. Dabei ist die Obergrenze 1 eine erfahrungsmäßige Abschätzung aber kein absolutes Maß.

2.3.2 Einschließung durch Verfeinerung

Die Einschließungen $F(X)$, $F^m(X)$ und $\hat{F}^m(X)$ des Wertebereichs $f(X)$ können durch die Verfeinerung des Intervalls X weiter verbessert werden.

Sei $X = \bigcup_{\nu=1}^L X^\nu$, $X^\nu \in (IR)^n$, eine beliebige Zerlegung von X . Wegen $f(X^\nu) \subseteq F(X^\nu)$ gilt

$$f(X) \subseteq \bigcup_{\nu=1}^L F(X^\nu). \quad (2.14)$$

Tritt jede Variable x_i in der Darstellung von f höchstens einmal auf, so folgt aus $f(X^\nu) = F(X^\nu)$:

$$f(X) = \bigcup_{\nu=1}^L F(X^\nu). \quad (2.15)$$

Die Eigenschaft (2.14) gilt auch für die Mittelwertformen F^m und \hat{F}^m unter den gleichen Voraussetzungen.

Darüberhinaus können die Konvergenzeigenschaften (2.7) und (2.13) auch durch die Verfeinerung von X beibehalten werden (vgl. [107]):

$$\begin{aligned} q(f(X), \bigcup_{\nu=1}^L F(X^\nu)) &\leq \gamma \cdot \max_{i,\nu} \{d(X_i^\nu)\}, \\ q(f(X), \bigcup_{\nu=1}^L F^m(X^\nu)) &\leq \beta \cdot \max_{i,\nu} \{d(X_i^\nu)^2\}, \\ q(f(X), \bigcup_{\nu=1}^L \hat{F}^m(X^\nu)) &\leq \alpha \cdot \max_{i,\nu} \{d(X_i^\nu)^2\}, \end{aligned} \quad (2.16)$$

da diese Eigenschaften ebenfalls für jeden Teilbereich X^ν gültig sind.

Die Ungleichungen in (2.16) deuten an, daß die Einschließungen $\bigcup F(X^\nu)$, $\bigcup F^m(X^\nu)$ und $\bigcup \hat{F}^m(X^\nu)$ um so besser sind, je feiner das Intervall X zerlegt wird.

2.4 Nullstellenbestimmung einer reellwertigen Funktion

In den Abschnitten 3.4.3 und 4.3.4 wird gezeigt, daß sich die Restgliedfaktoren der Quadratur- und Kubaturformeln mit Hilfe von Nullstellen der Kernfunktionen verifiziert berechnen lassen. Die Kernfunktionen sind ein- und mehrdimensionale reellwertige Funktionen. Zur Bestimmung ihrer Nullstellen wird das gemäß (2.10) verallgemeinerte Intervall-Newton-Verfahren eingesetzt. Im folgenden wird zunächst der eindimensionale Fall betrachtet und danach die Verallgemeinerung auf mehrere Dimensionen.

2.4.1 Erweitertes Intervall-Newton-Verfahren

Sei $f : D \subseteq \mathbb{R} \rightarrow \mathbb{R}$ auf D stetig differenzierbar. Ferner sei das Intervall $X \in I\mathbb{R}$ eine Teilmenge von D . Ist $x^* \in X$ eine Nullstelle von f , dann gilt laut Mittelwertsatz

$$0 = f(x^*) = f(m(X)) + f'(\xi) \cdot (x^* - m(X)), \quad \xi \in X.$$

Daraus folgt die nächste Nullstelleneinschließung (vgl. [61]) :

$$\begin{aligned} x^* &= m(X) \Leftrightarrow \frac{f(m(X))}{f'(\xi)} \\ &\in m(X) \Leftrightarrow \frac{f(m(X))}{f'(X)} \left(\subseteq m(X) \Leftrightarrow \frac{F(m(X))}{F'(X)} \right) =: \mathcal{N}(X). \end{aligned}$$

Hierbei ist $0 \in f'(X)$ zugelassen.

Um die Nullstellen von f in X zu bestimmen, fängt das Intervall-Newton-Verfahren mit dem Startintervall $X^{\{0\}} := X$ an und der *Intervall-Newton-Schritt*

$$X^{\{k+1\}} := X^{\{k\}} \cap \mathcal{N}(X^{\{k\}}), \quad k = 0, 1, 2, \dots \quad (2.17)$$

wiederholt sich solange bis die Schnittmenge $X^{\{k+1\}}$ die Bedingung $d(X^{\{k+1\}}) \leq \varepsilon$ erfüllt, wobei ε eine vorgegebene Schranke ist. Stellt $\mathcal{N}(X)$ wegen $0 \in f'(X)$ eine Vereinigungsmenge $\mathcal{N}(X) = \mathcal{N}_1 \cup \mathcal{N}_2$ dar, mit $\mathcal{N}_1, \mathcal{N}_2 \in I\overline{\mathbb{R}}$, so wird das Intervall-Newton-Verfahren jeweils für die Teilbereiche $X \cap \mathcal{N}_1$ und $X \cap \mathcal{N}_2$ fortgeführt.

Durch die Schnittbildung (2.17) wird das Intervall-Newton-Verfahren nie divergieren und somit können alle Nullstellen, die in X liegen, in die Teilbereiche von X einbezogen werden. Die Existenz der Nullstellen von f in X führt zu $X^{\{k\}} \cap \mathcal{N}(X^{\{k\}}) \neq \emptyset$, d. h., die Funktion f besitzt in X keine Nullstelle, wenn $X^{\{k\}} \cap \mathcal{N}(X^{\{k\}}) = \emptyset$ ist. Ferner wird die Existenz einer eindeutiger Nullstelle in $X^{\{k\}}$ durch $\mathcal{N}(X^{\{k\}}) \neq \emptyset$ und $\mathcal{N}(X^{\{k\}}) \overset{\circ}{\subset} X^{\{k\}}$ festgestellt.

2.4.2 Die Methode von Hansen/Neumaier

Das erweiterte Intervall-Newton-Verfahren kann auf die mehrdimensionalen reellwertigen Funktionen $f = f(x_1, \dots, x_n) : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ verallgemeinert werden. Die Idee besteht darin, daß der Mittelwertsatz sequentiell auf jede Komponente x_i anzuwenden ist. Diese Idee wurde erstmals von Hansen [65] 1968 für die Lösung des nichtlinearen Gleichungssystems $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ entwickelt und später hat Neumaier [114] 1988 diese Anwendung für das allgemeinere Gleichungssystem $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $n \geq m$, ausgebaut. Im folgenden beschränkt sich die Betrachtung auf die reellwertigen Funktionen $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

Es sei f auf D stetig differenzierbar und $X = (X_i) \in (I\mathbb{R})^n$ mit $X \subseteq D$. Gemäß (2.10) gilt für die Nullstelle $x^* \in X$ von f

$$\begin{aligned}
0 &= f(x_1^*, \dots, x_n^*) = f(m(X_1), \dots, m(X_n)) \\
&\quad + f_{x_1}(\xi_1, m(X_2), \dots, m(X_n)) \cdot (x_1^* \Leftrightarrow m(X_1)) \\
&\quad \vdots \\
&\quad + f_{x_n}(x_1^*, \dots, x_{n-1}^*, \xi_n) \cdot (x_n^* \Leftrightarrow m(X_n)).
\end{aligned}$$

Mit $x_*^{(k)} := (x_1^*, \dots, x_{k-1}^*, \xi_k, m(X_{k+1}), \dots, m(X_n))$ läßt sich die i -te Komponente der Nullstelle x^* durch

$$x_i^* = m(X_i) \Leftrightarrow \frac{f(m(X)) + \sum_{j \neq i} f_{x_j}(x_*^{(j)}) \cdot (x_j^* \Leftrightarrow m(X_j))}{f_{x_i}(x_*^{(i)})} \quad (2.18)$$

formulieren. Sei $X^{(k)} := (X_1, \dots, X_k, m(X_{k+1}), \dots, m(X_n))$. Ersetzt man in (2.18) die unbekanntenen Komponenten x_ν^* und die unbekanntenen Zwischenstellen ξ_ν durch die einschließenden Intervalle X_ν , so ergibt sich wegen der Teilmengeneigenschaft:

$$x_i^* \in m(X_i) \Leftrightarrow \frac{f(m(X)) + \sum_{j \neq i} f_{x_j}(X^{(j)}) \cdot (X_j \Leftrightarrow m(X_j))}{f_{x_i}(X^{(i)})} =: \mathcal{N}_i(X). \quad (2.19)$$

Hierbei ist ebenfalls $0 \in f_{x_i}(X^{(i)})$ zulässig. In diesem Fall stellt $\mathcal{N}_i(X)$ die Vereinigungsmenge zweier Intervalle $\mathcal{N}_{i1}, \mathcal{N}_{i2} \in I\overline{\mathbb{R}}$ dar, $\mathcal{N}_i(X) = \mathcal{N}_{i1} \cup \mathcal{N}_{i2}$.

Soll f in X eine Nullstelle $x^* \in X$ haben, so muß gelten

$$x^* \in \mathcal{N}(X) \cap X := (\mathcal{N}_i(X)) \cap (X_i) = (\mathcal{N}_i(X) \cap X_i).$$

Existiert hingegen ein j , so daß $\mathcal{N}_j(X) \cap X_j = \emptyset$ gilt, dann besitzt f in X keine Nullstelle. Die Eindeutigkeit der Lösungsmannigfaltigkeit $0 = f(x)$ wird durch $\mathcal{N}_i(X) \neq \emptyset$ und $\mathcal{N}_i(X) \overset{\circ}{\subset} X_i$, für $i = 1(1)n$, identifiziert (Beweis siehe [115]).

Bei intervallmäßiger Auswertung kann $\mathcal{N}_i(X)$ verbessert werden, falls $d(X_j)$, $1 \leq j \leq n$, klein genug sind und sich $f_{x_{kj}}$ direkt formelmäßig auswerten lassen. Es sei $f_{x_k}(x^{(k)}) := f_{x_k}(x_1, \dots, x_k, m(X_{k+1}), \dots, m(X_n))$ und $\xi^{(k)} := (x_1, \dots, x_{k-1}, \xi_k, m(X_{k+1}), \dots, m(X_n))$. Unter sequentieller Anwendung des Mittelwertsatzes jeweils auf x_j , $j = k(\Leftrightarrow 1)1$, ergibt sich

$$\begin{aligned}
f_{x_k}(x^{(k)}) &= f_{x_k}(m(X)) + f_{x_{k1}}(\xi^{(1)}) \cdot (x_1 \Leftrightarrow m(X_1)) + \dots + f_{x_{kk}}(\xi^{(k)}) \cdot (x_k \Leftrightarrow m(X_k)) \\
&\in F_{x_k}(m(X)) + F_{x_{k1}}(X^{(1)}) \cdot (X_1 \Leftrightarrow m(X_1)) + \dots + F_{x_{kk}}(X^{(k)}) \cdot (X_k \Leftrightarrow m(X_k)) \\
&=: \hat{F}_{x_k}^m(X^{(k)}).
\end{aligned} \quad (2.20)$$

Die Voraussetzungen für (2.20) sind die Existenz von $f_{x_{kj}}(x^{(j)}) := \frac{\partial^2 f}{\partial x_k \partial x_j}(x^{(j)})$ und $f_{x_{kj}}(X^{(j)}) \subseteq F_{kj}(X^{(j)})$, $j = 1(1)k$. Genügen $F_{x_{kj}}(X^{(j)})$ der Lipschitzbedingung, dann konvergiert $\hat{F}_{x_k}^m(X^{(k)})$ quadratisch gegen $f_{x_k}(X^{(k)})$. Im Eindimensionalen ist (2.20) äquivalent zu

$$\hat{F}_{x_1}^m(X^{(1)}) = (F')^m(X) := F'(m(X)) + F''(X) \cdot (X \Leftrightarrow m(X)), \quad (2.21)$$

wobei F' und F'' die Intervallerweiterungen von f' bzw. f'' darstellen.

Sei ε eine vorgegebene Schranke. Die Einschließungen sämtlicher Nullstellen von f in X können durch die folgenden Iterationsschritte verfeinert werden (vgl. auch [72]).

Algorithmus 2.1 : VerallgemeinXIN($f, X, NListe, \varepsilon$)

1. $X_{old} := X$.
2. **if** $\max_{1 \leq i \leq n} d(X_i) < 1$ **then** $W := \hat{F}^m(X)$ **else** $W := F(X)$.
3. **if** $0 \notin W$ **then** **exit**.
4. **if** $\max_{1 \leq i \leq n} d(X_i) \leq \varepsilon$ **then** $X \rightarrow NListe$; **exit**.
5. **for** $i = 1$ **to** n **do**
 - if** $\max_{1 \leq j \leq i} d(X_j) < 1$ **then** $W := \hat{F}_i^m(X^{(i)})$ **else** $W := F_i(X^{(i)})$.
 - if** $(0 \notin W)$ **and** $(d(X_i) > \varepsilon)$ **then** Berechne $\mathcal{N}_i(X)$ gemäß (2.19);
 $X_i := X_i \cap \mathcal{N}_i(X)$;
if $X_i = \emptyset$ **then** **exit**.
6. **for** $i = 1$ **to** n **do**
 - if** $\max_{1 \leq j \leq i} d(X_j) < 1$ **then** $W := \hat{F}_i^m(X^{(i)})$ **else** $W := F_i(X^{(i)})$.
 - if** $(0 \in W)$ **and** $(d(X_i) > \varepsilon)$ **then**
 - Berechne $\mathcal{N}_i(X) := \mathcal{N}_{i1} \cup \mathcal{N}_{i2}$ gemäß (2.19);
 - if** $\mathcal{N}_i(X) \neq [\Leftrightarrow \infty, \infty]$ **then**
 - $X_{i1} := X_i \cap \mathcal{N}_{i1}$; $X_{i2} := X_i \cap \mathcal{N}_{i2}$;
 - if** $X_{i1} = \emptyset$ **and** $X_{i2} = \emptyset$ **then** **exit**;
 - if** $X_{i1} \neq \emptyset$ **and** $X_{i2} \neq \emptyset$ **then** $X_i := X_{i2}$;
 - VerallgemeinXIN($f, X, NListe, \varepsilon$);
 - $X_i := X_{i1}$
 - else** $X_i := X_{i1} \cup X_{i2}$.
7. **if** $X_{old} \neq X$ **then** VerallgemeinXIN($f, X, NListe, \varepsilon$)
else Bisektion(X, X_1, X_2);
VerallgemeinXIN($f, X_1, NListe, \varepsilon$);
VerallgemeinXIN($f, X_2, NListe, \varepsilon$).

Die Laufzeit kann dadurch optimiert werden, daß in den Schritten 5 und 6 die Wertebereichseinschließung W erst dann erneut berechnet wird, wenn diese noch nicht vorhanden ist bzw. wenn sich X verändert hat. Für die Unterteilungsrichtung der Bisektion(X, X_1, X_2) im Schritt 7 sind in [126] verschiedene Auswahlkriterien diskutiert.

Für $n = 1$ reduziert sich der Algorithmus 2.1 auf das eindimensionale erweiterte Intervall-Newton-Verfahren.

2.5 Differentiationsarithmetik

Die (partiellen) Ableitungen reellwertiger Funktionen kommen oft bei der Behandlung von numerischen Problemen vor. Z. B. sind in der Mittelwertform, Abschnitt 2.3.1, und bei dem Intervall-Newton-Verfahren, Abschnitt 2.4, die (partiellen) Ableitungen erster bzw. zweiter Ordnung zu berücksichtigen.

Für die Aufgaben, die speziell auf bestimmte Funktionsgestalten zugeschnitten sind, können die Funktionswerte der Ableitungen formelmäßig berechnet werden, falls nur Ableitungen von gewissen Ordnungen zu betrachten sind und die Formeln der Ableitungen ohne Problematik hergeleitet werden können. Beispielsweise ist es für die Bestimmung der Fehlerkonstanten von numerischen Integrationsformeln zweckmäßig, die Ableitungen der Kernfunktionen formelmäßig auszuwerten (s. Abschnitt 3.4.2). Im Falle, daß Funktionen von beliebiger Gestalt betrachtet werden sollen, bzw. die Ordnung der benötigten Ableitung hoch ist oder variiert, ist es nicht mehr einfach, die Ableitungen formelmäßig auszuwerten. Dies ist z. B. bei der Fehlerabschätzung zur numerischen Quadratur und Kubatur (s. Abschnitte 3.4.1 und 4.3) der Fall. Für diese Aufgaben wurden die analytisch hergeleiteten Darstellungen der Quadratur- und Kubaturfehler in der Praxis nie angewendet, bis die Differentiationsarithmetik in Kombination mit der Intervallarithmetik eingesetzt wurde (vgl. [111], [38]).

Bei der Differentiationsarithmetik, die auch *automatische Differentiation* genannt wird, werden die Taylorkoeffizienten berechnet. D. h. die Werte der Ableitungen werden direkt berechnet. Daher ist die Differentiationsarithmetik effizienter als die symbolische Differentiation und zuverlässiger als die numerische Differentiation. Denn bei der symbolischen Differentiation müssen zuerst die expliziten Darstellungen der Ableitungen berechnet werden, und bei der numerischen Differentiation können nur Näherungen der Ableitungen bestimmt werden.

Im folgenden werden die Rechenregeln für die ein- und zweidimensionalen Taylorkoeffizienten zusammengestellt bzw. hergeleitet. Diese Diskussion beruht auf der sogenannten Vorwärtsmethode (vgl. [111]).

2.5.1 Eindimensionale Taylorkoeffizienten

Die rekursive Berechnung der Taylorkoeffizienten eindimensionaler reellwertiger Funktionen wurde bereits von Gibbons 1960 und Moore [111] 1966 in Programme implementiert (vgl. [60], [87]). In diesem Abschnitt werden ausschließlich einige wichtige Eigenschaften und Resultate zusammengestellt. Um die Division durch Null bzw. durch ein Intervall, das Null enthält, zu vermeiden, wird aus heuristischer Überlegung eine Modifikation der Rechenregeln der Taylorkoeffizienten für die Potenzfunktion f^a angegeben.

Ist $f : \mathbb{R} \rightarrow \mathbb{R}$ in einer Umgebung von t_0 analytisch, so läßt sich f als Taylor-Reihe

$$f(t) = \sum_{k=0}^{\infty} \frac{f^{(k)}(t_0)}{k!} (t \Leftrightarrow t_0)^k \quad (2.22)$$

darstellen. Für die Taylorkoeffizienten $\frac{f^{(k)}(t_0)}{k!}$ wird häufig die Abkürzung $(f)_k$ verwendet. Es gilt

$$(f')_k = \frac{f^{(k+1)}(t_0)}{k!} = (k+1) \cdot (f)_{k+1}. \quad (2.23)$$

Gemäß (2.22) lassen sich die folgenden Formeln für konstante Funktion $f(t) = c$ und die unbekannte $f(t) = t$ direkt angeben :

$$(c)_0 = c, \quad (c)_k = 0 \quad \text{für } k \geq 1, \quad (2.24)$$

$$(t)_0 = t_0, \quad (t)_1 = 1 \quad (t)_k = 0 \quad \text{für } k \geq 2. \quad (2.25)$$

Für die Verknüpfungen zweier Funktionen $f(t)$ und $g(t)$ folgen die nachstehenden Formeln ebenfalls direkt aus (2.22).

$$(f \pm g)_k = (f)_k \pm (g)_k \quad (2.26)$$

$$(f \cdot g)_k = \sum_{j=0}^k (f)_j \cdot (g)_{k-j} \left(= \sum_{j=0}^k (f)_{k-j} \cdot (g)_j \right). \quad (2.27)$$

Sei $h = f/g$. Dann kann die rekursive Formel für die Division durch Anwendung der Rechenregel (2.27) auf $h \cdot g = f$ erhalten werden :

$$(f/g)_k = \frac{1}{(g)_0} \left((f)_k \Leftrightarrow \sum_{j=1}^k (g)_j \cdot \left(\frac{f}{g} \right)_{k-j} \right). \quad (2.28)$$

Die rekursiven Formeln der Taylorkoeffizienten für die Quadratwurzel \sqrt{f} und die Quadratfunktion f^2 lassen sich ebenfalls durch Anwendung der Rechenregel (2.27) auf $\sqrt{f} \cdot \sqrt{f} = f$ bzw. $f \cdot f = f^2$ herleiten. Für $k \geq 1$ gilt

$$(\sqrt{f})_k = \begin{cases} \frac{1}{2(\sqrt{f})_0} \left((f)_k \Leftrightarrow 2 \sum_{j=1}^{\frac{k-1}{2}} (\sqrt{f})_j (\sqrt{f})_{k-j} \right), & k \text{ ungerade} \\ \frac{1}{2(\sqrt{f})_0} \left((f)_k \Leftrightarrow 2 \sum_{j=1}^{\frac{k-2}{2}} (\sqrt{f})_j (\sqrt{f})_{k-j} \Leftrightarrow (\sqrt{f})_{k/2}^2 \right), & k \text{ gerade} \end{cases} \quad (2.29)$$

bzw.

$$(f^2)_k = \begin{cases} 2 \sum_{j=0}^{\frac{k-1}{2}} (f)_j (f)_{k-j}, & k \text{ ungerade} \\ 2 \sum_{j=0}^{\frac{k-2}{2}} (f)_j (f)_{k-j} + (f)_{k/2}^2, & k \text{ gerade} \end{cases} \quad (2.30)$$

mit $(f^2)_0 = (f)_0^2$ und $(\sqrt{f})_0 = \sqrt{(f)_0}$.

Diese Technik gilt auch für die Potenzfunktion $f^a =: g$. Unter Berücksichtigung von $g' f = a f^a f'$ und (2.23) ergibt sich

$$(f^a)_k = \frac{1}{k (f)_0} \sum_{j=0}^{k-1} (a(k \Leftrightarrow j) \Leftrightarrow j) (f^a)_j (f)_{k-j}, \quad k \geq 1, \quad (2.31)$$

wobei $(f^a)_0 = (f)_0^a$ ist.

Bei Anwendung der Kettenregel, des Zusammenhangs (2.23) und der Rechenregel (2.27) können noch weitere rekursive Formeln für verschiedene Standardfunktionen g hergeleitet werden. Beginnend mit $(g(f))_0 = g((f)_0)$ gelten z. B. für $k \geq 1$

$$(e^f)_k = \frac{1}{k} \sum_{j=0}^{k-1} (k \Leftrightarrow j) (e^f)_j (f)_{k-j}, \quad (2.32)$$

$$(\sin f)_k = \frac{1}{k} \sum_{j=0}^{k-1} (k \Leftrightarrow j) (\cos f)_j (f)_{k-j}, \quad (2.33)$$

$$(\cos f)_k = \Leftrightarrow \frac{1}{k} \sum_{j=0}^{k-1} (k \Leftrightarrow j) (\sin f)_j (f)_{k-j}, \quad (2.34)$$

$$(\sinh f)_k = \frac{1}{k} \sum_{j=0}^{k-1} (k \Leftrightarrow j) (\cosh f)_j (f)_{k-j}, \quad (2.35)$$

$$(\cosh f)_k = \frac{1}{k} \sum_{j=0}^{k-1} (k \Leftrightarrow j) (\sinh f)_j (f)_{k-j}. \quad (2.36)$$

Die rekursiven Formeln der Taylorkoeffizienten für weitere Standardfunktionen sind in [87] zusammengestellt.

In der Formel (2.31) wird der k -te Taylorkoeffizient $(f^a)_k$ durch die Division durch $(f)_0$ erzielt. Gilt $(f)_0 = 0$ bzw. $0 \in f([t_0])$, so ist es unmöglich, $(f^a)_k$, $k \geq 1$, zu bestimmen. Sei $w_0 := 1$. Für $a \geq k$ kann dieses Problem vermieden werden. Gemäß (2.31) gilt

$$(f^a)_1 = \frac{1}{(f)_0} a (f^a)_0 (f)_1 = (f)_0^{a-1} (a(f)_1) =: (f)_0^{a-1} \cdot w_1,$$

$$\begin{aligned} (f^a)_2 &= \frac{1}{2(f)_0} \{2a (f^a)_0 (f)_2 + (a \Leftrightarrow 1) (f^a)_1 (f)_1\} \\ &= \frac{(f)_0^{a-2}}{2} \{2a (f)_0 w_0 (f)_2 + (a \Leftrightarrow 1) w_1 (f)_1\} =: (f)_0^{a-2} \cdot w_2, \end{aligned}$$

$$\begin{aligned} (f^a)_3 &= \frac{1}{3(f)_0} \{3a (f^a)_0 (f)_3 + (2a \Leftrightarrow 1) (f^a)_1 (f)_2 + (a \Leftrightarrow 2) (f^a)_2 (f)_1\} \\ &= \frac{(f)_0^{a-3}}{3} \{3a (f)_0^2 w_0 (f)_3 + (2a \Leftrightarrow 1) (f)_0 w_1 (f)_2 + (a \Leftrightarrow 2) w_2 (f)_1\} \\ &=: (f)_0^{a-3} \cdot w_3, \\ &\vdots \end{aligned}$$

Durch Einführung von w_k (siehe auch [115], S. 46f) mit

$$\begin{cases} w_0 & := 1, \\ w_k & := \frac{1}{k} \sum_{j=0}^{k-1} (a(k \Leftrightarrow j) \Leftrightarrow j) (f)_0^{k-j-1} w_j (f)_{k-j}, \quad k \geq 1 \end{cases} \quad (2.37)$$

läßt sich $(f^a)_k$ wie folgt berechnen :

$$(f^a)_k = (f)_0^{a-k} \cdot w_k. \quad (2.38)$$

2.5.2 Zweidimensionale Taylorkoeffizienten

Beruhend auf der Idee von Moore werden in diesem Abschnitt die rekursiven Formeln der zweidimensionalen Taylorkoeffizienten hergeleitet (siehe auch [83], [85]).

Es sei $D_1 := \frac{\partial}{\partial x}$ und $D_2 := \frac{\partial}{\partial y}$. Für die Funktionen f , die in einer Umgebung von (x_0, y_0) analytisch ist, gilt

$$\begin{aligned} f(x, y) &= \sum_{k=0}^{\infty} \frac{1}{k!} (h_1 D_1 + h_2 D_2)^k f(x_0, y_0), \quad h_1 := x \Leftrightarrow x_0, \quad h_2 := y \Leftrightarrow y_0 \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} \left(\sum_{i=0}^k \binom{k}{i} (h_1 D_1)^i (h_2 D_2)^{k-i} \right) f(x_0, y_0) \\ &= \sum_{k=0}^{\infty} \sum_{i=0}^k \frac{f^{(i, k-i)}(x_0, y_0)}{i! (k \Leftrightarrow i)!} (x \Leftrightarrow x_0)^i (y \Leftrightarrow y_0)^{k-i} \\ &= \sum_{k=0}^{\infty} \sum_{i=0}^k (f)_{i, k-i} (x \Leftrightarrow x_0)^i (y \Leftrightarrow y_0)^{k-i}. \end{aligned} \quad (2.39)$$

Dabei bezeichnet $(f)_{i, k-i} := \frac{f^{(i, k-i)}(x_0, y_0)}{i! (k \Leftrightarrow i)!}$ den $(i, k \Leftrightarrow i)$ -ten Taylorkoeffizienten von f an der Stelle (x_0, y_0) .

Wenn nicht anders ausdrücklich gemacht ist, seien im Rest dieses Abschnittes $0 \leq k$ und $0 \leq i \leq k$.

Ist f eine konstante Funktion $f(x, y) = c$, so gilt

$$(f)_{0,0} = c, \quad (f)_{i, k-i} = 0 \quad \text{für } k \geq 1. \quad (2.40)$$

Für die Unbekannten $f(x, y) = x$ und $g(x, y) = y$ lassen sich die Taylorkoeffizienten folgendermaßen angeben :

$$(f)_{0,0} = x_0, \quad (f)_{1,0} = 1, \quad (f)_{0,1} = 0, \quad (f)_{i, k-i} = 0 \quad \text{für } k \geq 2, \quad (2.41)$$

bzw.

$$(g)_{0,0} = y_0, \quad (g)_{1,0} = 0, \quad (g)_{0,1} = 1, \quad (g)_{i,k-i} = 0 \quad \text{für } k \geq 2. \quad (2.42)$$

Es seien nun $f(x, y)$ und $g(x, y)$ zwei beliebige Funktionen. Die nachstehenden Rechenregeln folgen direkt aus der Taylor-Reihe (2.39)

$$(f \pm g)_{i,k-i} = (f)_{i,k-i} \pm (g)_{i,k-i} \quad (2.43)$$

$$(f \cdot g)_{i,k-i} = \sum_{l=0}^i \sum_{m=0}^{k-i} (f)_{l,m} \cdot (g)_{i-l,k-i-m} \left(= \sum_{l=0}^i \sum_{m=0}^{k-i} (f)_{i-l,k-i-m} \cdot (g)_{l,m} \right). \quad (2.44)$$

Für Division $h = f/g$ werden die Taylorkoeffizienten unter Verwendung der Rechenregel (2.44) auf $f = h \cdot g$ mit Hilfe der Formel

$$(f)_{i,k-i} = \sum_{l=0}^i \sum_{m=0}^{k-i} (g)_{l,m} (h)_{i-l,k-i-m} = (h)_{i,k-i} (g)_{0,0} + \sum_{l=0}^i \sum_{m=0}^{k-i} (g)_{l,m} (h)_{i-l,k-i-m}$$

hergeleitet :

$$(f/g)_{i,k-i} = \frac{1}{(g)_{0,0}} \left((f)_{i,k-i} \Leftrightarrow \sum_{l=0}^i \sum_{m=0}^{k-i} (g)_{l,m} \cdot \left(\frac{f}{g} \right)_{i-l,k-i-m} \right), \quad (2.45)$$

wobei \sum^* bedeutet, daß der Term mit dem Index $(l, m) = (0, 0)$ nicht aufsummiert wird.

Die Taylorkoeffizienten der Quadratfunktion $f(x, y)^2$ lassen sich ebenfalls mit Hilfe der Rechenregel (2.44) herleiten. Für $k \geq 1$ gilt :

$$(f^2)_{i,k-i} = \begin{cases} 2 \sum_{l=0}^{\frac{i-1}{2}} \sum_{m=0}^{k-i} (f)_{l,m} (f)_{i-l,k-i-m}, & i \text{ ungerade,} \\ 2 \sum_{l=0}^{\frac{i-2}{2}} \sum_{m=0}^{k-i} (f)_{l,m} (f)_{i-l,k-i-m} + 2 \sum_{m=0}^{\frac{k-i-1}{2}} (f)_{i/2,m} (f)_{i/2,k-i-m}, & i \text{ gerade, } k \Leftrightarrow i \text{ ungerade,} \\ 2 \sum_{l=0}^{\frac{i-2}{2}} \sum_{m=0}^{k-i} (f)_{l,m} (f)_{i-l,k-i-m} + 2 \sum_{m=0}^{\frac{k-i-2}{2}} (f)_{i/2,m} (f)_{i/2,k-i-m} \\ + (f)_{i/2,(k-i)/2} (f)_{i/2,(k-i)/2}, & i \text{ gerade, } k \Leftrightarrow i \text{ gerade.} \end{cases} \quad (2.46)$$

Durch Anwendung der Rechenvorschrift (2.44) können rekursive Formeln der Taylorkoeffizienten für weitere rationale Funktionen $f(x, y)$ hergeleitet werden. Für die Quadratwurzel $\sqrt{f(x, y)}$, $f(x, y) \geq 0$ ergibt sich, wegen $\sqrt{f} \cdot \sqrt{f} = f$,

$$(\sqrt{f})_{i,k-i} = \frac{1}{2(\sqrt{f})_{0,0}} \left((f)_{i,k-i} \Leftrightarrow \sum_{l=0}^i \sum_{m=0}^{k-i} (\sqrt{f})_{l,m} (\sqrt{f})_{i-l,k-i-m} \right), \quad k \geq 1.$$

Hierbei bedeutet \sum^* , daß die Terme mit den Indizes $(l, m) = (0, 0)$ und $(l, m) = (i, k \Leftrightarrow i)$ nicht aufsummiert werden. Analog zu der Quadratfunktion besitzt diese Formel auch effizientere Darstellungen :

- i ungerade

$$(\sqrt{f})_{i,k-i} = \frac{1}{2(\sqrt{f})_{0,0}} \left((f)_{i,k-i} \Leftrightarrow 2 \sum_{l=0}^{\frac{i-1}{2}} \sum_{m=0}^{k-i} (\sqrt{f})_{l,m} (\sqrt{f})_{i-l,k-i-m} \right), \quad (2.47)$$

- i gerade, $k \Leftrightarrow i$ ungerade

$$\begin{aligned} (\sqrt{f})_{i,k-i} &= \frac{1}{2(\sqrt{f})_{0,0}} \left((f)_{i,k-i} \Leftrightarrow 2 \sum_{l=0}^{\frac{i-2}{2}} \sum_{m=0}^{k-i} (\sqrt{f})_{l,m} (\sqrt{f})_{i-l,k-i-m} \right. \\ &\quad \left. \Leftrightarrow 2 \sum_{m=0}^{\frac{k-i-1}{2}} (\sqrt{f})_{i/2,m} (\sqrt{f})_{i/2,k-i-m} \right), \end{aligned} \quad (2.48)$$

- i gerade, $k \Leftrightarrow i$ gerade

$$\begin{aligned} (\sqrt{f})_{i,k-i} &= \frac{1}{2(\sqrt{f})_{0,0}} \left((f)_{i,k-i} \Leftrightarrow 2 \sum_{l=0}^{\frac{i-2}{2}} \sum_{m=0}^{k-i} (\sqrt{f})_{l,m} (\sqrt{f})_{i-l,k-i-m} \right. \\ &\quad \Leftrightarrow 2 \sum_{m=0}^{\frac{k-i-2}{2}} (\sqrt{f})_{i/2,m} (\sqrt{f})_{i/2,k-i-m} \\ &\quad \left. \Leftrightarrow (\sqrt{f})_{i/2,(k-i)/2} (\sqrt{f})_{i/2,(k-i)/2} \right). \end{aligned} \quad (2.49)$$

Für $k = 0$ gilt $(f^2)_{0,0} = (f)_{0,0}^2$ und $(\sqrt{f})_{0,0} = \sqrt{(f)_{0,0}}$.

Um die rekursiven Formeln der Taylorkoeffizienten für andere Standardfunktionen herzuleiten, werden noch die folgenden Zusammenhänge zwischen $(f)_{i,k-i}$ und $(f^{(1,0)})_{i,k-i}$ bzw. zwischen $(f)_{i,k-i}$ und $(f^{(0,1)})_{i,k-i}$ benötigt :

$$\begin{cases} (f^{(1,0)})_{i,k-i} = \frac{f^{(i+1,k-i)}(x_0, y_0)}{i! (k \Leftrightarrow i)!} = (i+1) \cdot (f)_{i+1,k-i}, \\ (f^{(0,1)})_{i,k-i} = \frac{f^{(i,k-i+1)}(x_0, y_0)}{i! (k \Leftrightarrow i)!} = (k \Leftrightarrow i + 1) \cdot (f)_{i,k-i+1}. \end{cases} \quad (2.50)$$

Für die Exponentialfunktion $e^{f(x,y)} =: g(x,y)$ gilt $g^{(1,0)} = g \cdot f^{(1,0)}$ bzw. $g^{(0,1)} = g \cdot f^{(0,1)}$. Bei Anwendung der Rechenregel (2.44) auf $g^{(0,1)}$ und $g^{(1,0)}$ ergibt sich

$$\begin{aligned} i \cdot (g)_{i,k-i} &= (g^{(1,0)})_{i-1,k-i} = (g \cdot f^{(1,0)})_{i-1,k-i} \\ &= \sum_{l=0}^{i-1} \sum_{m=0}^{k-i} (g)_{l,m} (f^{(1,0)})_{i-1-l,k-i-m} \\ &= \sum_{l=0}^{i-1} \sum_{m=0}^{k-i} (g)_{l,m} \cdot (i \Leftrightarrow l) \cdot (f)_{i-l,k-i-m} \\ &= \sum_{l=0}^i \sum_{m=0}^{k-i} (g)_{l,m} \cdot (i \Leftrightarrow l) \cdot (f)_{i-l,k-i-m} \end{aligned}$$

bzw.

$$\begin{aligned}
(k \Leftrightarrow i) \cdot (g)_{i,k-i} &= (g^{(0,1)})_{i,k-i-1} = (g \cdot f^{(0,1)})_{i,k-i-1} \\
&= \sum_{l=0}^i \sum_{m=0}^{k-i-1} (g)_{l,m} (f^{(0,1)})_{i-l,k-i-1-m} \\
&= \sum_{l=0}^i \sum_{m=0}^{k-i-1} (g)_{l,m} \cdot (k \Leftrightarrow i \Leftrightarrow m) \cdot (f)_{i-l,k-i-m} \\
&= \sum_{l=0}^i \sum_{m=0}^{k-i} (g)_{l,m} \cdot (k \Leftrightarrow i \Leftrightarrow m) \cdot (f)_{i-l,k-i-m} .
\end{aligned}$$

Daraus folgt

$$(e^f)_{i,k-i} = \frac{1}{k} \sum_{l=0}^i \sum_{m=0}^{k-i} (k \Leftrightarrow l \Leftrightarrow m) \cdot (e^f)_{l,m} \cdot (f)_{i-l,k-i-m}, \quad k \geq 1. \quad (2.51)$$

Die Rechenregeln der Taylorkoeffizienten für $\sin(f(x, y))$ und $\cos(f(x, y))$ können auf gleiche Weise hergeleitet werden.

$$(\sin f)_{i,k-i} = \frac{1}{k} \sum_{l=0}^i \sum_{m=0}^{k-i} (k \Leftrightarrow l \Leftrightarrow m) \cdot (\cos f)_{l,m} \cdot (f)_{i-l,k-i-m}, \quad k \geq 1, \quad (2.52)$$

$$(\cos f)_{i,k-i} = \Leftrightarrow \frac{1}{k} \sum_{l=0}^i \sum_{m=0}^{k-i} (k \Leftrightarrow l \Leftrightarrow m) \cdot (\sin f)_{l,m} \cdot (f)_{i-l,k-i-m}, \quad k \geq 1. \quad (2.53)$$

Es ist erforderlich, die Taylorkoeffizienten von $\sin f$ und $\cos f$ paarweise zu berechnen. Dies gilt auch für die Hyperbelfunktionen $\sinh(f(x, y))$ und $\cosh(f(x, y))$:

$$(\sinh f)_{i,k-i} = \frac{1}{k} \sum_{l=0}^i \sum_{m=0}^{k-i} (k \Leftrightarrow l \Leftrightarrow m) \cdot (\cosh f)_{l,m} \cdot (f)_{i-l,k-i-m}, \quad k \geq 1, \quad (2.54)$$

$$(\cosh f)_{i,k-i} = \frac{1}{k} \sum_{l=0}^i \sum_{m=0}^{k-i} (k \Leftrightarrow l \Leftrightarrow m) \cdot (\sinh f)_{l,m} \cdot (f)_{i-l,k-i-m}, \quad k \geq 1. \quad (2.55)$$

Die Formeln (2.51), (2.52), (2.53), (2.54) und (2.55) gelten nur für $k \geq 1$. Ist $k = 0$, dann gibt es nur den $(0, 0)$ -ten Taylorkoeffizienten zu berechnen. Für beliebige Kompositionen $g \circ f$ gilt stets $(g(f))_{0,0} = g((f)_{0,0})$. Außerdem verschwindet jeweils der Term für $(l, m) = (i, k \Leftrightarrow i)$ in den Summen.

Die Rechenregel der Taylorkoeffizienten der Potenzfunktion $f(x, y)^a$ lassen sich auf ähnliche Weise herleiten. Es gilt

$$\begin{cases} (f^a)^{(1,0)} \cdot f &= a \cdot f^a \cdot f^{(1,0)} \\ (f^a)^{(0,1)} \cdot f &= a \cdot f^a \cdot f^{(0,1)}. \end{cases}$$

Verwendet man die Rechenregel (2.44) für diese Gleichungen, so ergibt sich

$$\left\{ \begin{array}{l} i \cdot (f^a)_{i,k-i} \cdot (f)_{0,0} = \sum_{l=0}^i \sum_{m=0}^{k-i} (a(i \Leftrightarrow l) \Leftrightarrow l) \cdot (f^a)_{l,m} \cdot (f)_{i-l,k-i-m} \\ (k \Leftrightarrow i) \cdot (f^a)_{i,k-i} \cdot (f)_{0,0} = \sum_{l=0}^i \sum_{m=0}^{k-i} (a(k \Leftrightarrow i \Leftrightarrow m) \Leftrightarrow m) \cdot (f^a)_{l,m} \cdot (f)_{i-l,k-i-m} . \end{array} \right.$$

Daraus folgt für $k \geq 1$

$$(f^a)_{i,k-i} = \frac{1}{k(f)_{0,0}} \sum_{l=0}^i \sum_{m=0}^{k-i} (a(k \Leftrightarrow l \Leftrightarrow m) \Leftrightarrow l \Leftrightarrow m) \cdot (f^a)_{l,m} \cdot (f)_{i-l,k-i-m} . \quad (2.56)$$

Gilt $f(x_0, y_0) = 0$ bzw. $0 \in f([x_0], [y_0])$, dann können $(f^a)_{i,k-i}$, für $k \geq 1$, gemäß (2.56) nicht berechnet werden. Deshalb werden $(f^a)_{i,k-i}$ analog zu (2.31) mit Hilfe von $w_{i,k-i}$, welche durch

$$\left\{ \begin{array}{l} w_{0,0} := 1, \\ w_{i,k-i} := \frac{1}{k} \sum_{l=0}^i \sum_{m=0}^{k-i} (a(k \Leftrightarrow l \Leftrightarrow m) \Leftrightarrow l \Leftrightarrow m) \cdot (f)_{0,0}^{k-l-m-1} \cdot w_{l,m} \cdot (f)_{i-l,k-i-m} , \quad k \geq 1, \end{array} \right.$$

definiert sind, wie folgt bestimmt :

$$(f^a)_{i,k-i} = (f)_{0,0}^{a-k} \cdot w_{i,k-i} . \quad (2.57)$$

Die Formel (2.57) ist nur für $k \leq a$ anwendbar, falls $f(x_0, y_0) = 0$ bzw. $0 \in f([x_0], [y_0])$ gilt.

Kapitel 3

Eindimensionale numerische Integration

Bei der numerischen Approximation des eindimensionalen bestimmten Integrals

$$I(f) := \int_a^b w(x) f(x) dx \quad (3.1)$$

wird vorausgesetzt, daß $w(x)$ eine vorgegebene zulässige *Gewichtsfunktion* ist, d. h., daß $w(x) \geq 0$ ist, die Momente $\mu_k := \int_a^b w(x) x^k dx$, $k = 0, 1, 2, \dots$, existieren und endlich sind und zusätzlich $\mu_0 = \int_a^b w(x) dx > 0$ gilt.

Um Näherungen des Integrals (3.1) zu berechnen, werden *Quadraturformeln* eingesetzt, welche Linearkombinationen von Punktfunktionalen darstellen :

$$S(f) := \sum_{i=1}^M w_i f(x_i) . \quad (3.2)$$

In der Summe (3.2) wird w_i als *Gewicht*, x_i als *Stützstelle* und M als *Knotenzahl* bezeichnet. Die Quadraturformel S hat den *Exaktheitsgrad* d , in Zeichen $\text{deg } S := d$, falls sie für alle Polynome vom Grad kleiner oder gleich d exakt und für mindestens ein Polynom vom Grad $d + 1$ nicht exakt ist.

Die Abweichung zwischen $I(f)$ und $S(f)$ wird durch das Fehlerfunktional E beschrieben

$$E(f) := I(f) - S(f) . \quad (3.3)$$

I , S und E sind stetige lineare Funktionale auf dem Funktionenraum $C[a, b]$, der mit der Supremum-Norm $\|\cdot\|_\infty$ versehen ist. Aus der Definition der Operatornorm $\|\mathcal{O}\| := \sup\{|\mathcal{O}(f)| \mid f \in C[a, b], \|f\|_\infty \leq 1\}$ folgt (vgl. [104])

$$\|I\| = \int_a^b w(x) dx , \quad \|S\| = \sum_{i=1}^M |w_i| , \quad \|E\| = \|I\| + \|S\| . \quad (3.4)$$

Für $\text{deg } S \geq 0$ gilt $\|I\| = I(1) = \sum w_i$. Ist darüber hinaus S positiv, d. h. $w_i \geq 0$ für $i = 1(1)M$, dann ergibt sich $\|I\| = \|S\|$.

3.1 Integration durch Transformation

Durch die affine Transformation

$$x = a + (\tilde{x} \leftrightarrow \tilde{a}) \frac{b \leftrightarrow a}{\tilde{b} \leftrightarrow \tilde{a}} =: \mathcal{T}(\tilde{x}),$$

wird $[\tilde{a}, \tilde{b}]$ auf $[a, b]$ abgebildet. Mit Hilfe von \mathcal{T} kann das bestimmte Integral (3.1) wie folgt erhalten werden :

$$I(f) := \int_a^b w(x) f(x) dx = \frac{b \leftrightarrow a}{\tilde{b} \leftrightarrow \tilde{a}} \cdot \int_{\tilde{a}}^{\tilde{b}} \tilde{w}(\tilde{x}) \tilde{f}(\tilde{x}) d\tilde{x} =: \frac{b \leftrightarrow a}{\tilde{b} \leftrightarrow \tilde{a}} \cdot \tilde{I}(\tilde{f}),$$

wobei $\tilde{w}(\tilde{x}) := w(\mathcal{T}(\tilde{x}))$ und $\tilde{f}(\tilde{x}) := f(\mathcal{T}(\tilde{x}))$ sind. Für $w(x) \equiv 1$ gilt $w(x) \equiv \tilde{w}(\tilde{x}) \equiv 1$.

Bestehen zwischen der Näherung $S(f) = \sum_{i=1}^M w_i f(x_i)$ von $I(f)$ und der Näherung $\tilde{S}(\tilde{f}) = \sum_{i=1}^M \tilde{w}_i \tilde{f}(\tilde{x}_i)$ von $\tilde{I}(\tilde{f})$ die folgenden Beziehungen :

$$x_i = a + (\tilde{x}_i \leftrightarrow \tilde{a}) \frac{b \leftrightarrow a}{\tilde{b} \leftrightarrow \tilde{a}}, \quad w_i = \tilde{w}_i \frac{b \leftrightarrow a}{\tilde{b} \leftrightarrow \tilde{a}},$$

dann heißt S die *affine Transformierte* von \tilde{S} und umgekehrt. Für $w \neq 1$ stellen die affinen Transformaten einer Quadraturformel die Integrationsformeln dar, die jeweils auf verschiedene Gewichtsfunktionen zugeschnitten sind.

In dieser Arbeit werden hauptsächlich die interpolatorischen Quadraturformeln, die mit den Orthogonalpolynomen zusammenhängen, betrachtet. Diese beschreibt der nächste Abschnitt.

3.2 Interpolationsquadraturen

S heißt eine *Interpolationsquadraturformel*, wenn sie für alle Polynome P aus \mathcal{IP}_{M-1} exakt ist, d. h. $E(P) = 0$ für $P \in \mathcal{IP}_{M-1}$ ([22]). Hierbei bezeichnet \mathcal{IP}_{M-1} die Menge aller Polynome vom Grad kleiner oder gleich $M \leftrightarrow 1$. Dies bedeutet $\deg S \geq M \leftrightarrow 1$, wenn S interpolatorisch ist. Darüber hinaus ist zu jedem System $a \leq x_1 < x_2 < \dots < x_M \leq b$ die Interpolationsquadraturformel S , die diese Punkte als Stützstellen besitzt, eindeutig bestimmt (vgl. Braß [22]).

In [40] (S. 74) haben Davis und Rabinowitz gezeigt, daß zu dem System $a \leq x_1 < x_2 < \dots < x_M \leq b$ zwei Möglichkeiten bestehen, die Gewichte w_i zu bestimmen :

1. Die w_i können unter Verwendung der Lagrangeschen Interpolationsformel wie folgt berechnet werden :

$$w_i = \int_a^b w(x) L_i(x) dx = \int_a^b w(x) \frac{\omega(x)}{(x \leftrightarrow x_i) \omega'(x_i)} dx,$$

wobei $\omega(x) = \prod_{i=1}^M (x \leftrightarrow x_i)$ das Knotenpolynom darstellt.

2. Die Gewichte w_1, w_2, \dots, w_M können durch Lösung des linearen Gleichungssystems

$$\int_a^b w(x) f(x) dx \Leftrightarrow \sum_{i=1}^M w_i f(x_i) = 0 ,$$

mit $f(x) = 1, x, x^2, \dots, x^{M-1}$, erhalten werden.

Die beiden Methoden führen zum gleichen Ergebnis.

3.2.1 Symmetrische Quadraturformeln

Ist die Interpolationsquadraturformel S *symmetrisch*, d. h. gilt

1. $w(x) = w(a + b \Leftrightarrow x)$, $x \in [a, b]$,
2. $w_i = w_{M+1-i}$ und $x_i \Leftrightarrow a = b \Leftrightarrow x_{M+1-i}$, $i = 1, 2, \dots, M$,

dann folgt $E(\mathbb{P}_{2s+1}) = 0$ stets aus $E(\mathbb{P}_{2s}) = 0$ (vgl. Braß [22], S. 76). Eine symmetrische Quadraturformel hat also stets einen ungeraden Exaktheitsgrad.

3.2.2 Orthogonalpolynome

Zur Konstruktion von Quadraturformeln mit nicht äquidistanten Stützstellen, werden häufig die Nullstellen von Orthogonalpolynomen verwendet.

Der Funktionenraum $C[a, b]$ sei mittels des Skalarproduktes

$$(f, g) := \int_a^b \bar{w}(x) f(x) g(x) dx$$

bzgl. der Gewichtsfunktion $\bar{w}(x)$ ein Skalarproduktraum, wobei $\bar{w}(x) \geq 0$ ($\bar{w} \not\equiv 0$) vorausgesetzt wird.

Erfüllen die Funktionen f_1, f_2, \dots die Bedingungen

$$(f_i, f_j) = 0 , \quad i \neq j \quad \text{und} \quad (f_i, f_i) \neq 0 , \quad i = 1, 2, \dots ,$$

dann bilden sie ein *Orthogonalsystem* (vgl. Szegö [148]). Gilt darüber hinaus $(f_i, f_i) = 1$, $i = 1, 2, \dots$, so spricht man von einem *Orthonormalsystem*.

Bekanntlich (s. z. B. [47]) gibt es zu jeder zulässigen Gewichtsfunktion $\bar{w}(x)$ stets ein System von Orthogonalpolynomen $\{P_k\}$, die bis auf einen Normierungsfaktor eindeutig bestimmt sind. Diese Polynome genügen der Rekursionsformel (vgl. [141])

$$P_{k+1} = (x \Leftrightarrow \delta_{k+1}) \cdot P_k(x) \Leftrightarrow \gamma_{k+1}^2 \cdot P_{k-1}(x) , \quad \text{für } k \geq 0 ,$$

mit $P_{-1}(x) := 0$, $P_0(x) = 1$ und den Koeffizienten

$$\delta_{k+1} := \frac{(xP_k, P_k)}{(P_k, P_k)}, \quad \text{für } k \geq 0,$$

$$\gamma_{k+1}^2 := \begin{cases} 0, & \text{für } k = 0, \\ \frac{(P_k, P_k)}{(P_{k-1}, P_{k-1})}, & \text{für } k \geq 1. \end{cases}$$

Die Nullstellen dieser Polynome sind reell, einfach und liegen im offenen Intervall (a, b) (vgl. [141]).

Für $[a, b] = [-1, 1]$ werden die Orthogonalpolynome bzgl. $\bar{w}(x) = (1 \mp x^2)^{-1/2}$ als *Tschebyscheff-Polynome erster Art* $T_n(x) := \cos(n \arccos x)$ bezeichnet. Ist $\bar{w}(x) = (1 \mp x^2)^{1/2}$, dann heißen die entsprechenden Orthogonalpolynome die *Tschebyscheff-Polynome zweiter Art* U_n , wobei $U_n(x) = \frac{1}{n+1}T'_{n+1}(x)$ gilt (vgl. [40]).

Verwendet man die Nullstellen x_i , $i = 1(1)M$, des M -ten Orthogonalpolynoms P_M als Stützstellen, so läßt sich eine Interpolationsquadraturformel konstruieren, für welche gilt:

$$\int_a^b w(x) f(x) dx = \sum_{i=1}^M w_i f(x_i), \quad \forall f \in \mathbb{P}_{M-1}, \quad (3.5)$$

wobei die Gewichte w_i die Lösung des linearen Gleichungssystems

$$\sum_{i=1}^M w_i P_k(x_i) = \begin{cases} (P_0, P_0), & \text{für } k = 0, \\ 0, & \text{für } k = 1, 2, \dots, M \mp 1 \end{cases}$$

sind (vgl. [141]). Diese Stützstellen x_i und die zugehörigen Gewichte w_i können auch mit Hilfe der Tridiagonalmatrix

$$J_M = \begin{bmatrix} \delta_1 & \gamma_2 & & & \\ \gamma_2 & \delta_2 & \cdot & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & \gamma_M \\ & & & \gamma_M & \delta_M \end{bmatrix}$$

berechnet werden, da x_i , $i = 1(1)M$, die Eigenwerte der Tridiagonalmatrix J_M darstellen und für w_i die Gleichung $w_i = (v_1^{(i)})^2$ gilt, wobei $v^{(i)} = (v_1^{(i)}, \dots, v_M^{(i)})$ der Eigenvektor zum Eigenwert x_i von J_M ist (vgl. [59], [141]).

Sind sämtliche Gewichte einer Folge von Interpolationsquadraturformel S_M positiv, dann konvergiert sie für alle Riemann-integrierbaren Funktionen f gegen $I(f)$ (vgl. Braß [22], S. 35):

$$\lim_{M \rightarrow \infty} S_M(f) = I(f).$$

3.3 Spezielle Quadraturformeln

Bei Anwendung eines adaptiven numerischen Integrationsverfahrens wird der Integrationsbereich, je nach dem Funktionsverhalten der Integranden, dynamisch in verschiedene Teilbereiche zerlegt. Für $w \neq 1$ bedeutet dies, daß viele verschiedene Quadraturformeln vorher konstruiert werden müssen, was in der Praxis nur schwierig zu realisieren ist. Aus diesem Grund wird die folgende Betrachtung, wie üblich bei der adaptiven numerischen Integration, auf $w \equiv 1$ beschränkt.

3.3.1 Gauß-Legendre-Quadratur

Im Fall $w(x) = \bar{w}(x)$ handelt es sich um die Gauß-Quadratur. Bei der Gauß-Quadratur gilt die Gleichung (3.5) für alle Polynome $f \in \mathbb{P}_{2M-1}$ (vgl. [40]). Keine Quadraturformel mit M Stützstellen kann alle Polynome des Grades $2M$ exakt integrieren, d. h. $\deg S_M \leq 2M \Leftrightarrow 1$. Die Gauß-Quadraturformeln mit M Stützstellen besitzen also den höchsten Exaktheitsgrad $2M \Leftrightarrow 1$ und sie sind interpolatorisch und positiv (vgl. [141]). Ist $w(x)$ symmetrisch, d. h. $w(x) = w(a + b \Leftrightarrow x)$, $\forall x \in [a, b]$, so sind die auf $w(x)$ zugeschnittenen Gauß-Formeln auch symmetrisch (vgl. [135], S. 6).

Für $[a, b] = [\Leftrightarrow 1, 1]$ und $\bar{w}(x) \equiv 1$ sind die Orthogonalpolynome P_k bis auf einen Normierungsfaktor gerade die *Legendre-Polynome*. Die Stützstellen x_1, x_2, \dots, x_M der Gauß-Legendre-Quadraturformeln sind die Nullstellen des M -ten Legendre-Polynoms. Verifizierte Berechnung von x_i und w_i der Gauß-Legendre-Formeln, die die Tridiagonalmatrix J_M heranzieht, wurde in [142] durchgeführt (vgl. auch [153]).

Für beliebige Integrationsbereiche $[a, b]$ können die benötigten Gewichte \tilde{w}_i und Stützstellen \tilde{x}_i bzgl. $w \equiv 1$ aus der Gauß-Legendre-Formeln mit $x_i \in [\Leftrightarrow 1, 1]$ transformiert werden:

$$\tilde{x}_i = \frac{b \Leftrightarrow a}{2} x_i + \frac{a + b}{2}, \quad \tilde{w}_i = \frac{b \Leftrightarrow a}{2} w_i. \quad (3.6)$$

Die Gauß-Quadraturformeln sind positiv und deshalb konvergent für Riemann-integrierbare Funktionen. Diese Konvergenz kann auch aus der Sicht der Riemann-Stieltjeschen Integrale erklärt werden. Da die Gauß-Quadraturformeln Riemann-Stieltjesche Summen darstellen (vgl. Stroud & Secrest [147]), gilt die Konvergenz für $f \in R[a, b]$ unter der Voraussetzung $0 \leq \int_a^b w(x) dx < \infty$.

3.3.2 Clenshaw-Curtis-Quadratur

Die Clenshaw-Curtis-Quadraturformeln sind u. a. bzgl. $w(x) \equiv 1$ konstruiert. Sie sind symmetrisch und interpolatorisch. Ihre Stützstellen sind die Extremalstellen des Tschebyscheff-Polynoms 1. Art T_{M-1} :

$$x_i = \Leftrightarrow \cos \frac{i \Leftrightarrow 1}{M \Leftrightarrow 1} \pi \in [\Leftrightarrow 1, 1], \quad i = 1(1)M.$$

Die Gewichte der Clenshaw-Curtis-Quadraturformeln sind positiv und lassen sich wie folgt berechnen

$$w_1 = w_M = \begin{cases} \frac{1}{(M \Leftrightarrow 1)^2}, & M \text{ gerade,} \\ \frac{1}{M(M \Leftrightarrow 2)}, & M \text{ ungerade,} \end{cases}$$

$$w_k = \left\{ 1 \Leftrightarrow \sum_{j=1}^{\lfloor \frac{M-1}{2} \rfloor^*} \frac{2}{4j^2 \Leftrightarrow 1} \cos \frac{2j(k \Leftrightarrow 1)}{M \Leftrightarrow 1} \pi \right\} \frac{2}{M \Leftrightarrow 1}, \quad k = 2(1)M \Leftrightarrow 1,$$

wobei \sum^* bedeutet, daß der letzte Summand halbiert werden soll, falls M ungerade ist (vgl. Braß [22]).

Der Exaktheitsgrad der Clenshaw-Curtis-Formeln beträgt $M \Leftrightarrow 1$ (falls M gerade) bzw. M (falls M ungerade). Für $M = 2$ erhält man die Trapezregel, für $M = 3$ die Simpsonregel.

Die Transformationsformel (3.6) gilt ebenfalls für Clenshaw-Curtis-Quadraturformeln.

3.4 Quadraturfehler

Bei Anwendung von Integrationsformeln ist es wichtig, die Güte der erzielten Integralnäherungen zu beurteilen. Für diese Aufgabe werden in diesem Abschnitt zunächst die von Peano aufgestellte Kern-Darstellung des Quadraturfehlers sowie die Eigenschaften der Peano-Kerne betrachtet. Anschließend wird zur Einschließung des Quadraturfehlers eine Diskussion durchgeführt, die sich auf den erweiterten Mittelwertsatz der Integralrechnung stützt und für beliebige Quadraturformeln allgemeingültig ist.

Zur verifizierten Berechnung der Restgliedfaktoren bzgl. der Kernfunktionen, die auf dem Integrationsbereich nicht definit sind bzw. deren Definitheit nicht bekannt ist, werden in dieser Arbeit die Nullstellen der Kernfunktionen berücksichtigt. Diese Methode wird als die exakte Methode bezeichnet (vgl. [147], S. 65). In [144] wurden die Restgliedfaktoren der Gauß-Quadraturformeln mit der exakten Methode verifiziert berechnet. Die Effektivität dieser Methode kann weiter erhöht werden, wenn im Intervall-Newton-Schritt die Mittelwertform für die erste Ableitung eingesetzt wird und bei der Berechnung der Restgliedfaktoren die Symmetrie und die lokale Definitheit der Kernfunktionen beachtet werden. Diese Teilaspekte werden im Abschnitt 3.4.3 besprochen. Die Restgliedfaktoren der Gauß-Legendre-Quadraturformeln, die mit diesem modifizierten Ansatz verifiziert berechnet wurden, sind in Tabelle 3.2 aufgeführt. In Tabelle 3.5 werden die von Braß/Förster [28] für die Restgliedfaktoren der Clenshaw-Curtis-Quadraturformeln angegebenen Abschätzungen und die verifiziert berechneten Werten zusammengestellt.

Um die Diskussion zu vereinfachen, werden zuerst einige Bezeichnungen eingeführt.

Definition 3.4.1

1. $(x \Leftrightarrow t)^{\langle r \rangle} := \frac{(x \Leftrightarrow t)^r}{r!}, \quad r \in \mathbb{N}_0.$
2. $(x \Leftrightarrow t)_+^r := \begin{cases} (x \Leftrightarrow t)^r, & \text{für } x \geq t, \\ 0, & \text{für } x < t. \end{cases} \quad (3.7)$
3. $(x \Leftrightarrow t)_-^r := \begin{cases} 0, & \text{für } x \geq t, \\ (x \Leftrightarrow t)^r, & \text{für } x < t. \end{cases}$

Dabei wird $0^0 := 1$ vereinbart. Aus diesen Bezeichnungen folgt unmittelbar der nächste Zusammenhang :

Bemerkung 3.4.1 $(x \Leftrightarrow t)^r = (x \Leftrightarrow t)_+^r + (x \Leftrightarrow t)_-^r.$

Darüber hinaus lassen sich die Intervallauswertungen dieser Funktionen wie folgt angeben:

Definition 3.4.2 Sei $[z] := [x] \Leftrightarrow [t]$. Es ist

1. $([x] \Leftrightarrow [t])_+ := \begin{cases} ([x] \Leftrightarrow [t]) \cap [0, \infty), & \text{für } \bar{z} > 0, \\ 0, & \text{für } \bar{z} \leq 0. \end{cases} \quad (3.8)$
2. $([x] \Leftrightarrow [t])_- := \begin{cases} 0, & \text{für } \underline{z} \geq 0, \\ ([x] \Leftrightarrow [t]) \cap (\Leftrightarrow\infty, 0], & \text{für } \underline{z} < 0. \end{cases}$

Die Funktionen in (3.8) sind inklusionsisoton, stetig und Lipschitz.

3.4.1 Peanosche Kerndarstellung

Nach der Definition ist der Quadraturfehler das lineare Funktional $E(f) := I(f) \Leftrightarrow S(f)$. Ersetzt man die Funktion f , $f \in C^r[a, b]$, durch ihre Taylor-Formel bzgl. des Entwicklungspunktes a mitsamt des Integralrestgliedes

$$f(x) = \sum_{\nu=0}^{r-1} f^{(\nu)}(a) \cdot (x \Leftrightarrow a)^{\langle \nu \rangle} + \int_a^x f^{(r)}(t) \cdot (x \Leftrightarrow t)^{\langle r-1 \rangle} dt,$$

so erhält man die folgende Integraldarstellung des Quadraturfehlers ([22]).

Satz 3.4.1 (PEANO [1913]) Für $1 \leq r \leq \deg S + 1$ und $f \in C^r[a, b]$ gilt

$$E(f) = \int_a^b K_r(t) f^{(r)}(t) dt \quad (3.9)$$

mit $K_r(t) := E\left((x \Leftrightarrow t)_+^{(r-1)}\right)$. Hierbei wird das Fehlerfunktional E auf x angewendet.

Die Funktion $K_r(t)$ heißt der r -te Peano-Kern der Quadraturformel S . Die Peano-Kerne hängen nur von der Quadraturformel und dem Integrationsbereich ab, nicht jedoch von der zu integrierenden Funktion und dem Entwicklungspunkt in der Taylor-Formel (vgl. [104], S. 15). Für die Anwendung der Peanoschen Fehlerdarstellung wird vorausgesetzt, daß f auf $[a, b]$ mindestens einmal stetig differenzierbar ist.

Verschiedene Eigenschaften der Peano-Kerne wurden von Braß [22] ausführlich untersucht. Diejenigen, die für die spätere Diskussion nützlich sind, werden hier dargestellt.

Stetigkeit und Differenzierbarkeit

Der Peano-Kern K_r ist auf $[a, b]$ $(r \Leftrightarrow 2)$ -mal stetig differenzierbar, $2 \leq r \leq \deg S + 1$. Für $r = 1$ besitzt K_r Sprungstellen in den Stützstellen x_i , die im halboffenen Intervall $[a, b)$ liegen. Ferner gilt

$$K_r^{(\nu)}(t) = (\Leftrightarrow 1)^\nu \cdot K_{r-\nu}(t), \quad \nu = 0, \dots, r \Leftrightarrow 2. \quad (3.10)$$

Symmetrie

Eine symmetrische Quadraturformel S hat symmetrische Peano-Kerne :

$$K_r(t) = (\Leftrightarrow 1)^r \cdot K_r(b + a \Leftrightarrow t) \quad \text{für } t \in [a, b] \quad \text{und} \quad 2 \leq r \leq \deg S + 1. \quad (3.11)$$

Im Fall $r = 1$ gilt die Gleichung in (3.11) nur für $t \in [a, b] \setminus \{x_1, \dots, x_M\}$ (vgl. [104]).

Definitheit

Hat eine Funktion keinen Vorzeichenwechsel, so heißt sie definit. Eine hinreichende und notwendige Bedingung für die Definitheit der Peano-Kerne enthält der nächste Satz (vgl. [22], S. 57).

Satz 3.4.2 $K_r(t)$ hat genau dann auf $[a, b]$ keinen Vorzeichenwechsel, wenn gilt :

$$E(f) = \frac{f^{(r)}(\xi)}{r!} E(x^r), \quad \xi \in [a, b]. \quad (3.12)$$

Für definite Kernfunktionen ist (3.12) eine direkte Folgerung aus der Peanoschen Fehlerdarstellung (3.9). Denn in diesem Fall läßt sich der erweiterte Mittelwertsatz der Integralrechnung auf (3.9) anwenden :

$$E(f) = f^{(r)}(\xi) \cdot \int_a^b K_r(t) dt, \quad \xi \in [a, b].$$

Es ergibt sich dann

$$E(x^r) = r! \cdot \int_a^b K_r(t) dt.$$

Die Definitheit gilt höchstens für die Kernfunktionen von höchster Ordnung, K_{d+1} , mit $d = \deg S$. Für $1 \leq r \leq d$ und $f(x) = x^r$ gilt wegen des Exaktheitsgrads

$$0 = E(f) = \int_a^b f^{(r)}(t) K_r(t) dt = r! \cdot \int_a^b K_r(t) dt.$$

Ist $K_r \equiv 0$, dann muß die Quadraturformel für beliebige Funktionen $f \in C^r[a, b]$ exakt sein, weil $E(f) \equiv 0$ ist. Dies widerspricht dem Exaktheitsgrad der Quadraturformel. Also müssen die Kernfunktionen K_r , $1 \leq r \leq d$, einen Vorzeichenwechsel haben.

Für die Clenshaw-Curtis-Formeln mit $M \geq 4$ sind die Kernfunktionen von höchster Ordnung nicht definit (Akrivis/Förster [3]).

Anzahl der Nullstellen

Die Peano-Kerne besitzen nur endlich viele Nullstellen. Zur Bestimmung der Nullstellen von Peano-Kernen soll das erweiterte Intervall-Newton-Verfahren in begrenzter Zeit beendet sein.

Die maximale Anzahl der Nullstellen eines Peano-Kerns formuliert der nächste Satz (vgl. [104], S. 16).

Satz 3.4.3 *K_r hat im Intervall $[a, b]$ höchstens $(M + 1)r$ verschiedene Nullstellen.*

Im Spezialfall $w \equiv 1$ besitzt K_r höchstens $r + 2M$ Nullstellen, wobei die Vielfachheiten mitgezählt sind ([104], S. 16 f).

Wegen $K_r(t) = E_x \left((x \Leftrightarrow t)_+^{(r-1)} \right)$ gilt $K_r(a) = K_r(b) = 0$, für $2 \leq r \leq d + 1$. Ist $x_1 > a$, so gilt $K_1(a) = 0$; ist $x_M < b$, so gilt $K_1(b) = 0$. Aus (3.10) folgt, daß in den Endpunkten des Integrationsbereichs $[a, b]$ der Peano-Kern K_r , $r = 2, \dots, d + 1$, jeweils eine mindestens $(r \Leftrightarrow 1)$ -fache Nullstelle besitzt (vgl. [104]).

Speziell für die Gauß-Quadratur existiert noch die nächste Aussage (vgl. Braß [22], S. 149).

Satz 3.4.4 *Für die Gauß-Quadraturformeln besitzt der Peano-Kern $K_{2M-\nu}$, $\nu = 0, \dots, 2M \Leftrightarrow 2$ auf dem offenen Intervall (a, b) ν Nullstellen.*

3.4.2 Bestimmung der Fehlerschranken

Bei der Abschätzung des Quadraturfehlers lassen sich zwei Fälle unterscheiden, nämlich Peano-Kerne mit Vorzeichenwechsel und Peano-Kerne ohne Vorzeichenwechsel.

Beiden Fällen liegt der erweiterte Mittelwertsatz der Integralrechnung zugrunde. Für die nicht definiten Kernfunktionen werden zusätzlich die folgenden Bezeichnungen verwendet:

$$\begin{aligned} K^+(t) &:= \max(K(t), 0) , \\ K^-(t) &:= \max(\ominus K(t), 0) . \end{aligned} \tag{3.13}$$

Es ist $K^+(t), K^-(t) \geq 0$. Mit diesen Bezeichnungen lassen sich die Funktionen $K(t)$ und $|K(t)|$ wie folgt darstellen :

$$\begin{aligned} K(t) &= K^+(t) \ominus K^-(t) , \\ |K(t)| &= K^+(t) + K^-(t) . \end{aligned} \tag{3.14}$$

Im Rest dieses Kapitels bezeichne d den Exaktheitsgrad der betrachteten Quadraturformel S , $d := \deg S$. Die Fehlerabschätzung für den Fall, daß die Kernfunktion K_{d+1} definit ist, wird im folgenden zuerst behandelt.

Definite Kernfunktion K_{d+1}

Für hinreichend oft stetig differenzierbare Funktionen führt Gleichung (3.12) direkt zu der Einschließungsformel (3.15).

Satz 3.4.5 *Hat $K_{d+1}(t)$ auf $[a, b]$ keinen Vorzeichenwechsel, so gilt für $f \in C^{d+1}[a, b]$:*

$$E(f) \in \frac{f^{(d+1)}([a, b])}{(d+1)!} \cdot E(x^{d+1}) \tag{3.15}$$

mit

$$E(x^{d+1}) := \begin{cases} \frac{b^{d+2} \ominus a^{d+2}}{d+2} \ominus S(x^{d+1}) , & \text{für } w(x) \equiv 1 , \\ \tilde{S}(x^{d+1}) \ominus S(x^{d+1}) , & \text{für beliebige } w(x) . \end{cases} \tag{3.16}$$

Dabei ist \tilde{S} eine auf die Gewichtsfunktion $w(x)$ zugeschnittene Quadraturformel, mit $\deg \tilde{S} \geq d+1$.

Der Wertebereich $f^{(d+1)}([a, b])$ in (3.15) kann mit Hilfe der eindimensionalen Intervall-Taylorkoeffizienten verifiziert abgeschätzt werden. Und für die Fehlerkonstante $E(x^{d+1})$

ist (3.16) eine direkte Folgerung aus der Definition $E(x^{d+1}) := I(x^{d+1}) \Leftrightarrow S(x^{d+1})$. Das bestimmte Integral $I(x^{d+1}) = \int_a^b w(x) x^{d+1} dx$ kann mit Hilfe der Stammfunktion bzw. einer Quadraturformel \tilde{S} , mit $\deg \tilde{S} \geq d+1$ und folglich $I(x^{d+1}) = \tilde{S}(x^{d+1})$, bestimmt werden.

Unter der Voraussetzung von Satz 3.4.5 ist der Restgliedfaktor $E(x^{d+1})$ in (3.15) für beliebige Quadraturformeln allgemeingültig. Für spezielle Quadraturverfahren kann man auch die einzeln hergeleiteten Restglieddarstellungen anwenden. Z. B. lassen sich die Restgliedfaktoren bzgl. K_{d+1} der Gauß-Quadraturformeln auch durch Berechnung des Skalarproduktes (P_M, P_M) erhalten ([141]), wobei P_M das M -te Orthogonalpolynom darstellt. In [147], [144] und [88] wurden die Restgliedfaktoren bzgl. K_{d+1} der Gauß-Legendre-Formeln unter Verwendung des Skalarproduktes bestimmt.

Kernfunktion K_r mit Vorzeichenwechsel

Für eine Kernfunktion K_r , die auf dem Integrationsbereich das Vorzeichen wechselt, wird häufig in der Literatur der Quadraturfehler zu

$$|E(f)| \leq \|f^{(r)}\|_\infty \cdot \int_a^b |K_r| dt \quad (3.17)$$

abgeschätzt. Die Güte dieser Abschätzung hängt davon ab, wie gut man die Maximumnorm $\|f^{(r)}\|_\infty$ und die L_1 -Norm $\int_a^b |K_r| dt$ bestimmen bzw. abschätzen kann.

Eine Obergrenze des Restgliedfaktors $\int_{-1}^1 |K_{d+1}(t)| dt$ für die Clenshaw-Curtis-Formeln hat Braß/Förster [28] angegeben.

Satz 3.4.6 Sei $[a, b] = [\Leftrightarrow 1, 1]$. Für die Clenshaw-Curtis-Formeln mit $M \geq 4$ gilt

$$|E(f)| \leq \frac{M \Leftrightarrow 1}{2^{M-4}(M^2 \Leftrightarrow 4)M(M \Leftrightarrow 4)} \cdot (1 + \varepsilon_M) \cdot \frac{\|f^{(M+1)}\|_\infty}{(M+1)!}, \quad M \text{ ungerade},$$

$$|E(f)| \leq \frac{1}{2^{M-4}(M^2 \Leftrightarrow 1)(M \Leftrightarrow 3)} \cdot (1 + \varepsilon_M) \cdot \frac{\|f^{(M)}\|_\infty}{M!}, \quad M \text{ gerade} \quad (3.18)$$

mit $\varepsilon_4 < \frac{1}{19}$, $\varepsilon_6 < \frac{1}{52}$, $\varepsilon_M < 10^{-6}$ für $M > 8$, $\varepsilon_M < 10^{-20}$ für $M > 20$ und $\varepsilon_M < 10^{-2}$ für andere M .

Während Braß ([22], [28]) für die Clenshaw-Curtis-Formeln die L_1 -Norm $\|K_{d+1}\|_1$ abgeschätzt hat, haben Stroud und Secrest [147] außer $\|K_{d+1}\|_1$ (s. o.) noch die L_1 -Norm $\|K_r\|_1$, $1 \leq r < d+1$, für die Gauß-Legendre-Formeln berechnet. Die Berechnungen verwendeten jeweils die exakte Methode für $r = 1, 2$ und die Approximationsmethode für gewisse $r \geq 3$. Bei der *exakten Methode* werden die Teilbereiche, auf welchen die Kernfunktionen keinen Vorzeichenwechsel haben, mit Hilfe der Nullstellen der Kernfunktionen identifiziert. Über solchen Bereichen läßt sich $\|K_r\|_1$ direkt berechnen. Bei der *Approximationsmethode* werden hingegen die Integrale $\int_a^b |K_r(t)| dt$ mit Hilfe von Quadraturformeln näherungsweise bestimmt, $I(|K_r|) \approx S(|K_r|)$.

Obwohl die Fehlerabschätzung (3.17) längst in der Literatur diskutiert wird, wurde sie dennoch in der Praxis noch nicht angewendet. Dies liegt daran, daß die Maximumnorm $\|f^{(r)}\|_\infty$ schwierig zu gewinnen ist. Mit Hilfe der eindimensionalen Intervall-Taylorkoeffizienten läßt sich der Wertebereich $f^{(r)}([a, b])$ zuverlässig und effizient abschätzen. Enge Wertebereichseinschließungen von $f^{(r)}$ können durch Verfeinerung des Definitionsbereichs $[a, b]$ erhalten werden (s. Kapitel 2). Diese Rechentechnik wurde bereits für die Einschließung der Quadraturfehler bzgl. der definiten Kernfunktionen eingesetzt.

Um den Quadraturfehler $E(f)$ bzgl. der Kernfunktionen, die nicht definit sind bzw. deren Definitheit unbekannt sind, einzuschließen, werden die am Anfang dieses Abschnitts eingeführten Bezeichnungen (3.13) und (3.14) herangezogen. Es folgt aus dem Satz von Peano

$$E(f) = \int_a^b K_r^+(t) f^{(r)}(t) dt \Leftrightarrow \int_a^b K_r^-(t) f^{(r)}(t) dt .$$

Unter Verwendung des erweiterten Mittelwertsatzes der Integralrechnung ergibt sich

$$E(f) = f^{(r)}(\xi) \cdot \int_a^b K_r^+(t) dt \Leftrightarrow f^{(r)}(\eta) \cdot \int_a^b K_r^-(t) dt , \quad \xi, \eta \in [a, b] .$$

Ersetzt man ξ und η durch das einschließende Intervall $[a, b]$, so läßt sich der Quadraturfehler wie (3.19) eingrenzen (vgl. [144]).

Satz 3.4.7 *Für den Quadraturfehler $E(f)$ mit der Darstellung (3.9) gilt*

$$E(f) \in P_r \cdot f^{(r)}([a, b]) + N_r \cdot f^{(r)}([a, b]) , \quad (3.19)$$

dabei sind $P_r := \int_a^b K_r^+(t) dt$ und $N_r := \int_a^b K_r^-(t) dt$.

Die Konstanten $E(x^{d+1})$, $\int_a^b |K_r|$, P_r und N_r werden gemeinsam als Restgliedfaktoren bzw. Fehlerkonstanten bezeichnet, es gilt $P_r \Leftrightarrow N_r = \int_a^b |K_r| dt$.

Für $1 \leq r \leq d$ haben die Restgliedfaktoren P_r und N_r , wegen $\int_a^b K_r(t) dt = 0$ und $K_r = K_r^+ \Leftrightarrow K_r^-$, den gleichen Betrag, d. h.

$$P_r = \Leftrightarrow N_r , \quad 1 \leq r < d + 1 . \quad (3.20)$$

Die Gleichung (3.20) ist für beliebige Quadraturverfahren gültig, auch wenn die betrachtete Quadraturformel S nicht symmetrisch ist.

Ist K_r nicht definit, so sind K_r^+ und K_r^- auf $[a, b]$ höchstens stetig aber an den Nullstellen von K_r nicht stetig differenzierbar. Für die verifizierte Bestimmung der Integrale $\int_a^b K_r^+(t) dt$ und $\int_a^b K_r^-(t) dt$ kann deshalb die Peanosche Fehlerdarstellung nicht eingesetzt werden. Um diese Fehlerkonstanten P_r und N_r zu berechnen, werden in dieser Arbeit ebenfalls die Nullstellen der Kernfunktionen berücksichtigt. Die konkrete Durchführung dieser Methode wird im nächsten Abschnitt beschrieben.

Wenn für $w(x) \not\equiv 1$ die Stammfunktion zu $w(x) (x \Leftrightarrow t)^{\langle r-1 \rangle}$ bekannt ist, dann kann die Kernfunktion $K_r(t)$ analog zum Fall $w(x) \equiv 1$ behandelt werden. Läßt sich hingegen das unbestimmte Integral $\int w(x) x^r dx$, $w(x) \not\equiv 1$, in keiner geschlossenen Form darstellen, so führt die Nullstellenbestimmung der Kernfunktion K_r zu einer schwierigen Aufgabe. Andererseits ist es meist nicht sinnvoll die Restgliedfaktoren zu $w(x) \not\equiv 1$ über $[a, b]$ für ein adaptives Integrationsverfahren zu verwenden, auch wenn sie sich berechnen lassen (vgl. auch [88]). Aus diesen Gründen wird die folgende Behandlung auf den Fall $w(x) \equiv 1$ beschränkt.

Satz 3.4.8 Für $w(x) \equiv 1$ besitzt der Peano-Kern K_r , $1 \leq r \leq d+1$, die folgenden beiden Darstellungen

$$K_r(t) = \begin{cases} (b \Leftrightarrow t)^{\langle r \rangle} \Leftrightarrow \sum_{i=1}^M w_i (x_i \Leftrightarrow t)_+^{\langle r-1 \rangle}, & (\star 1) \\ (a \Leftrightarrow t)^{\langle r \rangle} + \sum_{i=1}^M w_i (x_i \Leftrightarrow t)_-^{\langle r-1 \rangle}. & (\star 2) \end{cases}$$

Beweis : Die Darstellung $(\star 1)$ ist eine ausgeschriebene Form von (3.21).

$$\begin{aligned} \text{Wegen } (b \Leftrightarrow t)^{\langle r \rangle} &= \int_t^b (x \Leftrightarrow t)^{\langle r-1 \rangle} dx \\ &= \int_a^b (x \Leftrightarrow t)^{\langle r-1 \rangle} dx \Leftrightarrow \int_a^t (x \Leftrightarrow t)^{\langle r-1 \rangle} dx \\ &= \sum_{i=1}^M w_i (x_i \Leftrightarrow t)^{\langle r-1 \rangle} + (a \Leftrightarrow t)^{\langle r \rangle} \\ &= \sum_{i=1}^M w_i \left((x_i \Leftrightarrow t)_+^{\langle r-1 \rangle} + (x_i \Leftrightarrow t)_-^{\langle r-1 \rangle} \right) + (a \Leftrightarrow t)^{\langle r \rangle} \end{aligned}$$

ergibt sich $(\star 2)$. □

Abhängig vom Wert r besitzen die Summen $\sum_{i=1}^M w_i (x_i \Leftrightarrow t)_\pm^{\langle r-1 \rangle} \in \sum_{i=1}^M [w_i] ([x_i] \Leftrightarrow t)_\pm^{\langle r-1 \rangle}$ und die Kernfunktion K_r unterschiedliche Funktionsverhalten :

r	$\sum_{i=1}^M w_i (x_i \Leftrightarrow t)_\pm^{\langle r-1 \rangle}$	K_r
1	Treppenfunktion	Sprungfunktion mit Sprungstellen x_i
2	Polygonzug ($\in C^0[a, b]$)	an x_i nicht stetig diff. Funktion
≥ 3	Splinefunktion vom Grad $r \Leftrightarrow 1$	Monospline ($\in C^{r-2}[a, b]$) (vgl. [20])

Die Einschränkung von K_r auf $[x_i, x_{i+1}]$, $i = 0(1)M$, mit $x_0 := a$ und $x_{M+1} := b$, ist ein Polynom vom Grad r . Für Polynome sind die Stammfunktionen bekannt.

Da einerseits das Intervall-Newton-Verfahren zur Bestimmung der Nullstellen eindimensionaler reellwertiger Funktionen nur für die stetig differenzierbare Funktionen einsetzbar ist und andererseits statt x_i nur die Einschließungen $[x_i]$ bekannt sind, wird das Intervall-Newton-Verfahren stückweise auf $[\bar{x}_i, \underline{x}_{i+1}]$ angewendet. Daher können alle Kernfunktionen einheitlich behandelt werden. Mit den dadurch erzielten Nullstelleneinschließungen $[z_{ik}] \subseteq [\bar{x}_i, \underline{x}_{i+1}]$, $z_{ik} < z_{i,k+1}$, ist man in der Lage, die Teilfaktoren

$$P_r^{ik} := \int_{\bar{z}_{ik}}^{z_{i,k+1}} K_r^+(t) dt, \quad N_r^{ik} := \Leftrightarrow \int_{\bar{z}_{ik}}^{z_{i,k+1}} K_r^-(t) dt$$

zu bestimmen, wobei gilt :

$$\int_{\bar{z}_{ik}}^{z_{i,k+1}} K_r^\pm(t) dt = \int_{\bar{z}_{ik}}^{z_{i,k+1}} K_r(t) dt \quad \text{bzw.} \quad 0.$$

Das bestimmte Integral $\int_{\bar{z}_{ik}}^{z_{i,k+1}} K_r(t) dt$ kann dann direkt mit Hilfe der Stammfunktion

$$\int K_r(t) dt = \begin{cases} \Leftrightarrow(b \Leftrightarrow t)^{\langle r+1 \rangle} + \sum_{i=1}^M w_i (x_i \Leftrightarrow t)_+^{\langle r \rangle} \\ \Leftrightarrow(a \Leftrightarrow t)^{\langle r+1 \rangle} \Leftrightarrow \sum_{i=1}^M w_i (x_i \Leftrightarrow t)_-^{\langle r \rangle} \end{cases} \quad \text{auf } [a, b]$$

gewonnen werden. Bei konkreter Durchführung wird die Einschließungsfunktion $[K_r(t)] := ([b] \Leftrightarrow t)^{\langle r \rangle} \Leftrightarrow \sum_{i=1}^M [w_i] ([x_i] \Leftrightarrow t)_+^{\langle r-1 \rangle}$ betrachtet. Unter Verwendung der Maschinenintervallarithmetik werden die Einschließungen der Restgliedfaktoren $[P_r^{ik}]$ und $[N_r^{ik}]$ erzielt.

Für die Teilbereiche $[x_i]$ sind die Restgliedfaktoren wie folgt zu berechnen:

$$P_r^{x_i} := \int_{[x_i]} K_r^+(t) dt \in [K_r^+([x_i])] \cdot d([x_i]) =: [P_r^{x_i}],$$

$$N_r^{x_i} := \Leftrightarrow \int_{[x_i]} K_r^-(t) dt \in \Leftrightarrow [K_r^-([x_i])] \cdot d([x_i]) =: [N_r^{x_i}].$$

Hierbei werden $K_r^+([x_i])$ und $K_r^-([x_i])$ anhand der Schnittbildungen

$$[K_r^+([x_i])] := [K_r([x_i])] \cap [0, \infty),$$

$$[K_r^-([x_i])] := \Leftrightarrow [K_r([x_i])] \cap [0, \infty)$$

bestimmt. $[P_r^{z_{ik}}]$ und $[N_r^{z_{ik}}]$ über $[z_{ik}]$ werden den Faktoren $[P_r^{x_i}]$ und $[N_r^{x_i}]$ entsprechend berechnet. Die gesuchten Fehlerkonstanten $[P_r]$ und $[N_r]$ können dann durch Aufsummieren der berechneten Teilfaktoren erhalten werden :

$$[P_r] = \sum_{i,k} [P_r^{ik}] + \sum_i [P_r^{x_i}] + \sum_{i,k} [P_r^{z_{ik}}],$$

$$[N_r] = \sum_{i,k} [N_r^{ik}] + \sum_i [N_r^{x_i}] + \sum_{i,k} [N_r^{z_{ik}}].$$
(3.22)

Das Prinzip (3.22) zur verifizierten Bestimmung der Restgliedfaktoren wurde in [144] erstmals angewendet. Die Effektivität des erweiterten Intervall-Newton-Verfahrens kann dennoch durch Anwendung der Mittelwertform (2.21) für die erste Ableitung K'_r weiter erhöht werden. Ferner braucht die Berechnung von P_r und N_r für symmetrische Quadraturformeln, wie z. B. die Gauß-Legendre- und die Clenshaw-Curtis-Formeln, wegen der Symmetrieeigenschaft (3.11), nur über $[a, \frac{b+a}{2}]$ oder $[\frac{b+a}{2}, b]$ durchgeführt zu werden. Seien \tilde{P}_r und \tilde{N}_r die gemäß (3.22) über $[a, \frac{b+a}{2}]$ (bzw. über $[\frac{b+a}{2}, b]$) erzielten Faktoren. Dann gilt wegen (3.11)

$$\begin{aligned} (1) \quad P_r &= \tilde{P}_r \Leftrightarrow \tilde{N}_r, & N_r &= \tilde{N}_r \Leftrightarrow \tilde{P}_r, & r &\text{ ungerade,} \\ (2) \quad P_r &= \tilde{P}_r + \tilde{P}_r, & N_r &= \tilde{N}_r + \tilde{N}_r, & r &\text{ gerade.} \end{aligned} \tag{3.23}$$

Für symmetrische Quadraturformeln wird der Rechenaufwand unter Berücksichtigung von (3.23) auf 1/2 reduziert. Dabei soll die Darstellung ($\star 1$) im Satz 3.4.8 für den Bereich $[\frac{b+a}{2}, b]$ bzw. die Darstellung ($\star 2$) für den Bereich $[a, \frac{b+a}{2}]$ angewendet werden, denn dann kann weiterer Rechenaufwand gespart werden.

Mit den dualen Darstellungen ($\star 1$) und ($\star 2$) im Satz 3.4.8 ist es leicht nachzuprüfen, daß die Peano-Kerne $K_r(t)$ auf $[a, x_1]$ und $[x_M, b]$ ihre Vorzeichen nicht wechseln. (Für $r = 1$ wird $t = x_M$ ausgenommen.) Das Intervall-Newton-Verfahren muß dann nicht für diese Bereiche eingesetzt werden, falls $a \neq x_1$ bzw. $x_M \neq b$ ist. Da für $r \geq 2$ die Grenzpunkte a, b ($r \Leftrightarrow 1$)-fache Nullstellen sind, können dadurch mehrere Iterationsschritte gespart werden. Für großes r würde dies eine einflußreiche Ersparnis des Rechenaufwands bedeuten.

Weiter ist zu bemerken, daß für $[t] = [x_k]$ der k -te Term in der Summe $\sum_{i=1}^M [w_i] ([x_i] \Leftrightarrow [t])_+^0$ gemäß $([x_k] \Leftrightarrow [x_k])_+^0 := [0, 1]$ bestimmt werden soll, und die Wertebereichseinschließung der Ableitungen K'_r formelmäßig auszuwerten ist, da die Kernfunktionen für beliebige Quadraturverfahren von gleicher Gestalt sind. Durch Einsatz der Mittelwertform von K'_r wird der Rechenaufwand beträchtlich verringert. Die Wertebereichseinschließung von K'_r kann zwar auch mit Hilfe der Intervall-Taylorkoeffizienten berechnet werden, jedoch ist das Ergebnis i. a. gröber als die Intervallerweiterung von K'_r . Deshalb sollten sie erst für Funktionen von beliebiger Gestalt angewendet werden. Darüber hinaus konnte durch Einsatz der Intervallsteigung bzw. der Intervallsteigungsarithmetik auch keine bessere Recheneffektivität für die Fehlerkonstanten geliefert werden. Für die stückweise polynomialen Kernfunktionen ist die Intervallsteigungsarithmetik sogar viel ineffizienter als die Differentiationsarithmetik.

Die Restgliedfaktoren der Gauß-Legendre- und der Clenshaw-Curtis-Formeln, die mit den oben beschriebenen Ideen berechnet wurden, sind in Tabelle 3.2 und Tabelle 3.4 aufgelistet. Der betrachtete Integrationsbereich war $[\Leftrightarrow 1, 1]$.

Speziell für die Restgliedfaktoren der Gauß-Legendre-Formeln werden die verschiedenen bekannten Rechenergebnisse (vgl. [147], [101], [144]) in Tabelle 3.3 zusammengestellt. Sie sind im allgemeinen von gleicher Größenordnung und ihre ersten Ziffern können miteinander übereinstimmen. Insbesondere liefert das Programm, das im Rahmen dieser Arbeit implementiert wurde, bei geringer Rechenzeit die beste Güte.

Für die Fehlerkonstanten der Clenshaw-Curtis-Formeln sind in Tabelle 3.5 die verifiziert berechneten Ergebnisse und die von Braß/Förster [28] aufgestellten Abschätzungen zum Vergleich zusammengestellt. Diese numerischen Ergebnisse zeigen, daß die von Braß/Förster [28] angegebenen Oberschranken im allgemeinen gut abgeschätzt sind.

Die in diesem Abschnitt beschriebenen Ideen zur Berechnung der L_1 -Norm der Peano-Kerne können auch für die eindimensionalen Sard-Kerne, die in der Fehlerdarstellungen der Kubaturfehler zu finden sind, angewendet werden. Die numerische Kubatur wird im nächsten Kapitel behandelt.

Tabelle 3.2: Restgliedfaktoren $r! \int_{-1}^1 K_r^+$ und $r! \int_{-1}^1 K_r^-$ der Gauß-Legendre-Formeln

M	r	$r! [P_r]$	$r! [N_r]$
2	1	$2.5598306414370 \binom{8}{7} E \Leftrightarrow 01$	$\Leftrightarrow 2.5598306414370 \binom{7}{8} E \Leftrightarrow 01$
2	2	$8.112906564241 \binom{82}{78} E \Leftrightarrow 02$	$\Leftrightarrow 8.112906564241 \binom{7}{9} E \Leftrightarrow 02$
2	3	$5.754991027012 \binom{6}{4} E \Leftrightarrow 02$	$\Leftrightarrow 5.754991027012 \binom{4}{6} E \Leftrightarrow 02$
2	4	$1.7777777777777 \binom{9}{7} E \Leftrightarrow 01$	$0.0000000000000000 E + 00$
3	1	$1.78669008626 \binom{500}{498} E \Leftrightarrow 01$	$\Leftrightarrow 1.78669008626 \binom{498}{500} E \Leftrightarrow 01$
3	2	$3.74355436143 \binom{32}{29} E \Leftrightarrow 02$	$\Leftrightarrow 3.743554361433 \binom{0}{1} E \Leftrightarrow 02$
3	3	$1.644551573889 \binom{4}{2} E \Leftrightarrow 02$	$\Leftrightarrow 1.644551573889 \binom{2}{4} E \Leftrightarrow 02$
3	4	$1.09059248140 \binom{40}{39} E \Leftrightarrow 02$	$\Leftrightarrow 1.09059248140 \binom{3}{5} E \Leftrightarrow 02$
3	5	$1.17473328183 \binom{8}{6} E \Leftrightarrow 02$	$\Leftrightarrow 1.17473328183 \binom{6}{8} E \Leftrightarrow 02$
3	6	$4.571428571428 \binom{7}{5} E \Leftrightarrow 02$	$0.0000000000000000 E + 00$
4	1	$1.3799698495825 \binom{8}{7} E \Leftrightarrow 01$	$\Leftrightarrow 1.3799698495825 \binom{7}{8} E \Leftrightarrow 01$
4	2	$2.185660935053 \binom{2}{0} E \Leftrightarrow 02$	$\Leftrightarrow 2.185660935053 \binom{0}{2} E \Leftrightarrow 02$
4	3	$7.05030204561 \binom{6}{4} E \Leftrightarrow 03$	$\Leftrightarrow 7.05030204561 \binom{4}{6} E \Leftrightarrow 03$
4	4	$3.23380262898 \binom{8}{7} E \Leftrightarrow 03$	$\Leftrightarrow 3.23380262898 \binom{7}{8} E \Leftrightarrow 03$
4	5	$2.0887884341 \binom{12}{09} E \Leftrightarrow 03$	$\Leftrightarrow 2.0887884341 \binom{09}{12} E \Leftrightarrow 03$
4	6	$1.89187913485 \binom{2}{0} E \Leftrightarrow 03$	$\Leftrightarrow 1.89187913485 \binom{0}{2} E \Leftrightarrow 03$
4	7	$2.50538225491 \binom{3}{2} E \Leftrightarrow 03$	$\Leftrightarrow 2.50538225491 \binom{2}{3} E \Leftrightarrow 03$
4	8	$1.160997732426 \binom{4}{2} E \Leftrightarrow 02$	$0.0000000000000000 E + 00$
5	1	$1.125829506307 \binom{1}{0} E \Leftrightarrow 01$	$\Leftrightarrow 1.125829506307 \binom{0}{1} E \Leftrightarrow 01$
5	2	$1.43896206101 \binom{2}{0} E \Leftrightarrow 02$	$\Leftrightarrow 1.43896206101 \binom{0}{2} E \Leftrightarrow 02$
5	3	$3.7016725888 \binom{4}{3} E \Leftrightarrow 03$	$\Leftrightarrow 3.7016725888 \binom{3}{4} E \Leftrightarrow 03$
5	4	$1.31743000686 \binom{5}{3} E \Leftrightarrow 03$	$\Leftrightarrow 1.31743000686 \binom{3}{5} E \Leftrightarrow 03$
5	5	$6.362993401 \binom{9}{6} E \Leftrightarrow 04$	$\Leftrightarrow 6.362993401 \binom{6}{9} E \Leftrightarrow 04$

Restgliedfaktoren der Gauß-Legendre-Formeln (Forts.)

M	r	$r! [P_r]$	$r! [N_r]$
5	6	$4.006894882 \binom{30}{27} E \Leftrightarrow 04$	$\Leftrightarrow 4.006894882 \binom{27}{29} E \Leftrightarrow 04$
5	7	$3.2888558211 \binom{4}{0} E \Leftrightarrow 04$	$\Leftrightarrow 3.2888558211 \binom{0}{4} E \Leftrightarrow 04$
5	8	$3.58564086647 \binom{7}{4} E \Leftrightarrow 04$	$\Leftrightarrow 3.5856408664 \binom{7}{9} E \Leftrightarrow 04$
5	9	$5.57270393592 \binom{7}{0} E \Leftrightarrow 04$	$\Leftrightarrow 5.57270393592 \binom{0}{7} E \Leftrightarrow 04$
5	10	$2.93181245562 \binom{3}{1} E \Leftrightarrow 03$	$0.0000000000000000 E + 00$
6	1	$9.512839671176 \binom{8}{5} E \Leftrightarrow 02$	$\Leftrightarrow 9.512839671176 \binom{5}{8} E \Leftrightarrow 02$
6	2	$1.0208633409 \binom{91}{89} E \Leftrightarrow 02$	$\Leftrightarrow 1.0208633409 \binom{89}{90} E \Leftrightarrow 02$
6	3	$2.191652812 \binom{1}{0} E \Leftrightarrow 03$	$\Leftrightarrow 2.191652812 \binom{0}{1} E \Leftrightarrow 03$
6	4	$6.4152420577 \binom{3}{1} E \Leftrightarrow 04$	$\Leftrightarrow 6.4152420577 \binom{1}{3} E \Leftrightarrow 04$
6	5	$2.5067105883 \binom{8}{4} E \Leftrightarrow 04$	$\Leftrightarrow 2.5067105883 \binom{4}{8} E \Leftrightarrow 04$
6	6	$1.2423240022 \binom{9}{7} E \Leftrightarrow 04$	$\Leftrightarrow 1.2423240022 \binom{6}{9} E \Leftrightarrow 04$
6	7	$7.697268215 \binom{9}{7} E \Leftrightarrow 05$	$\Leftrightarrow 7.697268215 \binom{7}{9} E \Leftrightarrow 05$
6	8	$5.933865104 \binom{6}{4} E \Leftrightarrow 05$	$\Leftrightarrow 5.933865104 \binom{4}{6} E \Leftrightarrow 05$
6	9	$5.747673862 \binom{4}{2} E \Leftrightarrow 05$	$\Leftrightarrow 5.747673862 \binom{2}{4} E \Leftrightarrow 05$
6	10	$7.2285933639 \binom{9}{6} E \Leftrightarrow 05$	$\Leftrightarrow 7.22859336 \binom{393}{401} E \Leftrightarrow 05$
6	11	$1.2670625483 \binom{7}{6} E \Leftrightarrow 04$	$\Leftrightarrow 1.2670625483 \binom{6}{7} E \Leftrightarrow 04$
6	12	$7.3807866015 \binom{70}{63} E \Leftrightarrow 04$	$0.0000000000000000 E + 00$
7	1	$8.238052155035 \binom{4}{1} E \Leftrightarrow 02$	$\Leftrightarrow 8.238052155035 \binom{1}{4} E \Leftrightarrow 02$
7	2	$7.62508037150 \binom{8}{5} E \Leftrightarrow 03$	$\Leftrightarrow 7.62508037150 \binom{6}{8} E \Leftrightarrow 03$
7	3	$1.4070381397 \binom{11}{09} E \Leftrightarrow 03$	$\Leftrightarrow 1.4070381397 \binom{09}{11} E \Leftrightarrow 03$
7	4	$3.5087997949 \binom{2}{0} E \Leftrightarrow 04$	$\Leftrightarrow 3.5087997949 \binom{0}{3} E \Leftrightarrow 04$
7	5	$1.157840152 \binom{2}{1} E \Leftrightarrow 04$	$\Leftrightarrow 1.157840152 \binom{1}{2} E \Leftrightarrow 04$
7	6	$4.77711538 \binom{61}{59} E \Leftrightarrow 05$	$\Leftrightarrow 4.77711538 \binom{596}{601} E \Leftrightarrow 05$
7	7	$2.417372631 \binom{9}{7} E \Leftrightarrow 05$	$\Leftrightarrow 2.417372631 \binom{7}{9} E \Leftrightarrow 05$
7	8	$1.48025476 \binom{20}{18} E \Leftrightarrow 05$	$\Leftrightarrow 1.48025476 \binom{1}{2} E \Leftrightarrow 05$
7	9	$1.092721891 \binom{3}{2} E \Leftrightarrow 05$	$\Leftrightarrow 1.092721891 \binom{2}{3} E \Leftrightarrow 05$
7	10	$9.77103353 \binom{5}{3} E \Leftrightarrow 06$	$\Leftrightarrow 9.77103353 \binom{37}{41} E \Leftrightarrow 06$
7	11	$1.0763421 \binom{9}{8} E \Leftrightarrow 05$	$\Leftrightarrow 1.0763421 \binom{8}{9} E \Leftrightarrow 05$
7	12	$1.513085200 \binom{5}{4} E \Leftrightarrow 05$	$\Leftrightarrow 1.51308520 \binom{0}{1} E \Leftrightarrow 05$
7	13	$2.9255381 \binom{701}{699} E \Leftrightarrow 05$	$\Leftrightarrow 2.9255381 \binom{699}{701} E \Leftrightarrow 05$
7	14	$1.8546591973 \binom{3}{1} E \Leftrightarrow 04$	$0.0000000000000000 E + 00$

Restgliedfaktoren der Gauß-Legendre-Formeln (Forts.)

M	r	$r! [P_r]$	$r! [N_r]$
8	1	$7.26547519371 \binom{11}{08} E \Leftrightarrow 02$	$\Leftrightarrow 7.26547519371 \binom{08}{11} E \Leftrightarrow 02$
8	2	$5.9147750865 \binom{81}{77} E \Leftrightarrow 03$	$\Leftrightarrow 5.9147750865 \binom{78}{80} E \Leftrightarrow 03$
8	3	$9.5787033421 \binom{3}{0} E \Leftrightarrow 04$	$\Leftrightarrow 9.5787033421 \binom{0}{3} E \Leftrightarrow 04$
8	4	$2.0842646798 \binom{3}{1} E \Leftrightarrow 04$	$\Leftrightarrow 2.0842646798 \binom{0}{3} E \Leftrightarrow 04$
8	5	$5.969542487 \binom{8}{6} E \Leftrightarrow 05$	$\Leftrightarrow 5.969542487 \binom{6}{8} E \Leftrightarrow 05$
8	6	$2.119450374 \binom{9}{7} E \Leftrightarrow 05$	$\Leftrightarrow 2.119450374 \binom{7}{9} E \Leftrightarrow 05$
8	7	$9.13119978 \binom{9}{7} E \Leftrightarrow 06$	$\Leftrightarrow 9.13119978 \binom{7}{9} E \Leftrightarrow 06$
8	8	$4.6914810 \binom{10}{08} E \Leftrightarrow 06$	$\Leftrightarrow 4.6914810 \binom{08}{10} E \Leftrightarrow 06$
8	9	$2.84869914 \binom{9}{7} E \Leftrightarrow 06$	$\Leftrightarrow 2.84869914 \binom{7}{9} E \Leftrightarrow 06$
8	10	$2.03693234 \binom{8}{6} E \Leftrightarrow 06$	$\Leftrightarrow 2.03693234 \binom{6}{8} E \Leftrightarrow 06$
8	11	$1.718980 \binom{3}{1} E \Leftrightarrow 06$	$\Leftrightarrow 1.718980 \binom{1}{3} E \Leftrightarrow 06$
8	12	$1.72815254 \binom{8}{6} E \Leftrightarrow 06$	$\Leftrightarrow 1.72815254 \binom{6}{8} E \Leftrightarrow 06$
8	13	$2.1105611 \binom{1}{0} E \Leftrightarrow 06$	$\Leftrightarrow 2.1105611 \binom{0}{1} E \Leftrightarrow 06$
8	14	$3.253173196 \binom{3}{0} E \Leftrightarrow 06$	$\Leftrightarrow 3.25317319 \binom{5}{7} E \Leftrightarrow 06$
8	15	$6.8294665 \binom{81}{79} E \Leftrightarrow 06$	$\Leftrightarrow 6.8294665 \binom{79}{81} E \Leftrightarrow 06$
8	16	$4.654830926 \binom{7}{5} E \Leftrightarrow 05$	$0.0000000000000000 E + 00$
9	1	$6.498745348793 \binom{8}{6} E \Leftrightarrow 02$	$\Leftrightarrow 6.498745348793 \binom{6}{8} E \Leftrightarrow 02$
9	2	$4.72312020667 \binom{8}{2} E \Leftrightarrow 03$	$\Leftrightarrow 4.72312020667 \binom{3}{7} E \Leftrightarrow 03$
9	3	$6.8185108495 \binom{5}{3} E \Leftrightarrow 04$	$\Leftrightarrow 6.8185108495 \binom{3}{5} E \Leftrightarrow 04$
9	4	$1.3172995041 \binom{4}{1} E \Leftrightarrow 04$	$\Leftrightarrow 1.317299504 \binom{1}{2} E \Leftrightarrow 04$
9	5	$3.33831851 \binom{41}{38} E \Leftrightarrow 05$	$\Leftrightarrow 3.33831851 \binom{38}{41} E \Leftrightarrow 05$
9	6	$1.04282 \binom{5}{4} E \Leftrightarrow 05$	$\Leftrightarrow 1.04282447 \binom{88}{90} E \Leftrightarrow 05$
9	7	$3.926616 \binom{60}{55} E \Leftrightarrow 06$	$\Leftrightarrow 3.926616 E \binom{55}{60} E \Leftrightarrow 06$
9	8	$1.7477208 \binom{3}{2} E \Leftrightarrow 06$	$\Leftrightarrow 1.7477208 \binom{2}{3} E \Leftrightarrow 06$
9	9	$9.0898058 \binom{4}{0} E \Leftrightarrow 07$	$\Leftrightarrow 9.0898058 \binom{0}{4} E \Leftrightarrow 07$
9	10	$5.4843309 \binom{4}{1} E \Leftrightarrow 07$	$\Leftrightarrow 5.4843 \binom{29}{31} E \Leftrightarrow 07$
9	11	$3.826695 \binom{5}{4} E \Leftrightarrow 07$	$\Leftrightarrow 3.826695 \binom{4}{5} E \Leftrightarrow 07$
9	12	$3.090481 \binom{2}{0} E \Leftrightarrow 07$	$\Leftrightarrow 3.0904811 \binom{5}{9} E \Leftrightarrow 07$
9	13	$2.904707 \binom{1}{0} E \Leftrightarrow 07$	$\Leftrightarrow 2.904707 \binom{0}{1} E \Leftrightarrow 07$
9	14	$3.214531 \binom{6}{4} E \Leftrightarrow 07$	$\Leftrightarrow 3.21453 \binom{1}{2} E \Leftrightarrow 07$
9	15	$4.280204 \binom{4}{1} E \Leftrightarrow 07$	$\Leftrightarrow 4.280204 \binom{1}{4} E \Leftrightarrow 07$
9	16	$7.13372165 \binom{5}{3} E \Leftrightarrow 07$	$\Leftrightarrow 7.1337216 \binom{4}{7} E \Leftrightarrow 07$
9	17	$1.6076081 \binom{5}{3} E \Leftrightarrow 06$	$\Leftrightarrow 1.6076081 \binom{3}{5} E \Leftrightarrow 06$
9	18	$1.16731054 \binom{3}{1} E \Leftrightarrow 05$	$0.0000000000000000 E + 00$

Restgliedfaktoren der Gauß-Legendre-Formeln (Forts.)

M	r	$r! [P_r]$	$r! [N_r]$
10	1	$5.878635205337 \binom{4}{2} E \Leftrightarrow 02$	$\Leftrightarrow 5.878635205337 \binom{2}{4} E \Leftrightarrow 02$
10	2	$3.85926976046 \binom{5}{0} E \Leftrightarrow 03$	$\Leftrightarrow 3.85926976046 \binom{1}{4} E \Leftrightarrow 03$
10	3	$5.0274739573 \binom{4}{1} E \Leftrightarrow 04$	$\Leftrightarrow 5.0274739573 \binom{1}{4} E \Leftrightarrow 04$
10	4	$8.73914888 \binom{90}{87} E \Leftrightarrow 05$	$\Leftrightarrow 8.73914888 \binom{87}{91} E \Leftrightarrow 05$
10	5	$1.98795273 \binom{8}{6} E \Leftrightarrow 05$	$\Leftrightarrow 1.98795273 \binom{6}{8} E \Leftrightarrow 05$
10	6	$5.55227032 \binom{8}{5} E \Leftrightarrow 06$	$\Leftrightarrow 5.55227032 \binom{5}{7} E \Leftrightarrow 06$
10	7	$1.86083533 \binom{4}{2} E \Leftrightarrow 06$	$\Leftrightarrow 1.86083533 \binom{2}{4} E \Leftrightarrow 06$
10	8	$7.3292574 \binom{7}{4} E \Leftrightarrow 07$	$\Leftrightarrow 7.3292574 \binom{4}{7} E \Leftrightarrow 07$
10	9	$3.3489199 \binom{8}{6} E \Leftrightarrow 07$	$\Leftrightarrow 3.3489199 \binom{6}{8} E \Leftrightarrow 07$
10	10	$1.75900 \binom{50}{48} E \Leftrightarrow 07$	$\Leftrightarrow 1.759004 \binom{89}{91} E \Leftrightarrow 07$
10	11	$1.056043 \binom{7}{6} E \Leftrightarrow 07$	$\Leftrightarrow 1.056043 \binom{6}{7} E \Leftrightarrow 07$
10	12	$7.22688 \binom{95}{89} E \Leftrightarrow 08$	$\Leftrightarrow 7.226889 \binom{2}{4} E \Leftrightarrow 08$
10	13	$5.63804 \binom{3}{2} E \Leftrightarrow 08$	$\Leftrightarrow 5.63804 \binom{2}{3} E \Leftrightarrow 08$
10	14	$5.030959 \binom{3}{1} E \Leftrightarrow 08$	$\Leftrightarrow 5.0309 \binom{58}{60} E \Leftrightarrow 08$
10	15	$5.1726 \binom{41}{39} E \Leftrightarrow 08$	$\Leftrightarrow 5.1726 \binom{39}{41} E \Leftrightarrow 08$
10	16	$6.20952 \binom{9}{7} E \Leftrightarrow 08$	$\Leftrightarrow 6.209527 \binom{85}{95} E \Leftrightarrow 08$
10	17	$8.904119 \binom{7}{5} E \Leftrightarrow 08$	$\Leftrightarrow 8.904119 \binom{5}{7} E \Leftrightarrow 08$
10	18	$1.58824075 \binom{9}{6} E \Leftrightarrow 07$	$\Leftrightarrow 1.588240 \binom{7}{8} E \Leftrightarrow 07$
10	19	$3.808900 \binom{12}{08} E \Leftrightarrow 07$	$\Leftrightarrow 3.808900 \binom{08}{12} E \Leftrightarrow 07$
10	20	$2.92559033 \binom{2}{0} E \Leftrightarrow 06$	$0.0000000000000000 E + 00$

Bemerkung 3.4.2 Für 108 Kernfunktionen benötigt die Berechnung der obigen Fehlerkonstanten auf der Maschine Sun 5/110 circa 29.89 CPU Sekunden, wenn in dem Intervall-Newton-Schritt die Mittelwertform (2.21) eingesetzt wird. Ohne diesen Einsatz wird die Rechenleistung um das 4-fache langsamer. Die Effizienz steht für die Bestimmung der Fehlerkonstanten zwar nicht im Vordergrund, dennoch erweist die Recheneffektivität, daß mit dem Ansatz aus der Verifikationsnumerik diese mühsame Aufgabe recht wirkungsvoll gelöst werden kann.

Tabelle 3.3: Vergleich der Restgliedfaktoren $r! \int_{-1}^1 K_r^+$ der Gauß-Legendre-Formeln

		$r![P_r]$		
M	r	Verifikation	STROUD & SECREST (LETHER [101])	STORCK [144]
2	1	$2.5598306414370 \binom{8}{7} E \Leftrightarrow 01$	$2.560 E \Leftrightarrow 01$	
2	2	$8.112906564241 \binom{82}{8} E \Leftrightarrow 02$	$8.115 E \Leftrightarrow 02$	
2	3	$5.754991027012 \binom{6}{4} E \Leftrightarrow 02$	($5.76 E \Leftrightarrow 02$)	
2	4	$1.77777777777777 \binom{9}{7} E \Leftrightarrow 01$	$1.778 E \Leftrightarrow 01$	
3	1	$1.78669008626 \binom{500}{498} E \Leftrightarrow 01$	$1.787 E \Leftrightarrow 01$	
3	2	$3.74355436143 \binom{32}{29} E \Leftrightarrow 02$	$3.744 E \Leftrightarrow 02$	
3	4	$1.09059248140 \binom{40}{39} E \Leftrightarrow 02$	$1.091 E \Leftrightarrow 02$	
3	5	$1.17473328183 \binom{8}{6} E \Leftrightarrow 02$	($1.18 E \Leftrightarrow 02$)	
3	6	$4.571428571428 \binom{7}{5} E \Leftrightarrow 02$	$4.571 E \Leftrightarrow 02$	
4	1	$1.3799698495825 \binom{8}{7} E \Leftrightarrow 01$	$1.380 E \Leftrightarrow 01$	
4	2	$2.185660935053 \binom{2}{0} E \Leftrightarrow 02$	$2.186 E \Leftrightarrow 02$	
4	4	$3.23380262898 \binom{8}{7} E \Leftrightarrow 03$	$3.235 E \Leftrightarrow 03$	
4	7	$2.50538225491 \binom{3}{2} E \Leftrightarrow 03$	($2.51 E \Leftrightarrow 03$)	
4	8	$1.160997732426 \binom{4}{2} E \Leftrightarrow 02$	$1.161 E \Leftrightarrow 02$	
5	1	$1.125829506307 \binom{1}{0} E \Leftrightarrow 01$	$1.126 E \Leftrightarrow 01$	
5	2	$1.43896206101 \binom{2}{0} E \Leftrightarrow 02$	$1.439 E \Leftrightarrow 02$	
5	4	$1.31743000686 \binom{5}{3} E \Leftrightarrow 03$	$1.318 E \Leftrightarrow 03$	
5	8	$3.58564086647 \binom{7}{4} E \Leftrightarrow 04$	$3.586 E \Leftrightarrow 04$	
5	9	$5.57270393592 \binom{7}{0} E \Leftrightarrow 04$	($5.57 E \Leftrightarrow 04$)	
5	10	$2.93181245562 \binom{3}{1} E \Leftrightarrow 03$	$2.932 E \Leftrightarrow 03$	
6	1	$9.512839671176 \binom{8}{5} E \Leftrightarrow 02$	$9.515 E \Leftrightarrow 02$	
6	2	$1.0208633409 \binom{91}{89} E \Leftrightarrow 02$	$1.021 E \Leftrightarrow 02$	$1.02086 \binom{5}{3} E \Leftrightarrow 02$
6	4	$6.4152420577 \binom{3}{1} E \Leftrightarrow 04$	$6.420 E \Leftrightarrow 04$	$6.41 \binom{7}{5} E \Leftrightarrow 04$
6	6	$1.2423240022 \binom{9}{7} E \Leftrightarrow 04$		$1. \binom{3}{2} E \Leftrightarrow 04$
6	8	$5.933865104 \binom{6}{4} E \Leftrightarrow 05$	$5.935 E \Leftrightarrow 05$	
6	11	$1.2670625483 \binom{7}{6} E \Leftrightarrow 04$	($1.27 E \Leftrightarrow 04$)	
6	12	$7.3807866015 \binom{70}{63} E \Leftrightarrow 04$	$7.381 E \Leftrightarrow 04$	

Bemerkung 3.4.3 In [101] hat Lether bewiesen, daß für $r = 2M \Leftrightarrow 1$ die Kernfunktionen $K_r(t)$ der Gauß-Legendre-Formeln nur eine einzige Nullstelle $t = 0$ in $(\Leftrightarrow 1, 1)$ besitzen. Daher läßt sich $\|K_r\|_1$, $r=2M-1$, direkt berechnen.

Vergleich der Restgliedfaktoren der Gauss-Legendre-Formeln (Forts.)

		$r![P_r]$		
M	r	Verifikation	STROUD & SECREST (LETHER [101])	STORCK [144]
7	1	$8.238052155035 \binom{4}{1} E \Leftrightarrow 02$	$8.240 E \Leftrightarrow 02$	
7	2	$7.62508037150 \binom{8}{5} E \Leftrightarrow 03$	$7.625 E \Leftrightarrow 03$	
7	4	$3.5087997949 \binom{2}{0} E \Leftrightarrow 04$	$3.508 E \Leftrightarrow 04$	
7	8	$1.48025476 \binom{20}{18} E \Leftrightarrow 05$	$1.480 E \Leftrightarrow 05$	
7	13	$2.9255381 \binom{701}{699} E \Leftrightarrow 05$	($2.93 E \Leftrightarrow 05$)	
7	14	$1.8546591973 \binom{3}{1} E \Leftrightarrow 04$	$1.855 E \Leftrightarrow 04$	
8	1	$7.26547519371 \binom{11}{08} E \Leftrightarrow 02$	$7.265 E \Leftrightarrow 02$	
8	2	$5.9147750865 \binom{81}{77} E \Leftrightarrow 03$	$5.915 E \Leftrightarrow 03$	$5.914775086 \binom{8}{5} E \Leftrightarrow 03$
8	4	$2.0842646798 \binom{3}{1} E \Leftrightarrow 04$	$2.085 E \Leftrightarrow 04$	$2.0842 \binom{8}{6} E \Leftrightarrow 04$
8	6	$2.119450374 \binom{9}{7} E \Leftrightarrow 05$		$2. \binom{3}{1} E \Leftrightarrow 05$
8	8	$4.6914810 \binom{10}{08} E \Leftrightarrow 06$	$4.693 E \Leftrightarrow 06$	
8	15	$6.8294665 \binom{81}{79} E \Leftrightarrow 06$	($6.80 E \Leftrightarrow 06$)	
8	16	$4.654830926 \binom{7}{5} E \Leftrightarrow 05$	$4.655 E \Leftrightarrow 05$	
9	1	$6.498745348793 \binom{8}{6} E \Leftrightarrow 02$	$6.500 E \Leftrightarrow 02$	
9	2	$4.72312020667 \binom{8}{2} E \Leftrightarrow 03$	$4.723 E \Leftrightarrow 03$	
9	4	$1.3172995041 \binom{4}{1} E \Leftrightarrow 04$	$1.317 E \Leftrightarrow 04$	
9	8	$1.7477208 \binom{3}{2} E \Leftrightarrow 06$	$1.746 E \Leftrightarrow 06$	
9	16	$7.13372165 \binom{5}{3} E \Leftrightarrow 07$	$7.130 E \Leftrightarrow 07$	
9	17	$1.6076081 \binom{5}{3} E \Leftrightarrow 06$	($1.61 E \Leftrightarrow 06$)	
9	18	$1.16731054 \binom{3}{1} E \Leftrightarrow 05$	$1.167 E \Leftrightarrow 05$	
10	1	$5.878635205337 \binom{4}{2} E \Leftrightarrow 02$	$5.880 E \Leftrightarrow 02$	
10	2	$3.85926976046 \binom{5}{0} E \Leftrightarrow 03$	$3.860 E \Leftrightarrow 03$	$3.859 \binom{4}{2} E \Leftrightarrow 03$
10	4	$8.73914888 \binom{90}{87} E \Leftrightarrow 05$	$8.705 E \Leftrightarrow 05$	$8. \binom{9}{7} E \Leftrightarrow 05$
10	6	$5.55227032 \binom{8}{5} E \Leftrightarrow 06$		$\binom{6,8}{5,5} E \Leftrightarrow 06$
10	8	$7.3292574 \binom{7}{4} E \Leftrightarrow 07$	$7.330 E \Leftrightarrow 07$	
10	16	$6.20952 \binom{9}{7} E \Leftrightarrow 08$	$6.215 E \Leftrightarrow 08$	
10	19	$3.808900 \binom{12}{08} E \Leftrightarrow 07$	($3.81 E \Leftrightarrow 07$)	
10	20	$2.92559033 \binom{2}{0} E \Leftrightarrow 06$	$2.926 E \Leftrightarrow 06$	

Bemerkung 3.4.4 Für $1 \leq r < d + 1$ gilt stets $\|K_r\|_1 = 2P_r$. Da für Gauß-Legendre-Formeln K_{d+1} positiv ist, gilt $\|K_{d+1}\|_1 = P_{d+1}$. Stroud und Secrest [147] haben $r! \|K_r\|_1$ berechnet.

Tabelle 3.4: Restgliedfaktoren $r! f_{-1}^1 K_r^+$ und $r! f_{-1}^1 K_r^-$ der Clenshaw-Curtis-Formeln

M	r	$r! [P_r]$	$r! [N_r]$
4	1	$2.067901234567 \binom{91}{89} E \Leftrightarrow 01$	$\Leftrightarrow 2.067901234567 \binom{89}{91} E \Leftrightarrow 01$
4	2	$5.304069501600 \binom{5}{3} E \Leftrightarrow 02$	$\Leftrightarrow 5.304069501600 \binom{3}{5} E \Leftrightarrow 02$
4	3	$2.983539094650 \binom{3}{1} E \Leftrightarrow 02$	$\Leftrightarrow 2.983539094650 \binom{1}{3} E \Leftrightarrow 02$
4	4	$6.84008196582 \binom{504}{496} E \Leftrightarrow 02$	$\Leftrightarrow 1.734152991583 \binom{2}{3} E \Leftrightarrow 03$
5	1	$1.5496873619571 \binom{5}{4} E \Leftrightarrow 01$	$\Leftrightarrow 1.5496873619571 \binom{4}{5} E \Leftrightarrow 01$
5	2	$2.846674953941 \binom{9}{6} E \Leftrightarrow 02$	$\Leftrightarrow 2.846674953941 \binom{7}{8} E \Leftrightarrow 02$
5	3	$1.093620989454 \binom{9}{7} E \Leftrightarrow 02$	$\Leftrightarrow 1.093620989454 \binom{7}{9} E \Leftrightarrow 02$
5	4	$6.30018223588 \binom{82}{78} E \Leftrightarrow 03$	$\Leftrightarrow 6.3001822358 \binom{87}{89} E \Leftrightarrow 03$
5	5	$5.81066503674 \binom{4}{2} E \Leftrightarrow 03$	$\Leftrightarrow 5.81066503674 \binom{2}{4} E \Leftrightarrow 03$
5	6	$1.912748923633 \binom{5}{4} E \Leftrightarrow 02$	$\Leftrightarrow 7.98701887155 \binom{89}{92} E \Leftrightarrow 05$
6	1	$1.240629401369 \binom{9}{7} E \Leftrightarrow 01$	$\Leftrightarrow 1.240629401369 \binom{7}{9} E \Leftrightarrow 01$
6	2	$1.77329139838 \binom{1}{0} E \Leftrightarrow 02$	$\Leftrightarrow 1.77329139838 \binom{0}{1} E \Leftrightarrow 02$
6	3	$5.1376007135 \binom{91}{88} E \Leftrightarrow 03$	$\Leftrightarrow 5.1376007135 \binom{88}{91} E \Leftrightarrow 03$
6	4	$2.08803109031 \binom{3}{1} E \Leftrightarrow 03$	$\Leftrightarrow 2.08803109031 \binom{1}{3} E \Leftrightarrow 03$
6	5	$1.20616921636 \binom{6}{4} E \Leftrightarrow 03$	$\Leftrightarrow 1.20616921636 \binom{4}{6} E \Leftrightarrow 03$
6	6	$2.40365781924 \binom{8}{6} E \Leftrightarrow 03$	$\Leftrightarrow 2.270543829 \binom{3}{6} E \Leftrightarrow 05$
7	1	$1.032498143996 \binom{4}{3} E \Leftrightarrow 01$	$\Leftrightarrow 1.032498143996 \binom{3}{4} E \Leftrightarrow 01$
7	2	$1.21350444789 \binom{3}{2} E \Leftrightarrow 02$	$\Leftrightarrow 1.21350444789 \binom{2}{3} E \Leftrightarrow 02$
7	3	$2.860324310 \binom{900}{898} E \Leftrightarrow 03$	$\Leftrightarrow 2.860324310 \binom{898}{900} E \Leftrightarrow 03$
7	4	$9.2413622546 \binom{7}{5} E \Leftrightarrow 04$	$\Leftrightarrow 9.2413622546 \binom{4}{7} E \Leftrightarrow 04$
7	5	$4.0196109422 \binom{7}{5} E \Leftrightarrow 04$	$\Leftrightarrow 4.0196109422 \binom{5}{7} E \Leftrightarrow 04$
7	6	$2.2736592671 \binom{4}{2} E \Leftrightarrow 04$	$\Leftrightarrow 2.2736592671 \binom{2}{4} E \Leftrightarrow 04$
7	7	$1.766177611 \binom{80}{78} E \Leftrightarrow 04$	$\Leftrightarrow 1.766177611 \binom{78}{80} E \Leftrightarrow 04$
7	8	$7.93702114965 \binom{8}{0} E \Leftrightarrow 04$	$\Leftrightarrow 5.132131450572 \binom{2}{4} E \Leftrightarrow 08$
8	1	$8.844221753772 \binom{6}{2} E \Leftrightarrow 02$	$\Leftrightarrow 8.844221753772 \binom{2}{6} E \Leftrightarrow 02$
8	2	$8.83408561898 \binom{7}{3} E \Leftrightarrow 03$	$\Leftrightarrow 8.83408561898 \binom{3}{7} E \Leftrightarrow 03$
8	3	$1.760461607 \binom{30}{28} E \Leftrightarrow 03$	$\Leftrightarrow 1.760461607 \binom{28}{30} E \Leftrightarrow 03$
8	4	$4.75573543 \binom{601}{598} E \Leftrightarrow 04$	$\Leftrightarrow 4.75573543 \binom{598}{601} E \Leftrightarrow 04$
8	5	$1.7085796017 \binom{6}{3} E \Leftrightarrow 04$	$\Leftrightarrow 1.7085796017 \binom{3}{6} E \Leftrightarrow 04$
8	6	$7.824840341 \binom{8}{6} E \Leftrightarrow 05$	$\Leftrightarrow 7.824840341 \binom{6}{8} E \Leftrightarrow 05$
8	7	$5.037383811 \binom{5}{3} E \Leftrightarrow 05$	$\Leftrightarrow 5.037383811 \binom{3}{5} E \Leftrightarrow 05$
8	8	$1.9841511680 \binom{3}{1} E \Leftrightarrow 04$	$\Leftrightarrow 2.418388871452 \binom{6}{8} E \Leftrightarrow 09$

Restgliedfaktoren der Clenshaw-Curtis-Formeln (Forts.)

M	r	$r![P_r]$	$r![N_r]$
9	1	$7.73315230513 \binom{81}{78} E \Leftrightarrow 02$	$\Leftrightarrow 7.73315230513 \binom{78}{81} E \Leftrightarrow 02$
9	2	$6.7240498247 \binom{62}{59} E \Leftrightarrow 03$	$\Leftrightarrow 6.7240498247 \binom{59}{61} E \Leftrightarrow 03$
9	3	$1.16326354702 \binom{8}{6} E \Leftrightarrow 03$	$\Leftrightarrow 1.16326354702 \binom{6}{8} E \Leftrightarrow 03$
9	4	$2.7094985515 \binom{9}{7} E \Leftrightarrow 04$	$\Leftrightarrow 2.709498551 \binom{57}{60} E \Leftrightarrow 04$
9	5	$8.331816916 \binom{8}{6} E \Leftrightarrow 05$	$\Leftrightarrow 8.331816916 \binom{6}{8} E \Leftrightarrow 05$
9	6	$3.20083498 \binom{90}{88} E \Leftrightarrow 05$	$\Leftrightarrow 3.200834988 \binom{87}{92} E \Leftrightarrow 05$
9	7	$1.526665430 \binom{8}{6} E \Leftrightarrow 05$	$\Leftrightarrow 1.526665430 \binom{6}{8} E \Leftrightarrow 05$
9	8	$9.642888888 \binom{5}{1} E \Leftrightarrow 06$	$\Leftrightarrow 9.64288888 \binom{79}{88} E \Leftrightarrow 06$
9	9	$1.341969299 \binom{8}{7} E \Leftrightarrow 05$	$\Leftrightarrow 1.341969299 \binom{7}{8} E \Leftrightarrow 05$
9	10	$7.215010731 \binom{82}{75} E \Leftrightarrow 05$	$\Leftrightarrow 3.51678290921 \binom{68}{71} E \Leftrightarrow 11$
10	1	$6.87077103753 \binom{71}{67} E \Leftrightarrow 02$	$\Leftrightarrow 6.87077103753 \binom{67}{71} E \Leftrightarrow 02$
10	2	$5.29115936615 \binom{6}{1} E \Leftrightarrow 03$	$\Leftrightarrow 5.29115936615 \binom{1}{5} E \Leftrightarrow 03$
10	3	$8.0934542827 \binom{8}{4} E \Leftrightarrow 04$	$\Leftrightarrow 8.0934542827 \binom{4}{8} E \Leftrightarrow 04$
10	4	$1.659061048 \binom{7}{6} E \Leftrightarrow 04$	$\Leftrightarrow 1.659061048 \binom{6}{7} E \Leftrightarrow 04$
10	5	$4.469057064 \binom{4}{1} E \Leftrightarrow 05$	$\Leftrightarrow 4.469057064 \binom{1}{4} E \Leftrightarrow 05$
10	6	$1.49071518 \binom{6}{4} E \Leftrightarrow 05$	$\Leftrightarrow 1.49071518 \binom{4}{6} E \Leftrightarrow 05$
10	7	$6.08543835 \binom{4}{2} E \Leftrightarrow 06$	$\Leftrightarrow 6.08543835 \binom{2}{4} E \Leftrightarrow 06$
10	8	$3.30890720 \binom{4}{1} E \Leftrightarrow 06$	$\Leftrightarrow 3.30890720 \binom{2}{4} E \Leftrightarrow 06$
10	9	$3.44630893 \binom{7}{5} E \Leftrightarrow 06$	$\Leftrightarrow 3.44630893 \binom{5}{7} E \Leftrightarrow 06$
10	10	$2.2546899 \binom{701}{699} E \Leftrightarrow 05$	$\Leftrightarrow 2.15281260569 \binom{29}{31} E \Leftrightarrow 12$
11	1	$6.18116236166 \binom{6}{5} E \Leftrightarrow 02$	$\Leftrightarrow 6.18116236166 \binom{5}{6} E \Leftrightarrow 02$
11	2	$4.27347094122 \binom{7}{1} E \Leftrightarrow 03$	$\Leftrightarrow 4.27347094122 \binom{2}{6} E \Leftrightarrow 03$
11	3	$5.8620633896 \binom{9}{6} E \Leftrightarrow 04$	$\Leftrightarrow 5.8620633896 \binom{6}{9} E \Leftrightarrow 04$
11	4	$1.074143057 \binom{7}{6} E \Leftrightarrow 04$	$\Leftrightarrow 1.074143057 \binom{6}{7} E \Leftrightarrow 04$
11	5	$2.579109676 \binom{7}{4} E \Leftrightarrow 05$	$\Leftrightarrow 2.579109676 \binom{4}{7} E \Leftrightarrow 05$
11	6	$7.62618029 \binom{7}{4} E \Leftrightarrow 06$	$\Leftrightarrow 7.62618029 \binom{5}{7} E \Leftrightarrow 06$
11	7	$2.72460158 \binom{8}{6} E \Leftrightarrow 06$	$\Leftrightarrow 2.72460158 \binom{6}{8} E \Leftrightarrow 06$
11	8	$1.17242353 \binom{3}{1} E \Leftrightarrow 06$	$\Leftrightarrow 1.17242353 \binom{1}{3} E \Leftrightarrow 06$
11	9	$6.7248045 \binom{9}{6} E \Leftrightarrow 07$	$\Leftrightarrow 6.7248045 \binom{6}{9} E \Leftrightarrow 07$
11	10	$7.46062208 \binom{3}{1} E \Leftrightarrow 07$	$\Leftrightarrow 7.4606220 \binom{78}{87} E \Leftrightarrow 07$
11	11	$1.2503329 \binom{7}{6} E \Leftrightarrow 06$	$\Leftrightarrow 1.2503329 \binom{6}{7} E \Leftrightarrow 06$
11	12	$8.67188369 \binom{36}{28} E \Leftrightarrow 06$	$\Leftrightarrow 2.1393218558 \binom{29}{31} E \Leftrightarrow 14$

Restgliedfaktoren der Clenshaw-Curtis-Formeln (Forts.)

M	r	$r![P_r]$	$r![N_r]$
12	1	$5.61759919979 \binom{5}{3} E \Leftrightarrow 02$	$\Leftrightarrow 5.61759919979 \binom{3}{5} E \Leftrightarrow 02$
12	2	$3.5241778131 \binom{3}{2} E \Leftrightarrow 03$	$\Leftrightarrow 3.5241778131 \binom{2}{3} E \Leftrightarrow 03$
12	3	$4.383179339 \binom{70}{66} E \Leftrightarrow 04$	$\Leftrightarrow 4.383179339 \binom{66}{70} E \Leftrightarrow 04$
12	4	$7.26445130 \binom{82}{78} E \Leftrightarrow 05$	$\Leftrightarrow 7.26445130 \binom{78}{82} E \Leftrightarrow 05$
12	5	$1.57422192 \binom{5}{3} E \Leftrightarrow 05$	$\Leftrightarrow 1.57422192 \binom{3}{5} E \Leftrightarrow 05$
12	6	$4.18405576 \binom{9}{6} E \Leftrightarrow 06$	$\Leftrightarrow 4.18405576 \binom{6}{9} E \Leftrightarrow 06$
12	7	$1.33531478 \binom{2}{0} E \Leftrightarrow 06$	$\Leftrightarrow 1.33531478 \binom{0}{2} E \Leftrightarrow 06$
12	8	$5.0713701 \binom{3}{0} E \Leftrightarrow 07$	$\Leftrightarrow 5.0713701 \binom{0}{3} E \Leftrightarrow 07$
12	9	$2.4998699 \binom{2}{0} E \Leftrightarrow 07$	$\Leftrightarrow 2.4998699 \binom{0}{2} E \Leftrightarrow 07$
12	10	$2.016975 \binom{4}{3} E \Leftrightarrow 07$	$\Leftrightarrow 2.016975 \binom{3}{4} E \Leftrightarrow 07$
12	11	$4.5774853 \binom{4}{0} E \Leftrightarrow 07$	$\Leftrightarrow 4.5774853 \binom{0}{4} E \Leftrightarrow 07$
12	12	$3.03515928 \binom{8}{6} E \Leftrightarrow 06$	$\Leftrightarrow 1.5309505475 \binom{798}{800} E \Leftrightarrow 15$
13	1	$5.148167635817 \binom{4}{0} E \Leftrightarrow 02$	$\Leftrightarrow 5.148167635817 \binom{0}{4} E \Leftrightarrow 02$
13	2	$2.9564684646 \binom{1}{0} E \Leftrightarrow 03$	$\Leftrightarrow 2.95646846460 \binom{4}{8} E \Leftrightarrow 03$
13	3	$3.364147284 \binom{21}{18} E \Leftrightarrow 04$	$\Leftrightarrow 3.364147284 \binom{18}{21} E \Leftrightarrow 04$
13	4	$5.09175374 \binom{40}{37} E \Leftrightarrow 05$	$\Leftrightarrow 5.09175374 \binom{37}{40} E \Leftrightarrow 05$
13	5	$1.00619993 \binom{5}{4} E \Leftrightarrow 05$	$\Leftrightarrow 1.00619993 \binom{4}{5} E \Leftrightarrow 05$
13	6	$2.4325726 \binom{42}{39} E \Leftrightarrow 06$	$\Leftrightarrow 2.4325726 \binom{39}{42} E \Leftrightarrow 06$
13	7	$7.0374032 \binom{8}{5} E \Leftrightarrow 07$	$\Leftrightarrow 7.0374032 \binom{5}{8} E \Leftrightarrow 07$
13	8	$2.3956997 \binom{3}{1} E \Leftrightarrow 07$	$\Leftrightarrow 2.3956997 \binom{1}{3} E \Leftrightarrow 07$
13	9	$9.735972 \binom{7}{5} E \Leftrightarrow 08$	$\Leftrightarrow 9.735972 \binom{5}{7} E \Leftrightarrow 08$
13	10	$5.337373 \binom{5}{3} E \Leftrightarrow 08$	$\Leftrightarrow 5.337373 \binom{3}{5} E \Leftrightarrow 08$
13	11	$5.576316 \binom{9}{6} E \Leftrightarrow 08$	$\Leftrightarrow 5.576316 \binom{6}{9} E \Leftrightarrow 08$
13	12	$7.6441745 \binom{5}{2} E \Leftrightarrow 08$	$\Leftrightarrow 7.644174 \binom{48}{58} E \Leftrightarrow 08$
13	13	$1.735909 \binom{4}{2} E \Leftrightarrow 07$	$\Leftrightarrow 1.735909 \binom{2}{4} E \Leftrightarrow 07$
13	14	$1.21406371 \binom{5}{3} E \Leftrightarrow 06$	$\Leftrightarrow 1.10405171463 \binom{0}{1} E \Leftrightarrow 17$

Restgliedfaktoren der Clenshaw-Curtis-Formeln (Forts.)

M	r	$r![P_r]$	$r![N_r]$
14	1	$4.75124380654 \binom{4}{2} E \Leftrightarrow 02$	$\Leftrightarrow 4.75124380654 \binom{2}{4} E \Leftrightarrow 02$
14	2	$2.5159097293 \binom{3}{1} E \Leftrightarrow 03$	$\Leftrightarrow 2.5159097293 \binom{1}{3} E \Leftrightarrow 03$
14	3	$2.638644472 \binom{9}{7} E \Leftrightarrow 04$	$\Leftrightarrow 2.638644472 \binom{7}{9} E \Leftrightarrow 04$
14	4	$3.675607711 \binom{6}{2} E \Leftrightarrow 05$	$\Leftrightarrow 3.675607711 \binom{2}{6} E \Leftrightarrow 05$
14	5	$6.67708337 \binom{4}{1} E \Leftrightarrow 07$	$\Leftrightarrow 6.67708337 \binom{1}{4} E \Leftrightarrow 07$
14	6	$1.4806845 \binom{6}{5} E \Leftrightarrow 06$	$\Leftrightarrow 1.4806845 \binom{5}{6} E \Leftrightarrow 06$
14	7	$3.9172373 \binom{4}{1} E \Leftrightarrow 07$	$\Leftrightarrow 3.9172373 \binom{1}{4} E \Leftrightarrow 07$
14	8	$1.212006 \binom{1}{0} E \Leftrightarrow 07$	$\Leftrightarrow 1.212006 \binom{0}{1} E \Leftrightarrow 07$
14	9	$4.40626 \binom{3}{2} E \Leftrightarrow 08$	$\Leftrightarrow 4.40626 \binom{2}{3} E \Leftrightarrow 08$
14	10	$2.0672 \binom{10}{08} E \Leftrightarrow 08$	$\Leftrightarrow 2.0672 \binom{08}{10} E \Leftrightarrow 08$
14	11	$1.6140 \binom{8}{6} E \Leftrightarrow 08$	$\Leftrightarrow 1.6140 \binom{6}{8} E \Leftrightarrow 08$
14	12	$2.9444506 \binom{9}{6} E \Leftrightarrow 08$	$\Leftrightarrow 2.94445 \binom{0}{1} E \Leftrightarrow 08$
14	13	$6.2100 \binom{20}{16} E \Leftrightarrow 08$	$\Leftrightarrow 6.2100 \binom{16}{20} E \Leftrightarrow 08$
14	14	$4.5527389 \binom{4}{2} E \Leftrightarrow 07$	$\Leftrightarrow 8.76886953428 \binom{6}{8} E \Leftrightarrow 19$
15	1	$4.41114009942 \binom{3}{2} E \Leftrightarrow 02$	$\Leftrightarrow 4.41114009942 \binom{2}{3} E \Leftrightarrow 02$
15	2	$2.1671480669 \binom{1}{0} E \Leftrightarrow 03$	$\Leftrightarrow 2.1671480669 \binom{0}{1} E \Leftrightarrow 03$
15	3	$2.108056786 \binom{8}{7} E \Leftrightarrow 04$	$\Leftrightarrow 2.108056786 \binom{7}{8} E \Leftrightarrow 04$
15	4	$2.72048522 \binom{30}{28} E \Leftrightarrow 05$	$\Leftrightarrow 2.7204852 \binom{28}{30} E \Leftrightarrow 05$
15	5	$4.5746053 \binom{60}{56} E \Leftrightarrow 06$	$\Leftrightarrow 4.5746053 \binom{56}{60} E \Leftrightarrow 06$
15	6	$9.3765762 \binom{5}{1} E \Leftrightarrow 07$	$\Leftrightarrow 9.3765762 \binom{2}{5} E \Leftrightarrow 07$
15	7	$2.2892914 \binom{9}{6} E \Leftrightarrow 07$	$\Leftrightarrow 2.2892914 \binom{6}{9} E \Leftrightarrow 07$
15	8	$6.51910 \binom{92}{89} E \Leftrightarrow 08$	$\Leftrightarrow 6.51910 \binom{89}{92} E \Leftrightarrow 08$
15	9	$2.153 \binom{901}{899} E \Leftrightarrow 08$	$\Leftrightarrow 2.153 \binom{899}{901} E \Leftrightarrow 08$
15	10	$8.5389 \binom{9}{6} E \Leftrightarrow 09$	$\Leftrightarrow 8.5389 \binom{6}{9} E \Leftrightarrow 09$
15	11	$4.968 \binom{4}{2} E \Leftrightarrow 09$	$\Leftrightarrow 4.968 \binom{2}{4} E \Leftrightarrow 09$
15	12	$5.01 \binom{400}{398} E \Leftrightarrow 09$	$\Leftrightarrow 5.01398 \binom{7}{9} E \Leftrightarrow 09$
15	13	$6.2 \binom{400}{398} E \Leftrightarrow 09$	$\Leftrightarrow 6.2 \binom{398}{400} E \Leftrightarrow 09$
15	14	$1.100594 \binom{9}{7} E \Leftrightarrow 08$	$\Leftrightarrow 1.10059 \binom{4}{5} E \Leftrightarrow 08$
15	15	$2.4989 \binom{9}{8} E \Leftrightarrow 08$	$\Leftrightarrow 2.4989 \binom{8}{9} E \Leftrightarrow 08$
15	16	$1.874657 \binom{3}{1} E \Leftrightarrow 07$	$\Leftrightarrow 4.79189873164 \binom{6}{7} E \Leftrightarrow 21$

Restgliedfaktoren der Clenshaw-Curtis-Formeln (Forts.)

M	r	$r! [P_r]$	$r! [N_r]$
16	1	$4.11652382586 \binom{3}{1} E \Leftrightarrow 02$	$\Leftrightarrow 4.11652382586 \binom{1}{3} E \Leftrightarrow 02$
16	2	$1.8862879340 \binom{3}{1} E \Leftrightarrow 03$	$\Leftrightarrow 1.8862879340 \binom{1}{2} E \Leftrightarrow 03$
16	3	$1.710917815 \binom{4}{3} E \Leftrightarrow 04$	$\Leftrightarrow 1.710917815 \binom{3}{4} E \Leftrightarrow 04$
16	4	$2.05691967 \binom{6}{4} E \Leftrightarrow 05$	$\Leftrightarrow 2.05691967 \binom{4}{6} E \Leftrightarrow 05$
16	5	$3.21986441 \binom{5}{1} E \Leftrightarrow 06$	$\Leftrightarrow 3.21986441 \binom{1}{5} E \Leftrightarrow 06$
16	6	$6.1358083 \binom{8}{3} E \Leftrightarrow 07$	$\Leftrightarrow 6.1358083 \binom{4}{7} E \Leftrightarrow 07$
16	7	$1.390393 \binom{6}{5} E \Leftrightarrow 07$	$\Leftrightarrow 1.390393 \binom{6}{5} E \Leftrightarrow 07$
16	8	$3.662990 \binom{5}{0} E \Leftrightarrow 08$	$\Leftrightarrow 3.662990 \binom{0}{5} E \Leftrightarrow 08$
16	9	$1.11071 \binom{2}{0} E \Leftrightarrow 08$	$\Leftrightarrow 1.11071 \binom{0}{2} E \Leftrightarrow 08$
16	10	$3.936 \binom{21}{18} E \Leftrightarrow 09$	$\Leftrightarrow 3.936 \binom{18}{22} E \Leftrightarrow 09$
16	11	$1.892 \binom{5}{3} E \Leftrightarrow 09$	$\Leftrightarrow 1.892 \binom{5}{3} E \Leftrightarrow 09$
16	12	$1.585 \binom{5}{3} E \Leftrightarrow 09$	$\Leftrightarrow 1.585 \binom{3}{5} E \Leftrightarrow 09$
16	13	$2.42 \binom{11}{08} E \Leftrightarrow 09$	$\Leftrightarrow 2.42 \binom{08}{11} E \Leftrightarrow 09$
16	14	$4.0198 \binom{80}{78} E \Leftrightarrow 09$	$\Leftrightarrow 4.0198 \binom{7}{9} E \Leftrightarrow 09$
16	15	$9.600 \binom{62}{51} E \Leftrightarrow 09$	$\Leftrightarrow 9.600 \binom{51}{62} E \Leftrightarrow 09$
16	16	$7.364724 \binom{8}{6} E \Leftrightarrow 08$	$\Leftrightarrow 4.10102797828 \binom{2}{4} E \Leftrightarrow 22$
17	1	$3.85880162882 \binom{23}{18} E \Leftrightarrow 02$	$\Leftrightarrow 3.85880162882 \binom{18}{23} E \Leftrightarrow 02$
17	2	$1.6567686868 \binom{6}{5} E \Leftrightarrow 03$	$\Leftrightarrow 1.6567686868 \binom{5}{6} E \Leftrightarrow 03$
17	3	$1.407745855 \binom{1}{0} E \Leftrightarrow 04$	$\Leftrightarrow 1.407745855 \binom{0}{1} E \Leftrightarrow 04$
17	4	$1.5842603 \binom{10}{09} E \Leftrightarrow 05$	$\Leftrightarrow 1.5842603 \binom{09}{10} E \Leftrightarrow 05$
17	5	$2.32020706 \binom{3}{0} E \Leftrightarrow 06$	$\Leftrightarrow 2.32020706 \binom{0}{3} E \Leftrightarrow 06$
17	6	$4.1326920 \binom{8}{5} E \Leftrightarrow 07$	$\Leftrightarrow 4.1326920 \binom{5}{8} E \Leftrightarrow 07$
17	7	$8.74528 \binom{62}{59} E \Leftrightarrow 08$	$\Leftrightarrow 8.74528 \binom{59}{62} E \Leftrightarrow 08$
17	8	$2.14911 \binom{4}{2} E \Leftrightarrow 08$	$\Leftrightarrow 2.14911 \binom{2}{4} E \Leftrightarrow 08$
17	9	$6.0646 \binom{4}{1} E \Leftrightarrow 09$	$\Leftrightarrow 6.0646 \binom{1}{4} E \Leftrightarrow 09$
17	10	$1.97 \binom{70}{69} E \Leftrightarrow 09$	$\Leftrightarrow 1.97 \binom{69}{70} E \Leftrightarrow 09$
17	11	$8.06 \binom{9}{7} E \Leftrightarrow 10$	$\Leftrightarrow 8.06 \binom{7}{9} E \Leftrightarrow 10$
17	12	$5.4 \binom{20}{18} E \Leftrightarrow 10$	$\Leftrightarrow 5.4 \binom{18}{20} E \Leftrightarrow 10$
17	13	$5.1 \binom{84}{79} E \Leftrightarrow 10$	$\Leftrightarrow 5.1 \binom{79}{84} E \Leftrightarrow 10$
17	14	$6.038 \binom{8}{8} E \Leftrightarrow 10$	$\Leftrightarrow 6.038 \binom{8}{8} E \Leftrightarrow 10$
17	15	$8.96 \binom{6}{1} E \Leftrightarrow 10$	$\Leftrightarrow 8.96 \binom{1}{6} E \Leftrightarrow 10$
17	16	$1.6032 \binom{5}{4} E \Leftrightarrow 09$	$\Leftrightarrow 1.603 \binom{2}{3} E \Leftrightarrow 09$
17	17	$3.935 \binom{3}{0} E \Leftrightarrow 09$	$\Leftrightarrow 3.935 \binom{0}{3} E \Leftrightarrow 09$
17	18	$3.100936 \binom{8}{6} E \Leftrightarrow 08$	$\Leftrightarrow 1.75507719079 \binom{5}{6} E \Leftrightarrow 24$

Restgliedfaktoren der Clenshaw-Curtis-Formeln (Forts.)

M	r	$r![P_r]$	$r![N_r]$
18	1	$3.63147449119 \binom{2}{1} E \Leftrightarrow 02$	$\Leftrightarrow 3.63147449119 \binom{1}{2} E \Leftrightarrow 02$
18	2	$1.466776752 \binom{41}{39} E \Leftrightarrow 03$	$\Leftrightarrow 1.466776752 \binom{39}{41} E \Leftrightarrow 03$
18	3	$1.17225971 \binom{21}{19} E \Leftrightarrow 04$	$\Leftrightarrow 1.17225971 \binom{19}{21} E \Leftrightarrow 04$
18	4	$1.24007833 \binom{1}{0} E \Leftrightarrow 05$	$\Leftrightarrow 1.24007833 \binom{0}{1} E \Leftrightarrow 05$
18	5	$1.706356 \binom{60}{59} E \Leftrightarrow 06$	$\Leftrightarrow 1.706356 \binom{59}{60} E \Leftrightarrow 06$
18	6	$2.853157 \binom{7}{5} E \Leftrightarrow 07$	$\Leftrightarrow 2.853157 \binom{58}{62} E \Leftrightarrow 07$
18	7	$5.66197 \binom{4}{3} E \Leftrightarrow 08$	$\Leftrightarrow 5.66197 \binom{3}{4} E \Leftrightarrow 08$
18	8	$1.30242 \binom{9}{7} E \Leftrightarrow 08$	$\Leftrightarrow 1.30242 \binom{7}{9} E \Leftrightarrow 08$
18	9	$3.425 \binom{6}{4} E \Leftrightarrow 09$	$\Leftrightarrow 3.425 \binom{4}{6} E \Leftrightarrow 09$
18	10	$1.026 \binom{4}{2} E \Leftrightarrow 09$	$\Leftrightarrow 1.02 \binom{6}{7} E \Leftrightarrow 09$
18	11	$3.65 \binom{6}{5} E \Leftrightarrow 10$	$\Leftrightarrow 3.6 \binom{5}{6} E \Leftrightarrow 10$
18	12	$1.9 \binom{80}{78} E \Leftrightarrow 10$	$\Leftrightarrow 1.9 \binom{78}{81} E \Leftrightarrow 10$
18	13	$1. \binom{9}{7} E \Leftrightarrow 10$	$\Leftrightarrow 1. \binom{7}{9} E \Leftrightarrow 10$
18	14	$2.3 \binom{3}{1} E \Leftrightarrow 10$	$\Leftrightarrow 2.3 \binom{19}{21} E \Leftrightarrow 10$
18	15	$3.3 \binom{4}{2} E \Leftrightarrow 10$	$\Leftrightarrow 3.3 \binom{2}{4} E \Leftrightarrow 10$
18	16	$6.2377 \binom{8}{1} E \Leftrightarrow 10$	$\Leftrightarrow 6.2377 \binom{1}{9} E \Leftrightarrow 10$
18	17	$1.5 \binom{60}{58} E \Leftrightarrow 09$	$\Leftrightarrow 1.56 \binom{58}{60} E \Leftrightarrow 09$
18	18	$1.25975 \binom{6}{5} E \Leftrightarrow 08$	$\Leftrightarrow 1.588861019 \binom{999}{101} E \Leftrightarrow 25$
19	1	$3.42944585198 \binom{1}{0} E \Leftrightarrow 02$	$\Leftrightarrow 3.42944585198 \binom{0}{1} E \Leftrightarrow 02$
19	2	$1.3077248002 \binom{9}{8} E \Leftrightarrow 03$	$\Leftrightarrow 1.3077248002 \binom{8}{9} E \Leftrightarrow 03$
19	3	$9.8656497 \binom{101}{95} E \Leftrightarrow 05$	$\Leftrightarrow 9.8656497 \binom{95}{101} E \Leftrightarrow 05$
19	4	$9.8462759 \binom{83}{79} E \Leftrightarrow 06$	$\Leftrightarrow 9.8462759 \binom{78}{84} E \Leftrightarrow 06$
19	5	$1.2777754 \binom{6}{5} E \Leftrightarrow 06$	$\Leftrightarrow 1.2777754 \binom{5}{6} E \Leftrightarrow 06$
19	6	$2.013688 \binom{2}{1} E \Leftrightarrow 07$	$\Leftrightarrow 2.013688 \binom{1}{2} E \Leftrightarrow 07$
19	7	$3.76401 \binom{6}{5} E \Leftrightarrow 08$	$\Leftrightarrow 3.76401 \binom{5}{6} E \Leftrightarrow 08$
19	8	$8.150 \binom{210}{196} E \Leftrightarrow 09$	$\Leftrightarrow 8.150 \binom{19}{22} E \Leftrightarrow 09$
19	9	$2.016 \binom{8}{6} E \Leftrightarrow 09$	$\Leftrightarrow 2.016 \binom{6}{8} E \Leftrightarrow 09$
19	10	$5.6 \binom{80}{77} E \Leftrightarrow 10$	$\Leftrightarrow 5.6 \binom{77}{80} E \Leftrightarrow 10$

Restgliedfaktoren der Clenshaw-Curtis-Formeln (Forts.)

M	r	$r! [P_r]$	$r! [N_r]$
19	11	$1.8 \binom{8}{7} E \Leftrightarrow 10$	$\Leftrightarrow 1.8 \binom{7}{8} E \Leftrightarrow 10$
19	12	$8.5 \binom{9}{3} E \Leftrightarrow 11$	$\Leftrightarrow 8.5 \binom{4}{8} E \Leftrightarrow 11$
19	13	$\binom{9.6}{6.1} E \Leftrightarrow 11$	$\Leftrightarrow \binom{6.1}{9.6} E \Leftrightarrow 11$
19	14	$5.9 \binom{8}{2} E \Leftrightarrow 11$	$\Leftrightarrow \binom{5.8}{6.1} E \Leftrightarrow 11$
19	15	$\binom{7.1}{6.5} E \Leftrightarrow 11$	$\Leftrightarrow \binom{6.5}{7.1} E \Leftrightarrow 11$
19	16	$8. \binom{6.1}{5.2} E \Leftrightarrow 11$	$\Leftrightarrow 8.5 \binom{5}{8} E \Leftrightarrow 11$
19	17	$1. \binom{4}{3} E \Leftrightarrow 10$	$\Leftrightarrow 1. \binom{3}{4} E \Leftrightarrow 10$
19	18	$2.531 \binom{4}{2} E \Leftrightarrow 10$	$\Leftrightarrow 2.53 \binom{0}{2} E \Leftrightarrow 10$
19	19	$6.5 \binom{5}{2} E \Leftrightarrow 10$	$\Leftrightarrow 6.5 \binom{2}{5} E \Leftrightarrow 10$
19	20	$5.3989 \binom{8}{3} E \Leftrightarrow 09$	$\Leftrightarrow 5.46537151922 \binom{6}{9} E \Leftrightarrow 28$
20	1	$3.24872614871 \binom{7}{6} E \Leftrightarrow 02$	$\Leftrightarrow 3.24872614871 \binom{6}{7} E \Leftrightarrow 02$
20	2	$1.173232529 \binom{41}{39} E \Leftrightarrow 03$	$\Leftrightarrow 1.173232529 \binom{39}{40} E \Leftrightarrow 03$
20	3	$8.38147083 \binom{13}{06} E \Leftrightarrow 05$	$\Leftrightarrow 8.38147083 \binom{06}{13} E \Leftrightarrow 05$
20	4	$7.9176837 \binom{40}{35} E \Leftrightarrow 06$	$\Leftrightarrow 7.9176837 \binom{34}{41} E \Leftrightarrow 06$
20	5	$9.7223866 \binom{6}{1} E \Leftrightarrow 07$	$\Leftrightarrow 9.7223866 \binom{1}{6} E \Leftrightarrow 07$
20	6	$1.448928 \binom{5}{4} E \Leftrightarrow 07$	$\Leftrightarrow 1.448928 \binom{4}{5} E \Leftrightarrow 07$
20	7	$2.55945 \binom{9}{7} E \Leftrightarrow 08$	$\Leftrightarrow 2.55945 \binom{7}{9} E \Leftrightarrow 08$
20	8	$5.2313 \binom{4}{1} E \Leftrightarrow 09$	$\Leftrightarrow 5.2313 \binom{1}{4} E \Leftrightarrow 09$
20	9	$1.21 \binom{90}{88} E \Leftrightarrow 09$	$\Leftrightarrow 1.21 \binom{88}{90} E \Leftrightarrow 09$
20	10	$3.20 \binom{9}{6} E \Leftrightarrow 10$	$\Leftrightarrow 3.20 \binom{6}{9} E \Leftrightarrow 10$
20	11	$9. \binom{8}{6} E \Leftrightarrow 11$	$\Leftrightarrow 9. \binom{6}{8} E \Leftrightarrow 11$
20	12	$3. \binom{8}{6} E \Leftrightarrow 11$	$\Leftrightarrow 3. \binom{5}{8} E \Leftrightarrow 11$
20	13	$2. \binom{8}{2} E \Leftrightarrow 11$	$\Leftrightarrow 2. \binom{2}{8} E \Leftrightarrow 11$
20	14	$2. \binom{4}{0} E \Leftrightarrow 11$	$\Leftrightarrow 2. \binom{0}{4} E \Leftrightarrow 11$
20	15	$\binom{3.5}{2.4} E \Leftrightarrow 11$	$\Leftrightarrow \binom{2.4}{3.5} E \Leftrightarrow 11$
20	16	$3.2 \binom{5}{0} E \Leftrightarrow 11$	$\Leftrightarrow 3.2 \binom{0}{5} E \Leftrightarrow 11$
20	17	$\binom{6.0}{5.1} E \Leftrightarrow 11$	$\Leftrightarrow \binom{5.1}{6.0} E \Leftrightarrow 11$
20	18	$1.0 \binom{10}{08} E \Leftrightarrow 10$	$\Leftrightarrow 1.0 \binom{0}{2} E \Leftrightarrow 10$
20	19	$2. \binom{70}{66} E \Leftrightarrow 10$	$\Leftrightarrow 2. \binom{66}{70} E \Leftrightarrow 10$
20	20	$2.2495 \binom{8}{5} E \Leftrightarrow 09$	$\Leftrightarrow 5.16954808982 \binom{6}{8} E \Leftrightarrow 29$

Tabelle 3.5: Vergleich der Restgliedfaktoren $r! \int_{-1}^1 |K_r|$ der Clenshaw-Curtis-Formeln

		$(d+1)! \cdot \ K_{d+1}\ _1$	
M	d	BRASS/FÖRSTER	Verifikation
4	3	$7.017543859649122E \Leftrightarrow 02$	$7.013497264983 \binom{37}{28} E \Leftrightarrow 02$
5	5	$1.923809523809524E \Leftrightarrow 02$	$1.92073594250 \binom{51}{49} E \Leftrightarrow 02$
6	5	$2.426739926739926E \Leftrightarrow 03$	$2.4263632575 \binom{43}{39} E \Leftrightarrow 03$
7	7	$8.015873015873014E \Leftrightarrow 04$	$7.937534362 \binom{803}{795} E \Leftrightarrow 04$
8	7	$2.003968253968254E \Leftrightarrow 04$	$1.984175351 \binom{91}{89} E \Leftrightarrow 04$
9	9	$7.215014430014429E \Leftrightarrow 05$	$7.215014248 \binom{60}{54} E \Leftrightarrow 05$
10	9	$2.254692009379510E \Leftrightarrow 05$	$2.254690185 \binom{4}{1} E \Leftrightarrow 05$
11	11	$8.671892343767343E \Leftrightarrow 06$	$8.67188371 \binom{50}{42} E \Leftrightarrow 06$
12	11	$3.035162320318570E \Leftrightarrow 06$	$3.03515928 \binom{9}{7} E \Leftrightarrow 06$
13	13	$1.214064928127428E \Leftrightarrow 06$	$1.21406371 \binom{5}{3} E \Leftrightarrow 06$
14	13	$4.552743480477855E \Leftrightarrow 07$	$4.5527389 \binom{4}{2} E \Leftrightarrow 07$
15	15	$1.874659080196764E \Leftrightarrow 07$	$1.874657 \binom{3}{1} E \Leftrightarrow 07$
16	15	$7.364732100773001E \Leftrightarrow 08$	$7.364724 \binom{8}{6} E \Leftrightarrow 08$
17	17	$3.100939831904421E \Leftrightarrow 08$	$3.100936 \binom{8}{6} E \Leftrightarrow 08$
18	17	$1.259756806711171E \Leftrightarrow 08$	$1.25975 \binom{6}{5} E \Leftrightarrow 08$
19	19	$5.398957743047877E \Leftrightarrow 09$	$5.3989 \binom{8}{3} E \Leftrightarrow 09$
20	19	$2.249565726269948E \Leftrightarrow 09$	$2.2495 \binom{8}{5} E \Leftrightarrow 09$

Bemerkung 3.4.5 *Im allgemeinen liefern die von Braß/Förster [28] für die Fehlerkonstanten der Clenshaw-Curtis-Formeln angegebenen Abschätzungsformeln relativ gute Oberschranken.*

Kapitel 4

Mehrdimensionale numerische Integration

Während es im Eindimensionalen bis auf affine Abbildungen nur einen kompakten und zusammenhängenden Bereich gibt, existieren im Mehrdimensionalen eine große Zahl von kompakten und zusammenhängenden Gebieten, die sich unter affinen Abbildungen nicht ineinander überführen lassen. Z. B. gibt es im Zweidimensionalen das Quadrat, den Kreis und das Dreieck usw., für welche die Kubaturformeln einzeln konstruiert werden müssen. Wegen dieser Vielfältigkeit ist die numerische Integration im Mehrdimensionalen viel komplizierter als im Eindimensionalen. Um die Diskussion zu vereinfachen, wird im folgenden nur die zweidimensionale numerische Integration betrachtet.

4.1 Integration durch Transformation

Für die Integrationsbereiche, die affin ähnlich sind, können die Integrale über solchen Bereichen mit Hilfe von affinen Transformationen aus Integralen über einfacheren Bereichen erhalten werden. Dies gilt z. B. für Rechteck und Quadrat. Es sei $\Omega^* = [a, b] \times [c, d]$ und $\tilde{\Omega}^* = [\Leftarrow 1, 1] \times [\Leftarrow 1, 1]$. Durch die Transformation $T(\tilde{x}, \tilde{y}) = (x, y)$ mit

$$\begin{cases} x = \frac{(b \Leftarrow a)\tilde{x} + a + b}{2}, \\ y = \frac{(d \Leftarrow c)\tilde{y} + c + d}{2}, \end{cases} \quad (4.1)$$

wird $\tilde{\Omega}^*$ auf Ω^* abgebildet. Umgekehrt wird das Rechteck Ω^* durch T^{-1} auf das Quadrat $\tilde{\Omega}^*$ zurückgeführt. Das Integral einer Funktion f bzgl. der Gewichtsfunktion w über Ω^* kann dann mit Hilfe der Transformation T aus dem Integral über $\tilde{\Omega}^*$ erhalten werden :

$$\int \int_{\Omega^*} w(x, y) f(x, y) dx dy = \int \int_{\tilde{\Omega}^*} w(T(\tilde{x}, \tilde{y})) f(T(\tilde{x}, \tilde{y})) |\det \mathbf{J}| d\tilde{x} d\tilde{y}, \quad (4.2)$$

wobei \mathbf{J} die Abbildungsmatrix von T darstellt.

Der Kreis ist zwar nicht affin äquivalent zum Rechteck, jedoch läßt er sich durch nichtaffine Abbildungen in ein Rechteck überführen. Sei nun $\Omega^* := \{ (x, y) \mid x^2 + y^2 \leq c^2 \}$ ein Kreis. Durch die Transformation

$$T(r, \theta) = (r \cos \theta, r \sin \theta) \quad (4.3)$$

bzw.

$$\tilde{T}(\tilde{x}, \tilde{y}) = (\tilde{x}, \Leftrightarrow\sqrt{c^2 - \tilde{x}^2} + 2\tilde{y}\sqrt{c^2 - \tilde{x}^2}) \quad (4.4)$$

wird das Rechteck $[0, c] \times [0, 2\pi]$ bzw. $[\Leftrightarrow c, c] \times [0, 1]$ auf Ω^* abgebildet. Das bestimmte Integral $\int \int_{\Omega^*} w(x, y) f(x, y) dx dy$ über dem Kreis Ω^* kann ebenfalls gemäß (4.2) mit Hilfe der Transformation aus Integralen über Rechteckbereichen erhalten werden. Analog lassen sich die Dreieckbereiche ebenfalls durch nichtaffine Transformationen in Rechtecke überführen (vgl. [146]).

4.2 Kubaturformeln

Um für zweidimensionale Funktionen f das bestimmte Integral

$$I(f) := \int \int_{\Omega^*} w(x, y) f(x, y) dx dy, \quad (4.5)$$

numerisch zu bestimmen, ist es häufig notwendig, Kubaturformeln der Form

$$S(f) := \sum_{i=1}^M w_i f(x_i, y_i) \quad (4.6)$$

mit

$$I(f) = S(f) + E(f)$$

einzusetzen. Hierbei wird vorausgesetzt, daß die Gewichtsfunktion $w(x, y)$ positiv ist und der Ungleichung $I(1) < \infty$ genügt, alle Stützstellen (x_i, y_i) der Integrationsformel (4.6) in Ω^* liegen und sämtliche Gewichte w_i reell sind. Das Funktional E wird in diesem Fall als Kubaturfehler bezeichnet.

Die Kubaturformel S hat genau dann den Exaktheitsgrad d , in Zeichen $\deg S := d$, wenn gilt

1. $E(p) = 0, \quad \forall p \in \mathbb{P}_d,$
2. $E(q) \neq 0, \quad \exists q \in \dot{\mathbb{P}}_{d+1}$

mit $\mathbb{P}_d := \text{Spann}(\{x^i y^j \mid 0 \leq i + j \leq d\})$ und $\dot{\mathbb{P}}_d := \text{Spann}(\{x^i y^j \mid i + j = d\})$.

Ist $\deg S = d$, dann gilt für die Basis $\{f_1, f_2, \dots, f_{\hat{d}}\}$ von \mathbb{P}_d mit $\hat{d} = \dim \mathbb{P}_d = \frac{(d+1)(d+2)}{2}$

$$S(f_i) = I(f_i), \quad i = 1(1)\hat{d}. \quad (4.7)$$

Um Kubaturformeln der Form (4.6) zu konstruieren, wurden verschiedene Methoden untersucht, die sich im allgemeinen in zwei Klassen unterscheiden. Die eine benutzt den

Zusammenhang zwischen den mehrdimensionalen Orthogonalpolynomen und den Kubaturformeln, die andere versucht, das nichtlineare Gleichungssystem (4.7) direkt zu lösen.

In dieser Arbeit werden die in [45] erzielten vollsymmetrischen Kubaturformeln, die durch die Lösung des nichtlinearen Gleichungssystems (4.7) unter Berücksichtigung der Konsistenzbedingung konstruiert wurden, für die zweidimensionale numerische Integration verwendet. Insbesondere wurden die Fehlerkonstanten dieser vollsymmetrischen Kubaturformeln im Rahmen dieser Arbeit verifiziert berechnet (s. Abschnitt 4.3).

4.2.1 Vollsymmetrische Kubaturformeln

Eine Kubaturformel heißt unter der Abbildung φ invariant, falls der Integrationsbereich Ω^* , die Gewichtsfunktion $w(x, y)$ und die Menge der Stützstellen $\{(x_i, y_i)\}$ bzgl. φ invariant sind und die Gewichte zu den Knoten, welche bzgl. φ auf der gleichen Bahn liegen, identisch sind. Es gilt nämlich

$$\left\{ \begin{array}{l} 1. \quad \varphi(\Omega^*) = \Omega^*, \\ 2. \quad w(x, y) = w \circ \varphi(x, y), \\ 3. \quad S(f) = S(f \circ \varphi), \quad \forall f. \end{array} \right. \quad (4.8)$$

Eine Kubaturformel über dem Quadrat bzw. Kreis heißt *vollsymmetrisch*, falls sie \mathcal{D}_4 -invariant ist, wobei \mathcal{D}_4 die Diedergruppe des regelmäßigen 4-Ecks ist (vgl. [45], S. 11). Für das Dreieck ist eine Kubaturformel vollsymmetrisch, wenn (4.8) für alle Permutationen der drei Ecken gilt (vgl. [32], [45]).

Speziell für den Bereich $R_2^* := [\Leftarrow a, a] \times [\Leftarrow c, c]$ bzw. $C_2^* := \{(x, y) \mid x^2 + y^2 \leq r^2\}$ wird eine Kubaturformel als *symmetrisch* bezeichnet, falls die Bedingungen in (4.8) für alle Abbildungen φ aus der Menge $\mathcal{A}_s := \{(x, y) \mapsto (\pm x, \pm y)\}$ gelten. Sind darüber hinaus die Kubaturformeln, die bzgl. R_2^* bzw. C_2^* konstruiert sind, auch invariant unter den Abbildungen $\varphi \in \mathcal{A}_{vs} := \{(x, y) \mapsto (\pm x, \pm y), (x, y) \mapsto (\pm y, \pm x)\}$, so heißen sie *vollsymmetrisch*.

Für den Fall $\Omega^* = R_2^*$ bzw. C_2^* wird häufig die Schreibweise

$$S(f) = \sum_{i=1}^M w_i f(x_i, y_i) = \sum_{i=1}^{M'} w_i \{ f(\pm x_i, \pm y_i) \}$$

für symmetrische Kubaturformeln und die Schreibweise

$$S(f) = \sum_{i=1}^M w_i f(x_i, y_i) = \sum_{i=1}^{M'} w_i \{ f(\pm x_i, \pm y_i) + f(\pm y_i, \pm x_i) \}$$

für vollsymmetrische Kubaturformeln verwendet, wobei die Abkürzung $f(\pm x_i, \pm y_i)$ durch

$$f(\pm x_i, \pm y_i) := f(x_i, y_i) + f(\Leftarrow x_i, y_i) + f(x_i, \Leftarrow y_i) + f(\Leftarrow x_i, \Leftarrow y_i)$$

definiert ist.

4.2.2 Produktformeln

Für die mehrdimensionale numerische Integration bilden die Produktregeln eine weitere wichtige Klasse von Approximationsformeln.

Sei $\Omega^* := [a, b] \times [c, d]$. Das Integral (4.5) kann z. B. durch Anwendung eindimensionaler Quadraturformeln näherungsweise berechnet werden, falls die Gewichtsfunktion in der Form $w(x, y) = u(x) \cdot v(y)$ zerlegbar ist. Ausgehend von den Quadraturformeln, die auf die Gewichtsfunktionen $u(x)$ und $v(y)$ über $[a, b]$ bzw. $[c, d]$ zugeschnitten sind :

$$\int_a^b u(x) g(x) dx = \sum_{i=1}^{M_1} a_i g(x_i) + E_x(g), \quad (4.9)$$

$$\int_c^d v(y) h(y) dy = \sum_{j=1}^{M_2} b_j h(y_j) + E_y(h), \quad (4.10)$$

erhält man die folgende $M_1 M_2$ -punktige Produktformel über Ω^* bzgl. der Gewichtsfunktion $w(x, y)$

$$\iint_{\Omega^*} w(x, y) f(x, y) dx dy = \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} a_i b_j f(x_i, y_j) + E(f). \quad (4.11)$$

Die Formeln (4.9), (4.10) und (4.11) können durch Anwendung der Operatorschreibweise übersichtlicher dargestellt werden (vgl. [104]):

$$\begin{aligned} I_x(g) &= S_x(g) + E_x(g) \\ I_y(h) &= S_y(h) + E_y(h) \\ I_y I_x(f) &= S_y S_x(f) + E(f). \end{aligned} \quad (4.12)$$

Nach der Definitionen von I , S und E sind diese Operatoren kommutativ. Es gilt nämlich für $O_x \in \{I_x, S_x, E_x\}$ und $O_y \in \{I_y, S_y, E_y\}$

$$O_x O_y(f) = O_y O_x(f),$$

wobei $O_x O_y(f)$ durch $O_x(O_y(f(x, y)))$ erklärt wird.

Der Exaktheitsgrad der so konstruierten Produktformel $S_y S_x$ ist gleich dem kleinsten Exaktheitsgrad der zugrundeliegenden Quadraturformeln, d. h. $\deg S_y S_x = \min(d_1, d_2)$, mit $\deg S_x =: d_1$ und $\deg S_y =: d_2$ (vgl. [146], S. 27).

Die Produktformeln lassen sich nicht nur für Rechteckbereiche sondern auch für andere Bereiche, wie z. B. Kreis, Dreieck und Parallelogramm, konstruieren (vgl. [146], Ch. 1, 2). Für zweidimensionale Bereiche, die durch affine bzw. nicht affine Transformationen in Rechteckbereiche überführt werden können, lassen sich die Produktformeln anwenden.

Die Approximationsfehler E der Produktformeln werden im Abschnitt 4.3 eingehend diskutiert.

4.3 Kubaturfehler

Für die Fehlerbetrachtung bei den Kubaturformeln werden in diesem Abschnitt zuerst die von Nikol'skii und Stancu für die Produktformeln aufgestellten Methoden diskutiert. Danach wird die Kernmethode, die ursprünglich von Peano für den eindimensionalen Quadraturfehler hergeleitet und später von Sard auf mehrere Dimensionen verallgemeinert wurde, betrachtet. Die Kernmethode gilt sowohl für Produkt- als auch für Nicht-Produktformeln. Obwohl die Sard-Kerndarstellung für die Fehlerabschätzung der Produktformeln weniger günstig ist, wird sie dennoch zur Vervollständigung der Diskussion betrachtet.

Um die Eigenschaften der Sard-Kerne, die für die Berechnung der Fehlerkonstante notwendig sind, herzuleiten, wird im Abschnitt 4.3.3 eine Diskussion durchgeführt, die allgemeiner und übersichtlicher ist als die Herleitung von Lether [102].

Trotz fehlender Kenntnisse über die Definitheit der Sard-Kerne können die Fehlerkonstanten bzgl. der Sard-Kerne mit Hilfe der Überdeckungsmethode verifiziert berechnet werden. In dieser Arbeit wurden für die eindimensionalen Sard-Kerne die Fehlerkonstanten, wie bei den Peano-Kernen, mit Hilfe des erweiterten Intervall-Newton-Verfahrens unter Berücksichtigung der lokalen Definitheit und der Symmetrieeigenschaft der Kernfunktionen berechnet. Darüber hinaus wurden die Fehlerkonstanten bzgl. der zweidimensionalen Sard-Kerne mit Hilfe des von Hansen/Neumaier verallgemeinerten Intervall-Newton-Verfahrens bestimmt, wobei die lokale Definitheit, die Symmetrie- und die Vollsymmetrieeigenschaft der Sard-Kerne ebenfalls beachtet wurden. Dies wird im Abschnitt 4.3.4 besprochen.

4.3.1 Der Approximationsfehler von Produktformeln

Bei Anwendung der Operatorschreibweise (4.12) gilt (vgl. [104])

$$\begin{aligned} I_y I_x(f) &= I_y(S_x(f) + E_x(f)) = I_y S_x(f) + I_y E_x(f) \\ &= S_y S_x(f) + E_y S_x(f) + I_y E_x(f) , \end{aligned}$$

bzw.

$$I_y I_x(f) = S_x S_y(f) + E_x S_y(f) + I_x E_y(f) .$$

Daraus folgen die auf Nikol'skii zurückgehenden Fehlerdarstellungen (vgl. Stroud & Secrest [147], Luik [104]).

Satz 4.3.1 *Für die Approximationsfehler der Produktformeln gilt*

$$E(f) = E_y S_x(f) + I_y E_x(f) , \tag{4.13}$$

bzw.

$$E(f) = E_x S_y(f) + I_x E_y(f) . \tag{4.14}$$

Ersetzt man in (4.13) S_x durch $I_x \Leftrightarrow E_x$ und in (4.14) S_y durch $I_y \Leftrightarrow E_y$, so ergeben sich die Fehlerdarstellungen von Stancu (vgl. [104], S. 64).

Satz 4.3.2 *Für den Approximationsfehler der Produktformeln gilt*

$$E(f) = E_y I_x(f) + I_y E_x(f) \Leftrightarrow E_y E_x(f) , \quad (4.15)$$

bzw. die Variante

$$E(f) = E_y S_x(f) + S_y E_x(f) + E_y E_x(f) . \quad (4.16)$$

Gemäß der Darstellungen (4.13), (4.14), (4.15) und (4.16) lassen sich die Fehler der Produktformeln auf die Quadraturfehler der zugrundeliegenden Quadraturformeln zurückführen.

Die Nikol'skiische Methode benötigt nur zwei Terme zur Berechnung. Sie ist damit günstiger als die Darstellung von Stancu. In [104], S. 65, hat Luik erwähnt, daß für Quadraturformeln mit negativen Gewichten die Methode von Stancu bessere Fehlerabschätzung liefern kann.

Ω_x^* und Ω_y^* seien die Projektionen von Ω^* auf die x - und y -Achse. Für positive Quadraturformeln führt die Nikol'skiische Fehlerdarstellung zu den Einschließungsformeln (4.17) und (4.18).

Satz 4.3.3 *Sind die zugrundeliegenden Quadraturformeln einer Produktregel positiv, dann läßt sich der Kubaturfehler wie folgt abschätzen :*

$$E(f) \in \|S_x\| \cdot E_y(f(\Omega_x^*, \cdot)) + E_x(\|I_y\| \cdot f(\cdot, \Omega_y^*)) , \quad (4.17)$$

bzw.

$$E(f) \in \|S_y\| \cdot E_x(f(\cdot, \Omega_y^*)) + E_y(\|I_x\| \cdot f(\Omega_x^*, \cdot)) . \quad (4.18)$$

Dabei muß $\|I_x\|$ bzw. $\|I_y\|$ für Nicht-Rechteckgebiete keine Konstante sein.

Beweis : Da die linearen Operatoren I , S und E kommutativ sind, folgt aus (4.13)

$$\begin{aligned} E(f) &= \sum_{i=1}^{M_1} a_i \cdot E_y(f(x_i, \cdot)) + E_x(I_y(f(\cdot, y))) \\ &\in \sum_{i=1}^{M_1} a_i \cdot E_y(f(\Omega_x^*, \cdot)) + E_x(\|I_y\| \cdot f(\cdot, \Omega_y^*)) \\ &= \|S_x\| \cdot E_y(f(\Omega_x^*, \cdot)) + E_x(\|I_y\| \cdot f(\cdot, \Omega_y^*)) . \end{aligned}$$

Die letzte Gleichheit gilt wegen $a_i \geq 0$ für $i = 1(1)M_1$. Die Einschließungsformel (4.18) läßt sich auf gleiche Weise herleiten. \square

Für den Inklusionsschritt im obigen Beweis sei angemerkt, daß die Einschließungsformel nur für die Funktionen gilt, für welche die intervallmäßige Auswertung $f(\Omega_x^*, \Omega_y^*)$

möglich ist. Da für E_y anstatt über $(x_i, [y(x_i)])$, $i = 1(1)M_1$, nur die Taylorkoeffizienten über $(\Omega_x^*, \Omega_y^*) \supseteq \Omega^*$ zu berechnen sind, kann die Einschließungsformel im konkreten Fall günstiger sein. Für achsenparallele Rechtecke Ω^* gilt $(\Omega_x^*, \Omega_y^*) = \Omega^*$ und in diesem Fall sind $\|I_x\|$ und $\|I_y\|$ Konstanten.

Ein anderer Zugang zur Darstellung der Kubaturfehler der Produktformeln ist die von Sard erweiterte Kernmethode, die im nächsten Abschnitt besprochen wird.

4.3.2 Sard-Kerndarstellungen und Fehlereinschließungen

Die Peanosche Fehlerdarstellung hat Sard [133] und [134] auf die Kubaturfehler verallgemeinert. Die vollständige Theorie von Sard ist sehr umfangreich ([134], Chap. 4). Im folgenden wird die Diskussion von Stroud [146] eingeschlagen und die Betrachtung beschränkt sich zunächst auf die folgenden Funktionenräume

$$\begin{aligned} C^{(r)} &:= \{ f : \Omega \rightarrow \mathbb{R} \mid f^{(i,j)} \in C(\Omega), \quad 0 \leq i + j \leq r \}, \\ C^{k,l} &:= \{ f : \Omega \rightarrow \mathbb{R} \mid f^{(i,j)} \in C(\Omega), \quad 0 \leq i \leq k, \quad 0 \leq j \leq l \}, \end{aligned}$$

wobei $\Omega := \Omega_x \times \Omega_y := [a, b] \times [c, d]$ ein Rechteckgebiet ist, welches den Integrationsbereich Ω^* einschließt.

Als Hilfsmittel für die Diskussion werden die folgenden Treppenfunktionen eingeführt (vgl. [134]).

Definition 4.3.1

$$\begin{aligned} 1. \quad \theta(x, t) &:= \begin{cases} 1 & \text{für } x \geq t \\ 0 & \text{für } x < t \end{cases} \\ 2. \quad \psi(\alpha, \beta, \gamma) &:= \begin{cases} 1 & \text{für } \alpha \leq \beta < \gamma \\ \Leftrightarrow 1 & \text{für } \gamma \leq \beta < \alpha \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

Aus den Definitionen 4.3.1 und 3.4.1 ergeben sich unmittelbar die nächsten Zusammenhänge.

Bemerkung 4.3.1

$$1. \quad \theta(t, x) = 1 \Leftrightarrow \theta(x, t) \quad \text{für } x \neq t$$

2. $\theta(\Leftrightarrow x, \Leftrightarrow t) = \theta(t, x)$
3. $(x \Leftrightarrow t) \cdot \theta(x, t) = (x \Leftrightarrow t)_+$
4. $(x \Leftrightarrow t) \cdot \theta(t, x) = (x \Leftrightarrow t)_-$
5. $(x \Leftrightarrow t) \cdot \theta(t, x) = (x \Leftrightarrow t) \cdot (1 \Leftrightarrow \theta(x, t))$
6. $\psi(\alpha, \beta, \gamma) = \theta(\beta, \alpha) \Leftrightarrow \theta(\beta, \gamma)$

Der Ausgangspunkt der Sard-Kerndarstellungen für die Kubaturfehler $E(f) := I(f) \Leftrightarrow S(f)$ ist wie bei den Peano-Kernen ebenfalls die Taylor-Formel der Funktion f bzgl. des Entwicklungspunktes $(a_0, c_0) \in \Omega$ mitsamt der Integralrestglieder.

Kubaturfehler im Raum $B_{p,q}(a_0, c_0)$

Sei $(a_0, c_0) \in \Omega$. Für $p, q \in \mathbb{N}$ und $r = p + q$ lassen sich die Funktionen f aus $C^{(r)}$ wie folgt entwickeln (vgl. Stroud [146] S. 145) :

$$\begin{aligned}
f(x, y) &= \sum_{i+j < r} (x \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y \Leftrightarrow c_0)^{\langle j \rangle} \cdot f^{(i,j)}(a_0, c_0) \\
&+ \sum_{j < q} (y \Leftrightarrow c_0)^{\langle j \rangle} \cdot \int_{a_0}^x (x \Leftrightarrow u)^{\langle r-j-1 \rangle} \cdot f^{(r-j,j)}(u, c_0) du \\
&+ \sum_{i < p} (x \Leftrightarrow a_0)^{\langle i \rangle} \cdot \int_{c_0}^y (y \Leftrightarrow v)^{\langle r-i-1 \rangle} \cdot f^{(i,r-i)}(a_0, v) dv \\
&+ \int_{a_0}^x \int_{c_0}^y (x \Leftrightarrow u)^{\langle p-1 \rangle} \cdot (y \Leftrightarrow v)^{\langle q-1 \rangle} \cdot f^{(p,q)}(u, v) du dv.
\end{aligned} \tag{4.19}$$

Die Taylor-Formel (4.19) gilt auch für die Funktionen f , für welche $f^{(i,j)}(a_0, c_0)$, $i + j < r$, existieren, die partiellen Ableitungen $f^{(r-j,j)}(u, c_0)$ auf $[a, b]$, $f^{(i,r-i)}(a_0, v)$ auf $[c, d]$ und $f^{(p,q)}(u, v)$ auf $[a, b] \times [c, d]$ Riemann-integrierbar sind, mit $0 \leq i \leq p \Leftrightarrow 1$ und $0 \leq j \leq q \Leftrightarrow 1$, und sich der Funktionswert $f(x, y)$ stets wie (4.19) darstellen läßt. Die Menge solcher Funktionen wird als $B_{p,q}(a_0, c_0)$ bezeichnet. Hierbei stellt $B_{p,q}(a_0, c_0)$ eine Obermenge von $C^{(r)}$ dar.

Wendet man das Fehlerfunktional $E = I \Leftrightarrow S$ auf (4.19) an, so läßt sich der Kubaturfehler $E(f)$, $f \in B_{p,q}(a_0, c_0)$, wie folgt darstellen ([146] S. 145 f) :

Satz 4.3.4 *Es seien $p, q \in \mathbb{N}$, $p + q =: r$ und $2 \leq r \leq d + 1$. Ist die Kubaturformel S exakt auf \mathbb{P}_d , dann gilt für $(a_0, c_0) \in \Omega$ und $f \in B_{p,q}(a_0, c_0)$*

$$\begin{aligned}
E(f) &= \sum_{j=0}^{q-1} \int_a^b K_{r-j,j}(u) \cdot f^{(r-j,j)}(u, c_0) du \\
&+ \sum_{i=0}^{p-1} \int_c^d K_{i,r-i}(v) \cdot f^{(i,r-i)}(a_0, v) dv \\
&+ \int_a^b \int_c^d K_{p,q}(u, v) \cdot f^{(p,q)}(u, v) du dv ,
\end{aligned} \tag{4.20}$$

mit den **Sard-Kernen**

$$\begin{aligned}
K_{p,q}(u, v) &= E \left((x \Leftrightarrow u)^{\langle p-1 \rangle} \cdot \psi(a_0, u, x) \cdot (y \Leftrightarrow v)^{\langle q-1 \rangle} \cdot \psi(c_0, v, y) \right), \\
K_{r-j,j}(u) &= E \left((x \Leftrightarrow u)^{\langle r-j-1 \rangle} \cdot \psi(a_0, u, x) \cdot (y \Leftrightarrow c_0)^{\langle j \rangle} \right), \\
K_{i,r-i}(v) &= E \left((x \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y \Leftrightarrow v)^{\langle r-i-1 \rangle} \cdot \psi(c_0, v, y) \right).
\end{aligned} \tag{4.21}$$

Die Sardschen Kernfunktionen hängen von der Kubaturformel, dem Integrationsbereich und dem Entwicklungspunkt (a_0, c_0) ab. Für symmetrische Betrachtung erweist sich $(a_0, c_0) = (\frac{a+b}{2}, \frac{c+d}{2})$ als günstig (s. Abschnitt 4.3.3).

Um den Kubaturfehler (4.20) einzuschließen, werden wiederum die Bezeichnungen

$$K^+(\cdot) := \max(K(\cdot), 0), \quad K^-(\cdot) := \max(\Leftrightarrow K(\cdot), 0) \tag{4.22}$$

verwendet. Es gilt $K(\cdot) = K^+(\cdot) \Leftrightarrow K^-(\cdot)$ und $|K(\cdot)| = K^+(\cdot) + K^-(\cdot)$.

Wegen $K^+, K^- \geq 0$ läßt sich der Kubaturfehler (4.20) unter Verwendung des erweiterten Mittelwertsatzes für die ein- und mehrfache Integralrechnung wie folgt einschränken :

Satz 4.3.5 Sei $f \in B_{p,q}(a_0, c_0)$. Unter den Bedingungen von Satz 4.3.4 gilt

$$\begin{aligned}
E(f) &\in \sum_{j=0}^{q-1} \left(f^{(r-j,j)}(\Omega_x, c_0) \cdot \int_{\Omega_x} K_{r-j,j}^+(u) du \Leftrightarrow f^{(r-j,j)}(\Omega_x, c_0) \cdot \int_{\Omega_x} K_{r-j,j}^-(u) du \right) \\
&+ \sum_{i=0}^{p-1} \left(f^{(i,r-i)}(a_0, \Omega_y) \cdot \int_{\Omega_y} K_{i,r-i}^+(v) dv \Leftrightarrow f^{(i,r-i)}(a_0, \Omega_y) \cdot \int_{\Omega_y} K_{i,r-i}^-(v) dv \right) \\
&+ f^{(p,q)}(\Omega) \cdot \int_{\Omega} K_{p,q}^+(u, v) du dv \Leftrightarrow f^{(p,q)}(\Omega) \cdot \int_{\Omega} K_{p,q}^-(u, v) du dv
\end{aligned} \tag{4.23}$$

$$\begin{aligned}
&\subseteq \sum_{j=0}^{q-1} \left(f^{(r-j,j)}(\Omega) \cdot \int_{\Omega_x} K_{r-j,j}^+(u) du \Leftrightarrow f^{(r-j,j)}(\Omega) \cdot \int_{\Omega_x} K_{r-j,j}^-(u) du \right) \\
&+ \sum_{i=0}^{p-1} \left(f^{(i,r-i)}(\Omega) \cdot \int_{\Omega_y} K_{i,r-i}^+(v) dv \Leftrightarrow f^{(i,r-i)}(\Omega) \cdot \int_{\Omega_y} K_{i,r-i}^-(v) dv \right)
\end{aligned} \tag{4.24}$$

$$+ f^{(p,q)}(\Omega) \cdot \int_{\Omega} K_{p,q}^+(u,v) du dv \Leftrightarrow f^{(p,q)}(\Omega) \cdot \int_{\Omega} K_{p,q}^-(u,v) du dv.$$

Die Einschließung (4.24) ist zwar gröber als (4.23), läßt sich jedoch effizienter berechnen. Denn bei (4.24) braucht man nur die Taylorkoeffizienten über Ω zu berechnen, während in (4.23) die Taylorkoeffizienten über $\Omega_x \times [c_0]$, $[a_0] \times \Omega_y$ und Ω zu bestimmen sind. Im konkreten kann damit bei Verwendung von (4.24) das Verfahren schneller zu Ziel führen als bei Verwendung von (4.23).

Die Fehlerdarstellung in Satz 4.3.4 sowie die Fehlereinschließung in Satz 4.3.5 gelten sowohl für Rechteck- als auch für Nicht-Rechteckgebiete. Ist der Integrationsbereich Ω^* ein Rechteck, dann soll Ω gleich Ω^* gewählt werden. Ist hingegen Ω^* ein Kreis, dann ist es zweckmäßig, Ω und (a_0, c_0) so zu wählen, daß Ω den Kreis Ω^* einschreibt und der Entwicklungspunkt gleich dem Mittelpunkt des Kreises ist, d. h. $(a_0, c_0) = (\frac{a+b}{2}, \frac{c+d}{2})$. Ferner, für das Dreieck $T_2^* = \{(x, y) \mid x \geq 0, y \geq 0, x + y \leq 1\}$, sollten $\Omega = [0, 1] \times [0, 1]$ und $(a_0, c_0) = (0, 0)$ gesetzt werden. Denn mit diesen Entwicklungspunkten verschwindet $K_{p,q}(u, v)$ identisch auf $\Omega \setminus \Omega^*$ (vgl. Stroud [146], S. 169f).

Kubaturfehler im Raum $B_{[k,l]}(a_0, c_0)$

Sei $\mathbb{P}_{d_1, d_2} := \text{Spann}(\{x^i y^j \mid 0 \leq i \leq d_1, 0 \leq j \leq d_2\})$. Die betrachtete Kubaturformel S erfülle $E(f) = 0$, für alle Polynome aus \mathbb{P}_{d_1, d_2} .

Für $k, l \in \mathbb{N}$ und $(a_0, c_0) \in \Omega$ lassen sich die Funktionen f aus $C^{k,l}$ wie folgt entwickeln (vgl. Stroud [146] S. 157)

$$\begin{aligned} f(x, y) &= \sum_{\substack{i < k \\ j < l}} (x \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y \Leftrightarrow c_0)^{\langle j \rangle} \cdot f^{(i,j)}(a_0, c_0) \\ &+ \sum_{j < l} (y \Leftrightarrow c_0)^{\langle j \rangle} \cdot \int_{a_0}^x (x \Leftrightarrow u)^{\langle k-1 \rangle} \cdot f^{(k,j)}(u, c_0) du \\ &+ \sum_{i < k} (x \Leftrightarrow a_0)^{\langle i \rangle} \cdot \int_{c_0}^y (y \Leftrightarrow v)^{\langle l-1 \rangle} \cdot f^{(i,l)}(a_0, v) dv \\ &+ \int_{a_0}^x \int_{c_0}^y (x \Leftrightarrow u)^{\langle p-1 \rangle} \cdot (y \Leftrightarrow v)^{\langle q-1 \rangle} \cdot f^{(p,q)}(u, v) du dv. \end{aligned} \tag{4.25}$$

Die Taylor-Formel (4.25) ist auch für Funktionen $f \in B_{[k,l]}(a_0, c_0)$ gültig, wobei $B_{[k,l]}(a_0, c_0)$ die Menge der Funktionen f bezeichnet, für welche $f^{(i,j)}(a_0, c_0)$, $i < k$, $j < l$, existieren, die partiellen Ableitungen $f^{(k,j)}(u, c_0)$ auf $[a, b]$, $f^{(i,l)}(a_0, v)$ auf $[c, d]$ und $f^{(k,l)}(u, v)$ auf $[a, b] \times [c, d]$ Riemann-integrierbar sind, mit $0 \leq i \leq k \Leftrightarrow 1$ und $0 \leq j \leq l \Leftrightarrow 1$, und sich der Funktionswert $f(x, y)$ stets wie (4.25) darstellen läßt. Der Funktionenraum $B_{[k,l]}(a_0, c_0)$ ist offenbar eine Obermenge von $C^{k,l}$.

Durch die Anwendung des Fehlerfunctionals $E := I \Leftrightarrow S$ auf (4.25) ergibt sich der nächste Satz ([146] S. 158).

Satz 4.3.6 *Es seien $1 \leq k \leq d_1 + 1$, $1 \leq l \leq d_2 + 1$ und $(a_0, c_0) \in \Omega$. Ist die Kubaturformel S exakt auf \mathbb{P}_{d_1, d_2} , dann gilt für $f \in B_{[k, l]}(a_0, c_0)$*

$$\begin{aligned}
E(f) &= \sum_{j=0}^{l-1} \int_a^b K_{k, j}(u) \cdot f^{(k, j)}(u, c_0) du \\
&+ \sum_{i=0}^{k-1} \int_c^d K_{i, l}(v) \cdot f^{(i, l)}(a_0, v) dv \\
&+ \int_a^b \int_c^d K_{k, l}(u, v) \cdot f^{(k, l)}(u, v) du dv,
\end{aligned} \tag{4.26}$$

mit den **Sard-Kernen**

$$\begin{aligned}
K_{k, l}(u, v) &= E \left((x \Leftrightarrow u)^{\langle k-1 \rangle} \cdot \psi(a_0, u, x) \cdot (y \Leftrightarrow v)^{\langle l-1 \rangle} \cdot \psi(c_0, v, y) \right), \\
K_{k, j}(u) &= E \left((x \Leftrightarrow u)^{\langle k-1 \rangle} \cdot \psi(a_0, u, x) \cdot (y \Leftrightarrow c_0)^{\langle j \rangle} \right), \\
K_{i, l}(v) &= E \left((x \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y \Leftrightarrow v)^{\langle l-1 \rangle} \cdot \psi(c_0, v, y) \right).
\end{aligned} \tag{4.27}$$

Die Funktionaldarstellungen der Kernfunktionen in (4.27) und (4.21) sind identisch, falls die Entwicklungspunkte (a_0, c_0) gleich gewählt sind und die Indizes der Kernfunktionen miteinander übereinstimmen. In diesem Fall unterscheiden sich die Kernfunktionen eigentlich durch den zugrundeliegenden Integrationsbereich Ω^* und die verwendeten Kubaturformeln wegen $E = I \Leftrightarrow S$.

Die Fehlerdarstellung (4.26) gilt ebenfalls für die Integrationsbereiche, die von Rechtecken verschieden sind (vgl. [146], S. 169f). Für Produktformeln ist die Fehlerdarstellung in Satz 4.3.6 adäquat. Lether [99] hat gezeigt, daß bei Produktformeln sich die Sard-Kerne (4.27) auf die Peano-Kerne K_k^x und K_l^y (s. Kapitel 3) der zugrundeliegenden Quadraturformeln S_x und S_y , mit $d_1 := \deg S_x$ und $d_2 := \deg S_y$, zurückführen lassen. Übersichtlichere Darstellungen für die Zusammenhänge zwischen den Sard-Kernen und den Peano-Kernen wurden von Luik [104] angegeben :

Satz 4.3.7 *Für die Produktformeln, die den Voraussetzungen in Satz 4.3.6 genügen, gilt*

$$\begin{aligned}
K_{k, l}(u, v) &= K_k^x(u) \cdot I_y \left((y \Leftrightarrow v)^{\langle l-1 \rangle} \cdot \psi(c_0, v, y) \right) \\
&+ K_l^y(v) \cdot I_x \left((x \Leftrightarrow u)^{\langle k-1 \rangle} \cdot \psi(a_0, u, x) \right) \\
&\Leftrightarrow K_k^x(u) \cdot K_l^y(v),
\end{aligned} \tag{4.28}$$

$$\begin{aligned}
K_{k,j}(u) &= K_k^x(u) \cdot I_y \left((y \Leftrightarrow c_0)^{\langle j \rangle} \right), \\
K_{i,l}(v) &= K_l^y(v) \cdot I_x \left((x \Leftrightarrow a_0)^{\langle i \rangle} \right).
\end{aligned} \tag{4.29}$$

Mit Hilfe der Darstellungen in (4.29) läßt sich unmittelbar nachweisen, daß die beiden univariaten Sard-Kerne identisch verschwinden, falls $(a_0, c_0) = (\frac{a+b}{2}, \frac{c+d}{2})$ und i, j ungerade sind.

Bemerkung 4.3.2 Sind i, j ungerade und $a_0 = \frac{a+b}{2}$ bzw. $c_0 = \frac{c+d}{2}$, dann gilt

$$K_{k,j}(u) \equiv 0, \quad \text{bzw.} \quad K_{i,l}(v) \equiv 0.$$

Analog zum Satz 4.3.5 läßt sich der Kubaturfehler (4.26) unter Verwendung des Mittelwertsatzes für mehrfache Integralrechnung wie folgt eingrenzen :

Satz 4.3.8 Sei $f \in B_{[k,l]}(a_0, c_0)$. Unter den Bedingungen vom Satz 4.3.6 gilt

$$\begin{aligned}
E(f) &\in \sum_{j=0}^{l-1} \left(f^{(k,j)}(\Omega_x, c_0) \cdot \int_{\Omega_x} K_{k,j}^+(u) du \Leftrightarrow f^{(k,j)}(\Omega_x, c_0) \cdot \int_{\Omega_x} K_{k,j}^-(u) du \right) \\
&+ \sum_{i=0}^{k-1} \left(f^{(i,l)}(a_0, \Omega_y) \cdot \int_{\Omega_y} K_{i,l}^+(v) dv \Leftrightarrow f^{(i,l)}(a_0, \Omega_y) \cdot \int_{\Omega_y} K_{i,l}^-(v) dv \right) \\
&+ f^{(k,l)}(\Omega) \cdot \int_{\Omega} K_{k,l}^+(u, v) du dv \Leftrightarrow f^{(k,l)}(\Omega) \cdot \int_{\Omega} K_{k,l}^-(u, v) du dv \tag{4.30}
\end{aligned}$$

$$\begin{aligned}
&\subseteq \sum_{j=0}^{l-1} \left(f^{(k,j)}(\Omega) \cdot \int_{\Omega_x} K_{k,j}^+(u) du \Leftrightarrow f^{(k,j)}(\Omega) \cdot \int_{\Omega_x} K_{k,j}^-(u) du \right) \\
&+ \sum_{i=0}^{k-1} \left(f^{(i,l)}(\Omega) \cdot \int_{\Omega_y} K_{i,l}^+(v) dv \Leftrightarrow f^{(i,l)}(\Omega) \cdot \int_{\Omega_y} K_{i,l}^-(v) dv \right) \tag{4.31} \\
&+ f^{(k,l)}(\Omega) \cdot \int_{\Omega} K_{k,l}^+(u, v) du dv \Leftrightarrow f^{(k,l)}(\Omega) \cdot \int_{\Omega} K_{k,l}^-(u, v) du dv.
\end{aligned}$$

Die Einschließung (4.31) ist zwar angesichts des Rechenaufwandes günstiger als (4.30), jedoch benötigt sie mehr Berechnungen als die Einschließung (4.17) nach Nikol'skii. Denn gemäß (4.17) werden nur zwei Reihen von Taylorkoeffizienten, $(f(\Omega))_{0,j}$ und $(f(\Omega))_{i,0}$ mit $\Omega := \Omega_x^* \times \Omega_y^*$, benötigt, während in der Darstellung (4.31) die Taylorkoeffizienten $(f(\Omega))_{i,j}$ mit $i = 0, \dots, k$ und $j = 0, \dots, l$ zu bestimmen sind. Ferner läßt sich die Fehlereinschließung (4.24) auch für Produktformeln anwenden, wobei $d = \min(d_1, d_2)$ ist. Mit (4.24) sind die Taylorkoeffizienten $(f(\Omega))_{i,j}$ mit $0 \leq i + j \leq r$ zu berechnen. Für $r = \min(k, l)$ benötigt (4.24) höchstens 1/2 Rechenaufwand als (4.31).

4.3.3 Eigenschaften der Sard-Kerne

Die Erfahrung für die Bestimmung der Restgliedfaktoren bzgl. der Peano-Kerne zeigt, daß es nützlich ist, die Eigenschaften der Kernfunktionen wie z. B. die lokale Definitheit und die Symmetrie zu berücksichtigen. Aus diesem Grund werden in diesem Abschnitt die Eigenschaften der Sard-Kernfunktionen untersucht. In [102] hat Lether zwar für die vollsymmetrischen Kubaturen die Symmetrie- und die Vollsymmetrieeigenschaften der Sard-Kerne hergeleitet, dennoch wurde die Betrachtung auf den Spezialfall mit $w \equiv 1$, $\Omega^* = [\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ und $(a_0, c_0) = (0, 0)$ beschränkt. Mit der dort ausgeführten Diskussion kann man die Eigenschaften der Sard-Kerne im Zusammenspiel mit dem Entwicklungspunkt schwer nachvollziehen. Außerdem wurde in der Diskussion von Lether eine vollsymmetrische Kubaturformel als gewichtete Summe von Produktformeln aufgefasst. Die dadurch hergeleiteten Darstellungen der Kernfunktionen lassen sich für diese Arbeit nicht direkt anwenden.

Im folgenden wird für Nicht-Produkt-Kubaturformeln eine allgemeine Diskussion über die Eigenschaften der Sard-Kerne durchgeführt. Insbesondere werden explizite Darstellungen der Sard-Kerne aufgestellt, die viel übersichtlicher und direkter anwendbar sind als es die von Lether angegebenen Darstellungen sind. Für die Diskussion wird vorausgesetzt, daß $(a_0, c_0) \in \Omega^*$ ist.

Abhängigkeit vom Entwicklungspunkt

Im Gegensatz zu den Peano-Kernen hängen die Sard-Kerne von dem Entwicklungspunkt der Taylor-Formel des Integranden ab. Um diese Abhängigkeit zu verdeutlichen, wird zunächst der Entwicklungspunkt $(a_0, c_0) \in [a, b] \times [c, d]$ ausgezeichnet :

$$\begin{aligned} K_{p,q}^{a_0,c_0}(u,v) &:= E \left((x \Leftrightarrow u)^{\langle p-1 \rangle} \cdot \psi(a_0, u, x) \cdot (y \Leftrightarrow v)^{\langle q-1 \rangle} \cdot \psi(c_0, v, y) \right) , \\ K_{r-j,j}^{a_0,c_0}(u) &:= E \left((x \Leftrightarrow u)^{\langle r-j-1 \rangle} \cdot \psi(a_0, u, x) \cdot (y \Leftrightarrow c_0)^{\langle j \rangle} \right) , \\ K_{i,r-i}^{a_0,c_0}(v) &:= E \left((x \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y \Leftrightarrow v)^{\langle r-i-1 \rangle} \cdot \psi(c_0, v, y) \right) , \end{aligned}$$

mit $p + q = r$, $i, j = 0(1)r \Leftrightarrow 2$, $2 \leq r \leq d + 1$ und $d := \deg S$.

Die univariaten Sard-Kerne $K_{r-j,j}^{a_0,c_0}(u)$ und $K_{i,r-i}^{a_0,c_0}(v)$ besitzen, wie die Peano-Kerne, auch eine Doppel-Darstellung. Diese Darstellungen sowie ihre Abhängigkeit vom Entwicklungspunkt sind in Satz 4.3.9 und Satz 4.3.10 zusammengefasst.

Satz 4.3.9 Für $(a_0, c_0) \in \Omega^*$ und $u \in [a, b]$ gelten die folgenden Eigenschaften :

1. $K_{r-j,j}^{a_0,c_0}(u)$ ist unabhängig von a_0 .

Schreibweise : $K_{r-j,j}^{-,c_0}(u) := K_{r-j,j}^{a_0,c_0}(u)$

$$2. \quad K_{r-j,j}^{-,c_0}(u) = \Leftrightarrow E \left((x \Leftrightarrow u)_-^{(r-j-1)} \cdot (y \Leftrightarrow c_0)^{(j)} \right)$$

$$3. \quad K_{r-j,j}^{-,c_0}(u) = E \left((x \Leftrightarrow u)_+^{(r-j-1)} \cdot (y \Leftrightarrow c_0)^{(j)} \right)$$

Beweis : Nach der Definition ist

$$K_{r-j,j}^{a_0,c_0}(u) = E \left((x \Leftrightarrow u)^{(r-j-1)} \psi(a_0, u, x) \cdot (y \Leftrightarrow c_0)^{(j)} \right) .$$

Ersetzt man $\psi(a_0, u, x)$ durch $\theta(u, a_0) \Leftrightarrow \theta(u, x)$, so ergibt sich

$$\begin{aligned} K_{r-j,j}^{a_0,c_0}(u) &= E \left((x \Leftrightarrow u)^{(r-j-1)} \theta(u, a_0) \cdot (y \Leftrightarrow c_0)^{(j)} \right) \\ &\Leftrightarrow E \left((x \Leftrightarrow u)^{(r-j-1)} \theta(u, x) \cdot (y \Leftrightarrow c_0)^{(j)} \right) \\ &= \Leftrightarrow E \left((x \Leftrightarrow u)^{(r-j-1)} \theta(u, x) \cdot (y \Leftrightarrow c_0)^{(j)} \right) . \end{aligned} \quad (4.32)$$

Wegen $(x \Leftrightarrow u) \cdot \theta(u, x) = (x \Leftrightarrow u) \cdot (1 \Leftrightarrow \theta(x, u))$ und $r \Leftrightarrow j \Leftrightarrow 1 \geq 1$ folgt aus (4.32)

$$\begin{aligned} K_{r-j,j}^{a_0,c_0}(u) &= \Leftrightarrow E \left((x \Leftrightarrow u)^{(r-j-1)} \cdot (y \Leftrightarrow c_0)^{(j)} \right) \\ &\quad + E \left((x \Leftrightarrow u)^{(r-j-1)} \theta(x, u) \cdot (y \Leftrightarrow c_0)^{(j)} \right) \\ &= E \left((x \Leftrightarrow u)^{(r-j-1)} \theta(x, u) \cdot (y \Leftrightarrow c_0)^{(j)} \right) . \end{aligned} \quad (4.33)$$

Die Darstellungen (4.32) und (4.33) für die Behauptungen 2 bzw. 3 sind unabhängig von a_0 . \square

Eine ähnliche Doppel-Darstellung läßt sich auch für $K_{i,r-i}^{a_0,c_0}(v)$ herleiten.

Satz 4.3.10 Für $(a_0, c_0) \in \Omega^*$ und $v \in [c, d]$ gelten die folgenden Eigenschaften :

$$1. \quad K_{i,r-i}^{a_0,c_0}(v) \text{ ist unabhängig von } c_0.$$

$$\underline{\text{Schreibweise}} : \quad K_{i,r-i}^{a_0,-}(v) := K_{i,r-i}^{a_0,c_0}(v)$$

$$2. \quad K_{i,r-i}^{a_0,-}(v) = \Leftrightarrow E \left((x \Leftrightarrow a_0)^{(i)} \cdot (y \Leftrightarrow v)_-^{(r-i-1)} \right)$$

$$3. \quad K_{i,r-i}^{a_0,-}(v) = E \left((x \Leftrightarrow a_0)^{(i)} \cdot (y \Leftrightarrow v)_+^{(r-i-1)} \right)$$

Beweis : Analog zu Satz 4.3.9 ersetzt man in der Definition

$$K_{i,r-i}^{a_0,c_0}(v) = E \left((x \Leftrightarrow a_0)^{(i)} \cdot (y \Leftrightarrow v)^{(r-i-1)} \psi(c_0, v, y) \right) .$$

$\psi(c_0, v, y)$ durch $\theta(v, c_0) \Leftrightarrow \theta(v, y)$, so ist die Darstellung im Punkt 2 zu erzielen.

Wegen $(y \Leftrightarrow v)_-^{(r-i-1)} = (y \Leftrightarrow v)^{(r-i-1)} \cdot (1 \Leftrightarrow \theta(y, v))$ folgt aus Punkt 2 die Darstellung 3. Die Darstellungen 2 und 3 sind offenbar unabhängig von c_0 . \square

Während $K_{r-j,j}^{a_0,c_0}(u)$ von a_0 und $K_{i,r-i}^{a_0,c_0}(v)$ von c_0 unabhängig ist, hängt die Darstellung des bivariaten Sard-Kerns $K_{p,q}^{a_0,c_0}(u,v)$ jedoch zugleich von der gegenseitigen Lage zwischen a_0 und u bzw. zwischen c_0 und v ab.

Satz 4.3.11 Für $(a_0, c_0) \in \Omega^*$, $(u, v) \in \Omega$ gilt :

$$K_{p,q}^{a_0,c_0}(u,v) = \begin{cases} E\left((x \Leftrightarrow u)_+^{(p-1)} \cdot (y \Leftrightarrow v)_+^{(q-1)}\right), & u \geq a_0, v \geq c_0 \\ \Leftrightarrow E\left((x \Leftrightarrow u)_-^{(p-1)} \cdot (y \Leftrightarrow v)_+^{(q-1)}\right), & u < a_0, v \geq c_0 \\ \Leftrightarrow E\left((x \Leftrightarrow u)_+^{(p-1)} \cdot (y \Leftrightarrow v)_-^{(q-1)}\right), & u \geq a_0, v < c_0 \\ E\left((x \Leftrightarrow u)_-^{(p-1)} \cdot (y \Leftrightarrow v)_-^{(q-1)}\right), & u < a_0, v < c_0 \end{cases}$$

Beweis : Ersetzt man, in der Definition von $K_{p,q}^{a_0,c_0}(u,v)$, $\psi(a_0, u, x)$ durch $\theta(u, a_0) \Leftrightarrow \theta(u, x)$ und $\psi(c_0, v, y)$ durch $\theta(v, c_0) \Leftrightarrow \theta(v, y)$, so ergibt sich

$$\begin{aligned} K_{p,q}^{a_0,c_0}(u,v) &= E\left((x \Leftrightarrow u)_-^{(p-1)} \cdot (y \Leftrightarrow v)_-^{(q-1)}\right) \\ &\Leftrightarrow E\left((x \Leftrightarrow u)^{(p-1)} \cdot (y \Leftrightarrow v)_-^{(q-1)}\right) \cdot \theta(u, a_0) \\ &\Leftrightarrow E\left((x \Leftrightarrow u)_-^{(p-1)} \cdot (y \Leftrightarrow v)^{(q-1)}\right) \cdot \theta(v, c_0). \end{aligned} \quad (4.34)$$

Die Funktionaldarstellung von $K_{p,q}^{a_0,c_0}(u,v)$ läßt sich dann in vier Fälle unterscheiden.

1. $u \geq a_0, v \geq c_0$:

Wegen $\theta(u, a_0) = \theta(v, c_0) = 1$ und

$$E\left((x \Leftrightarrow u)_-^{(p-1)} \cdot (y \Leftrightarrow v)^{(q-1)}\right) = \Leftrightarrow E\left((x \Leftrightarrow u)_+^{(p-1)} \cdot (y \Leftrightarrow v)^{(q-1)}\right)$$

$$\begin{aligned} \text{gilt} \quad K_{p,q}^{a_0,c_0}(u,v) &= \Leftrightarrow E\left((x \Leftrightarrow u)_+^{(p-1)} \cdot (y \Leftrightarrow v)_-^{(q-1)}\right) \\ &\quad + E\left((x \Leftrightarrow u)_+^{(p-1)} \cdot (y \Leftrightarrow v)^{(q-1)}\right) \\ &= E\left((x \Leftrightarrow u)_+^{(p-1)} \cdot (y \Leftrightarrow v)_+^{(q-1)}\right). \end{aligned}$$

2. $u < a_0, v \geq c_0$:

Die Behauptung gilt wegen $\theta(u, a_0) = 0$ und $\theta(v, c_0) = 1$.

3. $u \geq a_0, v < c_0$:

Die Behauptung gilt wegen $\theta(u, a_0) = 1$ und $\theta(v, c_0) = 0$.

4. $u < a_0, v < c_0$:

Die Behauptung gilt wegen $\theta(u, a_0) = \theta(v, c_0) = 0$. □

Der bivariate Sard-Kern $K_{p,q}^{a_0,c_0}(u,v)$ besitzt eventuell Sprünge entlang der Gerade $u = a_0$ bzw. $v = c_0$, falls er auf beiden Seiten dieser Geraden verschiedene Vorzeichen hat.

Stetigkeit und Differenzierbarkeit

Die univariaten Sard-Kerne $K_{r-j,j}^{a_0,c_0}(u)$ und $K_{i,r-i}^{a_0,c_0}(v)$ sind jeweils auf $[a, b]$ ($r \Leftrightarrow j \Leftrightarrow 2$)-mal bzw. auf $[c, d]$ ($r \Leftrightarrow i \Leftrightarrow 2$)-mal stetig differenzierbar. Und für den bivariaten Sard-Kern $K_{p,q}^{a_0,c_0}(u, v)$ mit $p, q \geq 2$ gilt $K_{p,q}(u, v) \in C^{p-2, q-2}(\Omega \setminus \{(u, v) \mid u = a_0 \text{ bzw. } v = c_0\})$. Ferner bestehen zwischen diesen Sard-Kernen und ihren (partiellen) Ableitungen die folgenden Beziehungen :

$$\begin{aligned} K_{r-j,j}^{(\alpha)}(u) &= (\Leftrightarrow 1)^\alpha \cdot K_{r-j-\alpha,j}(u), & \alpha = 0(1)r \Leftrightarrow j \Leftrightarrow 2, \\ K_{i,r-i}^{(\beta)}(v) &= (\Leftrightarrow 1)^\beta \cdot K_{i,r-i-\beta}(v), & \beta = 0(1)r \Leftrightarrow i \Leftrightarrow 2, \\ K_{p,q}^{(\alpha,\beta)}(u, v) &= (\Leftrightarrow 1)^{\alpha+\beta} \cdot K_{p-\alpha,q-\beta}(u, v), & \alpha = 0(1)p \Leftrightarrow 2, \quad \beta = 0(1)q \Leftrightarrow 2. \end{aligned} \tag{4.35}$$

Aussagen über die Nullstellen

Aus den Darstellungen in den Sätzen 4.3.9 und 4.3.10 folgen unmittelbar für die univariaten Sard-Kerne die nächsten Zusammenhänge :

$$\begin{aligned} K_{r-j,j}(a) &= K_{r-j,j}(b) = 0, \\ K_{i,r-i}(c) &= K_{i,r-i}(d) = 0. \end{aligned}$$

Für den bivariaten Sard-Kern gilt gemäß Satz 4.3.11

$$\begin{aligned} K_{p,q}(b, v) &= 0 \quad \text{für } 2 \leq p \text{ bzw. } p = 1 \text{ und } \hat{x} < b, \\ K_{p,q}(u, d) &= 0 \quad \text{für } 2 \leq q \text{ bzw. } q = 1 \text{ und } \hat{y} < d \end{aligned}$$

mit $\hat{x} := \max_{1 \leq i \leq M} x_i$ und $\hat{y} := \max_{1 \leq i \leq M} y_i$.

Ist darüber hinaus $a < a_0$ bzw. $c < c_0$, dann gilt zusätzlich

$$\begin{aligned} K_{p,q}(a, v) &= 0 \quad \text{für } a < a_0, \\ K_{p,q}(u, c) &= 0 \quad \text{für } c < c_0. \end{aligned}$$

Die Vielfachheit dieser Nullstellen bzw. Nullstellengeraden der Sard-Kerne läßt sich direkt aus den Zusammenhängen in (4.35) ablesen. Die univariaten Sard-Kerne $K_{r-j,j}(u)$ und $K_{i,r-i}(v)$ besitzen in den Punkten $u = a, b$ bzw. $v = c, d$ mindestens ($r \Leftrightarrow j \Leftrightarrow 1$)- bzw. ($r \Leftrightarrow i \Leftrightarrow 1$)-fache Nullstellen und der bivariate Sard-Kern $K_{p,q}(u, v)$ ($p, q \geq 2$) besitzt entlang der Geraden $u = a, b$ bzw. $v = c, d$, mit $a < a_0$ und $c < c_0$, mindestens ($p \Leftrightarrow 1$)- bzw. ($q \Leftrightarrow 1$)-fache Nullstellen.

Die Symmetrieeigenschaften

Zur Vereinfachung der Diskussion wird im folgenden der Integrationsbereich auf das Rechteck R_2^* und den Kreis C_2^* beschränkt. Ohne Beschränkung der Allgemeinheit seien $R_2^* := [\Leftrightarrow a, a] \times [\Leftrightarrow c, c]$ und $C_2^* := \{(x, y) \mid x^2 + y^2 \leq r^2\}$.

Gemäß Satz 4.3.9, Satz 4.3.10 und Satz 4.3.11 lassen sich die Sard-Kerne gemeinsam in der Form

$$E \left(f_{(\pm)}(x, u) \cdot g_{(\pm)}(y, v) \right),$$

darstellen, wobei f und g Polynome von der Gestalt $(x \Leftrightarrow u)^{\langle s \rangle}$, $(y \Leftrightarrow v)^{\langle t \rangle}$, $(x \Leftrightarrow a_0)^{\langle i \rangle}$ bzw. $(y \Leftrightarrow c_0)^{\langle j \rangle}$ sind. Nach der Definition ist das Fehlerfunktional $E(\cdot)$ gleich der Differenz des bestimmten Integrals $I(\cdot)$ und der gewichteten Summe $S(\cdot)$

$$E(\cdot) := I(\cdot) \Leftrightarrow S(\cdot),$$

$$\text{mit } I(f(x, u) \cdot g(y, v)) = \int \int_{\Omega^*} w(x, y) \cdot f(x, u) \cdot g(y, v) dx dy$$

$$\text{und } S(f(x, u) \cdot g(y, v)) = \sum_{k=1}^M w_k \cdot f(x_k, u) \cdot g(y_k, v).$$

Für symmetrische Kubaturformeln S über dem symmetrischen Bereich $\Omega^* = R_2^*$ bzw. C_2^* besitzen die Stützstellen $(\pm x_k, \pm y_k)$ ein gemeinsames Gewicht w_k . Weiter sind auf der Knotenmenge $\mathcal{M}_s = \{(\pm x_k, \pm y_k), k = 1(1)M'\}$ die folgenden Selbstabbildungen

$$\left. \begin{array}{l} (x, y) \mapsto (\Leftrightarrow x, y) \\ (x, y) \mapsto (x, \Leftrightarrow y) \\ (x, y) \mapsto (\Leftrightarrow x, \Leftrightarrow y) \end{array} \right\} \quad (4.36)$$

bijektiv. Daraus folgt das nächste Lemma.

Lemma 4.3.1 *Für symmetrische Kubaturformeln bleibt die gewichtete Summe $S(\cdot)$ in $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ unverändert unter den Abbildungen (4.36).*

Diese Invarianz gilt ebenfalls für das bestimmte Integral $I(\cdot)$.

Lemma 4.3.2 *Für symmetrische Kubaturformeln über dem symmetrischen Bereich R_2^* bzw. C_2^* bleibt das bestimmte Integral $I(\cdot)$ in $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ unverändert unter den Abbildungen (4.36).*

Beweis : Für den univariaten Sard-Kern $K_{r-j,j}(u)$ einer symmetrischen Kubaturformel gilt

$$\begin{aligned} & \int \int_{\Omega^*} w(x, y) (x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (y \Leftrightarrow c_0)^{\langle j \rangle} dy dx \\ &= \int \int_{\Omega^*} w(\Leftrightarrow x, y) (\Leftrightarrow x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (y \Leftrightarrow c_0)^{\langle j \rangle} dy dx \\ &= \int \int_{\Omega^*} w(x, y) (\Leftrightarrow x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (y \Leftrightarrow c_0)^{\langle j \rangle} dy dx \end{aligned}$$

und

$$\begin{aligned}
& \int \int_{\Omega^*} w(x, y) (x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (y \Leftrightarrow c_0)^{\langle j \rangle} dy dx \\
&= \int \int_{\Omega^*} w(x, \Leftrightarrow y) (x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (\Leftrightarrow y \Leftrightarrow c_0)^{\langle j \rangle} dy dx \\
&= \int \int_{\Omega^*} w(x, y) (x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (\Leftrightarrow y \Leftrightarrow c_0)^{\langle j \rangle} dy dx .
\end{aligned}$$

Daraus folgt die Behauptung für $K_{r-j,j}(u)$. Die Beweise für die Sard-Kerne $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ sind analog. \square

Man erhält also aus Lemma 4.3.1 und Lemma 4.3.2, in Bezug auf x und y , die symmetrische Invarianz der Sard-Kerne.

Satz 4.3.12 *Für symmetrische Kubaturformeln über dem symmetrischen Bereich R_2^* bzw. C_2^* sind die Sard-Kerne $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ invariant unter den Abbildungen (4.36).*

Mit der auf x und y bezogenen symmetrischen Invarianz der Sard-Kerne und aufgrund der Eigenschaft

$$(\Leftrightarrow x + u) \cdot \psi(a_0, \Leftrightarrow u, \Leftrightarrow x) = \Leftrightarrow(x \Leftrightarrow u) \cdot \Leftrightarrow\psi(\Leftrightarrow a_0, u, x), \quad u \neq \Leftrightarrow a_0$$

ist man in der Lage, die folgenden wichtigen symmetrischen Eigenschaften der Sard-Kerne herzuleiten.

Satz 4.3.13 *Es seien $\Omega^* = R_2^*$ bzw. C_2^* und $(a_0, c_0) \in \Omega^*$. Für symmetrische Kubaturformeln gilt*

1. $K_{r-j,j}^{-,c_0}(\Leftrightarrow u) = (\Leftrightarrow 1)^{r-j} \cdot K_{r-j,j}^{-,c_0}(u)$
2. $K_{i,r-i}^{a_0,-}(\Leftrightarrow v) = (\Leftrightarrow 1)^{r-i} \cdot K_{i,r-i}^{a_0,-}(v)$
3. $K_{p,q}^{a_0,c_0}(\Leftrightarrow u, v) = (\Leftrightarrow 1)^p \cdot K_{p,q}^{-a_0,c_0}(u, v), \quad u \neq \Leftrightarrow a_0$
4. $K_{p,q}^{a_0,c_0}(u, \Leftrightarrow v) = (\Leftrightarrow 1)^q \cdot K_{p,q}^{a_0,-c_0}(u, v), \quad v \neq \Leftrightarrow c_0$
5. $K_{p,q}^{a_0,c_0}(\Leftrightarrow u, \Leftrightarrow v) = (\Leftrightarrow 1)^r \cdot K_{p,q}^{-a_0,-c_0}(u, v), \quad u \neq \Leftrightarrow a_0 \wedge v \neq \Leftrightarrow c_0$

Im Fall $p = 1$ gilt die Behauptung 3 nur auf $[a, b] \setminus \{x_1, \dots, x_M\}$ und für $q = 1$ gilt die Behauptung 4 nur auf $[c, d] \setminus \{y_1, \dots, y_M\}$. Ist $p = q = 1$, dann gilt die Behauptung 5 nur für $(u, v) \in [a, b] \times [c, d] \setminus \{(u, v) \mid u = x_k \vee v = y_k, k = 1(1)M\}$.

Beweis vom Satz 4.3.13 :

zu 1 : Aus der symmetrischen Invarianz unter der Abbildung $(x, y) \mapsto (\Leftrightarrow x, y)$ folgt

$$K_{r-j,j}^{-,c_0}(u) = \Leftrightarrow E \left((\Leftrightarrow x \Leftrightarrow u)_-^{(r-j-1)} \cdot (y \Leftrightarrow c_0)^{(j)} \right) .$$

Wegen $r \Leftrightarrow j \Leftrightarrow 1 \geq 1$ und $(\Leftrightarrow x + y)_- = \Leftrightarrow (x \Leftrightarrow y)_+$ gilt

$$\begin{aligned} K_{r-j,j}^{-,c_0}(\Leftrightarrow u) &= \Leftrightarrow E \left((\Leftrightarrow x + u)_-^{(r-j-1)} \cdot (y \Leftrightarrow c_0)^{(j)} \right) \\ &= (\Leftrightarrow 1)^{r-j} E \left((x \Leftrightarrow u)_+^{(r-j-1)} \cdot (y \Leftrightarrow c_0)^{(j)} \right) . \end{aligned}$$

Nach Punkt 3 in Satz 4.3.9 ergibt sich

$$K_{r-j,j}^{-,c_0}(\Leftrightarrow u) = (\Leftrightarrow 1)^{r-j} \cdot K_{r-j,j}^{-,c_0}(u) .$$

zu 2 : Analog zu 1 gilt die Behauptung wegen der symmetrischen Invarianz unter der Abbildung $(x, y) \mapsto (x, \Leftrightarrow y)$.

zu 3 : Aus der Invarianz von $K_{p,q}(u, v)$ unter der Abbildung $(x, y) \mapsto (\Leftrightarrow x, y)$ folgt

$$K_{p,q}^{a_0,c_0}(\Leftrightarrow u, v) = E \left((\Leftrightarrow x + u)^{(p-1)} \psi(a_0, \Leftrightarrow u, \Leftrightarrow x) \cdot (y \Leftrightarrow v)^{(q-1)} \psi(c_0, v, y) \right) .$$

Wegen $E(\cdot) = I(\cdot) \Leftrightarrow S(\cdot)$ und für $p \geq 2$ bzw. für $p = 1 \wedge x \neq u$

$$\begin{aligned} &(\Leftrightarrow x + u)^{(p-1)} \psi(a_0, \Leftrightarrow u, \Leftrightarrow x) \\ &= (\Leftrightarrow x + u)^{(p-1)} (\theta(\Leftrightarrow u, a_0) \Leftrightarrow \theta(\Leftrightarrow u, \Leftrightarrow x)) \\ &= (\Leftrightarrow x + u)^{(p-1)} \left((1 \Leftrightarrow \theta(u, \Leftrightarrow a_0)) \Leftrightarrow (1 \Leftrightarrow \theta(u, x)) \right), \quad u \neq \Leftrightarrow a_0 \\ &= (\Leftrightarrow x + u)^{(p-1)} (\theta(u, x) \Leftrightarrow \theta(u, \Leftrightarrow a_0)) \\ &= (\Leftrightarrow x + u)^{(p-1)} \cdot \Leftrightarrow \psi(\Leftrightarrow a_0, u, x) \end{aligned}$$

gilt daher $K_{p,q}^{a_0,c_0}(\Leftrightarrow u, v) = (\Leftrightarrow 1)^p \cdot K_{p,q}^{-a_0,c_0}(u, v)$ für $u \neq \Leftrightarrow a_0$ unter der Voraussetzung $p \geq 2$ bzw. $p = 1 \wedge x_i \neq u$ für $i = 1(1)M$.

zu 4 : Analog zu 3 gilt die Behauptung wegen $\psi(c_0, \Leftrightarrow v, \Leftrightarrow y) = \Leftrightarrow \psi(\Leftrightarrow c_0, v, y)$ und der symmetrischen Invarianz unter der Abbildung $(x, y) \mapsto (x, \Leftrightarrow y)$.

zu 5 : Wegen $\psi(a_0, \Leftrightarrow u, \Leftrightarrow x) = \Leftrightarrow \psi(\Leftrightarrow a_0, u, x)$ und $\psi(c_0, \Leftrightarrow v, \Leftrightarrow y) = \Leftrightarrow \psi(\Leftrightarrow c_0, v, y)$ folgt die Behauptung, analog zu 3, aus der symmetrischen Invarianz unter der Abbildung $(x, y) \mapsto (\Leftrightarrow x, \Leftrightarrow y)$. □

Der Satz 4.3.13 impliziert, daß bei Anwendung der symmetrischen Kubaturformeln der Rechenaufwand für die Bestimmung der Restgliedfaktoren bzgl. der univariaten Sard-Kerne $K_{r-j,j}^{a_0,c_0}(u)$ und $K_{i,r-i}^{a_0,c_0}(v)$ mit $a_0 = \Leftrightarrow a_0$ und $c_0 = \Leftrightarrow c_0$ auf 1/2 reduziert werden kann. Dies gilt auch für den bivariaten Sard-Kern $K_{p,q}^{a_0,c_0}(u, v)$, falls entweder $\Leftrightarrow a_0 = a_0$ oder $\Leftrightarrow c_0 = c_0$ ist. Sind zugleich $\Leftrightarrow a_0 = a_0$ und $\Leftrightarrow c_0 = c_0$, dann kann der Rechenaufwand für die Bestimmung der Fehlerkonstanten der bivariaten Sard-Kern $K_{p,q}^{a_0,c_0}(u, v)$ sogar auf 1/4 reduziert werden.

Fazit : Seien $\Omega = R_2^*$ bzw. C_2^* und $(a_0, c_0) = (0, 0)$. Zur Berechnung der Restgliedfaktoren von symmetrischen Kubaturformeln sind nur die folgenden Bearbeitungen nötig:

1. $K_{r-j,j}(u)$ über $[0, a]$ bzw. $[0, r]$
2. $K_{i,r-i}(v)$ über $[0, c]$ bzw. $[0, r]$
3. $K_{p,q}(u, v)$ über $\{(x, y) \in \Omega \mid x \geq 0 \wedge y \geq 0\}$

Die Vollsymmetrieeigenschaften

Da vollsymmetrische Kubaturformeln auch symmetrisch sind, gilt der Satz 4.3.13 ebenfalls für vollsymmetrische Kubaturformeln.

Sei nun $R_2^* := [\Leftrightarrow a, a] \times [\Leftrightarrow a, a]$ und S eine vollsymmetrische Kubaturformel über dem vollsymmetrischen Bereich R_2^* bzw. C_2^* , dann besitzen die Stützstellen $(\pm x_k, \pm y_k)$ und $(\pm y_k, \pm x_k)$ von S ein gemeinsames Gewicht w_k . Auf der Knotenmenge $\mathcal{M}_{v_s} = \{(\pm x_k, \pm y_k), (\pm y_k, \pm x_k), k = 1(1)M'\}$ ist die Selbstabbildung

$$(x, y) \mapsto (y, x) \quad (4.37)$$

bijektiv, daher gilt das nächste Lemma.

Lemma 4.3.3 Für vollsymmetrische Kubaturformeln über R_2^* bzw. C_2^* bleibt die gewichtete Summe $S(\cdot)$ in $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ unverändert unter der Abbildung (4.37).

Ebenfalls gilt diese Invarianz für das bestimmte Integral $I(\cdot)$

Lemma 4.3.4 Für vollsymmetrische Kubaturformeln über dem vollsymmetrischen Bereich R_2^* bzw. C_2^* bleibt das bestimmte Integral $I(\cdot)$ in $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ unverändert unter der Abbildung (4.37).

Beweis : Für den univariaten Sard-Kern $K_{r-j,j}(u)$ einer vollsymmetrischen Kubaturformel gilt

$$\begin{aligned} & \int \int_{\Omega^*} w(x, y) (x \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (y \Leftrightarrow c_0)^{\langle j \rangle} dy dx \\ &= \int \int_{\Omega^*} w(y, x) (y \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (x \Leftrightarrow c_0)^{\langle j \rangle} dx dy \\ &= \int \int_{\Omega^*} w(x, y) (y \Leftrightarrow u)_+^{\langle r-j-1 \rangle} (x \Leftrightarrow c_0)^{\langle j \rangle} dx dy . \end{aligned}$$

Daraus folgt die Behauptung für $K_{r-j,j}(u)$. Die Beweise für die Sard-Kerne $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ sind analog. \square

Man erhält also aus Lemma 4.3.3 und Lemma 4.3.4, in Bezug auf x und y , die vollsymmetrische Invarianz der Sard-Kerne.

Satz 4.3.14 Für vollsymmetrische Kubaturformeln über dem vollsymmetrischen Bereich R_2^* bzw. C_2^* sind die Sard-Kerne $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ invariant unter der Abbildung (4.37).

Aus dem obigen Satz und der Definition von $K_{r-j,j}(u)$, $K_{i,r-i}(v)$ und $K_{p,q}(u, v)$ folgt der nächste Satz.

Satz 4.3.15 Es seien $\Omega^* = R_2^*$ bzw. C_2^* und $(a_0, c_0) \in \Omega^*$. Für vollsymmetrische Kubaturformeln über Ω^* gilt

1. $K_{r-j,j}^{-,c_0}(u) = K_{j,r-j}^{c_0,-}(u)$ für $j = 0(1)r \Leftrightarrow 2$
2. $K_{i,r-i}^{a_0,-}(v) = K_{r-i,i}^{-,a_0}(v)$ für $i = 0(1)r \Leftrightarrow 2$
3. $K_{p,q}^{a_0,c_0}(u, v) = K_{q,p}^{c_0,a_0}(v, u)$ mit $p + q = r$

Der Satz 4.3.15 impliziert, daß für eine vollsymmetrische Kubaturformel S die univariaten Sard-Kerne $K_{r-j,j}(u)$ und $K_{i,r-i}(v)$ das gleiche Funktionsverhalten besitzen, falls $a_0 = c_0$, $i = j$ und $r \Leftrightarrow i = r \Leftrightarrow j$ sind. Darüber hinaus lassen sich die Funktionswerte des bivariaten Sard-Kerns $K_{p,q}(u, v)$ aus den Funktionswerten von $K_{q,p}(u, v)$ erhalten, falls $a_0 = c_0$ ist, da in diesem Fall die Funktionswerte der beiden bivariaten Sard-Kerne unter der Abbildung (4.37) aufeinander abgebildet werden.

Aus Satz 4.3.13 und dem obigen Satz kann das nächste Fazit gezogen werden.

Fazit : Es seien $\Omega^* = R_2^*$ bzw. C_2^* und $(a_0, c_0) = (0, 0)$. Zur Berechnung der Restgliedfaktoren für vollsymmetrische Kubaturformeln über Ω^* sind nur die folgenden Bearbeitungen nötig :

1. $K_{r-j,j}(u)$ über $[0, a]$ bzw. $[0, r]$, $j = 0(1)r \Leftrightarrow 2$
2. $K_{r-q,q}(u, v)$ über $\{(x, y) \in \Omega \mid x \geq 0 \wedge y \geq 0\}$ $q = 1(1)\lfloor \frac{r}{2} \rfloor$

Explizite Darstellungen und lokale Definitheiten

In den Sätzen 4.3.9, 4.3.10 und 4.3.11 wurden verschiedene Funktionaldarstellungen für die Sard-Kerne angegeben. Auf dieser Basis werden in diesem Abschnitt explizite Darstellungen hergeleitet, die durch die Gewichte und die Koordinaten der Stützstellen der betrachteten Kubaturformel ausgedrückt sind. Mit Hilfe dieser expliziten Darstellungen können speziell für die univariaten Sard-Kerne der symmetrischen Kubaturformeln bzgl. des Entwicklungspunktes $(a_0, c_0) = (0, 0)$ zusätzliche günstige Eigenschaften hergeleitet werden.

Im folgenden wird die Betrachtung auf $w(x, y) \equiv 1$ beschränkt.

Satz 4.3.16 Seien $w(x, y) \equiv 1$, $\Omega^* = [a, b] \times [c, d]$ und $(a_0, c_0) \in \Omega^*$. Es gilt für $j = 0(1)r \Leftrightarrow 2$, $2 \leq r \leq d + 1$, $d := \deg S$

$$1. \quad K_{r-j,j}^{-,c_0}(u) = (a \Leftrightarrow u)^{\langle r-j \rangle} \cdot k_{1,j}^{c_0} + \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_-^{\langle r-j-1 \rangle} \cdot (y_k \Leftrightarrow c_0)^{\langle j \rangle}$$

$$2. \quad K_{r-j,j}^{-,c_0}(u) = (b \Leftrightarrow u)^{\langle r-j \rangle} \cdot k_{1,j}^{c_0} \Leftrightarrow \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_+^{\langle r-j-1 \rangle} \cdot (y_k \Leftrightarrow c_0)^{\langle j \rangle}$$

$$\text{mit } k_{1,j}^{c_0} = (d \Leftrightarrow c_0)^{\langle j+1 \rangle} \Leftrightarrow (c \Leftrightarrow c_0)^{\langle j+1 \rangle}.$$

Beweis : Wegen $E(\cdot) = I(\cdot) \Leftrightarrow S(\cdot)$ folgt die Darstellung 1 aus Satz 4.3.9, Punkt 2, und die Darstellung 2 folgt aus Satz 4.3.9, Punkt 3, wobei gilt

$$k_{1,j}^{c_0} = \int_c^d (y \Leftrightarrow c_0)^{\langle j \rangle} dy. \quad \square$$

Seien $\tilde{x} := \min_{1 \leq i \leq M} x_i$ und $\hat{x} := \max_{1 \leq i \leq M} x_i$. Mit den Darstellungen in Satz 4.3.16 sind zwei lokale Definitheiten von $K_{r-j,j}^{a_0, c_0}(u)$ trivialerweise zu erkennen.

Korollar 4.3.16 Unter den Bedingungen des obigen Satzes ist $K_{r-j,j}^{-,c_0}(u)$ definit auf $[a, \tilde{x}]$ und $[\hat{x}, b]$.

Zur Berechnung der Restgliedfaktoren bzgl. $K_{r-j,j}^{-,c_0}(u)$ braucht keine Nullstelle im Inneren der beiden Teilbereiche bestimmt zu werden.

Der univariate Sard-Kern $K_{i,r-i}^{a_0, -}(v)$ läßt sich ähnlich wie $K_{r-j,j}^{-,c_0}(u)$ darstellen.

Satz 4.3.17 Es seien $w(x, y) \equiv 1$, $\Omega^* = [a, b] \times [c, d]$ und $(a_0, c_0) \in \Omega^*$. Für $i = 0(1)r \Leftrightarrow 2$, $2 \leq r \leq d + 1$ und $d := \deg S$ gilt

$$1. \quad K_{i,r-i}^{a_0, -}(v) = k_{2,i}^{a_0} \cdot (c \Leftrightarrow v)^{\langle r-i \rangle} + \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y_k \Leftrightarrow v)_-^{\langle r-i-1 \rangle}$$

$$2. \quad K_{i,r-i}^{a_0, -}(v) = k_{2,i}^{a_0} \cdot (d \Leftrightarrow v)^{\langle r-i \rangle} \Leftrightarrow \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow a_0)^{\langle i \rangle} \cdot (y_k \Leftrightarrow v)_+^{\langle r-i-1 \rangle}$$

$$\text{mit } k_{2,i}^{a_0} = (b \Leftrightarrow a_0)^{\langle i+1 \rangle} \Leftrightarrow (a \Leftrightarrow a_0)^{\langle i+1 \rangle}.$$

Beweis : Die Darstellung 1 folgt aus Satz 4.3.10, Punkt 2, und die Darstellung 2 folgt aus Satz 4.3.10, Punkt 3, wobei gilt

$$k_{2,i}^{a_0} = \int_a^b (x \Leftrightarrow a_0)^{\langle i \rangle} dx. \quad \square$$

Seien $\check{y} := \min_{1 \leq i \leq M} y_i$ und $\hat{y} := \max_{1 \leq i \leq M} y_i$. In Analogie zu $K_{r-j,j}^{-,c_0}(u)$ besitzt $K_{i,r-i}^{a_0,-}(v)$ ebenfalls zwei triviale lokale Definitheiten.

Korollar 4.3.17 *Unter den Voraussetzungen des letzten Satzes ist $K_{i,r-i}^{a_0,-}(v)$ definit auf $[c, \check{y}]$ und $[\hat{y}, d]$.*

Die explizite Darstellung des bivariaten Sard-Kerns folgt direkt aus Satz 4.3.11, welche im nächsten Satz formuliert wird.

Satz 4.3.18 *Es seien $w(x, y) \equiv 1$, $\Omega^* = [a, b] \times [c, d]$ und $(a_0, c_0) \in \Omega^*$. Für $p+q = r$, $2 \leq r \leq d+1$ und $d := \deg S$ gilt*

$$K_{p,q}^{a_0,c_0}(u, v) = \begin{cases} (b \Leftrightarrow u)^{\langle p \rangle} \cdot (d \Leftrightarrow v)^{\langle q \rangle} \Leftrightarrow \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_+^{\langle p-1 \rangle} \cdot (y_k \Leftrightarrow v)_+^{\langle q-1 \rangle}, \\ \hspace{15em} u \geq a_0, v \geq c_0, \\ (a \Leftrightarrow u)^{\langle p \rangle} \cdot (d \Leftrightarrow v)^{\langle q \rangle} + \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_-^{\langle p-1 \rangle} \cdot (y_k \Leftrightarrow v)_+^{\langle q-1 \rangle}, \\ \hspace{15em} u < a_0, v \geq c_0, \\ (b \Leftrightarrow u)^{\langle p \rangle} \cdot (c \Leftrightarrow v)^{\langle q \rangle} + \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_+^{\langle p-1 \rangle} \cdot (y_k \Leftrightarrow v)_-^{\langle q-1 \rangle}, \\ \hspace{15em} u \geq a_0, v < c_0, \\ (a \Leftrightarrow u)^{\langle p \rangle} \cdot (c \Leftrightarrow v)^{\langle q \rangle} \Leftrightarrow \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_-^{\langle p-1 \rangle} \cdot (y_k \Leftrightarrow v)_-^{\langle q-1 \rangle}, \\ \hspace{15em} u < a_0, v < c_0. \end{cases}$$

Beweis : Nach Satz 4.3.11 braucht man, wegen $E(\cdot) = I(\cdot) \Leftrightarrow S(\cdot)$, nur noch die explizite Darstellung des bestimmten Integrals $I(\cdot)$ herzuleiten.

1. Für $u \geq a_0$, $v \geq c_0$ gilt

$$\begin{aligned} I\left((x \Leftrightarrow u)_+^{\langle p-1 \rangle} \cdot (y \Leftrightarrow v)_+^{\langle q-1 \rangle}\right) &= \int_a^b (x \Leftrightarrow u)^{\langle p-1 \rangle} dx \cdot \int_v^d (y \Leftrightarrow v)^{\langle q-1 \rangle} dy \\ &= (b \Leftrightarrow u)^{\langle p \rangle} \cdot (d \Leftrightarrow v)^{\langle q \rangle}. \end{aligned}$$

2. Die Diskussionen für die anderen drei Fälle sind analog zu 1. □

Seien $\check{x} := \min_{1 \leq i \leq M} x_i$, $\hat{x} := \max_{1 \leq i \leq M} x_i$, $\check{y} := \min_{1 \leq i \leq M} y_i$ und $\hat{y} := \max_{1 \leq i \leq M} y_i$. Ferner seien $\hat{\hat{x}} := \max(\hat{x}, a_0)$, $\hat{\hat{y}} := \max(\hat{y}, c_0)$, $\check{\check{x}} := \min(\check{x}, a_0)$ und $\check{\check{y}} := \min(\check{y}, c_0)$. Mit den Darstellungen in Satz 4.3.18 läßt sich unmittelbar prüfen, daß der bivariate Sard-Kern die folgenden lokalen Definitheiten besitzt.

Korollar 4.3.18 *Unter der Bedingungen des obigen Satzes besitzt der bivariate Sard-Kern die folgenden Definitheiten.*

- $K_{p,q}^{a_0,c_0}(u,v)$ ist positiv auf $[\hat{x}, b] \times [c_0, d]$ und $[a_0, b] \times [\hat{y}, d]$.
- $K_{p,q}^{a_0,c_0}(u,v)$ ist definit auf $[a, \check{x}] \times [c_0, d]$ und $[a, a_0] \times [\hat{y}, d]$.
- $K_{p,q}^{a_0,c_0}(u,v)$ ist definit auf $[\hat{x}, b] \times [c, c_0]$ und $[a_0, b] \times [c, \check{y}]$.
- $K_{p,q}^{a_0,c_0}(u,v)$ ist definit auf $[a, \check{x}] \times [c, c_0]$ und $[a, a_0] \times [c, \check{y}]$.

Ist $p = 1$, dann gelten die Definitheiten nur für $u \neq \hat{x}$. Ebenfalls gelten die Definitheiten nur für $v \neq \hat{y}$, falls $q = 1$ ist.

Aus den Darstellungen in Satz 4.3.16 und Satz 4.3.17 können weitere günstige Eigenschaften für die univariaten Sard-Kerne der symmetrischen Kubaturformeln bzgl. des Entwicklungspunktes $(a_0, c_0) = (0, 0)$ hergeleitet werden.

Satz 4.3.19 *Es seien $w(x, y) \equiv 1$, $\Omega^* := [\Leftrightarrow a, a] \times [\Leftrightarrow c, c]$ und $(a_0, c_0) = (0, 0)$. Für symmetrische Kubaturformel S , mit $\deg S =: d$, gilt*

$$\begin{aligned}
1. \quad k_{1,j} &:= k_{1,j}^0 = \begin{cases} 0, & j \text{ ungerade} \\ 2c^{\langle j+1 \rangle}, & j \text{ gerade} \end{cases} \\
2. \quad k_{2,i} &:= k_{2,i}^0 = \begin{cases} 0, & i \text{ ungerade} \\ 2a^{\langle i+1 \rangle}, & i \text{ gerade} \end{cases} \\
3. \quad K_{r-j,j}(u) &= \begin{cases} 0, & j \text{ ungerade} \\ (a \Leftrightarrow u)^{\langle r-j \rangle} \cdot k_{1,j} \Leftrightarrow \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_+^{\langle r-j-1 \rangle} \cdot y_k^{\langle j \rangle}, & j \text{ gerade} \end{cases} \\
4. \quad K_{i,r-i}(v) &= \begin{cases} 0, & i \text{ ungerade} \\ k_{2,i} \cdot (c \Leftrightarrow v)^{\langle r-i \rangle} \Leftrightarrow \sum_{k=1}^M w_k \cdot x_k^{\langle i \rangle} \cdot (y_k \Leftrightarrow v)_+^{\langle r-i-1 \rangle}, & i \text{ gerade}, \end{cases}
\end{aligned}$$

mit $2 \leq r \leq d+1$ und $i, j = 0(1)r \Leftrightarrow 2$.

Beweis : Die Behauptungen in Punkten 1 und 2 folgen unmittelbar aus Satz 4.3.16 und Satz 4.3.17. Für den Sard-Kern $K_{r-j,j}^{0,0}(u)$ gilt nach Satz 4.3.16

$$K_{r-j,j}^{0,0}(u) = (b \Leftrightarrow u)^{\langle r-j \rangle} \cdot k_{1,j} \Leftrightarrow \sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_+^{\langle r-j-1 \rangle} \cdot y_k^{\langle j \rangle}.$$

Der erste Term verschwindet für ungerade j , wegen $k_{1,j} = 0$. Für symmetrische Kubaturformeln kann der zweite Term in die Form

$$\sum_{k=1}^M w_k \cdot (x_k \Leftrightarrow u)_+^{\langle r-j-1 \rangle} \cdot y_k^{\langle j \rangle} = \sum_{k'=1}^{M'} w_{k'} \cdot (\pm x_{k'} \Leftrightarrow u)_+^{\langle r-j-1 \rangle} \cdot (\pm y_{k'})^{\langle j \rangle}$$

umgeschrieben werden. Diese Summe verschwindet auch für ungerade j , wegen $(\Leftrightarrow y_{k'})^j = \Leftrightarrow y_{k'}^j$. Daraus folgt die Behauptung 3.

Für die Behauptung 4 ist die Diskussion ähnlich. \square

Der Satz 4.3.19 zeigt, daß für symmetrische Kubaturformeln über dem symmetrischen Integrationsbereich $\Omega^* := [\Leftrightarrow a, a] \times [\Leftrightarrow c, c]$ die univariaten Sard-Kerne bzgl. des Entwicklungspunkts $(a_0, c_0) = (0, 0)$ identisch verschwinden, falls sie ungerade Indizes i, j besitzen. Der Übersichtlichkeit halber werden für diesen Spezialfall die Symmetrieeigenschaften der univariaten Sard-Kerne im nächsten Korollar zusammengefasst.

Satz 4.3.20 *Es seien $w(x, y) \equiv 1$, $\Omega^* := [\Leftrightarrow a, a] \times [\Leftrightarrow c, c]$ und $(a_0, c_0) = (0, 0)$. Für symmetrische Kubaturformeln gilt :*

1. $K_{r-j,j}(u) \equiv K_{i,r-i}(v) \equiv 0$, falls i, j ungerade,
2. $K_{r-j,j}(\Leftrightarrow u) = (\Leftrightarrow 1)^r \cdot K_{r-j,j}(u)$, falls j gerade,
3. $K_{i,r-i}(\Leftrightarrow v) = (\Leftrightarrow 1)^r \cdot K_{i,r-i}(v)$, falls i gerade,
4. $K_{p,q}(\Leftrightarrow u, v) = (\Leftrightarrow 1)^p \cdot K_{p,q}(u, v)$, $u \neq 0$
5. $K_{p,q}(u, \Leftrightarrow v) = (\Leftrightarrow 1)^q \cdot K_{p,q}(u, v)$, $v \neq 0$
6. $K_{p,q}(\Leftrightarrow u, \Leftrightarrow v) = (\Leftrightarrow 1)^r \cdot K_{p,q}(u, v)$, mit $r = p + q$ und $u \neq 0 \wedge v \neq 0$.

Für $p = 1$ bzw. $q = 1$ gelten die Behauptungen 4, 5 und 6 wie in Satz 4.3.13. Unter den Voraussetzungen in Satz 4.3.20 kann der Aufwand für die Bestimmung der Restgliedfaktoren der symmetrischen Kubaturformeln um den Faktor 4 reduziert werden.

Sind darüber hinaus S und Ω^* vollsymmetrisch, so kann der Rechenaufwand, gemäß dem Satz 4.3.15, insgesamt um den Faktor 8 verringert werden. Denn in diesem Fall erfüllen die Kernfunktionen zusätzlich die folgenden Eigenschaften.

Korollar 4.3.15 *Es seien $w(x, y) \equiv 1$, $\Omega^* := [\Leftrightarrow a, a] \times [\Leftrightarrow a, a]$ und $(a_0, c_0) = (0, 0)$. Für vollsymmetrische Kubaturformeln S gilt :*

1. $K_{r-j,j}(u) = K_{i,r-i}(v)$, falls $i = j$ und $u = v$ sind,
2. $K_{p,q}(u, v) = K_{q,p}(v, u)$, mit $p + q = r$.

Bemerkung 4.3.3

Für Kubaturformeln, die bzgl. des Kreises $\Omega^* = C_2^* = \{(x, y) \mid x^2 + y^2 \leq 1\}$ konstruiert sind, lassen sich die Sard-Kerne nicht in einer allgemeinen expliziten Formel darstellen. Die

expliziten Darstellungen können nur einzeln angegeben werden, wenn die Indizes bekannt sind. Beispielsweise gilt für den univariaten Sard-Kern $K_{r-j,j}^{0,0}(u)$ einer symmetrischen Kubaturformel S bzgl. $w \equiv 1$ über C_2^* :

$$\begin{aligned} K_{6,2}^{0,0}(u) &= \frac{1}{2 \cdot 5!} \left(\Leftrightarrow \frac{5}{128} u \pi \Leftrightarrow \frac{5}{24} u^3 \pi \Leftrightarrow \frac{1}{8} u^5 \pi + \frac{5}{12} \sqrt{1 \Leftrightarrow u^2} u^4 + \frac{5}{96} (1 \Leftrightarrow u^2)^{3/2} u^2 \right. \\ &+ \frac{325}{1512} (1 \Leftrightarrow u^2)^{5/2} u^2 + \frac{5}{64} \sqrt{1 \Leftrightarrow u^2} u^2 + \frac{1}{6} u^6 (1 \Leftrightarrow u^2)^{3/2} + \frac{5}{64} u \arcsin(u) \\ &+ \frac{1}{4} u^6 \sqrt{1 \Leftrightarrow u^2} + \frac{5}{18} u^4 (1 \Leftrightarrow u^2)^{3/2} + \frac{1}{4} u^5 \arcsin(u) + \frac{16}{945} (1 \Leftrightarrow u^2)^{5/2} \\ &+ \left. \frac{5}{12} u^3 \arcsin(u) + \frac{125}{756} (1 \Leftrightarrow u^2)^{5/2} u^4 \right) \\ &\Leftrightarrow S((x \Leftrightarrow u)_+^{(5)} \cdot y^{(2)}), \quad u \geq 0, \end{aligned}$$

$$\begin{aligned} K_{2,2}^{0,0}(u) &= \frac{1}{2} \left(\Leftrightarrow \frac{1}{8} u \pi + \frac{2}{15} (1 \Leftrightarrow u^2)^{5/2} + \frac{1}{4} \sqrt{1 \Leftrightarrow u^2} u^2 + \frac{1}{6} (1 \Leftrightarrow u^2)^{3/2} + \frac{1}{4} u \arcsin(u) \right) \\ &\Leftrightarrow S((x \Leftrightarrow u)_+^{(1)} \cdot y^{(2)}), \quad u \geq 0 \end{aligned}$$

und

$$\begin{aligned} K_{2,0}^{0,0}(u) &= \left(\Leftrightarrow \frac{1}{2} u \pi + \frac{2}{3} (1 \Leftrightarrow u^2)^{3/2} + \sqrt{1 \Leftrightarrow u^2} u^2 + u \arcsin(u) \right) \\ &\Leftrightarrow S((x \Leftrightarrow u)_+^{(1)} \cdot y^{(0)}), \quad u \geq 0 \end{aligned}$$

usw..

Die expliziten Darstellungen der Sard-Kerne $K_{i,r-i}^{0,0}(v)$ und $K_{p,q}^{0,0}(u, v)$ können ebenfalls nur einzeln angegeben werden.

Aus dem Grund, daß diese expliziten Darstellungen recht komplizierte Gestalt besitzen können, und daß bei Anwendung eines adaptiven numerischen Integrationsverfahrens die kreisförmigen Intergrationsbereiche in Rechteckgebiete transformiert werden müssen, ist es entbehrlich, die Fehlerkonstanten für die Kubaturformeln über dem Kreis C_2^* zu berechnen.

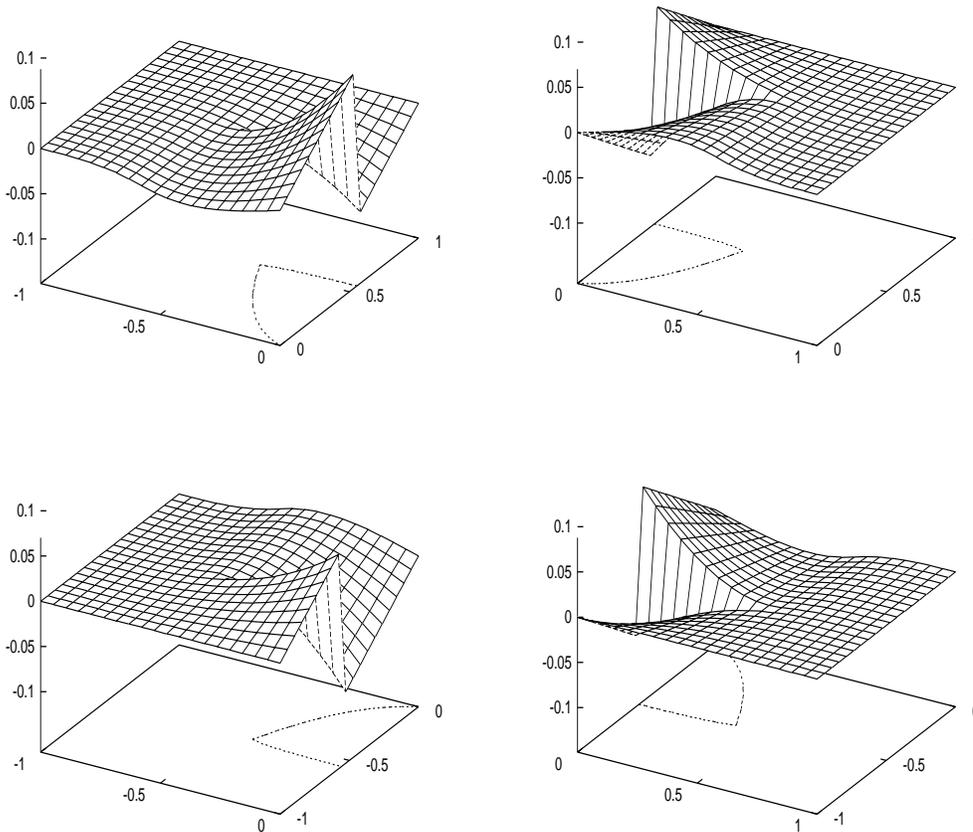
Die Betrachtung in diesem Kapitel beschränkt sich zwar auf das Zweidimensionale, jedoch lassen sich sowohl die Fehlerdarstellungen der Kubaturformeln als auch die Herleitung der Symmetrie- und Vollsymmetrieeigenschaften auf höhere Dimensionen verallgemeinern.

Graphische Darstellungen

Um die Eigenschaften des bivariaten Sard-Kerns, die in Satz 4.3.20 und Korollar 4.3.15 erwähnt wurden, zu veranschaulichen, werden im folgenden die Graphen von $K_{p,q}^{0,0}(u, v)$, für vollsymmetrische Kubaturformeln, über dem Integrationsbereich $\Omega := [\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$

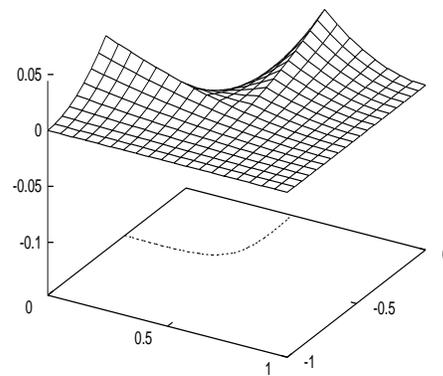
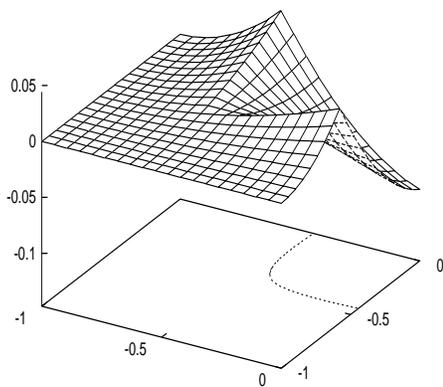
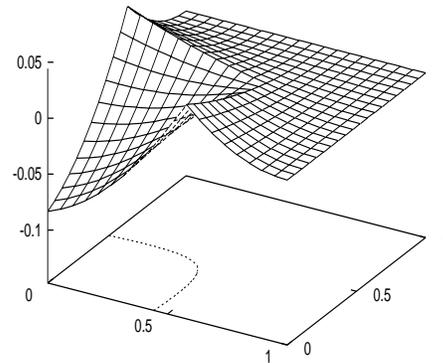
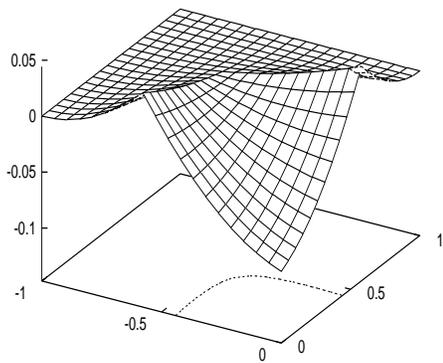
dargestellt. Dabei wird im Rest dieser Arbeit die Abkürzung VSK_M für die vollsymmetrische Kubaturformel, PGL_M für die Produkt-Gauß-Legendre-Formel und PCC_M für die Produkt-Clenshaw-Curtis-Formel, welche M Stützstellen besitzen, verwendet.

Graphen des bivariaten Sard-Kerns $K_{p,q}^{0,0}(u, v)$ von VSK_4 mit Exaktheitsgrad 3



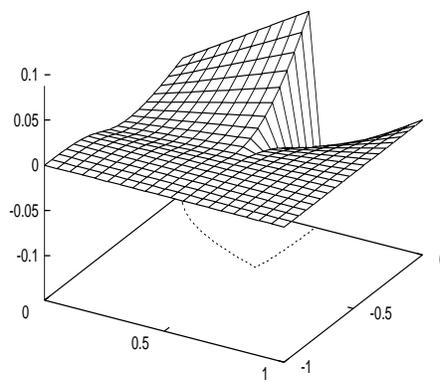
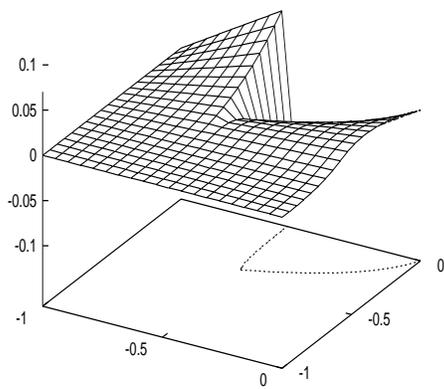
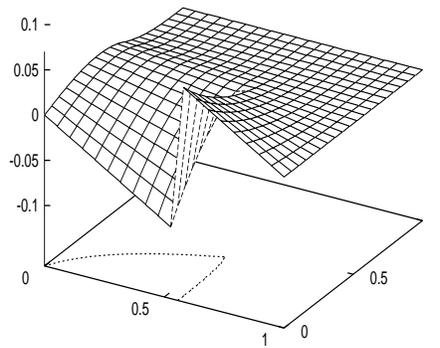
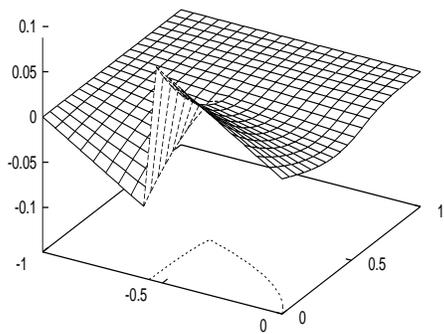
$$K_{3,1}^{0,0}(u, v), \quad \deg S = 3$$

Der Sard-Kern $K_{3,1}(u, v)$ ist eine ungerade Funktion, sowohl bzgl. u als auch bzgl. v . Wegen $q = 1$ ist $K_{3,1}(u, v)$ bzgl. v eine Sprungfunktion.



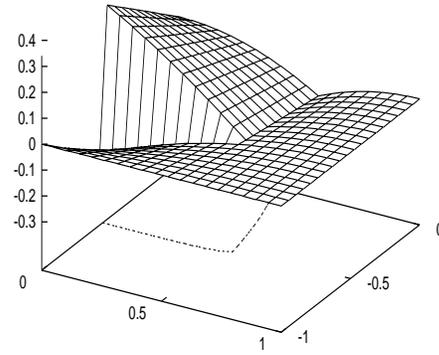
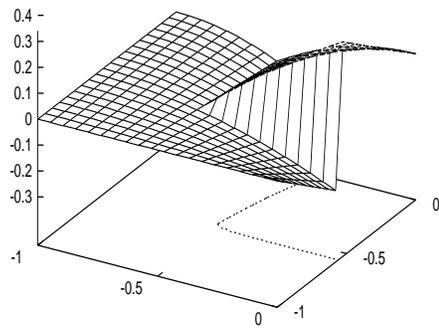
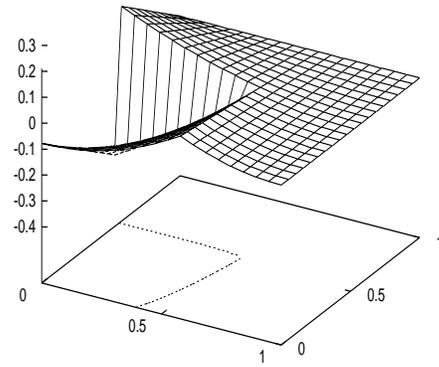
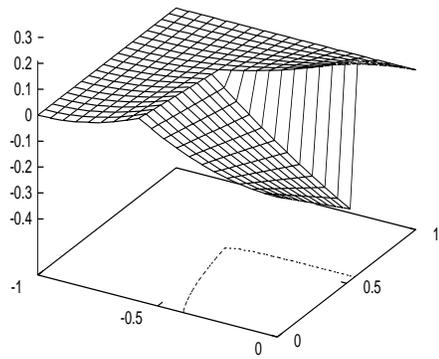
$$K_{2,2}^{0,0}(u,v), \quad \deg S = 3$$

Für $p = 2$ und $q = 2$ ist der Sard-Kern $K_{2,2}(u,v)$ bzgl. u bzw. v jeweils eine gerade Funktion.



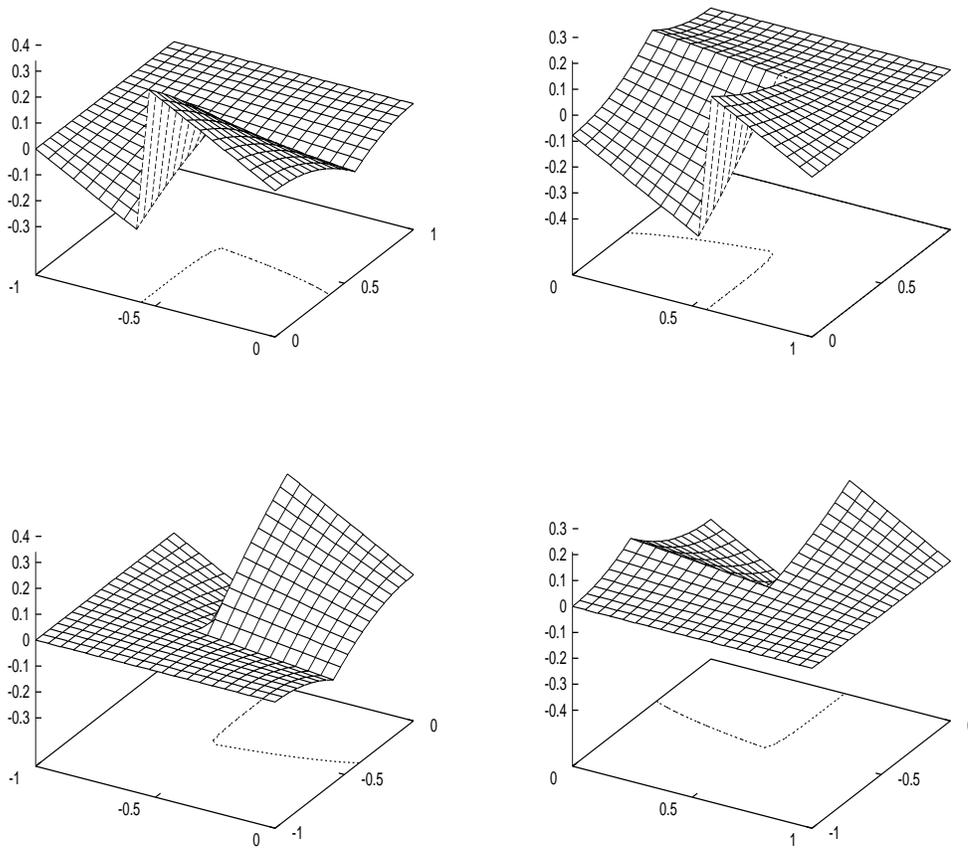
$$K_{1,3}^{0,0}(u,v), \quad \deg S = 3$$

Das Schaubild von $K_{1,3}^{0,0}(u,v)$ ist identisch zu dem Spiegelungsbild von $K_{3,1}^{-0,0}(u,v)$ zu der Achse $u = v$.



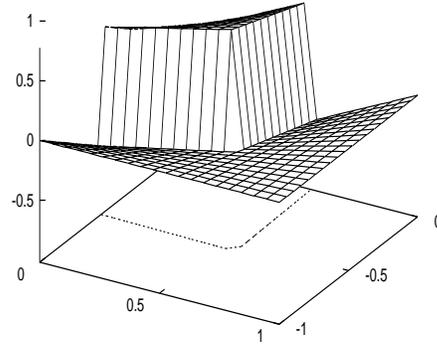
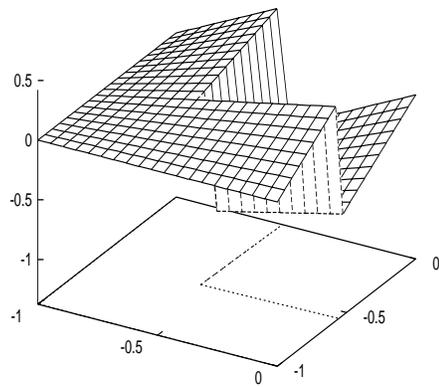
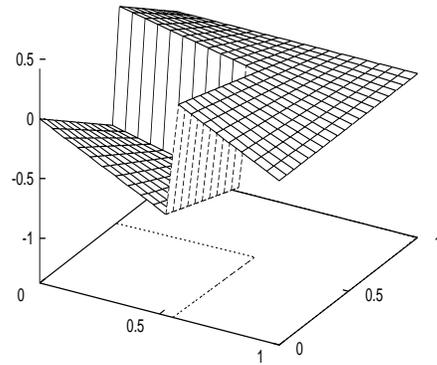
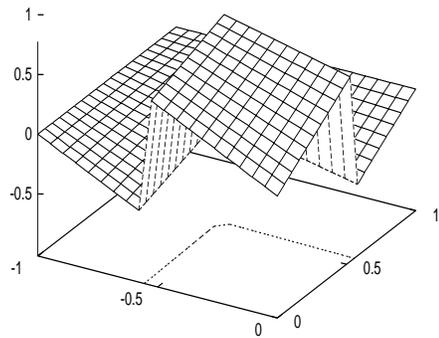
$$K_{2,1}^{0,0}(u,v), \quad \deg S = 3$$

In diesem Fall ist der bivariate Sard-Kern bzgl. u eine gerade und bzgl. v eine ungerade Funktion ist.



$$K_{1,2}^{0,0}(u,v), \quad \deg S = 3$$

Die Funktionswerte von $K_{1,2}^{0,0}(u,v)$ können ebenfalls durch die Spiegelung von $K_{2,1}^{0,0}(u,v)$ zu der Achse $u = v$ erhalten werden, nämlich $K_{1,2}^{0,0}(u,v) = K_{2,1}^{0,0}(v,u)$.



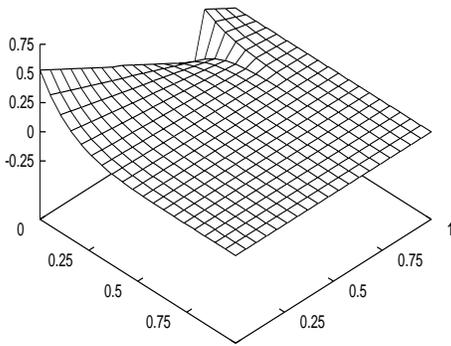
$$K_{1,1}^{0,0}(u,v), \quad \deg S = 3$$

Für $(p,q) = (1,1)$ ist der bivariate Sard-Kern $K_{1,1}(u,v)$ eine Sprungfunktion, die bzgl. u bzw. v eine ungerade Funktion ist.

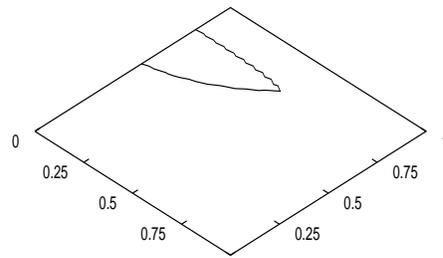
Um die Informationen über die bivariaten Sard-Kern $K_{p,q}^{0,0}(u,v)$, $p = r \Leftrightarrow q$, einer vollsymmetrischen Kubaturformel zu erfahren, braucht man $K_{r-q,q}(u,v)$ nur über $[0, 1] \times [0, 1]$, für $q = 1(1)\lfloor \frac{r}{2} \rfloor$, zu betrachten (vgl. auch das Fazit zum Satz 4.3.15).

Im folgenden werden die Graphen des bivariaten Sard-Kerns $K_{r-q,q}(u,v)$ der vollsymmetrischen Kubaturformel VSK_{12} mit dem Exaktheitsgrad 7 zusammen mit den Nullstellenkurven $K_{r-q,q}(u,v) = 0$ dargestellt.

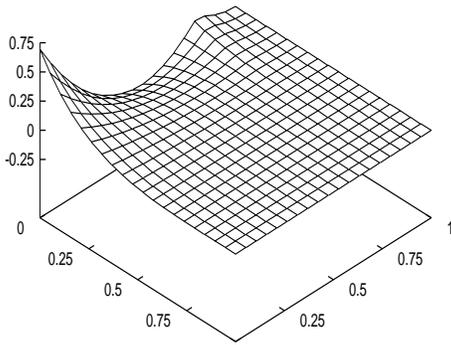
Graphen des bivariaten Sard-Kerns $K_{p,q}^{0,0}(u,v)$ von VSK_{12} mit dem Exaktheitsgrad 7



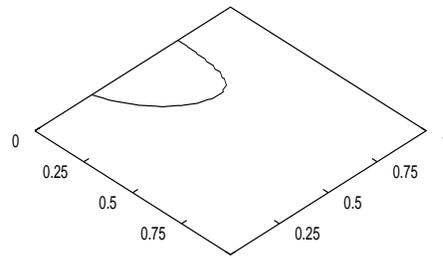
$$\underline{5000 \cdot K_{7,1}^{0,0}(u,v)}$$



$$\underline{K_{7,1}^{0,0}(u,v) = 0}$$

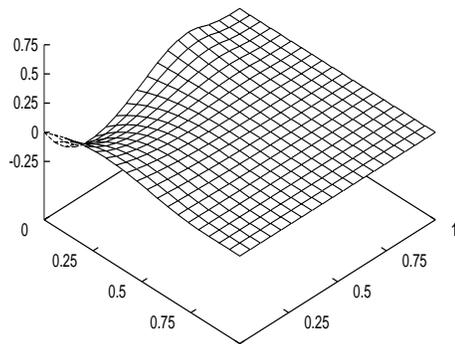


$$\underline{5000 \cdot K_{6,2}^{0,0}(u,v)}$$

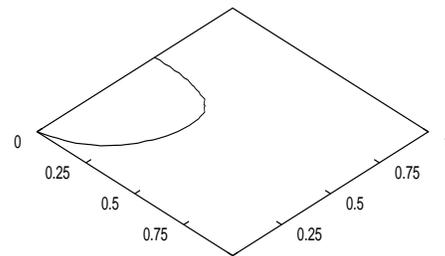


$$\underline{K_{6,2}^{0,0}(u,v) = 0}$$

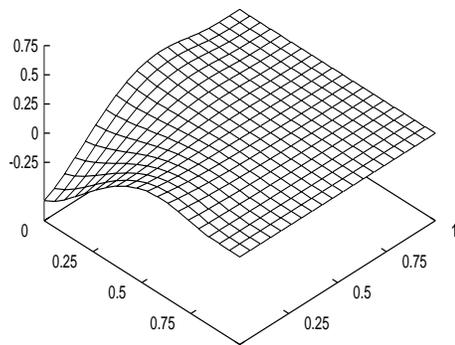
$$K_{p,q}^{0,0}(u,v), \quad \deg S = 7$$



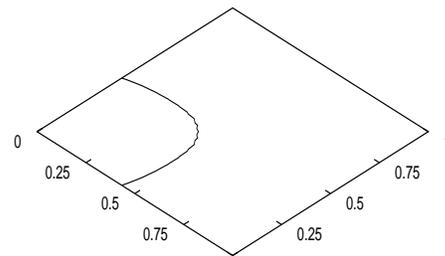
$$\underline{5000 \cdot K_{5,3}^{0,0}(u,v)}$$



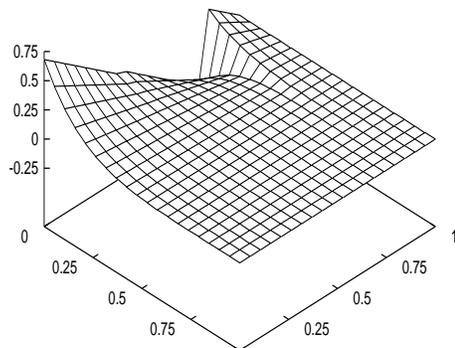
$$\underline{K_{5,3}^{0,0}(u,v) = 0}$$



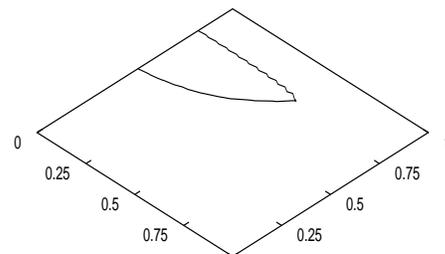
$$\underline{5000 \cdot K_{4,4}^{0,0}(u,v)}$$



$$\underline{K_{4,4}^{0,0}(u,v) = 0}$$

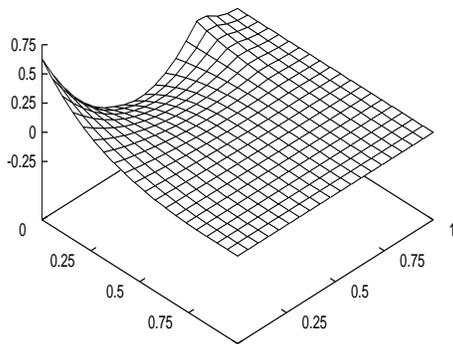


$$\underline{1000 \cdot K_{6,1}^{0,0}(u,v)}$$

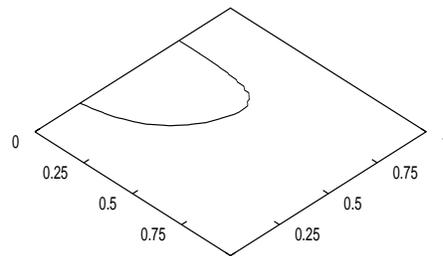


$$\underline{K_{6,1}^{0,0}(u,v) = 0}$$

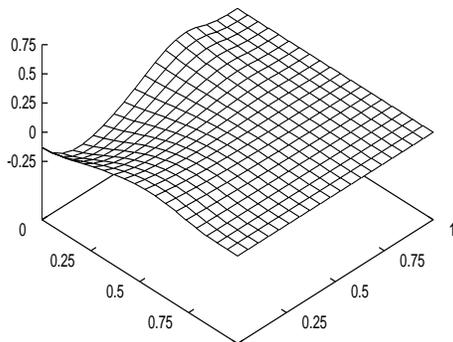
$$K_{p,q}^{0,0}(u,v), \quad \deg S = 7$$



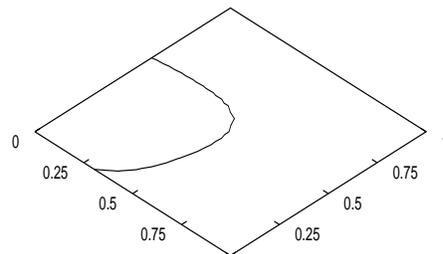
$$\underline{1000 \cdot K_{5,2}^{0,0}(u,v)}$$



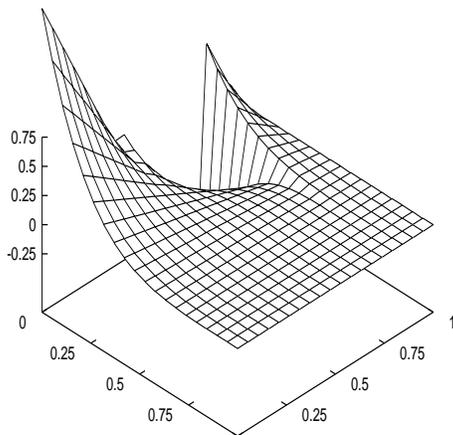
$$\underline{K_{5,2}^{0,0}(u,v) = 0}$$



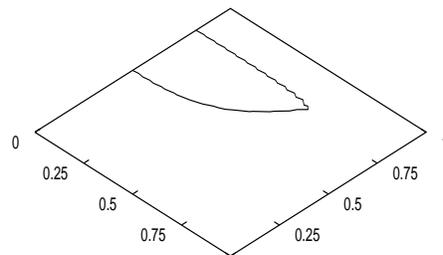
$$\underline{1000 \cdot K_{4,3}^{0,0}(u,v)}$$



$$\underline{K_{4,3}^{0,0}(u,v) = 0}$$

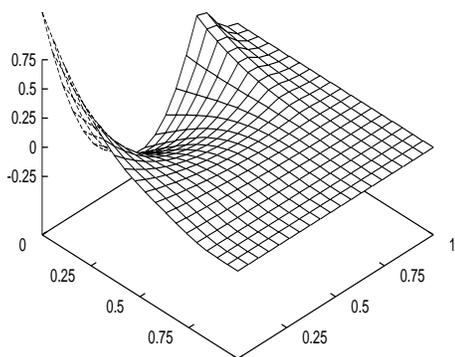


$$\underline{500 \cdot K_{5,1}^{0,0}(u,v)}$$

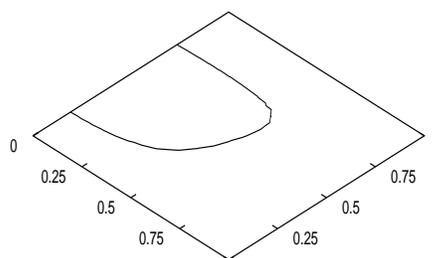


$$\underline{K_{5,1}^{0,0}(u,v) = 0}$$

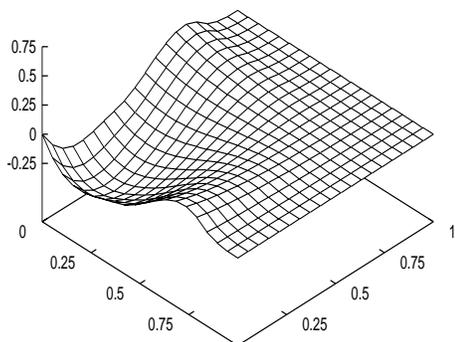
$$K_{p,q}^{0,0}(u,v), \quad \deg S = 7$$



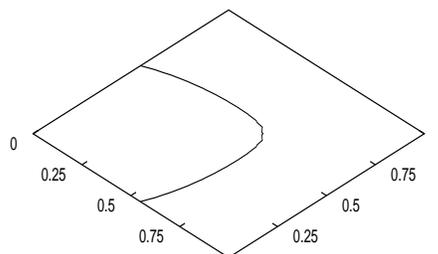
$500 \cdot K_{4,2}^{0,0}(u, v)$



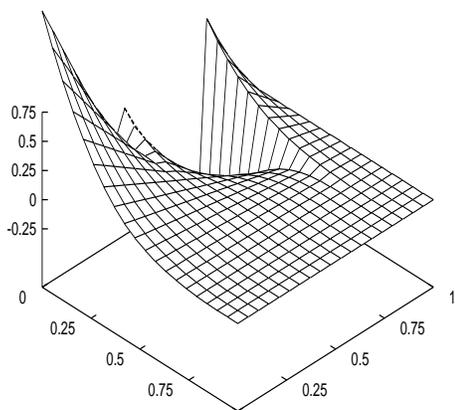
$K_{4,2}^{0,0}(u, v) = 0$



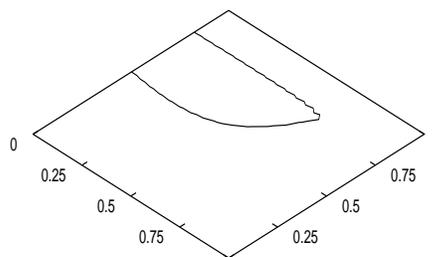
$500 \cdot K_{3,3}^{0,0}(u, v)$



$K_{3,3}^{0,0}(u, v) = 0$

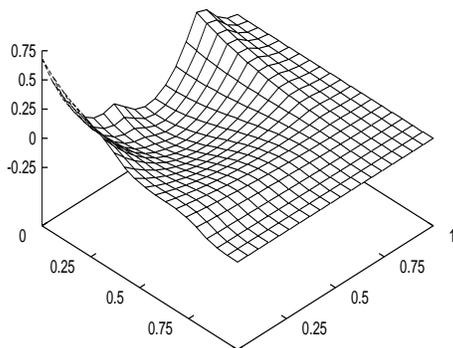


$100 \cdot K_{4,1}^{0,0}(u, v)$

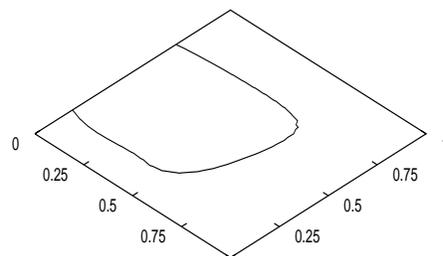


$K_{4,1}^{0,0}(u, v) = 0$

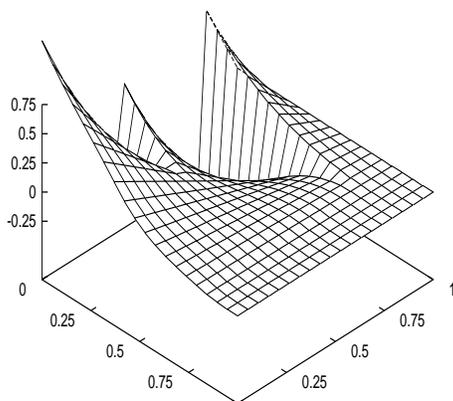
$K_{p,q}^{0,0}(u, v), \quad \deg S = 7$



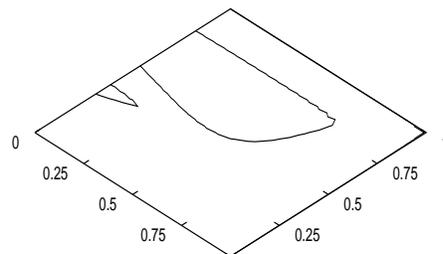
$$\underline{100 \cdot K_{3,2}^{0,0}(u,v)}$$



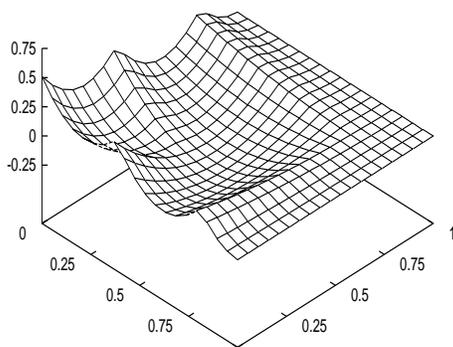
$$\underline{K_{3,2}^{0,0}(u,v) = 0}$$



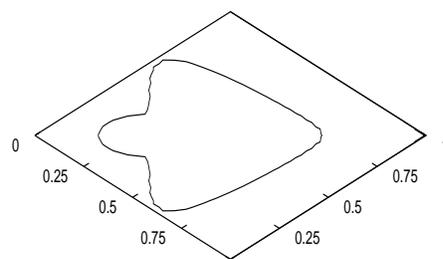
$$\underline{25 \cdot K_{3,1}^{0,0}(u,v)}$$



$$\underline{K_{3,1}^{0,0}(u,v) = 0}$$

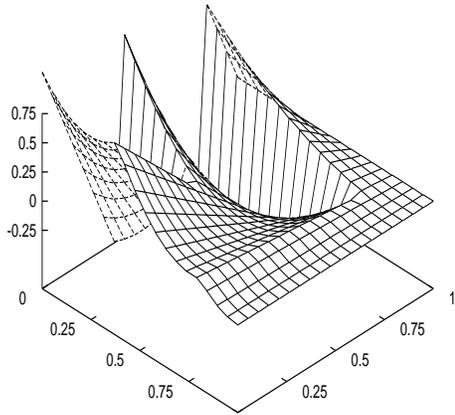


$$\underline{25 \cdot K_{2,2}^{0,0}(u,v)}$$

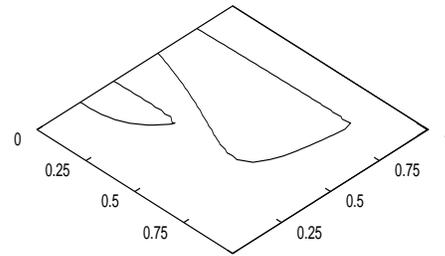


$$\underline{K_{2,2}^{0,0}(u,v) = 0}$$

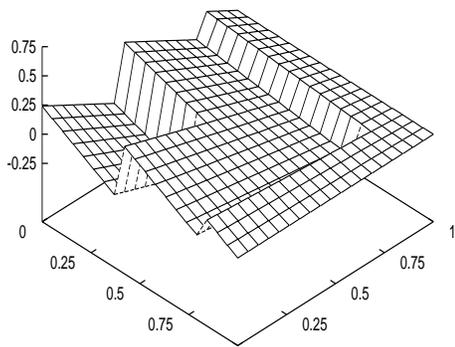
$$K_{p,q}^{0,0}(u,v), \quad \deg S = 7$$



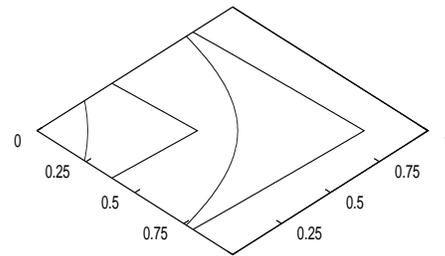
$$\underline{10 \cdot K_{2,1}^{0,0}(u, v)}$$



$$\underline{K_{2,1}^{0,0}(u, v) = 0}$$



$$\underline{K_{1,1}^{0,0}(u, v)}$$



$$\underline{K_{1,1}^{0,0}(u, v) = 0}$$

$$K_{p,q}^{0,0}(u, v), \quad \deg S = 7$$

Für die betrachtete vollsymmetrische Kubaturformel mit dem Exaktheitsgrad 7, bestehen die Konturlinien zur Höhe null bzgl. des bivariaten Sard-Kerns $K_{p,q}^{0,0}(u, v)$ aus einer bzw. mehreren (geschlossenen) Kurven.

4.3.4 Verifizierte Berechnung der Restgliedfaktoren

Es erweist sich als schwierige Aufgabe, die Fehlerkonstanten der Integrationsformeln mit Hilfe der Nullstellen der Kernfunktionen zu bestimmen. Die Komplexität nimmt insbesondere im Mehrdimensionalen zu. Bereits für die eindimensionalen Quadraturfehler haben Stroud und Secrest [147] nur die Fehlerkonstanten bzgl. der Peano-Kerne erster und zweiter Ordnung mit Hilfe der Nullstellen der Kernfunktionen berechnet. Für die Peano-Kerne höherer Ordnungen und die Sard-Kerne (vgl. [146]) hat er jedoch die L_1 -Norm näherungsweise bestimmt. Weiter hat Lether [101], [102] die L_1 -Norm der Sard-Kerne ebenfalls mit Hilfe von Kubaturformeln approximativ errechnet. Bislang bestanden noch keine verifizierten numerischen Ergebnisse für die L_1 -Norm der Sard-Kerne.

Im Abschnitt 3.4.3 wurde zwar anhand von numerischen Ergebnissen gezeigt, daß die L_1 -Norm der Peano-Kerne mit Hilfe des erweiterten Intervall-Newton-Verfahrens bestimmt werden kann, jedoch läßt sich dieser Ansatz, die Fehlerkonstanten mittels der Nullstellen der Kernfunktionen zu berechnen, nicht problemlos auf das Mehrdimensionale erweitern, da folgende Schwierigkeiten auftreten :

1. Die Nullstellenbestimmung für die mehrdimensionalen reellwertigen Funktionen ist viel aufwendiger als dieselbe Aufgabe im Eindimensionalen.
2. Im Eindimensionalen ist es einfach, die Integrale $\int K^+$ und $\int K^-$ direkt über den größten zusammenhängenden definiten Teilbereichen zu berechnen, wodurch sowohl der Rechenaufwand als auch der Rundungsfehler reduziert werden kann. Im Mehrdimensionalen ist es jedoch schwierig, möglichst große zusammenhängende Teilbereiche, auf denen die Kernfunktionen keinen Vorzeichenwechsel haben, zu finden.

Außerdem sind für die stückweise Betrachtung mehrdimensionaler Kernfunktionen mehr Sortierungen der einzelnen Koordinaten der Stützstellen notwendig. Trotz dieser Schwierigkeiten bzw. des Zusatzaufwandes wurden in dieser Arbeit die Restgliedfaktoren bzgl. der Sard-Kerne auch mit Hilfe der Nullstellen der Kernfunktionen berechnet und zwar aus zwei Gründen. Zum einen waren für die herangezogenen vollsymmetrischen Kubaturformeln noch keine Fehlerkonstanten vorhanden, und zum anderen kann garantiert werden, daß unter Berücksichtigung der Nullstellen sich die Fehlerkonstanten ohne Über- und Unterschätzung bestimmen lassen, falls der Rundungsfehler nicht überwiegend groß ist. Für weitere Betrachtungen stehen dann die in dieser Arbeit erzielten numerischen Ergebnisse als Referenzdaten zur Verfügung.

Im allgemeinen lassen sich die im Abschnitt 3.4.3 beschriebenen Berechnungsideen für die Bestimmung der Fehlerkonstanten bzgl. der Sard-Kerne anwenden. Im Abschnitt 3.4.3 wurde erklärt, daß die Bestimmung der Restgliedfaktoren nur für $w \equiv 1$ sinnvoll ist. Aus gleichem Grund wird die folgende Diskussion ebenfalls auf $w \equiv 1$ beschränkt. Unter dieser Voraussetzung reicht es aus, ausschließlich den Standardintegrationsbereich $\Omega := [\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ zu betrachten. Für andere Rechteckbereiche können die Fehlerkonstanten durch Transformationen aus den berechneten Werten erhalten werden. Über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$

sind die Extrema der Kernfunktionen von der Größenordnung $O(1)$, was sich mit Hilfe der expliziten Darstellung der Kernfunktionen einfach prüfen läßt.

Die Fehlerkonstanten $\int K^+$ und $\int K^-$ bzgl. der eindimensionalen Sard-Kerne können, wie bei den Peano-Kernen, mit Hilfe des erweiterten Intervall-Newton-Verfahrens berechnet werden.

Für die mehrdimensionalen Sard-Kerne lassen sich die Nullstellen durch Einsatz des von Hansen und Neumaier verallgemeinerten Intervall-Newton-Verfahrens (s. Abschnitt 2.4.2) bestimmen. Um die Integrale $\int K^+$ und $\int K^-$ über möglichst große definite Teilbereiche zu berechnen, wird der Intervall-Newton-Schritt iteriert durchgeführt, bis der bearbeitete Teilbereich, auf dem die Kernfunktion das Vorzeichen wechselt, nicht mehr verkleinert werden kann. Aus der Überlegung der Stetigkeit und der Differenzierbarkeit der Kernfunktionen wird die konkrete Berechnung, wie im Eindimensionalen, jeweils über den Teilbereichen, die durch die Grenzen der Stützstelleneinschließungen $[x_i]$ und $[y_i]$ begrenzt werden, durchgeführt. Für die Stützstellen (x_i, y_i) wird vorausgesetzt, daß die Einschließungen $[x_i], [y_j]$ hinreichend fein sind.

Es sei $a =: x_0 \leq x_1 < \dots < x_M \leq x_{M+1} := b$ und $c =: y_0 \leq y_1 < \dots < y_M \leq y_{M+1} := d$. Dabei muß (x_i, y_i) keine Stützstelle sein. Mit der oben erwähnten Zerlegung werden die folgenden Integraleinschließungen

$$\int_{\underline{x}_i}^{\bar{x}_i} \int_{\underline{y}_j}^{\bar{y}_{j+1}} K^{+/-} dy dx \in K^{+/-}([x_i], [\bar{y}_j, \underline{y}_{j+1}]) \cdot d([x_i]) \cdot (\underline{y}_{j+1} \Leftrightarrow \bar{y}_j), \quad (4.38)$$

$$\int_{\bar{x}_i}^{\underline{x}_{i+1}} \int_{\underline{y}_j}^{\bar{y}_j} K^{+/-} dy dx \in K^{+/-}([\bar{x}_i, \underline{x}_{i+1}], [y_j]) \cdot d([y_j]) \cdot (\underline{x}_{i+1} \Leftrightarrow \bar{x}_i) \quad (4.39)$$

und

$$\int_{\underline{x}_i}^{\bar{x}_i} \int_{\underline{y}_j}^{\bar{y}_j} K^{+/-} dy dx \in K^{+/-}([x_i], [y_j]) \cdot d([x_i]) \cdot d([y_j]) \quad (4.40)$$

berechnet.

Durch Anwendung des verallgemeinerten Intervall-Newton-Verfahrens auf $[\bar{x}_i, \underline{x}_{i+1}] \times [\bar{y}_j, \underline{y}_{j+1}]$ sind neue Teilbereiche entstanden, auf welchen die Sard-Kerne entweder definit sind oder das Vorzeichen wechseln. Über den definiten Teilbereichen lassen sich die Integrale $\int K^{+/-}$ mit Hilfe der Stammfunktion berechnen. Für die verbleibenden Teilbereiche $[z_x] \times [z_y]$ mit $0 \in K([z_x], [z_y])$ und $d([z_x]), d([z_y]) \leq \varepsilon$ werden die Integraleinschließungen

$$\int_{[z_x]} \int_{[z_y]} K^{+/-} dy dx \in K^{+/-}([z_x], [z_y]) \cdot d([z_x]) \cdot d([z_y])$$

bestimmt.

Damit können die gesuchten Restgliedfaktoren $[\int_{\Omega} K^{+/-}]$ durch Aufsummieren der berechneten Teilintegraleinschließungen gewonnen werden. Hierbei ist zu bemerken, daß die Einschließungen (4.38), (4.39) und (4.40) wegen der kleinen Größenordnung der Extrema der Kernfunktionen und aufgrund der kleinen Durchmesser $d([x_i])$ und $d([y_j])$, praktisch sehr kleinen Beitrag zu den Fehlerkonstanten $\int_{\Omega} K^{+/-}$ liefern.

Gemäß der oben beschriebenen Ideen wurden die Fehlerkonstanten bzgl. der ein- und zweidimensionalen Sard-Kerne für die in [45] konstruierten vollsymmetrischen Kubaturformeln von Exaktheitsgraden 3 und 7, unter Berücksichtigung der Symmetrie-, der Vollsymmetrieeigenschaften und der lokalen Definitheit, verifiziert berechnet. Um die Effizienz zu erhöhen, wurden in dem Intervall-Newton-Schritt die verbesserte Mittelwertform für die partiellen Ableitungen eingesetzt. Diese numerischen Ergebnisse sind in Tabellen 4.1, 4.2, 4.3 und 4.4 aufgelistet.

Aus gleichem Grund wie bei den Peano-Kernen sind $\int_{-1}^1 K_{r-j,j}(u)du = 0$, $\int_{-1}^1 K_{i,r-i}(v)dv = 0$ und $\int_{-1}^1 \int_{-1}^1 K_{p,q}(u,v) du dv = 0$, für $i, j = 0(1)r \Leftrightarrow 2$, $p + q = r$ und $2 \leq r < d + 1$. Daher gelten die folgenden Gleichungen

$$\begin{aligned}
 P_{r-j,j} &:= \int_{-1}^1 K_{r-j,j}^+(u) du &= \int_{-1}^1 K_{r-j,j}^-(u) du &=: \Leftrightarrow N_{r-j,j}, \\
 P_{i,r-i} &:= \int_{-1}^1 K_{i,r-i}^+(v) dv &= \int_{-1}^1 K_{i,r-i}^-(v) dv &=: \Leftrightarrow N_{i,r-i}, \\
 P_{p,q} &:= \int_{-1}^1 \int_{-1}^1 K_{p,q}^+(u,v) du dv &= \int_{-1}^1 \int_{-1}^1 K_{p,q}^-(u,v) du dv &=: \Leftrightarrow N_{p,q}
 \end{aligned} \tag{4.41}$$

mit $i, j = 0(1)r \Leftrightarrow 2$, $p + q = r$, $2 \leq r < d + 1$ und $d := \deg S$.

Diese Identitäten sind für beliebige Kubaturformeln S gültig, auch wenn S nicht symmetrisch ist.

In [146] hat Stroud für verschiedene Kubaturformeln die L_1 -Norm der Sard-Kerne K berechnet, es gilt $\|K\|_1 = P \Leftrightarrow N$. Da die Fehlerkonstanten ein weiteres Maß zur Beurteilung der Integrationsformeln bieten, werden die Restgliedfaktoren, die im Rahmen dieser Arbeit verifiziert berechnet wurden, und die von Stroud angegebenen Fehlerkonstanten in den Tabellen 4.5, 4.6, 4.7 und 4.8 zum Vergleich zusammengestellt. Die betrachteten Kubaturformeln haben zwar verschiedene Knotenzahl, jedoch ist die Differenz nicht groß. Ferner sind die Fehlerkonstanten bzgl. Kubaturformeln mit größerer Knotenzahl nicht unbedingt kleiner.

Tabelle 4.1: 1-dim. Fehlerkonstanten : VSK_4 über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 3

r	j	$(r \Leftrightarrow j)! j! [P_{r-j,j}]$	$(r \Leftrightarrow j)! j! [N_{r-j,j}]$
2	0	$1.6225813128483 \binom{7}{5} E \Leftrightarrow 01$	$\Leftrightarrow 1.6225813128483 \binom{5}{7} E \Leftrightarrow 01$
3	0	$1.150998205402 \binom{6}{4} E \Leftrightarrow 01$	$\Leftrightarrow 1.150998205402 \binom{4}{6} E \Leftrightarrow 01$
4	0	$3.55555555555555 \binom{7}{5} E \Leftrightarrow 01$	$0.0000000000000000 E + 00$
4	2	$5.408604376161 \binom{22}{19} E \Leftrightarrow 02$	$\Leftrightarrow 5.408604376161 \binom{1}{3} E \Leftrightarrow 02$

Tabelle 4.2: 2-dim. Fehlerkonstanten : VSK_4 über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 3

p	q	$p! q! [P_{p,q}]$	$p! q! [N_{p,q}]$
1	1	$4.9202258114172 \binom{5}{2} E \Leftrightarrow 01$	$\Leftrightarrow 4.9202258114172 \binom{2}{5} E \Leftrightarrow 01$
1	2	$2.27700 \binom{8}{6} E \Leftrightarrow 01$	$\Leftrightarrow 2.27700 \binom{6}{8} E \Leftrightarrow 01$
1	3	$1.1890 \binom{3}{1} E \Leftrightarrow 01$	$\Leftrightarrow 1.1890 \binom{1}{3} E \Leftrightarrow 01$
2	2	$9.3743 \binom{50}{43} E \Leftrightarrow 02$	$\Leftrightarrow 9.3743 \binom{43}{50} E \Leftrightarrow 02$

Tabelle 4.3: 1-dim. Fehlerkonstanten von VSK_{12} über $[\Leftarrow 1, 1] \times [\Leftarrow 1, 1]$ vom Grad 7

r	j	$(r \Leftarrow j)! j! [P_{r-j,j}]$	$(r \Leftarrow j)! j! [N_{r-j,j}]$
2	0	$2.5696377921 \binom{5}{3} E \Leftarrow 02$	$\Leftarrow 2.5696377921 \binom{3}{5} E \Leftarrow 02$
3	0	$7.735598956 \binom{24}{17} E \Leftarrow 03$	$\Leftarrow 7.735598956 \binom{17}{24} E \Leftarrow 03$
4	0	$3.559164873 \binom{7}{6} E \Leftarrow 03$	$\Leftarrow 3.559164873 \binom{6}{7} E \Leftarrow 03$
4	2	$1.4041060935 \binom{6}{5} E \Leftarrow 02$	$\Leftarrow 1.4041060935 \binom{5}{6} E \Leftarrow 02$
5	0	$2.301854937 \binom{6}{5} E \Leftarrow 03$	$\Leftarrow 2.301854937 \binom{5}{6} E \Leftarrow 03$
5	2	$6.470633179 \binom{63}{59} E \Leftarrow 03$	$\Leftarrow 6.470633179 \binom{59}{63} E \Leftarrow 03$
6	0	$2.1085230146 \binom{2}{0} E \Leftarrow 03$	$\Leftarrow 2.108523014 \binom{5}{7} E \Leftarrow 03$
6	2	$4.36905157 \binom{802}{799} E \Leftarrow 03$	$\Leftarrow 4.36905157 \binom{798}{804} E \Leftarrow 03$
6	4	$1.5148887692 \binom{3}{2} E \Leftarrow 02$	$\Leftarrow 1.5148887692 \binom{2}{3} E \Leftarrow 02$
7	0	$2.8040832105 \binom{3}{0} E \Leftarrow 03$	$\Leftarrow 2.8040832105 \binom{0}{3} E \Leftarrow 03$
7	2	$4.992542858 \binom{41}{38} E \Leftarrow 03$	$\Leftarrow 4.992542858 \binom{38}{41} E \Leftarrow 03$
7	4	$7.7362999664 \binom{5}{1} E \Leftarrow 03$	$\Leftarrow 7.7362999664 \binom{1}{5} E \Leftarrow 03$
8	0	$1.318511144 \binom{30}{29} E \Leftarrow 02$	$0.0000000000000000 E + 00$
8	2	$2.0441347270 \binom{7}{5} E \Leftarrow 02$	$0.0000000000000000 E + 00$
8	4	$1.6218454827 \binom{5}{4} E \Leftarrow 03$	$\Leftarrow 1.1656688688 \binom{2}{4} E \Leftarrow 02$
8	6	$2.4556440235 \binom{2}{0} E \Leftarrow 02$	$\Leftarrow 4.115092964 \binom{47}{50} E \Leftarrow 03$

Tabelle 4.4: 2-dim. Fehlerkonstanten von VSK_{12} über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ vom Grad 7

p	q	$p!q! [P_{p,q}]$	$p!q! [N_{p,q}]$
1	1	$1.8245 \binom{4}{2} E \Leftrightarrow 01$	$\Leftrightarrow 1.8245 \binom{2}{4} E \Leftrightarrow 01$
1	2	$8.8110 \binom{5}{1} E \Leftrightarrow 02$	$\Leftrightarrow 8.8110 \binom{1}{5} E \Leftrightarrow 02$
1	3	$7.5258 \binom{55}{43} E \Leftrightarrow 02$	$\Leftrightarrow 7.5258 \binom{43}{55} E \Leftrightarrow 02$
2	2	$2.1273 \binom{7}{4} E \Leftrightarrow 02$	$\Leftrightarrow 2.1273 \binom{4}{7} E \Leftrightarrow 02$
1	4	$6.83982 \binom{6}{1} E \Leftrightarrow 02$	$\Leftrightarrow 6.83982 \binom{1}{6} E \Leftrightarrow 02$
2	3	$1.7703 \binom{12}{09} E \Leftrightarrow 02$	$\Leftrightarrow 1.7703 \binom{09}{12} E \Leftrightarrow 02$
1	5	$6.1998 \binom{51}{47} E \Leftrightarrow 02$	$\Leftrightarrow 6.1998 \binom{47}{51} E \Leftrightarrow 02$
2	4	$1.6366 \binom{12}{09} E \Leftrightarrow 02$	$\Leftrightarrow 1.6366 \binom{09}{12} E \Leftrightarrow 02$
3	3	$1.2427 \binom{9}{8} E \Leftrightarrow 02$	$\Leftrightarrow 1.2427 \binom{8}{9} E \Leftrightarrow 02$
1	6	$5.68319 \binom{7}{2} E \Leftrightarrow 02$	$\Leftrightarrow 5.68319 \binom{2}{7} E \Leftrightarrow 02$
2	5	$1.5228 \binom{8}{7} E \Leftrightarrow 02$	$\Leftrightarrow 1.5228 \binom{7}{8} E \Leftrightarrow 02$
3	4	$9.833 \binom{20}{16} E \Leftrightarrow 03$	$\Leftrightarrow 9.833 \binom{16}{20} E \Leftrightarrow 03$
1	7	$5.32166 \binom{8}{2} E \Leftrightarrow 02$	$\Leftrightarrow 5.32166 \binom{2}{8} E \Leftrightarrow 02$
2	6	$2.5540 \binom{4}{2} E \Leftrightarrow 02$	$\Leftrightarrow 5.09 \binom{894}{900} E \Leftrightarrow 03$
3	5	$6.6096 \binom{7}{1} E \Leftrightarrow 03$	$\Leftrightarrow 6.6096 \binom{1}{7} E \Leftrightarrow 03$
4	4	$3.085 \binom{62}{59} E \Leftrightarrow 03$	$\Leftrightarrow 1.3120 \binom{4}{5} E \Leftrightarrow 02$

Tabelle 4.5: Vergleich 1-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 3$

		$(r \Leftrightarrow j)! j! \ K_{r-j,j}(u)\ _1$			
		Verifikation	STROUD		
r	j	VSK_4	PGL_4	VSK_5	VSK_9
2	0	$3.2451626256967 \binom{4}{1} E \Leftrightarrow 01$	$3.25E \Leftrightarrow 01$	$3.95E \Leftrightarrow 01$	$3.95E \Leftrightarrow 01$
3	0	$2.30199641080 \binom{51}{49} E \Leftrightarrow 01$	$2.30E \Leftrightarrow 01$	$3.33E \Leftrightarrow 01$	$3.33E \Leftrightarrow 01$
4	0	$3.55555555555555 \binom{60}{52} E \Leftrightarrow 01$	$3.56E \Leftrightarrow 01$	$5.33E \Leftrightarrow 01$	$5.33E \Leftrightarrow 01$
4	2	$1.0817208752322 \binom{45}{36} E \Leftrightarrow 01$	$1.08E \Leftrightarrow 01$	$8.89E \Leftrightarrow 01$	$1.67E \Leftrightarrow 01$

Tabelle 4.6: Vergleich 2-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 3$

		$p! q! \ K_{p,q}(u, v)\ _1$			
		Verifikation	STROUD		
p	q	VSK_4	PGL_4	VSK_5	VSK_9
1	1	$9.8404516228344 \binom{9}{4} E \Leftrightarrow 01$	$9.89E \Leftrightarrow 01$	$8.21E \Leftrightarrow 01$	$7.77E \Leftrightarrow 01$
1	2	$4.55401 \binom{5}{2} E \Leftrightarrow 01$	$4.56E \Leftrightarrow 01$	$7.16E \Leftrightarrow 01$	$4.38E \Leftrightarrow 01$
1	3	$2.3780 \binom{5}{3} E \Leftrightarrow 01$	$2.38E \Leftrightarrow 01$	$8.33E \Leftrightarrow 01$	$2.90E \Leftrightarrow 01$
2	2	$1.8748 \binom{70}{68} E \Leftrightarrow 01$	$1.87E \Leftrightarrow 01$	$8.89E \Leftrightarrow 01$	$2.06E \Leftrightarrow 01$

Bemerkung 4.3.4 Die in [45] erzielte vollsymmetrische Kubaturformel VSK_4 hat ungefähr die gleiche Güte wie die Produkt-Gauß-Legendre-Formel PGL_4 . Die anderen beiden vollsymmetrischen Kubaturformeln VSK_5 und VSK_9 besitzen nicht nur mehr Stützstellen, sondern auch ihre Fehlerkonstanten sind i. a. etwas größer.

Tabelle 4.7: Vergleich 1-dim. Fehlerkonstanten über $[\Leftarrow 1, 1] \times [\Leftarrow 1, 1]$ mit $\deg S = 7$

		$(r \Leftarrow j)! j! \ K_{r-j,j}(u)\ _1$			
		Verifikation	STROUD		
r	j	VSK_{12}	PGL_{16}	VSK_{12}	VSK_{13}
2	0	$5.139275584 \binom{30}{27} E \Leftarrow 02$	$8.74E \Leftarrow 02$	$5.14E \Leftarrow 02$	$6.81E \Leftarrow 02$
3	0	$1.5471197912 \binom{5}{3} E \Leftarrow 02$	$2.82E \Leftarrow 02$	$1.55E \Leftarrow 02$	$1.96E \Leftarrow 02$
4	0	$7.118329747 \binom{4}{2} E \Leftarrow 03$	$1.29E \Leftarrow 02$	$7.12E \Leftarrow 03$	$8.02E \Leftarrow 03$
4	2	$2.8082121871 \binom{2}{0} E \Leftarrow 02$	$2.91E \Leftarrow 02$	$2.81E \Leftarrow 02$	$2.24E \Leftarrow 02$
5	0	$4.603709875 \binom{2}{0} E \Leftarrow 03$	$8.36E \Leftarrow 03$	$4.60E \Leftarrow 03$	$4.60E \Leftarrow 03$
5	2	$1.2941266359 \binom{3}{1} E \Leftarrow 02$	$9.40E \Leftarrow 03$	$1.29E \Leftarrow 02$	$6.40E \Leftarrow 03$
6	0	$4.217046029 \binom{3}{1} E \Leftarrow 03$	$7.57E \Leftarrow 03$	$4.22E \Leftarrow 03$	$3.64E \Leftarrow 03$
6	2	$8.73810315 \binom{605}{598} E \Leftarrow 03$	$4.31E \Leftarrow 03$	$8.74E \Leftarrow 03$	$2.58E \Leftarrow 03$
6	4	$3.0297775384 \binom{6}{5} E \Leftarrow 02$	$1.75E \Leftarrow 02$	$3.03E \Leftarrow 02$	$3.01E \Leftarrow 02$
7	0	$5.6081664210 \binom{6}{0} E \Leftarrow 03$	$1.00E \Leftarrow 02$	$5.61E \Leftarrow 03$	$4.23E \Leftarrow 03$
7	2	$9.985085716 \binom{82}{76} E \Leftarrow 03$	$2.79E \Leftarrow 03$	$9.99E \Leftarrow 03$	$1.44E \Leftarrow 03$
7	4	$1.5472599932 \binom{9}{8} E \Leftarrow 02$	$5.64E \Leftarrow 03$	$1.55E \Leftarrow 02$	$1.93E \Leftarrow 02$
8	0	$1.318511144 \binom{30}{29} E \Leftarrow 02$	$2.32E \Leftarrow 02$	$1.32E \Leftarrow 02$	$8.53E \Leftarrow 03$
8	2	$2.0441347270 \binom{7}{5} E \Leftarrow 02$	$5.22E \Leftarrow 03$	$2.04E \Leftarrow 02$	$1.13E \Leftarrow 03$
8	4	$1.3278534171 \binom{1}{0} E \Leftarrow 02$	$2.59E \Leftarrow 03$	$1.33E \Leftarrow 02$	$3.10E \Leftarrow 02$
8	6	$2.8671533199 \binom{7}{5} E \Leftarrow 02$	$1.25E \Leftarrow 02$	$2.87E \Leftarrow 02$	$3.46E \Leftarrow 02$

Bemerkung 4.3.5 Die in [45] erzielte vollsymmetrische Kubaturformel VSK_{12} hat ungefähr die gleiche Güte wie die von Stroud [146] betrachtete vollsymmetrische Kubaturformel VSK_{12} . Die anderen beiden Kubaturformeln PGL_{16} und VSK_{13} besitzen mehr Stützstellen und ihre Fehlerkonstanten sind nicht unbedingt kleiner als die Fehlerkonstanten der beiden VSK_{12} (vgl. auch Tabelle 4.8).

Tabelle 4.8: Vergleich 2-dim. Fehlerkonstanten über $[\Leftrightarrow 1, 1] \times [\Leftrightarrow 1, 1]$ mit $\deg S = 7$

		$p!q! \ K_{p,q}(u, v)\ _1$			
		Verifikation	STROUD		
p	q	VSK_{12}	PGL_{16}	VSK_{12}	VSK_{13}
1	1	$3.6490\binom{7}{4}E \Leftrightarrow 01$	$4.11E \Leftrightarrow 01$	$3.64E \Leftrightarrow 01$	$3.67E \Leftrightarrow 01$
1	2	$1.7622\binom{1}{0}E \Leftrightarrow 01$	$1.91E \Leftrightarrow 01$	$1.75E \Leftrightarrow 01$	$1.79E \Leftrightarrow 01$
1	3	$1.5051\binom{71}{68}E \Leftrightarrow 01$	$1.36E \Leftrightarrow 01$	$1.49E \Leftrightarrow 01$	$1.38E \Leftrightarrow 01$
2	2	$4.254\binom{74}{69}E \Leftrightarrow 02$	$4.39E \Leftrightarrow 02$	$4.23E \Leftrightarrow 02$	$3.88E \Leftrightarrow 02$
1	4	$1.36796\binom{6}{4}E \Leftrightarrow 01$	$1.10E \Leftrightarrow 01$	$1.36E \Leftrightarrow 01$	$1.23E \Leftrightarrow 01$
2	3	$3.5406\binom{24}{19}E \Leftrightarrow 02$	$2.31E \Leftrightarrow 02$	$3.53E \Leftrightarrow 02$	$2.75E \Leftrightarrow 02$
1	5	$1.2399\binom{71}{69}E \Leftrightarrow 01$	$9.29E \Leftrightarrow 02$	$1.23E \Leftrightarrow 01$	$1.17E \Leftrightarrow 01$
2	4	$3.2732\binom{24}{18}E \Leftrightarrow 02$	$1.84E \Leftrightarrow 02$	$3.27E \Leftrightarrow 02$	$3.03E \Leftrightarrow 02$
3	3	$2.4855\binom{7}{6}E \Leftrightarrow 02$	$1.17E \Leftrightarrow 02$	$2.49E \Leftrightarrow 02$	$1.61E \Leftrightarrow 02$
1	6	$1.1366\binom{40}{38}E \Leftrightarrow 01$	$7.88E \Leftrightarrow 02$	$1.13E \Leftrightarrow 01$	$1.14E \Leftrightarrow 01$
2	5	$3.0457\binom{6}{4}E \Leftrightarrow 02$	$1.56E \Leftrightarrow 02$	$3.04E \Leftrightarrow 02$	$3.32E \Leftrightarrow 02$
3	4	$1.9666\binom{4}{3}E \Leftrightarrow 02$	$7.07E \Leftrightarrow 03$	$1.97E \Leftrightarrow 02$	$1.98E \Leftrightarrow 02$
1	7	$1.06433\binom{4}{2}E \Leftrightarrow 01$	$6.64E \Leftrightarrow 02$	$1.06E \Leftrightarrow 01$	$1.10E \Leftrightarrow 01$
2	6	$3.0639\binom{4}{2}E \Leftrightarrow 02$	$1.33E \Leftrightarrow 02$	$3.05E \Leftrightarrow 02$	$3.46E \Leftrightarrow 02$
3	5	$1.3219\binom{4}{2}E \Leftrightarrow 02$	$4.75E \Leftrightarrow 03$	$1.31E \Leftrightarrow 02$	$2.31E \Leftrightarrow 02$
4	4	$1.6206\binom{1}{0}E \Leftrightarrow 02$	$4.13E \Leftrightarrow 03$	$1.62E \Leftrightarrow 02$	$3.10E \Leftrightarrow 02$

Kapitel 5

Verifizierte automatische Integration

Die Aufgabe eines adaptiven numerischen Integrationsverfahrens, welches auch *automatischer Integrator* genannt wird, ist, für das bestimmte Integral

$$I := I(f) := \int_{\Omega} f(\vec{x}) d\vec{x}, \quad \Omega \subseteq \mathbb{R}^n,$$

eine Näherung I^* derart zu liefern, daß (vgl. de Boor [41], [42])

$$|I \Leftrightarrow I^*| \leq \max(\varepsilon_a, \varepsilon_r \cdot |I|) \quad (5.1)$$

gilt, wobei ε_a und ε_r die vom Benutzer vorgegebenen Obergrenzen der absoluten Genauigkeit bzw. der relativen Genauigkeit sind. Weiter wird vorausgesetzt, daß sich Ω durch Transformationen in einen Standardbereich, für welchen numerische Integrationsformeln vorhanden sind, überführen läßt.

Da das exakte Integral I im allgemeinen unbekannt ist, ist die Genauigkeitsanforderung (5.1) mit der reellen Arithmetik nicht realisierbar. Dieses Problem kann jedoch mit der Verifikationsnumerik zuverlässig gelöst werden. Mit Hilfe der verifiziert berechneten Approximationssumme $[S]$ und der Schranken des Diskretisierungsfehlers $[E]$ wird eine Einschließung des exakten Integrals I gewonnen :

$$[I] := [S] + [E].$$

Für $I^* \in [I]$ kann die Genauigkeitsanforderung (5.1) gesichert werden, wenn gilt

$$d([I]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle). \quad (5.2)$$

Denn aus (5.2) folgt für $I^* \in [I]$

$$|I \Leftrightarrow I^*| \leq d([I]) \stackrel{(5.2)}{\leq} \underbrace{\max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)}_{\leq |I|} \leq \max(\varepsilon_a, \varepsilon_r \cdot |I|).$$

D. h. unter der Voraussetzung (5.2) ist jedes $I^* \in [I]$ eine Näherung von I , welche (5.1) erfüllt.

Um die geforderte Genauigkeit effektiv erreichen zu können, wird der Integrationsbereich Ω abhängig vom Funktionsverhalten des Integranden f und von der erwünschten Genauigkeit dynamisch in verschiedene Teilbereiche Ω_j zerlegt :

$$\Omega = \bigcup_{j=1}^L \Omega_j \quad \text{mit} \quad \overset{\circ}{\Omega}_i \cap \overset{\circ}{\Omega}_j = \emptyset \quad \text{für} \quad i \neq j .$$

Im allgemeinen lassen sich diese Teilbereiche durch affine Abbildungen auf einen Standardbereich Ω_s zurückführen. Für jeden Teilbereich Ω_j wird dann jeweils eine Integrationsformel $S^{(j)}$ angewendet, die die affine Transformierte einer der vorhandenen Integrationsformeln S_1, \dots, S_T darstellt. Hierbei sind die Integrationsformeln S_1, \dots, S_T , bzgl. des Standardbereichs Ω_s konstruiert. Durch Summation der Approximationen $[S^{(j)}]$ bzw. der Approximationsfehler $[E^{(j)}]$ werden

$$[S] := \sum_{j=1}^L [S^{(j)}(f)] \quad \text{bzw.} \quad [E] := \sum_{j=1}^L [E^{(j)}(f)]$$

erhalten. Im konkreten Fall kann es auch vorkommen, daß der Integrationsbereich Ω in Teilbereiche zerlegt wird, welche durch affine Transformationen in unterschiedliche Standardbereiche überführt werden.

Für den Fall, daß die Integraleinschließung $[I] = [S] + [E]$ gröber als die geforderte Genauigkeit ist, d. h.

$$d([I]) > \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle) ,$$

muß die Zerlegung $\{\Omega_1, \dots, \Omega_L\}$ weiter verfeinert werden, bis sie der Bedingung (5.2) genügt bzw. bis festgestellt wird, daß die Genauigkeitsanforderung unerreichbar ist.

Die unerreichbare Genauigkeit rechtzeitig zu erkennen und darauffolgend die Berechnung zu unterbrechen, ist ein sehr wichtiges Kriterium für die Zuverlässigkeit. Dies ist bei der gewöhnlichen Numerik schwierig zu verwirklichen, da ohne Hilfe der maximal genauen Maschinenintervallarithmetik es kaum möglich ist, den Fehler richtig zu berücksichtigen. Wie die unerreichbaren Genauigkeiten erkannt werden können, wird im Abschnitt 5.3 behandelt.

Die Fragen, welche Integrationsformel für den Teilbereich Ω_j angewendet werden soll bzw. wie die Zerlegung $\{\Omega_1, \dots, \Omega_L\}$ weiter verfeinert werden kann, werden im nächsten Abschnitt diskutiert.

5.1 Doppelte adaptive Strategien

Die adaptive numerische Integration, die mittels der dynamischen Verfeinerung des Integrationsbereichs durchgeführt wird, wird seit Anfang der 60er Jahre untersucht und entwickelt. Die verwendeten Verfeinerungsmethoden unterteilen sich in zwei Klassen. Die eine wird als lokale und die andere als globale Strategie bezeichnet. Bei diesen Strategien werden die Teilbereiche, die noch zu unterteilen sind, nach verschiedenen Prinzipien

ausgewählt. Die Untersuchung der lokalen Verfeinerungsstrategie (vgl. [108]) liegt circa 10 Jahre vor der ersten Anwendung der globalen Methode ([39]). Nachdem Malcolm und Simpson [106] 1975 diese beiden Methoden anhand konkreter Beispiele verglichen haben, wird fast nur noch die globale Strategie betrachtet. Für die Effizienzüberlegung spielt bei Anwendung der globalen Strategie die Datenstruktur zur Registrierung der lokalen Informationen der Teilbereiche eine wichtige Rolle. Beispielsweise ist für das Suchen nach dem Element mit dem größten Fehler ein partiell geordneter balancierter Baum günstiger als eine linear geordnete Liste. Dies wird später besprochen.

Numerische Integrationsverfahren, die außer der adaptiven Bereichsverfeinerung noch verschiedene Exaktheitsgrade für die lokalen Integrationsformeln zulassen, heißen doppelt adaptiv. Diese Verfahren wurden am Anfang der 70er Jahre eingeführt (vgl. Oliver [118]). Wenn man heute von adaptiver numerischer Integration spricht, bezieht man sich fast immer auf die doppelte Adaption.

In der Verifikationsnumerik wurde der erste Beitrag zur eindimensionalen adaptiven numerischen Integration von Corliss und Rall [38] 1987 geleistet, in dem die globale Strategie und die adaptiven Exaktheitsgrade berücksichtigt wurden. Jedoch weicht die Implementierung der Methoden in [38] von der Implementierung im Softwareprogramm VANI ab, welches in Rahmen dieser Arbeit für die ein- und zweidimensionale automatische Integration entwickelt wurde. Diese Abweichungen werden in den Abschnitten 5.1.1 und 5.1.2 beschrieben. Weitere Literatur über die doppelt adaptive numerische Integration mit automatischer Ergebnisverifikation sind z. B. in [46], [83] und [144] zu finden, in denen das lokale Kriterium betrachtet wird. Konkret verwenden sie unterschiedliche Reihenfolgen, den Integrationsbereich zu unterteilen. In [143] wurde zweidimensionale numerische Integration mit automatischer Ergebnisverifikation anhand vom Extrapolationsverfahren durchgeführt, wobei adaptive Exaktheitsgrade berücksichtigt wurden.

Bemerkenswert ist, daß im Extremfall, wenn z. B. der Graph des Integranden eine sehr scharfe Spitze besitzt, das lokale Kriterium Schwierigkeit hat, Integraleinschließungen zu liefern. Es sei denn, zusätzliche Maßnahmen werden ergriffen, um eine unendliche Bereichsverfeinerung zu verhindern.

Im folgenden werden zuerst die adaptiven Verfeinerungsstrategien behandelt, und danach die Adaption der Exaktheitsgrade.

5.1.1 Adaptive Verfeinerungsstrategien

Die lokale Strategie für die Bereichsverfeinerung ist zwar gemäß den numerischen Untersuchungsergebnissen i. a. weniger effizient als die globale Strategie (insbesondere wenn bei der globalen Strategie eine günstige Datenstruktur für die Bereichsverwaltung verwendet ist), dennoch bietet sie eine praktische Richtlinie für die Auswahl der lokalen Integrationsformeln und deshalb wird sie an dieser Stelle beschrieben.

Die lokale Strategie

Bei dem lokalen Kriterium wird verlangt, daß für jeden Teilbereich Ω_j die nächste Bedingung

$$d([I^{(j)}]) \leq \varepsilon_j, \quad j = 1(1)L, \quad (5.3)$$

erfüllt werden muß. Hierbei ist $[I^{(j)}] := [S^{(j)}] + [R^{(j)}]$ und ε_j wird durch

$$\varepsilon_j := \frac{|\Omega_j|}{|\Omega|} \cdot \varepsilon, \quad \varepsilon := \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle), \quad (5.4)$$

bestimmt. Gilt (5.3) für einen Teilbereich nicht, dann muß er weiter unterteilt werden. Die Reihenfolge, welches Teilgebiet als das nächste behandelt wird, ist unwesentlich, wenn (5.2) erreicht werden kann. Üblicherweise wird ein Stapel (Stack) verwendet, um die Teilbereiche festzuhalten. In der Praxis wird jedoch eine rekursive Durchführung bevorzugt.

Die globale Strategie

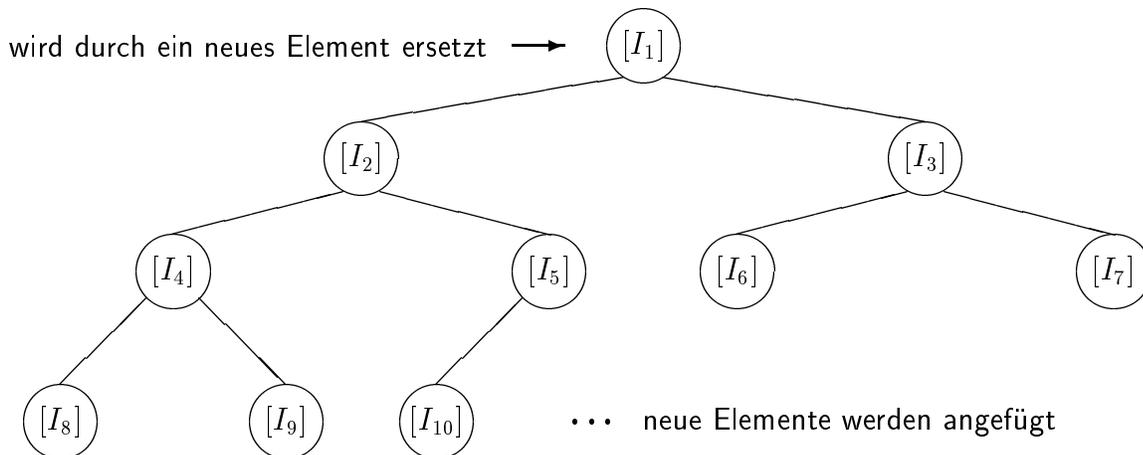
Im Gegensatz zu (5.3) wird bei der globalen Methode die Bedingung

$$\sum_{j=1}^L d([I^{(j)}]) \leq \varepsilon \quad (5.5)$$

geprüft. Gilt (5.5) mit der aktuellen Zerlegung nicht, dann wird das Teilgebiet Ω_m , das den maximalen Fehler besitzt

$$d([I^{(m)}]) \geq d([I^{(j)}]), \quad j = 1(1)L, \quad j \neq m,$$

unterteilt. Um auf dieses Gebiet schnell zugreifen zu können, ist es zweckmäßig, die Teilbereiche in einer Baumstruktur so festzuhalten, daß das in der Wurzel jedes Teilbaums platzierte Element einen größeren Fehler als alle seine unmittelbaren Nachkommen besitzt. Zwischen den benachbarten Elementen ist die Relation unwesentlich. Dies soll am Beispiel des balancierten binären Baumes erläutert werden.



Der Baum ist so partiell geordnet, daß für den Durchmesser gilt $d([I_1]) \geq \max(d([I_2]), d([I_3]))$, $d([I_2]) \geq \max(d([I_4]), d([I_5]))$ und $d([I_3]) \geq \max(d([I_6]), d([I_7]))$ usw.. Die Relation zwischen $d([I_2])$ und $d([I_3])$ bzw. zwischen den anderen benachbarten Elementen ist irrelevant.

Das Element, das an der Spitze des Baumes sitzt, hat also den größten Fehler. Für die weitere Verfeinerung wird dieses Element entfernt und unterteilt. Das neue Element mit dem größten Fehler wird in die Wurzel des gesamten Baumes gesetzt. Die anderen neuen Elemente werden dann an den Baum so angefügt, daß der Baum balanciert ist. D. h. die neuen Elemente werden immer von links nach rechts (bzw. in umgekehrter Richtung) innerhalb der tiefsten Schicht angefügt. Sie werden erst in eine tiefere Schicht plaziert, wenn die letzte Schicht voll ist (vgl. [86]). Nachdem ein neues Element eingefügt ist, wird es gleich an seinen richtigen Platz herunter- bzw. hochsortiert. Die Sortierung eines neuen Elementes ist fertig, sobald der aktuell betrachtete Teilbaum der partiellen Ordnung genügt. Für einen balancierten Baum beträgt der gesamte Sortierungsaufwand $O(L \log L)$ (vgl. [106]).

Bei manchen Anwendungen (z. B. [62], [19]) wird ein ternärer Baum eingesetzt. In [38] wurde hingegen eine linear geordnete Liste zum Festhalten der Teilbereiche verwendet, für die der gesamte Sortierungsaufwand $O(L^2)$ ist.

Die Tatsache, daß die lokale Strategie strenger als die globale Strategie ist, kann auch analytisch erklärt werden. Die Ungleichung (5.5) ist eine notwendige Bedingung für (5.3). Im Extremfall kann das lokale Kriterium (5.3) über dem problematischen Teilbereich u. U. nie erfüllt werden. Der unendliche Programmlauf läßt sich z. B. dadurch abbrechen, daß eine Oberschranke für die Anzahl von Funktionsauswertungen bzw. eine Unterschranke für die Durchmesser der Teilbereiche gesetzt werden. Die Integraleinschließungen, die in dieser Situation erzielt werden, können u. U. sehr grob sein bzw. sehr lange Rechenzeit benötigen. Hingegen müssen bei der globalen Verfeinerungsstrategie sämtliche Teilbereiche abgespeichert werden. Der Speicherbedarf, der von dem Integranden, der geforderten Genauigkeit und der Dimension abhängig ist, könnte insbesondere im mehrdimensionalen beträchtlich sein. Zum Vergleich der beiden Verfeinerungsstrategien seien die Bemerkungen von Malcolm und Simpson ([106], S. 145, 146) zu zitieren :

”The advantage of global strategies over local strategies were seen to be in some sense a space-time tradeoff. The reduced numbers of required function evaluations and the apparent increase in domain of applicability using the global strategy comes at the expense of additional memory requirements. However, the memory requirements of ... are not unreasonable for many applications, and when a higher order local quadrature rule is used, such as ..., the additional memory requirements are very small.”

” It demonstrates that interval acceptance criteria need not be totally local as ... or totally global as ... and doubtless a variety of such intermediate stages are possible.”

5.1.2 Adaptive Exaktheitsgrade

Um (5.2) mit möglichst geringem Aufwand zu erreichen, soll vermieden werden, daß das adaptive numerische Integrationsverfahren eine wesentlich bessere Integraleinschließung $[I]$ Genauigkeit liefert, d. h. $d([I]) \ll \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$. Es ist daher wichtig, die Integrationsformeln mit den am besten geeigneten Exaktheitsgraden zu verwenden. Dadurch lassen sich überflüssige Funktionsauswertungen vermeiden.

Zur Auswahl der lokalen Integrationsformeln stellt die Bedingung (5.3) eine geeignete Richtlinie dar, obwohl für die Erfüllung des globalen Kriteriums (5.5) die Bedingung (5.3) nicht gelten muß. Da, bevor entschieden wird, welche Integrationsformel für den Teilbereich Ω_j anzuwenden ist, keine Approximationssumme $S^{(j)}$ zur Verfügung steht, wird für die Auswahl der lokalen Regel anstatt $[I^{(j)}]$ die Fehlereinschließung $[E^{(j)}]$ betrachtet.

Ist für jede Integrationsformel S_i , $i = 1(1)T$, der lokale Diskretisierungsfehler größer als die Genauigkeit ε_j , d. h.

$$d([E_i^{(j)}]) > \varepsilon_j, \quad i = 1(1)T,$$

so wird diejenige Integrationsformel, deren Diskretisierungsfehler und Knotenzahl am kleinsten ist, zur Approximation des Teilintegrals ausgewählt. Im Falle, daß Integrationsformeln S_i existieren, für welche

$$d([E_i^{(j)}]) \leq \varepsilon_j$$

gilt, wird aus diesen Integrationsformeln diejenige, die die geringste Knotenzahl hat, für die Berechnung der lokalen Approximation ausgewählt.

if $d([E_i^{(j)}]) > \varepsilon_j, \quad \forall S_i \in \{S_1, \dots, S_T\}$ **then**
wähle $S^{(j)}$ so, daß $d([E^{(j)}]) = \min_{1 \leq i \leq T} \{d([E_i^{(j)}])\}$ (★)
else if $\exists S_{\lambda_j} \in \{S_1, \dots, S_T\}$ mit $d([E_{\lambda_j}^{(j)}]) \leq \varepsilon_j$ **then**
wähle $S^{(j)} \in \{S_{\lambda_j}\}$ so, daß $S^{(j)}$ minimale Knotenzahl hat.

Erfüllen mehrere Integrationsformeln die Gleichung in (★), dann soll diejenige mit minimaler Knotenzahl ausgewählt.

Gemäß der Genauigkeitsanforderung (5.1) bzw. (5.2) kann ε_j für die Auswahl lokaler Integrationsformeln zwar eine strenge Fehlerschranke sein, jedoch wird dadurch die Anzahl der Funktionsauswertungen $f([\vec{x}_i])$ wesentlich stärker reduziert als bei dem Ansatz, bei dem die lokalen Integrationsformeln direkt gemäß (5.2) ausgewählt sind.

Dieses Prinzip wurde bereits in [46], [83] und [144] berücksichtigt. Hingegen wurde in [38] das strengere Kriterium (★) angewendet. Diese Anwendung benötigt viele Funktionsauswertungen und hat damit zur Folge, daß die dadurch erzielten Integraleinschließungen viel feiner als erwünscht sein werden.

5.2 Unterteilung eines Integrationsbereichs

Bei Anwendung einer Verfeinerungsstrategie wird jedes Teilgebiet Ω_j in q Teilbereiche zerlegt, welche i. a. von gleichem Volumen sind und sich durch affine Abbildungen in einen (oder mehreren) Standardbereich(e) überführen lassen.

Üblicherweise wird $q = 2$ bzw. 2^n gesetzt. Im Eindimensionalen sind die binäre und die 2^n -Unterteilung identisch. Hingegen sind die Zerlegungsmöglichkeit von mehrdimensionalen Gebieten vielfältig. Beispielsweise gibt es jeweils 3, 2 und 0 Möglichkeiten, ein Dreieck in 2 Teildreiecke, ein Rechteck in 2 Teilrechtecke bzw. einen Kreis in Teilkreise, wobei die Teilbereiche vom gleichen Volumen sind, zu zerlegen. Weiter ist die 2^n -Zerlegung für Rechtecke eindeutig. Dennoch gibt es mehrere Möglichkeiten, ein Dreieck in 2^2 Teildreiecke zu unterteilen. Da Kreise und Dreiecke sich durch nichtaffine Transformationen in Rechteckbereiche überführen lassen, wird die folgende Betrachtung auf das Rechteckgebiet beschränkt, welches auch das am besten untersuchte Gebiet ist.

5.2.1 Zerlegung in 2^n Teilbereiche

Die 2^n -Unterteilung eines n -dimensionalen Quaders läßt sich durch rekursive Aufrufe der binären Unterteilung in jeder Richtung realisieren.

Algorithmus 5.1 : AllBisection(*Quader*, *QListe*)

1. $dir := 1; \quad no := 1.$
2. Bisections(*Quader*, *QListe*, *dir*, *no*).

Algorithmus 5.2 : Bisections(*Quader*, *QListe*, *dir*, *no*)

1. Bisection(*Quader*, *TeilQuader*, *dir*)
2. **if** $dir = n$ **then**
 - $QListe[no] := TeilQuader[1];$
 - $QListe[no + 1] := TeilQuader[2];$
 - $no := no + 2$
- else** Bisections(*TeilQuader*[1], *QListe*, $dir + 1$, *no*);
- Bisections(*TeilQuader*[2], *QListe*, $dir + 1$, *no*).

Die tatsächliche binäre Unterteilung des Quaders in der *dir* Richtung wird von der Prozedur Bisection(*Quader*, *TeilQuader*, *dir*) gemacht.

5.2.2 Binäre Unterteilung

Für die binäre Unterteilung eines Rechtecks wird die Unterteilungsrichtung mit Hilfe der Abschätzung des Approximationsfehlers der angewendeten Integrationsformel bestimmt.

Bei den Produktformeln benötigt die Nikol'skiische Fehlerdarstellung am wenigsten Rechenaufwand. Und für die Nicht-Produktformeln ist die Sard-Darstellung adäquat. Im allgemeinen lassen sich diese Fehlerbetrachtungen als die Kernmethode bezeichnen. Bei der Kernmethode wird der Diskretisierungsfehler einer Integrationsformel anhand der Taylorkoeffizienten und der Fehlerkonstanten abgeschätzt. Für große Integrationsbereiche sind die unmittelbaren Wertebereichseinschließungen der (partiellen) Ableitungen der zu integrierenden Funktion, die mit Hilfe der Differentiations- und Intervallarithmetik berechnet werden, wegen des Aufblähungseffekts der Intervallrechnung oft sehr grob. Diese Überschätzung kann durch Verfeinerung des Integrationsbereichs reduziert werden. Die adaptive Verfeinerungsstrategie ist also nicht nur für gute Integralnäherungen in vielen Fällen wichtig, sondern auch für eine angemessene Fehlereinschließung unentbehrlich.

Bei Anwendung von Produktformeln

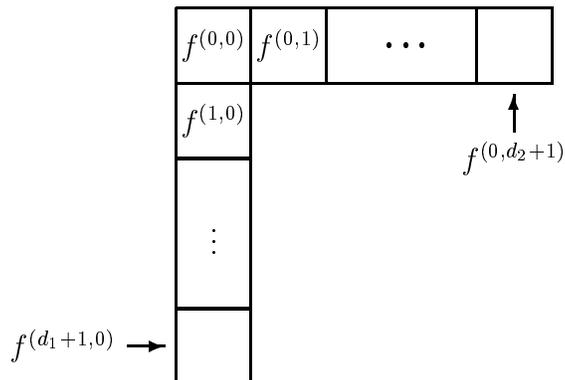
Gemäß (4.17) läßt sich der Approximationsfehler einer Produktregel als Summe von den Fehlern in der x - und y -Richtung darstellen :

$$[E^{(j)}] = [E_{\Omega_x}] + [E_{\Omega_y}] .$$

Die Unterteilungsrichtung wird anhand der Durchmesser der Einschließungen $[E_{\Omega_x}]$ und $[E_{\Omega_y}]$ entschieden :

$$\mathbf{if} \ d([E_{\Omega_x}]) \geq d([E_{\Omega_y}]) \ \mathbf{then} \ dir := 1 \ \mathbf{else} \ dir := 2$$

Bei Anwendung von Produktformeln gibt es insgesamt $d_1 + d_2 + 3$ Taylorkoeffizienten bzgl. Ω_j zu berechnen.



Bei Anwendung von Nicht-Produktformeln

Nach der Einschließungsformel (4.24) ist der Diskretisierungsfehler $[E]$ einer Nicht-Produktformel gleich der Summe der Fehler in x -, y -Richtung und auf Ω :

$$[E^{(j)}] = [E_{\Omega_x}] + [E_{\Omega_y}] + [E_{\Omega}] .$$

In welcher Richtung ein Teilgebiet weiter unterteilt wird, beschreibt das nächste Schema:

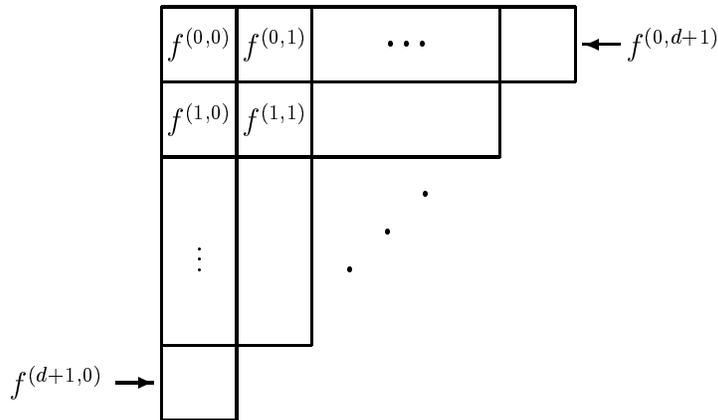
if $d([E_{\Omega_x}]) \geq \max(d([E_{\Omega_y}], d([E_{\Omega}]))$ **then** $dir := 1$

else if $d([E_{\Omega_y}]) \geq \max(d([E_{\Omega_x}], d([E_{\Omega}]))$ **then** $dir := 2$

else if $d(f^{(p,0)}(\Omega_j)) \geq d(f^{(0,q)}(\Omega_j))$ **then** $dir := 1$ **else** $dir := 2$

Die letzte Fallunterscheidung liegt daran, daß für $[E_{\Omega}]$ die Ableitung $f^{(p,q)}$ zu berücksichtigen ist.

Für Nicht-Produktformeln gibt es insgesamt $O(d^2)$ Taylorkoeffizienten bzgl. Ω_j zu berechnen. Der Rechenaufwand für einen lokalen Approximationsfehler ist bei Anwendung der Nicht-Produktformeln deshalb wesentlich größer als bei Anwendung der Produktformeln von gleichen Exaktheitsgraden.



5.3 Zuverlässige und effiziente Fehlerbehandlung

Bei der Intervallrechnung gilt für $[S] + [E] = [I]$

$$d([I]) = d([S]) + d([E]) .$$

Sei $S := \sum [w_i] \cdot F_{\diamond}([\vec{x}_i]) = [m] \cdot b^e$ mit $|m| := 0.m_1m_2 \dots$ und $m_1 \neq 0$, dann besitzt der Durchmesser von $[S]$ eine Unterschranke, $d([S]) \geq b^{e-l}$, die nicht unterschritten werden kann. Hingegen läßt sich die Fehlereinschließung $[E]$ durch feinere Zerlegung des Integrationsbereichs und darauffolgende Anwendungen der lokalen Integrationsformeln, im Rahmen des Gleitkommasystems, quasi beliebig verfeinern.

Wegen der Unterschranke von $d([S])$ kann es vorkommen, daß mit beliebiger Zerlegung $\{\Omega_1, \dots, \Omega_L\}$ die Ungleichung $d([I]) > \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ gilt. Ein zuverlässiger automatischer Integrator soll die unerreichbare Genauigkeitsanforderung rechtzeitig erkennen können und in diesem Fall die Berechnung abbrechen.

Außer der Zuverlässigkeit ist die Effizienz ebenfalls ein wichtiges Kriterium für die automatische Integration. Es ist daher wichtig, das Genauigkeitskriterium (5.2) sowie das Abfangen der unerreichbaren Genauigkeit mit möglichst kleinem Aufwand zu behandeln.

In der folgenden Diskussion wird vorausgesetzt, daß für die betrachteten Integrationsformeln die Einschließungen $[w_i]$ und $[\vec{x}_i]$ hinreichend fein berechnet sind.

5.3.1 Reine absolute Fehlerbehandlung

Für den Rechenaufwand erweist sich die Anzahl der Funktionsaufrufe als eine wichtige Metrik. Um für eine reine absolute Genauigkeitsanforderung, d. h. $\varepsilon_r = 0$, die Anzahl der Funktionsaufrufe gering zu halten, wird bei der absoluten Fehlerbehandlung die Näherung $S(f)$ erst dann berechnet, wenn gilt

$$d([E(f)]) \leq \varepsilon_a . \quad (5.6)$$

In [83], [144] und [88] wurde für die absolute Genauigkeitsprüfung nur die Bedingung (5.6) berücksichtigt. Diese Fehlerbehandlung ist jedoch nicht vollständig, insbesondere wenn die globale Verfeinerungsstrategie eingesetzt wird. Denn es kann durchaus passieren, daß unter der Voraussetzung (5.6) die Einschließung $[S] + [E]$ immer noch zu grob ist.

Gilt unter der Voraussetzung (5.6)

$$d([S(f)]) + d([E(f)]) \leq \varepsilon_a ,$$

so ist die erwünschte absolute Genauigkeit erreicht.

Ist umgekehrt

$$d([S(f)]) + d([E(f)]) > \varepsilon_a ,$$

dann soll geprüft werden, ob die Einschließung $[S(f)]$ zu grob ist. Im Fall

$$d([S]) \geq \varepsilon_a \quad (5.7)$$

kann die geforderte absolute Genauigkeit mit den Gegebenheiten $[\vec{x}_i]$, $[w_i]$ und $F_\diamond([\vec{x}_i])$ nicht erreicht werden. Denn unter der Bedingung (5.7) läßt sich $d([S])$ nicht wesentlich verbessern (vgl. auch [38]). Der automatische Integrator soll diesen Tatbestand berichten und einen Abbruch veranlassen. Andernfalls, d. h. wenn

$$d([S]) < \varepsilon_a ,$$

soll das adaptive numerische Integrationsverfahren weiter durchgeführt werden, bis eine der folgenden Situationen zustande kommt :

- $d([S]) + d([E]) \leq \varepsilon_a$.
- Die maximal zulässige Anzahl der Funktionsaufrufe ist überschritten.
- Die maximal zulässige Anzahl der Teilbereiche ist überschritten.

5.3.2 Absolute und relative Fehlerbehandlung

Ist $\varepsilon_r \neq 0$, dann wird für die Fehlerbehandlung stets die Bedingung (5.2) geprüft. Es ist jedoch schwierig, für die relative Genauigkeit ein Kriterium zu finden, so daß die automatische Integration in einer ersten Phase ohne Berechnung von $[S]$ durchgeführt werden kann. Dieser Sachverhalt läßt sich wie folgt darlegen.

Für $[I] = [S] + [E]$ besitzt $\langle [I] \rangle$ eine Oberschranke

$$\begin{aligned} \langle [S] + [E] \rangle &= \min\{ |s + r| \mid s \in [S], r \in [R] \} \\ &\leq \min\{ |s| + |r| \mid s \in [S], r \in [R] \} \\ &= \langle [S] \rangle + \langle [E] \rangle . \end{aligned}$$

Ist $d([S]) + d([E]) \leq \varepsilon_r \cdot \langle [I] \rangle$, dann muß gelten

$$d([S]) + d([E]) \leq \varepsilon_r \cdot (\langle [S] \rangle + \langle [E] \rangle) .$$

Sei $[S] = [m] \cdot b^e$ mit $|m| = 0.m_1m_2 \dots$ und $m_1 \neq 0$. Da für $\varepsilon_r > b^{1-l}$ und $0 \notin [S]$ ($\Leftrightarrow 0 \notin [m]$) häufig gilt

$$d([S]) = d([m]) \cdot b^e \leq \varepsilon_r \cdot \langle [m] \rangle \cdot b^e = \varepsilon_r \cdot \langle [S] \rangle \quad (\text{sogar } d([S]) \ll \varepsilon_r \cdot \langle [S] \rangle) ,$$

ist erforderlich, daß für $[E]$ zumindestens die Bedingung

$$d([E]) \leq \varepsilon_r \cdot \langle [S] \rangle \quad \text{für } \langle [E] \rangle = 0 \quad (5.8)$$

$$\text{bzw.} \quad d([E]) \leq \varepsilon_r \cdot \langle [E] \rangle \quad \text{für } \langle [E] \rangle \neq 0 \quad (5.9)$$

erfüllt wird, damit die relative Genauigkeitsanforderung möglicherweise erreicht werden kann. Bei dem häufig auftretenden Fall $\langle [E] \rangle = 0$ bedeutet dies, daß die Näherung S stets berechnet werden muß. Die Situation $\langle [E] \rangle \neq 0$ wird wegen der Anwendung von adaptiven Exaktheitsgraden in der Praxis selten vorkommen. Wird in (5.8) $d([E])$ mit einer beliebigen Abschätzung $[\tilde{S}]$ geprüft, dann hat dies in vielen Fällen, z. B. für oszillierende Funktionen bzw. für Funktionen mit Peak, $d([E]) > \varepsilon_r \cdot \langle [S] \rangle$ zur Folge. Aus diesem Grund und da (5.8) und (5.9) notwendige aber keine hinreichenden Bedingungen für eine erreichbare relative Genauigkeit darstellen, soll bei einer zuverlässigen Fehlerbehandlung stets die Bedingung

$$d([S]) + d([E]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle) \quad (5.10)$$

mit den wahren Werten $[E]$, $[S]$ und $[I]$ geprüft werden.

Die unerreichbare Genauigkeit ist nach dem im Abschnitt 5.3.1 beschriebenen Grund durch

$$d([S]) > \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle) \quad \text{und} \quad d([E]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle) \quad (5.11)$$

zu erkennen. Im Spezialfall $I \doteq 0$ wird das Verfahren wegen $0 \in [I]$, und folglich $0 = \langle [I] \rangle$, in einen unendlichen Lauf eintreten. In diesem Fall wird die Berechnung abgebrochen, wenn die maximal zulässige Anzahl von Funktionsauswertungen $f(x_i)$ bzw. die maximal zulässige Anzahl von Teilbereichen Ω_j überschritten wird.

5.4 Verifikation über dem Bereichsrand

Die bisherige Diskussion setzt voraus, daß die Grenzen des Integrationsbereichs in Gleitkommazahlen exakt dargestellt werden können. Für Integrationsbereiche, deren Grenzen keine Gleitkommazahlen sind, wie z. B. 0.1 bzgl. eines binären Zahlenformates, ist der Konvertierungsfehler zu berücksichtigen. Mit Hilfe der Maschinenintervallarithmetik kann man solche reellen Zahlen in das engst mögliche Gleitkommaintervall einschließen. Beispiele für die richtige Handhabung des Konvertierungsfehlers sind in [61] angegeben. Über diesen Randbereichen soll die Integration anders behandelt werden.

5.4.1 Eindimensionaler Bereich

Im Eindimensionalen ist die Verifikation über den Intervallgrenzen direkt. Nach der Definition (vgl. [38])

$$\int_{[a]}^{[b]} f(x) dx := \left\{ \int_s^t f(x) dx \mid s \in [a], t \in [b] \right\}$$

läßt sich das Integral $\int_{[a]}^{[b]} f(x) dx$ durch die folgende Summe darstellen :

$$\int_{[a]}^{[b]} f(x) dx = \int_{[a]}^{\bar{a}} f(x) dx + \int_{\bar{a}}^{\underline{b}} f(x) dx + \int_{\underline{b}}^{[b]} f(x) dx .$$

Für enge Intervalle $[a]$, $[b]$ kann man das erste und dritte Integral mit Hilfe der Riemannschen Summe einschließen. Es gilt

$$\begin{aligned} \int_{[a]}^{\bar{a}} f(x) dx &:= \left\{ \int_s^{\bar{a}} f(x) dx \mid s \in [a] \right\} \\ &\left(= \left\{ f(\xi_s) \cdot (\bar{a} \Leftrightarrow s) \mid s \in [a], \xi_s \in [s, \bar{a}] \right\}, \quad f \in C[s, \bar{a}] \right) \\ &\subseteq \left\{ f([s, \bar{a}]) \cdot (\bar{a} \Leftrightarrow s) \mid s \in [a] \right\} \\ &\subseteq f([a]) \cdot [0, \bar{a} \Leftrightarrow \underline{a}] , \end{aligned}$$

und analog

$$\int_{\underline{b}}^{[b]} f(x) dx \subseteq f([b]) \cdot [0, \bar{b} \Leftrightarrow \underline{b}] .$$

5.4.2 Mehrdimensionaler Bereich

Im folgenden wird vorausgesetzt, daß $b \Leftrightarrow a$, $d \Leftrightarrow c \geq b^{1-l}$ ist. Die Verifikation über mehrdimensionalen Randbereichen unterscheidet mehrere Fälle. Für ein zweidimensionales Rechteckgebiet gibt es z. B. 2 Klassen von Randbereichen zu behandeln.

$$\begin{aligned}
& \int_{[a]}^{[b]} \int_{[c]}^{[d]} f(x, y) dy dx & [d] & \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline & & \\ \hline 1 & \int_{\underline{a}}^{\underline{b}} \int_{\underline{c}}^{\underline{d}} f(x, y) dy dx & 1 \\ \hline & & \\ \hline 0 & 1 & 0 \\ \hline \end{array} \\
& = \int_{[a]}^{\bar{a}} \int_{[c]}^{\bar{c}} f dy dx + \int_{[a]}^{\bar{a}} \int_{\underline{c}}^{\underline{d}} f dy dx + \int_{[a]}^{\bar{a}} \int_{\underline{d}}^{[d]} f dy dx \\
& + \int_{\underline{a}}^{\underline{b}} \int_{[c]}^{\bar{c}} f dy dx + \left[\int_{\underline{a}}^{\underline{b}} \int_{\underline{c}}^{\underline{d}} f dy dx \right] + \int_{\underline{a}}^{\underline{b}} \int_{\underline{d}}^{[d]} f dy dx \\
& + \int_{\underline{b}}^{[b]} \int_{[c]}^{\bar{c}} f dy dx + \int_{\underline{b}}^{[b]} \int_{\underline{c}}^{\underline{d}} f dy dx + \int_{\underline{b}}^{[b]} \int_{\underline{d}}^{[d]} f dy dx & [c] & \\
& & [a] & [b]
\end{aligned}$$

Für die Randbereiche vom Typ 0 (siehe Figur) ist die Behandlung ähnlich wie im Eindimensionalen. Beispielsweise gilt für $[[a], \bar{a}] \times [[c], \bar{c}]$

$$\int_{[a]}^{\bar{a}} \int_{[c]}^{\bar{c}} f(x, y) dy dx \subseteq f([a], [c]) \cdot [0, \bar{a} \Leftrightarrow \underline{a}] \cdot [0, \bar{c} \Leftrightarrow \underline{c}] . \quad (5.12)$$

Die Integraleinschließungen über den Randbereichen vom Typ 1 führt zu eindimensionaler Integration. Zum Beispiel lassen sich die Integrale über $[[a], \bar{a}] \times [\bar{c}, \underline{d}]$ bzw. $[\bar{a}, \underline{b}] \times [[c], \bar{c}]$ wie folgt einschließen :

$$\begin{aligned}
\int_{[a]}^{\bar{a}} \int_{\bar{c}}^{\underline{d}} f(x, y) dy dx & \subseteq [0, \bar{a} \Leftrightarrow \underline{a}] \cdot \underbrace{\int_{\bar{c}}^{\underline{d}} f([a], y) dy}_{=: I_y} , \\
\int_{\bar{a}}^{\underline{b}} \int_{[c]}^{\bar{c}} f(x, y) dy dx & \subseteq [0, \bar{c} \Leftrightarrow \underline{c}] \cdot \underbrace{\int_{\bar{a}}^{\underline{b}} f(x, [c]) dx}_{=: I_x} .
\end{aligned}$$

Sei $I_{inn} := \int_{\bar{a}}^{\underline{b}} \int_{\bar{c}}^{\underline{d}} f(x, y) dx dy$. Ferner seien $[I_i^{(0)}]$ und $[I_j^{(1)}]$ die Teilintegraleinschließungen über dem Randbereich vom Typ 0 bzw. Typ 1. Wegen $[I] = [I_{inn}] + \sum_{i=1}^{n_0} [I_i^{(0)}] + \sum_{j=1}^{n_1} [I_j^{(1)}]$ folgt aus der Genauigkeitsanforderung (5.2)

$$\begin{aligned}
d([I]) & = d([I_{inn}]) + \sum_{i=1}^{n_0} d([I_i^{(0)}]) + \sum_{j=1}^{n_1} d([I_j^{(1)}]) \\
& \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle) \\
& \leq \max \left(\varepsilon_a, \varepsilon_r \cdot \left\langle \langle [I_{inn}] \rangle + \sum_{i=1}^{n_0} \langle [I_i^{(0)}] \rangle + \sum_{j=1}^{n_1} \langle [I_j^{(1)}] \rangle \right\rangle \right) ,
\end{aligned}$$

wobei n_0 die Anzahl der Randbereiche vom Typ 0 und n_1 die Anzahl der Randbereiche vom Typ 1 ist.

Da $[I_{inn}]$, $[I_i^{(0)}]$ und $[I_j^{(1)}]$ getrennt berechnet werden müssen, ist es unmöglich die Genauigkeitsanforderung $\max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ direkt mit $[I]$ zu prüfen. Um für die gesamte Integraleinschließung $[\int_{[a]}^{[b]} \int_{[c]}^{[d]} f(x, y) dx dy] =: [I]$ die Genauigkeitsanforderung (5.2) möglichst

gut erfüllen zu können, wird I_{inn} gemäß

$$d([I_{inn}]) \leq \max(\varepsilon_a \cdot \frac{A}{A}, \varepsilon_r \cdot \langle [I_{inn}] \rangle) \quad (5.13)$$

mit $[A] := ([b] \Leftrightarrow [a]) \cdot ([d] \Leftrightarrow [c])$ bestimmt.

Für das Integral $[I_j^{(1)}]$, ohne Beschränkung der Allgemeinheit sei $[I_j^{(1)}] := [0, \bar{a} \Leftrightarrow \underline{a}] \cdot [I_y]$, gilt

$$d([I_j^{(1)}]) = \begin{cases} (\bar{a} \Leftrightarrow \underline{a}) \cdot |[I_y]|, & 0 \notin [I_y] \\ (\bar{a} \Leftrightarrow \underline{a}) \cdot d([I_y]), & 0 \in [I_y] \end{cases}$$

und

$$\langle [I_j^{(1)}] \rangle = 0 \leq \langle [I_y] \rangle \cdot (\bar{a} \Leftrightarrow \underline{a}) \leq |[I_y]| \cdot (\bar{a} \Leftrightarrow \underline{a}).$$

Wegen $\langle [I_j^{(1)}] \rangle = 0$ wird die Bedingung

$$d([I_j^{(1)}]) \leq \max(c_j \cdot \varepsilon_a, \varepsilon_r \cdot \langle [I_j^{(1)}] \rangle)$$

mit $c_j := \frac{(\bar{a} \Leftrightarrow \underline{a})}{(\bar{b} \Leftrightarrow \underline{a})} \cdot \frac{(\underline{d} \Leftrightarrow \bar{c})}{(\bar{d} \Leftrightarrow \underline{c})} \approx \frac{(\bar{a} \Leftrightarrow \underline{a})}{(\bar{b} \Leftrightarrow \underline{a})}$ für $\varepsilon_a = 0$ nie erfüllt. Im Fall $0 \notin [I_y]$ läßt sich $[I_y]$ im Prinzip gemäß

$$|[I_y]| \leq \max\left(\frac{1}{(\bar{b} \Leftrightarrow \underline{a})} \cdot \varepsilon_a, \varepsilon_r \cdot \langle [I_y] \rangle\right)$$

bestimmen. Da aber die beiden Ungleichungen $|[I_y]| \leq \varepsilon_r \cdot \langle [I_y] \rangle$ und $|[I_y]| \leq \varepsilon_r \cdot |[I_y]|$ nicht erfüllbar sind, und da für ein festes I_y die Bedingung $|[I_y]| \leq \frac{1}{(\bar{b} - \underline{a})} \cdot \varepsilon_a$ u. U. nie gelten kann, wird deshalb $[I_y]$ gemäß

$$d([I_y]) \leq \max\left(\frac{1}{(\bar{b} \Leftrightarrow \underline{a})} \cdot \varepsilon_a, \varepsilon_r \cdot |[I_y]|\right) \quad (5.14)$$

berechnet, damit das Softwareprogramm für die Verifikation über den Randbereichen nicht in einen unendlichen Lauf eintritt, und alle Integraleinschließungen $[I_j^{(1)}]$ berechnet werden können, wenn $\varepsilon_a \neq 0$ oder $\varepsilon_r \neq 0$ ist und die maximal zulässige Anzahl von Funktionsauswertungen groß genug ist.

Die Integraleinschließung $[I_x]$ wird entsprechend gemäß

$$d([I_x]) \leq \max\left(\frac{1}{(\bar{d} \Leftrightarrow \underline{c})} \cdot \varepsilon_a, \varepsilon_r \cdot |[I_x]|\right) \quad (5.15)$$

bestimmt.

Im konkreten werden zuerst $[I_i^{(0)}]$, $[I_j^{(1)}]$ und danach $[I_{inn}]$ berechnet, da die Bedingung (5.13) im Fall $\langle [I_{inn}] \rangle = 0$ und $\varepsilon_r \neq 0$ einen unendlichen Programmlauf veranlassen kann, welcher dann mit Hilfe der maximalen Anzahl von Funktionsauswertungen bzw. Teilbereichen abgebrochen wird.

5.5 Das Programm VANI

Das Softwareprogramm VANI (Validated Adaptive Numerical Integration) wurde für die ein- und zweidimensionale adaptive numerische Integration mit automatischer Ergebnisverifikation entwickelt. Für die zweidimensionale Integration werden zur Zeit in VANI nur Rechteckgebiete berücksichtigt. Integrale über Kreise und Dreiecke können in Integrale über Rechtecke transformiert werden (s. Kapitel 4). Diese Transformationen werden zur Zeit dem Benutzer überlassen.

5.5.1 Benutzerschnittstellen

Zur Durchführung der automatischen Integration muß der Benutzer zuerst in dem Modul **User_Func** die zu integrierenden Funktionen definieren. Der Benutzer kann gleichzeitig mehrere Integranden angeben. Durch Überladen von Funktionen unter PASCAL-XSC werden die ein- und zweidimensionalen Funktionen gleich benannt. Die Variable **x** einer eindimensionalen Funktion ist vom Typ `itaylor`, und die Variablen **x** und **y** einer zweidimensionalen Funktion sind vom Typ `ditaylor2`.

Beispielsweise lassen sich folgende ein- und zweidimensionale Funktionen

$$f_1(x) = \frac{100}{(1 + (10 \cdot x)^2)}$$

$$f_2(x) = 2x \cdot e^{x^2} \cdot \sin(e^{x^2})$$

$$f_3(x) = \frac{1}{a^2 + (3x \Leftrightarrow 1)^2} \Leftrightarrow \frac{1}{a^2 + (3x \Leftrightarrow 4)^2} + \frac{1}{a^2 + (3x \Leftrightarrow 7)^2} \Leftrightarrow \frac{1}{a^2 + (3x \Leftrightarrow 10)^2}, \quad a = 0.1$$

$$f_1(x, y) = \frac{1}{(6 \Leftrightarrow 2x \Leftrightarrow y)^2}$$

$$f_2(x, y) = \cos(2\pi\beta + \alpha_1 x + \alpha_2 y), \quad \beta = 0.25, \alpha_1 = 75, \alpha_2 = 25$$

$$f_3(x, y) = \sum_{k=1}^{10} (2 + 2k \Leftrightarrow e^{kx} \Leftrightarrow e^{ky})^2$$

durch die nächsten zwei Programmabschnitte definieren.

```
GLOBAL FUNCTION f( x : itaylor; no : INTEGER ) : itaylor[lb(x)..ub(x)];
VAR ex : itaylor[lb(x)..ub(x)];
    a : INTERVAL;
BEGIN
  CASE no OF
    1 : f := 100 / ( 1 + sqr(10*x) );
    2 : BEGIN
          ex := exp(sqr(x));  f := 2*x*ex*sin(ex);
        END;
```

```

    3 : BEGIN
      a := sqr(intval(1)/10);
      f := ( 1/(a+sqr(3*x-1)) - 1/(a+sqr(3*x-4))
            + 1/(a+sqr(3*x-7)) - 1/(a+sqr(3*x-10)));
    END;
  END;
END;

GLOBAL FUNCTION f( x,y : ditaylor2; no : INTEGER )
                  : ditaylor2[lb(x,1)..ub(x,1), lb(x,2)..ub(x,2)];
VAR k      : INTEGER;
    tmp : ditaylor2[lb(x,1)..ub(x,1), lb(x,2)..ub(x,2)];
BEGIN
  CASE no OF
    1 : f:= 1/sqr(6-2*x-y);
    2 : f:= cos(2*pi*0.25+75*x+25*y);
    3 : BEGIN
          tmp:= 0.0;
          FOR k:= 1 TO 10 DO
            tmp := tmp + sqr( 2 + 2*k - exp(k*x) - exp(k*y) );
          f:= tmp;
        END;
    END;
END;

```

Hierbei ist `pi` eine globale Konstante, die in dem Modul **User_Func** definiert ist (s. Seite 127). Ferner ist das Auswahlargument `no` für die Syntax notwendig, auch wenn in der `CASE`-Anweisung nur eine Funktion definiert ist. Die zugehörigen Integrationsbereiche dieser Integranden müssen zusammen in der Funktion `IntegrationArea()` mit der folgenden Syntax angegeben werden.

```

GLOBAL PROCEDURE IntegrationArea( VAR a, b : IVECTOR; dim, no : INTEGER );
VAR aa,bb : IVECTOR[1..dim];
    i      : INTEGER;
BEGIN
  CASE dim OF
    1 : CASE no OF
          1 : BEGIN
              aa[1] := intval(-1);
              bb[1] := intval(1);
            END;
          2 : BEGIN
              aa[1] := intval(0);
              bb[1] := intval(2);
            END;
          3 : BEGIN

```

```

        aa[1] := intval(0);
        bb[1] := intval(4);
    END;
END;
2 : CASE no OF
    1 : BEGIN
        aa[1] := intval(-1);
        bb[1] := intval(1);
        aa[2] := intval(0);
        bb[2] := intval(1.6);
    END;
    2 : FOR i := 1 TO dim DO
        BEGIN
            aa[i] := intval(0);
            bb[i] := intval(1);
        END;
    3 : FOR i := 1 TO dim DO
        BEGIN
            aa[i] := intval(-1);
            bb[i] := intval(1);
        END;
    END;
END;
a := aa;
b := bb;
END;

```

Da der eigentliche Integrationsbereich in $[a, b]$ festgehalten wird, sind die letzten zwei Zuweisungen unentbehrlich. Hierbei stellen a und b die dynamischen Intervallvektoren dar, deren Länge von dem Parameter `dim` bestimmt wird. Die Anzahl der Dimensionen wird durch interaktive Abfrage gesetzt.

Außer der Variable `dim` hat der Benutzer noch die Freiheit, die folgenden Attributen während der Laufzeit durch interaktive Auswahl bzw. Angabe zu setzen :

<code>dim</code>	1- oder 2-Dimension
<code>bd</code>	Integrationsbereich (wird von <code>no</code> bestimmt)
<code>no</code>	Funktionsnummer
<code>cat</code>	Kategorie der Integrationsformeln
	<code>dim = 1</code> : Gauß-Legendre- bzw. Clenshaw-Curtis-Quadraturformeln
	<code>dim = 2</code> : Produkt-Gauß-Legendre-, -Clenshaw-Curtis-Formeln
	bzw. vollsymmetrische Kubaturformeln
<code>part</code>	2 oder 2^{dim} -Unterteilung
<code>abs_err</code>	Oberschranke der absoluten Genauigkeit
<code>rel_err</code>	Oberschranke der relativen Genauigkeit

N_max	maximal zulässige Anzahl von Funktionsaufrufen
L_max	maximal zulässige Anzahl von Teilbereichen
BinTerT	binärer oder ternärer Baum

Als Ergebnis des Programms VANI werden die folgenden Informationen ausgegeben.

[I]	die erzielte Integraleinschließung
N_total	die Anzahl aller Funktionsaufrufe während des Programmablaufs
L_total	die Anzahl der zu der letzten Zerlegung zugehörigen Teilbereiche
LL_tot	die Anzahl der insgesamt bearbeiteten Teilbereiche
err	Typ des Fehlers
0	kein Fehler
1	die erwünschte Genauigkeit ist wegen des Rundungsfehlers nicht erreichbar
2	N_max ist überschritten
3	L_max ist überschritten

All die obigen Informationen werden zusammen in einer strukturierten Variable **data** festgehalten, die als Ganzes an die nächste Prozedur übergeben wird.

Weitere Ausgaben sind noch die erreichte absolute Genauigkeit $d([I])$ sowie die erreichte relative Genauigkeit, welche wie folgt angegeben wird :

$$\begin{cases} \frac{d([I])}{\langle [I] \rangle}, & 0 \notin [I], \\ d([I]), & 0 \in [I]. \end{cases}$$

5.5.2 Der verwendete Baum

Die Teilintegrationsbereiche, die während der automatischen Integration durch Zerlegung entstanden sind, werden zur Zeit in dem Programm VANI in einem Baum mit statischer Struktur festgehalten. In dem Baum enthält jeder Knoten die folgenden Informationen :

bd	Integrationsteilbereich
S	Einschließung der lokalen Approximationssumme $S^{(j)}$
R	Einschließung des lokalen Fehlers $E^{(j)}$
d	Durchmesser der lokalen Integraleinschließung $[S^{(j)}] + [E^{(j)}]$
dir	die Unterteilungsrichtung Ω_j
N	die Anzahl der Stützstellen der verwendeten lokalen Integrationsformel

Nach der Beschreibung im Abschnitt 5.1.1 sollen die in dem partiell geordneten balancierten Baum neu eingefügten Knoten an ihre richtigen Plätze sortiert werden. Diese Sortierungen machen die Prozeduren **BubbleDown**(n) und **BubbleUp**(n), wobei n die Anzahl der

gesamten Knoten des aktuellen Baums darstellt. In VANI hat der Benutzer die Möglichkeit, den Baum binär bzw. ternär zu gestalten. Um diesen Unterschied in den beiden Prozeduren zu erkennen, werden 2 zusätzliche Variablen *nnodes* und *counter* verwendet. Für den binären Baum gilt

$$nnodes := 2; \quad counter := 1;$$

und für den ternären Baum

$$nnodes := 3; \quad counter := 2;$$

Algorithmus 5.2 : BubbleDown(*n*)

1. $i := 1$.
2. **if** ($nnodes = 3$) **then** $j := nnodes \cdot i \Leftrightarrow 1$ **else** $j := nnodes \cdot i$.
3. $changed := \text{TRUE}$.
4. **while** ($j \leq n$) **and** $changed$ **do**
 - if** ($j = n$) **then** ExchangeNode($i, j, changed$)
 - else** $m := j + counter$;
 - if** ($n < m$) **then** $m := n$;
 - for** $k := j + 1$ **to** m **do**
 - if** ($Node[j].d < Node[k].d$) **then** $j := k$;
 - ExchangeNode($i, j, changed$).
- if** $changed$ **then**
 - $i := j$;
 - if** ($nnodes = 3$) **then** $j := nnodes \cdot i \Leftrightarrow 1$ **else** $j := nnodes \cdot i$.

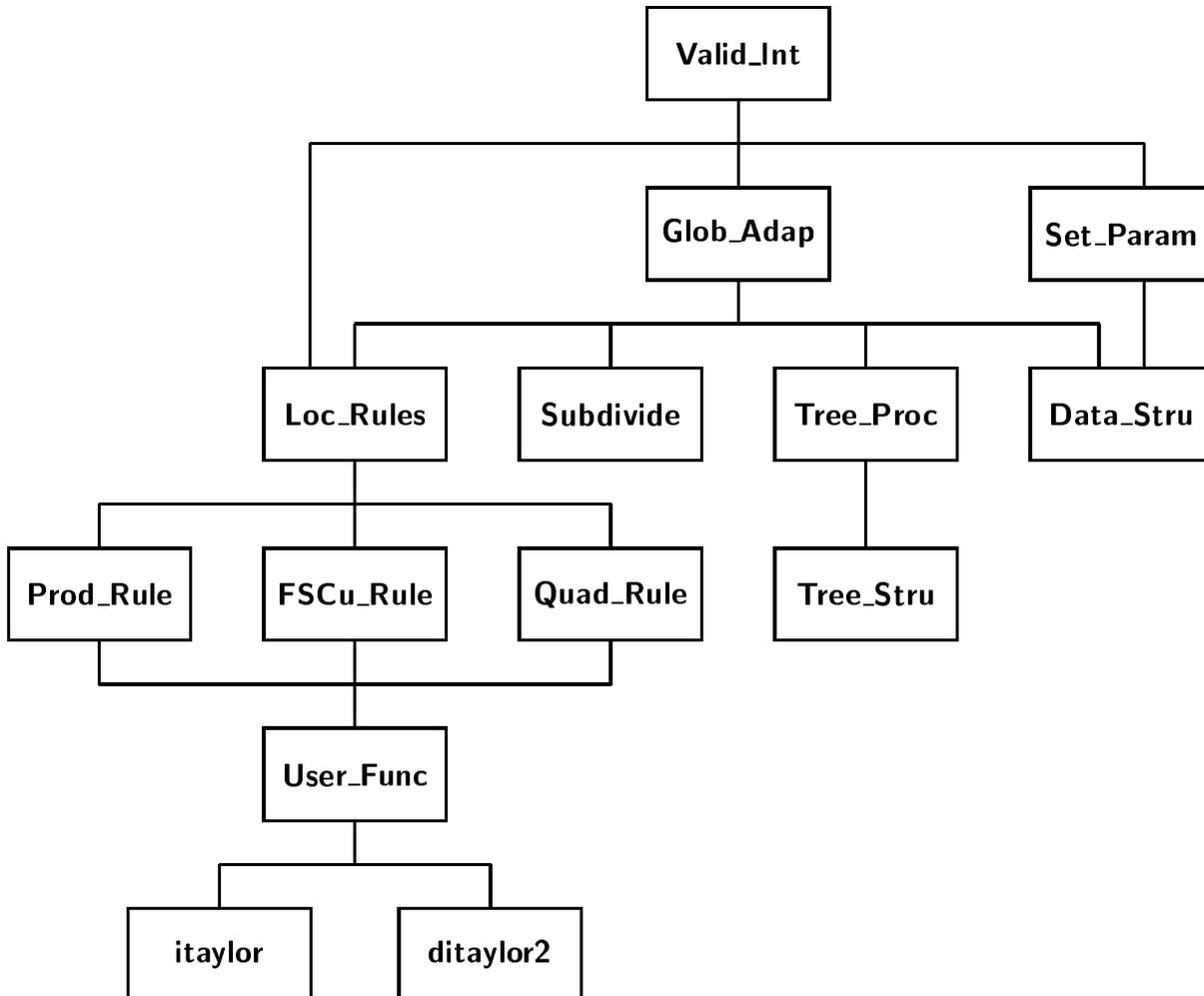
In der Prozedur ExchangeNode($i, j, changed$) werden die Knoten $Node[i]$ und $Node[j]$ mit $i < j$ nur dann vertauscht, wenn $Node[i].d < Node[j].d$ gilt.

Algorithmus 5.2 : BubbleUp(*n*)

1. $i := n$.
2. **if** ($nnodes = 2$) **then** $j := \lceil i/nnodes \rceil$ **else** $j := \lceil (i + 1)/nnodes \rceil$.
3. $changed := \text{TRUE}$.
4. **while** ($j > 0$) **and** $changed$ **do**
 - ExchangeNode($j, i, changed$);
 - if** $changed$ **then**
 - $i := j$;
 - if** ($nnodes = 2$) **then** $j := \lceil i/nnodes \rceil$ **else** $j := \lceil (i + 1)/nnodes \rceil$.

5.5.3 Programmaufbau

Die verschiedenen Teilaspekte, die in diesem Kapitel behandelt werden, sind in dem Softwareprogramm VANI berücksichtigt. Das nächste Diagramm zeigt die hierarchische Struktur der 14 Bausteine des Programms VANI.



- **Data_Struc** und **Set_Param**

Im Modul **Data_Struc** werden die im Abschnitt 5.5.1 erwähnte Datenstruktur zum Abspeichern der Eingabeparameter und der Ausgabedaten definiert.

Die Prozeduren und Funktionen für die Abfrage und das Abspeichern der Werte der einzelnen Eingabeparameter sind im Modul **Set_Param** implementiert.

- **Tree_Struc** und **Tree_Proc**

Im Modul **Tree_Struc** wird die Datenstruktur des angewendeten Baums definiert.

Die notwendigen Baumoperationen sind im Modul **Tree_Proc** enthalten. Unter anderem sind dies die Initialisierung des Baums, das Vertauschen von Baumknoten,

die Definition von neuer Wurzel, die Anfügung, die Hoch- und Heruntersortierung von Knoten.

- **Subdivide**

In diesem Modul sind die Prozeduren für die binäre und die 2^n -Unterteilung eines Teilintegrationsbereichs definiert.

- **Loc_Rules**

Dieses Modul enthält die Prozeduren für die Initialisierung von Integrationsformeln. D. h. die Stützstellen-, die Gewichte- und die Fehlerkonstanteneinschließungen des ausgewählten Integrationsverfahrens werden von Dateien in Matrizen eingelesen und der maximale Exaktheitsgrad dieser Integrationsformeln wird gesetzt.

Darüber hinaus enthält dieses Modul die Zugänge zu den verschiedenen vorhandenen Quadratur- bzw. Kubaturverfahren.

Die in VANI zur Verfügung gestellten Integrationsformeln sind die Gauß-Legendre-Formeln, die in [144] erzielt wurden, die Clenshaw-Curtis-Formeln, die sich direkt berechnen lassen (s. Abschnitt 3.3.2), und die vollsymmetrischen Kubaturformeln, die in [45] konstruiert wurden. Dabei wurden ihre zugehörigen Fehlerkonstanten im Rahmen dieser Arbeit verifiziert berechnet.

- **Valid_Int**

Im Hauptprogramm **Valid_Int** werden die folgenden Teilaufgaben durchgeführt :

1. Die notwendigen Parameter werden interaktiv abgefragt und in einer strukturierten Variablen festgehalten.
2. Die vom Benutzer ausgewählten Integrationsformeln werden initialisiert.
3. Die Integrale über den Randbereichen werden berechnet, falls die Grenzen des Integrationsbereichs keine Gleitkommazahlen sind. In diesem Fall werden die zweidimensionalen Funktionen als eindimensionale Funktionen betrachtet, für welche dann das adaptive numerische Integrationsverfahren eingesetzt wird.
4. Für das Integral über dem inneren Integrationsbereich wird die adaptive numerische Integration, die der Dimension des Integranden entspricht, durchgeführt.
5. Die numerischen Ergebnisse und der Fehlerzustand werden berichtet.

- **Glob_Adap**

In diesem Modul ist die globale Verfeinerungsstrategie unter Berücksichtigung der im Abschnitt 5.3 beschriebenen Fehlerbehandlung implementiert. Diese ist im nächsten Algorithmus zusammengefasst.

Algorithmus 5.2 : GlobalAdaption($N_{\max}, L_{\max}, \varepsilon_a, \varepsilon_r, S, E$)

1. **if** $\varepsilon_r = 0$ **then** $flag := 0$ **else** $flag := 1$.
2. **if** $flag = 0$ **then** berechne $[E]$ über Ω ; $[S] := 0$
else berechne $[S]$ und $[E]$ über Ω .
3. Füge $(\Omega, [S], [E], d([S] + [E]))$ im Baum ein.
4. $return := \text{FALSE}$.
5. **if** $flag = 0$ **and** $d([E]) \leq \varepsilon_a$ **then** $return := \text{TRUE}$
else if $flag = 1$ **and** $d([I]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ **then** $return := \text{TRUE}$.
6. $err := 0$.
7. **if** $\#(\text{Funktionsaufrufe}) > N_{\max}$ **then** $err := 2$
else if $\#(\text{Teilbereiche}) > L_{\max}$ **then** $err := 3$
else if $flag = 1$ **and**
 $(d([E]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ **and** $d([S]) \geq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$
then $err := 1$.
8. **while not** $return$ **and** $err = 0$ **do**
 - 8.1. Entferne die Wurzel $(\Omega_w, [S_w], [E_w], d([S_w] + [E_w]))$ aus dem Baum.
 - 8.2. $\underline{E} := \underline{E} \Leftrightarrow \underline{E}_w$; $\overline{E} := \overline{E} \Leftrightarrow \overline{E}_w$.
 - 8.3. **if** $flag = 1$ **then** $\underline{S} := \underline{S} \Leftrightarrow \underline{S}_w$; $\overline{S} := \overline{S} \Leftrightarrow \overline{S}_w$.
 - 8.4. Zerlege Ω_w in q Teilbereiche Ω_j von gleicher Volume.
 - 8.5. Berechne für jeden neuen Teilbereich $[E^{(j)}]$; $[S^{(j)}] := 0$.
 - 8.6. $[E] := [E] + \sum_{j=1}^q [E^{(j)}]$.
 - 8.7. **if** $flag = 1$ **then** berechne $[S^{(j)}]$; $[S] := [S] + \sum_{j=1}^q [S^{(j)}]$.
 - 8.8. Füge die neuen Teilbereiche im Baum ein.
 - 8.9. Sortiere den Baum nach $d([I^{(j)}])$.
 - 8.10. Bestimme $return$ gemäß 5.
 - 8.11. Bestimme err gemäß 7.
9. **if** $flag = 0$ **then**
Traversiere den Baum und berechne $[S] = \sum_{j=1}^q [S^{(j)}]$;
 $flag := 1$; $return := \text{FALSE}$;
Bestimme $return$ gemäß 5;
Bestimme err gemäß 7;
goto 8.

• **Prod_Rule** und **FSCu_Rule** und **Quad_Rule**

Diese Module enthalten jeweils zwei Prozeduren.

Die eine berechnet den lokalen Diskretisierungsfehler, bestimmt die best geeignete Integrationsformel, die für den betrachteten Teilbereich angewendet werden soll, und legt die Unterteilungsrichtung fest.

Die andere verwendet die ausgewählte Integrationsformel, um die lokale Approximationssumme zu berechnen.

- **User_Func**

In diesem Modul werden die zu integrierenden Funktionen definiert. Es besitzt das folgende Syntaxskelett.

```

MODULE User_Func;

USE GLOBAL itaylor;
USE GLOBAL ditaylor2;

VAR pi : INTERVAL;

GLOBAL FUNCTION f( x : itaylor; no : INTEGER )
                    : itaylor[lb(x)..ub(x)];

BEGIN
END;

GLOBAL FUNCTION f( x,y : ditaylor2; no : INTEGER )
                    : ditaylor2[lb(x,1)..ub(x,1), lb(x,2)..ub(x,2)];

BEGIN
END;

GLOBAL PROCEDURE IntegrationArea( VAR a, b : IVECTOR;
                                   dim, no : INTEGER );

BEGIN
END;

BEGIN
  pi := 2*arcsin(intval(1.0));
END.

```

Die beiden Funktionen `f()` und die Prozedur `IntegrationArea()` müssen nach der im Abschnitt 5.5.1 beschriebenen Syntax definiert werden.

- **itaylor** und **ditaylor2**

Im Modul **itaylor** sind die Funktionen und Operatoren zur Berechnung eindimensionaler Taylorkoeffizienten $(f)_i$, $i = 0(1)UB(x)$ implementiert. Die unabhängige Variable x und die Konstante c werden definiert durch

$$\text{expand}(f, x) \quad \text{bzw.} \quad f := c .$$

Für die Berechnung zweidimensionaler Taylorkoeffizienten $(f)_{i,j}$ enthält das Modul **ditaylor2** die notwendigen Operatoren und Funktionen. Die unabhängigen Variablen x und y werden nach der folgenden Syntax

$$\text{expand}(fx, x, 1) \quad \text{bzw.} \quad \text{expand}(fy, y, 2)$$

definiert. Konstanten sind ebenfalls durch Zuweisungen zu definieren.

In **ditaylor2** werden nur die Taylorkoeffizienten mit den folgenden Indizes berechnet.

$$\begin{cases} i = 0(1)UB_1, & j = 0(1)UB_2 \Leftrightarrow i & \text{für } UB_1 \leq UB_2 \\ i = 0(1)UB_1, & j = 0(1) \min(UB_2, UB_1 \Leftrightarrow i) & \text{für } UB_1 > UB_2 \end{cases}$$

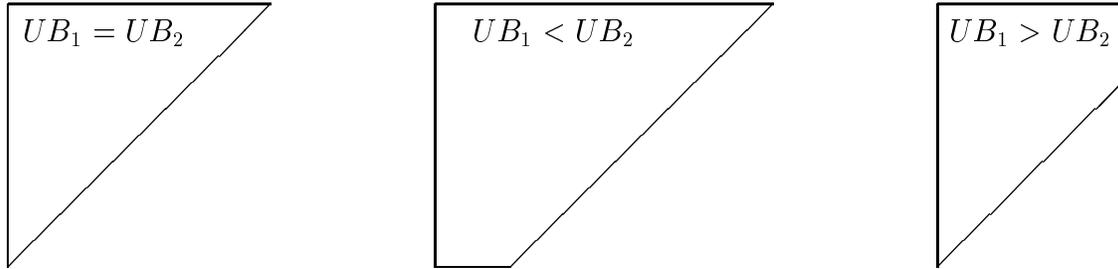


Abbildung 5.1: Zweidimensionale Taylorkoeffizienten in Dreieckanordnungen

Kapitel 6

Numerische Beispiele

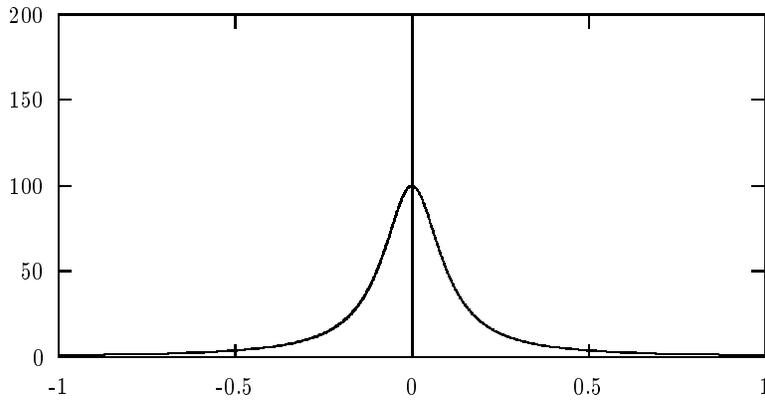
In diesem Kapitel wird die ein- und zweidimensionale adaptive numerische Integration mit automatischer Ergebnisverifikation, die zur Zeit in dem Programm VANI implementiert ist, anhand verschiedener Beispiele (glatte Funktionen, Funktionen mit Peak und oszillierende Funktionen) demonstriert. Für die Integrationsbereiche, deren Grenzen keine Gleitkommazahlen sind, ist die Verifikation über dem Bereichsrand notwendig, dies wird ebenfalls betrachtet. Die Berechnungen, die auf der Maschine Sun Ultra 1 Model 170E durchgeführt wurden, verwendeten die binäre Unterteilung für die Bereichsverfeinerung und den binären Baum für die Informationsregistrierung der Teilbereiche. Ferner wurden für die Betrachtung die Fehlereinschließungen $[E(f)]$ bzgl. der Kernfunktionen von maximalen Ordnungen $r = d + 1$ berechnet, wobei $d := \deg S$ ist (s. Kapitel 3 und 4).

6.1 Numerische Quadratur

In dem Programm VANI werden für die eindimensionale automatische Integration die Clenshaw-Curtis- und die Gauß-Legendre-Quadraturformeln zur Verfügung gestellt. Die Exaktheitsgrade dieser Formeln sind $3(2)19$. Die verwendeten Knotenzahlen bei den Clenshaw-Curtis-Formeln sind $M = 4, 5(2)19$ und bei den Gauß-Legendre-Formeln $M = 2(1)10$. Die Effizienz der Gauß-Legendre-Quadratur läßt sich weiter erhöhen, wenn die Gauß-Legendre-Formeln von höheren Exaktheitsgraden eingesetzt werden. Es ist dennoch schwierig, ihre kompletten Fehlerkonstanten zu berechnen. Die folgenden numerischen Ergebnisse beabsichtigen deshalb nicht, die Effizienz der beiden Quadraturverfahren zu vergleichen, vielmehr wird der Anwendungsbereich der zur Zeit in VANI implementierten Quadraturverfahren demonstriert.

6.1.1 Glatte Funktionen

Für glatte Funktionen haben, bei Anwendung von VANI, die Gauß-Legendre- und die Clenshaw-Curtis-Quadratur ungefähr die gleiche Effektivität (vgl. die Bemerkung oben). Diese Ergebnisse zeigen die nächsten beiden Beispiele.

Beispiel 1 (vgl. [61])

$$f(x) = \frac{10^2}{1 + (10x)^2}$$

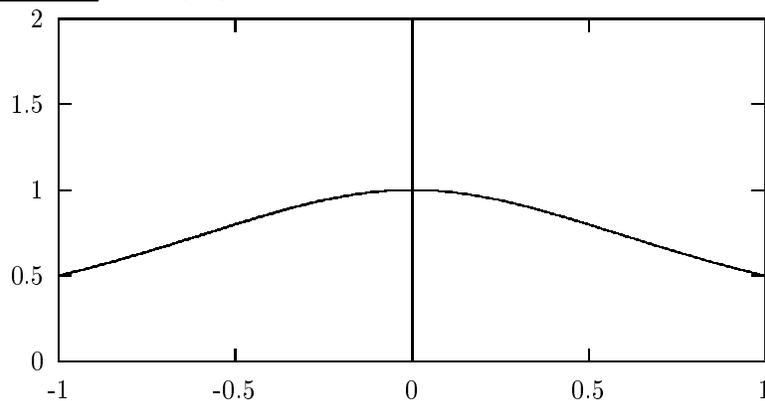
$$\Omega = [-1, 1]$$

Gauß-Legendre :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$29. \binom{5}{4}$	$4.15 \cdot 10^{-2}$	52	14	0.12	0
10^{-3}	$29.42 \binom{3}{2}$	$5.36 \cdot 10^{-4}$	76	16	0.12	0
10^{-6}	$29.42255 \binom{4}{3}$	$3.86 \cdot 10^{-7}$	120	16	0.12	0
10^{-9}	$29.42255348 \binom{7}{5}$	$3.93 \cdot 10^{-10}$	170	22	0.17	0
10^{-12}	$29.42255348607 \binom{6}{4}$	$7.57 \cdot 10^{-13}$	240	28	0.19	0
10^{-15}	$29.422553486074 \binom{8}{6}$	$3.20 \cdot 10^{-14}$	292	32	0.23	1

Clenshaw-Curtis :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$29. \binom{5}{3}$	$8.22 \cdot 10^{-2}$	104	8	0.07	0
10^{-3}	$29.42 \binom{3}{2}$	$4.93 \cdot 10^{-4}$	110	14	0.12	0
10^{-6}	$29.422553 \binom{6}{4}$	$1.57 \cdot 10^{-7}$	176	16	0.13	0
10^{-9}	$29.42255348 \binom{62}{59}$	$2.20 \cdot 10^{-10}$	252	16	0.15	0
10^{-12}	$29.42255348607 \binom{5}{4}$	$5.47 \cdot 10^{-13}$	334	22	0.20	0
10^{-15}	$29.422553486074 \binom{8}{6}$	$4.97 \cdot 10^{-14}$	457	29	0.26	1

Beispiel 2 (vgl. [37])

$$f(x) = \frac{1}{1 + x^2}$$

$$\Omega = [-1, 1]$$

Gauß-Legendre :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$1.\binom{6}{5}$	$4.08 \cdot 10^{-2}$	8	2	0.02	0
10^{-3}	$1.57\binom{2}{0}$	$7.38 \cdot 10^{-4}$	16	2	0.03	0
10^{-6}	$1.57079\binom{7}{6}$	$1.33 \cdot 10^{-7}$	22	4	0.04	0
10^{-9}	$1.57079632\binom{7}{6}$	$3.31 \cdot 10^{-10}$	30	4	0.04	0
10^{-12}	$1.57079632679\binom{6}{4}$	$9.75 \cdot 10^{-13}$	38	4	0.04	0
10^{-15}	$1.570796326794\binom{90}{89}$	$1.78 \cdot 10^{-15}$	62	8	0.07	1
10^{-16}	$1.57079632679489\binom{8}{5}$	$1.33 \cdot 10^{-15}$	68	8	0.07	1

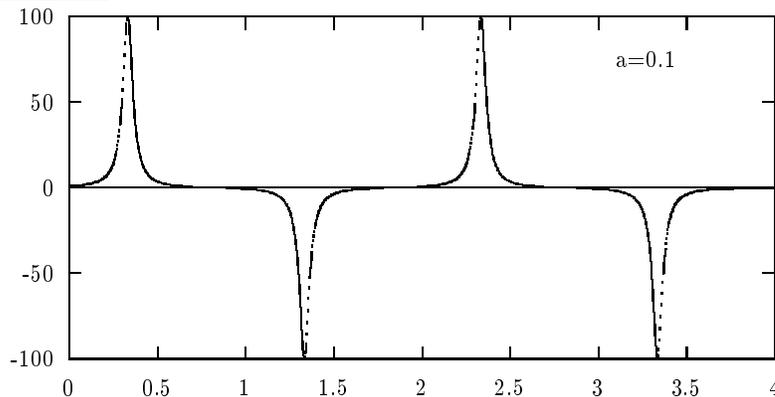
Clenshaw-Curtis :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$1.\binom{6}{5}$	$4.64 \cdot 10^{-2}$	10	2	0.03	0
10^{-3}	$1.57\binom{1}{0}$	$3.70 \cdot 10^{-4}$	18	2	0.04	0
10^{-6}	$1.57079\binom{7}{5}$	$7.16 \cdot 10^{-7}$	34	2	0.04	0
10^{-9}	$1.570796326\binom{9}{7}$	$6.11 \cdot 10^{-11}$	48	4	0.04	0
10^{-12}	$1.57079632679\binom{50}{48}$	$5.73 \cdot 10^{-14}$	64	4	0.05	0
10^{-15}	$1.570796326794\binom{90}{89}$	$3.33 \cdot 10^{-15}$	83	5	0.07	1
10^{-16}	$1.570796326794\binom{90}{89}$	$2.66 \cdot 10^{-15}$	94	6	0.07	1

6.1.2 Funktionen mit Peak

Zu dieser Funktionsklasse wird die Funktion $f(x) = \frac{1}{a^2+(3x-1)^2} \Leftrightarrow \frac{1}{a^2+(3x-4)^2} + \frac{1}{a^2+(3x-7)^2} \Leftrightarrow \frac{1}{a^2+(3x-10)^2}$ mit $a = 0.1$ und $a = 0.001$ betrachtet. Den Funktionsgraphen für $a = 0.1$ zeigt das nächste Bild. Für $a = 0.001$ werden die Spitzen an den Stellen $x = \frac{1}{3}, \frac{4}{3}, \frac{7}{3}, \frac{10}{3}$ um das 10^4 -fache verstärkt.

Im allgemeinen ist die in VANI implementierte Clenshaw-Curtis-Quadratur für diese Funktionen effizienter als die Gauß-Legendre-Quadratur in VANI.

Beispiel 3 (vgl. [61])

$$f(x) = \frac{1}{a^2 + (3x \Leftrightarrow 1)^2} \Leftrightarrow \frac{1}{a^2 + (3x \Leftrightarrow 4)^2} + \frac{1}{a^2 + (3x \Leftrightarrow 7)^2} \Leftrightarrow \frac{1}{a^2 + (3x \Leftrightarrow 10)^2}$$

$$a = 0.1, \quad \Omega = [0, 4]$$

Gauß-Legendre :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$\Leftrightarrow 0.1(\frac{1}{9})$	$6.70 \cdot 10^{-2}$	269	52	1.07	0
10^{-3}	$\Leftrightarrow 0.15(\frac{1}{3})$	$9.92 \cdot 10^{-4}$	388	63	1.30	0
10^{-6}	$\Leftrightarrow 0.15196(\frac{3}{5})$	$8.72 \cdot 10^{-7}$	561	84	1.78	0
10^{-9}	$\Leftrightarrow 0.15196394(\frac{1}{3})$	$7.85 \cdot 10^{-10}$	769	92	1.99	0
10^{-12}	$\Leftrightarrow 0.15196394223(\frac{2}{4})$	$9.58 \cdot 10^{-13}$	1108	126	2.80	0
10^{-15}	$\Leftrightarrow 0.15196394223(\frac{2}{4})$	$4.69 \cdot 10^{-13}$	1425	161	3.56	1

Clenshaw-Curtis :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$\Leftrightarrow 0.1(\frac{2}{8})$	$5.15 \cdot 10^{-2}$	337	52	1.10	0
10^{-3}	$\Leftrightarrow 0.15(\frac{1}{3})$	$4.49 \cdot 10^{-4}$	546	52	1.19	0
10^{-6}	$\Leftrightarrow 0.15196(\frac{3}{5})$	$9.90 \cdot 10^{-7}$	855	65	1.54	0
10^{-9}	$\Leftrightarrow 0.15196394(\frac{1}{3})$	$8.48 \cdot 10^{-10}$	1174	86	2.07	0
10^{-12}	$\Leftrightarrow 0.15196394223(\frac{2}{4})$	$9.79 \cdot 10^{-13}$	1624	98	2.50	0
10^{-15}	$\Leftrightarrow 0.15196394223(\frac{2}{4})$	$4.44 \cdot 10^{-13}$	2060	124	3.16	1

Beispiel 4

$$\int_0^4 \frac{1}{a^2 + (3x \Leftrightarrow 1)^2} \Leftrightarrow \frac{1}{a^2 + (3x \Leftrightarrow 4)^2} + \frac{1}{a^2 + (3x \Leftrightarrow 7)^2} \Leftrightarrow \frac{1}{a^2 + (3x \Leftrightarrow 10)^2} dx, \quad a = 0.001$$

Gauß-Legendre :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$(\begin{smallmatrix} -1.1 \\ -2.1 \end{smallmatrix})$	$9.75 \cdot 10^{-2}$	1033	154	3.34	0
10^{-3}	$\Leftrightarrow 1.5(\frac{2}{4})$	$9.87 \cdot 10^{-4}$	1276	177	3.88	0
10^{-6}	$\Leftrightarrow 1.5292(\frac{1}{3})$	$8.83 \cdot 10^{-7}$	1764	223	4.93	0
10^{-9}	$\Leftrightarrow 1.529219(\frac{7}{9})$	$4.76 \cdot 10^{-9}$	2429	282	6.34	1
10^{-12}	$\Leftrightarrow 1.529219(\frac{7}{9})$	$3.69 \cdot 10^{-9}$	3136	346	7.89	1

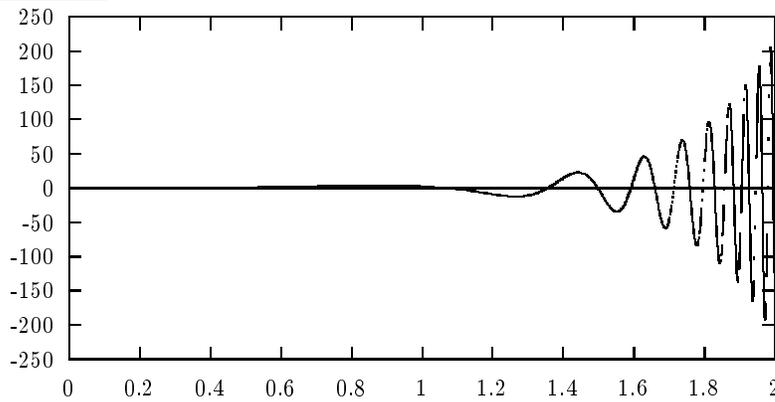
Clenshaw-Curtis :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$\Leftrightarrow 1.(\frac{3}{8})$	$4.18 \cdot 10^{-2}$	1457	136	3.19	0
10^{-3}	$\Leftrightarrow 1.5(\frac{2}{4})$	$9.60 \cdot 10^{-4}$	1992	138	3.48	0
10^{-6}	$\Leftrightarrow 1.5292(\frac{1}{3})$	$9.44 \cdot 10^{-7}$	2624	182	4.59	0
10^{-9}	$\Leftrightarrow 1.529219(\frac{7}{9})$	$4.44 \cdot 10^{-9}$	3532	224	5.75	1
10^{-12}	$\Leftrightarrow 1.529219(\frac{7}{9})$	$3.89 \cdot 10^{-9}$	4777	287	7.65	1

6.1.3 Oszillierende Funktionen

Für oszillierende Funktionen benötigen die Clenshaw-Curtis-Quadratur von VANI mehr Rechenzeit als die Gauß-Legendre-Quadratur von VANI, auch wenn die Integranden einfache Funktionen sind.

Beispiel 5 (vgl. [61])



$$f(x) = 2xe^{x^2} \sin(e^{x^2})$$

$$\Omega = [0, 2]$$

Gauß-Legendre :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$\Leftrightarrow 0.(\frac{95}{88})$	$5.93 \cdot 10^{-2}$	57	8	0.79	0
10^{-3}	$\Leftrightarrow 0.91(\frac{12}{67})$	$3.60 \cdot 10^{-4}$	73	10	1.03	0
10^{-6}	$\Leftrightarrow 0.91096(\frac{45}{35})$	$9.35 \cdot 10^{-7}$	102	12	1.35	0
10^{-9}	$\Leftrightarrow 0.91096403(\frac{96}{89})$	$6.02 \cdot 10^{-10}$	139	16	1.85	0
10^{-12}	$\Leftrightarrow 0.91096403926(\frac{8}{4})$	$2.20 \cdot 10^{-12}$	192	21	2.51	1
10^{-15}	$\Leftrightarrow 0.91096403926(\frac{7}{5})$	$1.48 \cdot 10^{-12}$	263	28	3.44	1

Clenshaw-Curtis :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$\Leftrightarrow 0.(\frac{93}{89})$	$2.27 \cdot 10^{-2}$	86	6	0.98	0
10^{-3}	$\Leftrightarrow 0.91(\frac{11}{68})$	$2.54 \cdot 10^{-4}$	108	8	1.28	0
10^{-6}	$\Leftrightarrow 0.91096(\frac{42}{39})$	$2.77 \cdot 10^{-7}$	144	10	1.69	0
10^{-9}	$\Leftrightarrow 0.910964039(\frac{6}{6})$	$5.21 \cdot 10^{-10}$	205	13	2.36	0
10^{-12}	$\Leftrightarrow 0.91096403926(\frac{7}{4})$	$1.96 \cdot 10^{-12}$	274	16	3.10	1
10^{-15}	$\Leftrightarrow 0.91096403926(\frac{7}{5})$	$1.35 \cdot 10^{-12}$	382	22	4.37	1

Beispiel 6

$$\int_{0.1}^{3.2} \sin x \, dx$$

Der Integrationsbereich dieses Beispiels kann nicht in Gleitkommazahlen exakt dargestellt

werden. Die Verifikation über dem Bereichsrand ist daher notwendig, wofür zusätzlich zwei Funktionsauswertungen benötigt werden.

Gauß-Legendre :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$(\begin{smallmatrix} 2.0 \\ 1.9 \end{smallmatrix})$	$7.01 \cdot 10^{-2}$	4	1	0.05	0
10^{-3}	$1.993(\begin{smallmatrix} 3 \\ 2 \end{smallmatrix})$	$1.57 \cdot 10^{-5}$	6	1	0.07	0
10^{-6}	$1.99329(\begin{smallmatrix} 91 \\ 89 \end{smallmatrix})$	$1.06 \cdot 10^{-7}$	7	1	0.08	0
10^{-9}	$1.99329894(\begin{smallmatrix} 2 \\ 0 \end{smallmatrix})$	$4.86 \cdot 10^{-10}$	8	1	0.09	0
10^{-12}	$1.9932989410727(\begin{smallmatrix} 9 \\ 7 \end{smallmatrix})$	$6.88 \cdot 10^{-15}$	10	1	0.10	0
10^{-15}	$1.9932989410727(\begin{smallmatrix} 8 \\ 7 \end{smallmatrix})$	$2.66 \cdot 10^{-15}$	11	1	0.11	1
10^{-16}	$1.9932989410727(\begin{smallmatrix} 8 \\ 7 \end{smallmatrix})$	$2.66 \cdot 10^{-15}$	11	1	0.11	1

Clenshaw-Curtis :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	sec	err
10^{-1}	$(\begin{smallmatrix} 2.0 \\ 1.9 \end{smallmatrix})$	$2.77 \cdot 10^{-2}$	6	1	0.08	0
10^{-3}	$1.99(\begin{smallmatrix} 4 \\ 3 \end{smallmatrix})$	$6.07 \cdot 10^{-4}$	7	1	0.10	0
10^{-6}	$1.9932989(\begin{smallmatrix} 5 \\ 4 \end{smallmatrix})$	$2.61 \cdot 10^{-9}$	11	1	0.14	0
10^{-9}	$1.9932989410(\begin{smallmatrix} 8 \\ 6 \end{smallmatrix})$	$5.72 \cdot 10^{-12}$	13	1	0.13	0
10^{-12}	$1.993298941072(\begin{smallmatrix} 8 \\ 7 \end{smallmatrix})$	$1.53 \cdot 10^{-14}$	15	1	0.15	0
10^{-15}	$1.9932989410727(\begin{smallmatrix} 9 \\ 7 \end{smallmatrix})$	$5.11 \cdot 10^{-15}$	17	1	0.18	1
10^{-16}	$1.9932989410727(\begin{smallmatrix} 9 \\ 7 \end{smallmatrix})$	$5.11 \cdot 10^{-17}$	17	1	0.18	1

Für die numerische Betrachtung in diesem Kapitel bezeichnet $\#f$ die Anzahl der gesamten Funktionsauswertungen $f([x_i])$, $\#\Omega_j$ die Anzahl der auf die letzte Zerlegung bezogenen Teilbereiche. Bei Anwendung einer binären Unterteilung beträgt für jede Durchführung adaptiver numerischer Integration die Anzahl der gesamten bearbeiteten Teilbereiche $\#\tilde{\Omega}_j = 2 \cdot \#\Omega_j \Leftrightarrow 1$.

Die numerischen Ergebnisse in diesem Abschnitt zeigen, daß das Programm VANI die eindimensionalen Integraleinschließungen recht effizient berechnen und die unerreichbaren Genauigkeiten in relativ kurzer Zeit ermitteln kann.

6.2 Numerische Kubatur

In diesem Abschnitt wird die zweidimensionale adaptive numerische Integration demonstriert. Für die Betrachtung werden zum Vergleich noch die von dem Softwarepaket Cubpack++ (vgl. [34], [35]) erzielten numerischen Ergebnisse zusammengestellt. Das Softwarepaket Cubpack++ ist speziell für die zweidimensionale automatische Integration entwickelt, wobei vollsymmetrische Kubaturformeln verwendet werden. Vollsymmetrische

Kubaturformeln besitzen im allgemeinen weniger Knoten als die Produktformeln. Da die Intervallarithmetik, auf der das Programm VANI beruht, in Software implementiert ist, ist es nicht gerechtfertigt, die benötigte Rechenzeit der beiden Programme direkt zu vergleichen.

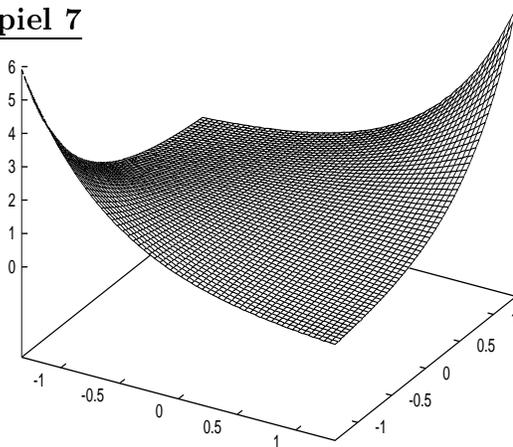
6.2.1 Zum Vergleich der vorhandenen Kubaturverfahren

Für die zweidimensionale automatische Integration stehen im Programm VANI die Produkt-Gauß-Legendre-, die Produkt-Clellshaw-Curtis- und die vollsymmetrische Kubaturformeln über dem Quadrat $[-1, 1] \times [-1, 1]$ zur Verfügung. Die Produktformeln besitzen ungerade Exaktheitsgrade 3(2)19. Hingegen haben die in VANI vorhandenen vollsymmetrischen Kubaturformeln die Exaktheitsgrade 3 und 7, welche sich für die meisten Integranden als unzureichend herausgestellt haben. Unter diesen Integrationsformeln zeigte das Produkt-Gauß-Legendre-Verfahren die beste Effizienz. Dies wird anhand des Beispiels 8 demonstriert. Aus diesem Grund wurden die numerischen Ergebnisse der weiteren Beispiele in diesem Abschnitt mit Hilfe der Produkt-Gauß-Legendre-Formeln berechnet.

Knotenzahl der in VANI herangezogenen Kubaturverfahren

deg S	3	5	7	9	11	13	15	17	19
M_{PGL}	4	9	16	25	36	49	64	81	100
M_{PCC}	16	25	49	81	121	169	225	289	361
M_{VSK}	4	-	12	-	28	-	44	-	68

Beispiel 7



$$\exp(x \cdot y), \quad \Omega = \left[-\frac{4}{3}, \frac{4}{3}\right]^2$$

Da die Grenzen von Ω keine Gleitkommazahlen sind, gibt es insgesamt 4 Randbereiche vom Typ 1 zu berücksichtigen. Die Integraleinschließungen $[I_j^{(1)}]$ und $[I_{inn}]$ werden jeweils mit Hilfe 1- bzw. 2-dimensionaler adaptiver numerischer Integrationsverfahren berechnet.

Für diese einfache Funktion benötigt Cubpack++ i. a. mehr Funktionsauswertungen als VANI. Die unerreichbare Genauigkeitsanforderung $\varepsilon_a = 10^{-15}$ kann Cubpack++ nicht innerhalb von angemessener Zeit erkennen. Das Programm wird erst abgebrochen, wenn die maximale Anzahl von Funktionsauswertungen überschritten ist.

VANI — PGL :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$8.48\binom{6}{3}$	$1.81 \cdot 10^{-3}$	30	5	5	0.13	0
10^{-3}	$8.4846\binom{8}{6}$	$1.61 \cdot 10^{-5}$	41	5	5	0.15	0
10^{-6}	$8.484671\binom{78}{67}$	$9.69 \cdot 10^{-8}$	56	5	5	0.18	0
10^{-9}	$8.48467172\binom{41}{36}$	$4.23 \cdot 10^{-10}$	71	5	5	0.22	0
10^{-12}	$8.4846717238619\binom{7}{3}$	$2.66 \cdot 10^{-14}$	105	5	5	0.28	0
10^{-15}	$8.4846717238619\binom{7}{3}$	$2.13 \cdot 10^{-14}$	128	5	5	0.36	1

VANI — PCC :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$8.\binom{52}{45}$	$5.32 \cdot 10^{-2}$	39	5	5	0.15	0
10^{-3}	$8.484\binom{8}{6}$	$1.24 \cdot 10^{-4}$	65	5	5	0.20	0
10^{-6}	$8.48467\binom{20}{15}$	$3.96 \cdot 10^{-7}$	101	5	5	0.29	0
10^{-9}	$8.48467172386\binom{4}{0}$	$2.80 \cdot 10^{-12}$	191	5	5	0.47	0
10^{-12}	$8.4846717238619\binom{7}{3}$	$3.91 \cdot 10^{-14}$	249	5	5	0.58	0
10^{-15}	$8.4846717238619\binom{7}{3}$	$3.02 \cdot 10^{-14}$	317	5	5	0.69	1

VANI — VSK :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$8.48\binom{7}{2}$	$4.32 \cdot 10^{-3}$	62	8	11	0.29	0
10^{-3}	$8.48\binom{51}{43}$	$7.70 \cdot 10^{-4}$	88	10	15	0.42	0
10^{-6}	$8.48467\binom{22}{12}$	$8.91 \cdot 10^{-7}$	296	27	49	1.23	0
10^{-9}	$8.48467172\binom{5}{3}$	$9.67 \cdot 10^{-10}$	1150	98	191	4.87	0
10^{-12}	$8.48467172386\binom{3}{1}$	$9.98 \cdot 10^{-13}$	6048	468	931	29.68	0
10^{-15}	$8.48467172386\binom{22}{17}$	$3.41 \cdot 10^{-13}$	23200	1935	3865	98.63	1

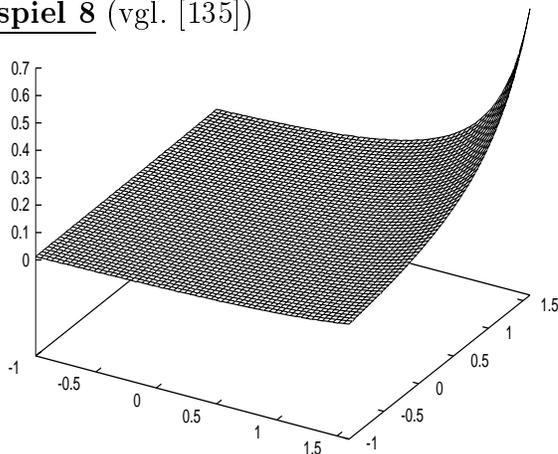
Cubpack++ :

ε_{abs}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	8.484671722887295	$4.15 \cdot 10^{-4}$	185	0.01	in
10^{-3}	8.484671722887295	$4.15 \cdot 10^{-4}$	185	0.01	in
10^{-6}	8.484671723812260	$2.66 \cdot 10^{-7}$	481	0.01	in
10^{-9}	8.484671723861949	$3.31 \cdot 10^{-10}$	1961	0.04	in
10^{-12}	8.484671723861936	$7.81 \cdot 10^{-13}$	6697	0.10	in
10^{-15}	8.484671723861327	$9.42 \cdot 10^{-14}$	10000101	142.16	out

Liegen die von Cubpack++ erzielten Integralnäherungen in den entsprechenden Integralschließungen, die gemäß gleichen Genauigkeitsanforderungen von VANI berechnet sind, werden diese als "in" bezeichnet, ansonsten "out".

6.2.2 Glatte Funktionen

Beispiel 8 (vgl. [135])



$$\frac{1}{(6 \Leftrightarrow 2x \Leftrightarrow y)^2}, \quad \Omega = [\Leftrightarrow 1, 1.6]^2$$

Für diese einfache Funktion kann das Programm VANI die Integraleinschlüsse relativ effizient berechnen. Da sich die Bereichsgrenze 1.6 im binären Zahlenformat nicht exakt darstellen läßt, ist die Verifikation über den Randbereichen vom Typ 0 und Typ 1 notwendig. In VANI wird die 1-dimensionale adaptive numerische Integration über den Randbereichen vom Typ 1 mit Hilfe von Gauß-Legendre-Formeln durchgeführt.

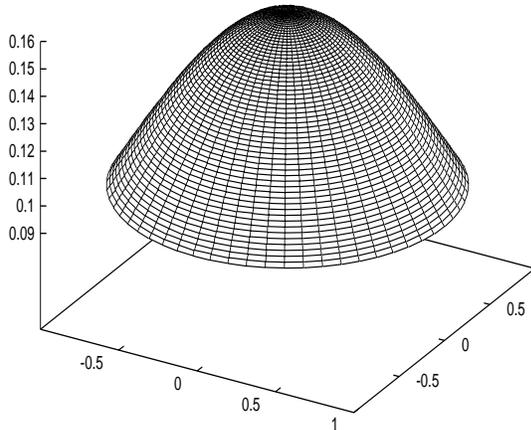
VANI — PGL :

ε_{abs}	$[I]$	$d([I])$	$\#f$		$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
			$\overset{\circ}{\Omega}$	$\overline{\Omega} \setminus \overset{\circ}{\Omega}$				
10^{-1}	$0.(\frac{45}{38})$	$6.37 \cdot 10^{-2}$	256	12	12	21	0.22	0
10^{-3}	$0.40(\frac{7}{5})$	$8.83 \cdot 10^{-4}$	435	24	14	25	0.33	0
10^{-6}	$0.40587(\frac{7}{6})$	$6.99 \cdot 10^{-7}$	881	37	20	37	0.50	0
10^{-9}	$0.40587646(\frac{2}{1})$	$7.76 \cdot 10^{-10}$	1237	52	27	51	0.71	0
10^{-12}	$0.40587646148(\frac{42}{36})$	$4.24 \cdot 10^{-13}$	2224	68	37	71	1.11	0
10^{-15}	$0.40587646148389(\frac{7}{5})$	$9.99 \cdot 10^{-16}$	9127	90	98	193	4.16	0
10^{-16}	$0.40587646148389(\frac{7}{5})$	$9.99 \cdot 10^{-16}$	4628	105	68	133	2.31	1

Die unerreichbare Genauigkeitsanforderung $\varepsilon_a = 10^{-15}$ kann Cubpack++ nicht rechtzeitig erkennen. Die dafür erzielte Näherung ist sogar schlechter als die Näherung bzgl. $\varepsilon_a = 10^{-12}$. In diesem Fall kann Cubpack++ den absoluten Fehler nicht angemessen abschätzen.

Cubpack++ :

ε_{abs}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	0.4058480953535884	$3.62 \cdot 10^{-3}$	37	0.01	in
10^{-3}	0.4058764521783579	$6.96 \cdot 10^{-6}$	259	0.01	in
10^{-6}	0.4058764613523304	$1.12 \cdot 10^{-7}$	407	0.02	in
10^{-9}	0.4058764614836839	$2.01 \cdot 10^{-10}$	999	0.02	in
10^{-12}	0.4058764614838951	$5.36 \cdot 10^{-13}$	2479	0.04	in
10^{-15}	0.4058764614832955	$4.51 \cdot 10^{-15}$	10000011*	1250.45	out

Beispiel 9

$$\frac{1}{2\pi} \cdot \exp\left(\frac{\Leftrightarrow x^2 \Leftrightarrow y^2}{2}\right),$$

$$\Omega = \{ (x, y) \mid x^2 + y^2 \leq 1 \}$$

Für dieses Beispiel wird in VANI das folgende transformierte Integral berechnet :

$$\int \int_{\Omega} \frac{1}{2\pi} \cdot \exp\left(\frac{\Leftrightarrow x^2 \Leftrightarrow y^2}{2}\right) dx dy = \int_0^{2\pi} \int_0^1 \frac{1}{2\pi} \cdot r \cdot \exp\left(\frac{\Leftrightarrow r^2}{2}\right) dr d\theta .$$

Da 2π keine Gleitkommazahl ist, wurde Verifikation über dem Randbereich von Typ 1 durchgeführt. Um die Genauigkeit $\varepsilon_a = 10^{-15}$ zu erfüllen, braucht das Programm VANI 162 Funktionsauswertungen über $\tilde{\Omega}$.

VANI — PGL :

ε_{abs}	$[I]$	$d([I])$	$\#f$		$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
			$\overset{\circ}{\Omega}$	$\overline{\Omega} \setminus \overset{\circ}{\Omega}$				
10^{-1}	$0.39\binom{4}{1}$	$1.56 \cdot 10^{-3}$	4	2	2	2	0.07	0
10^{-3}	$0.393\binom{50}{46}$	$2.94 \cdot 10^{-5}$	9	3	2	2	0.08	0
10^{-6}	$0.393469\binom{4}{0}$	$3.73 \cdot 10^{-7}$	16	5	2	2	0.08	0
10^{-9}	$0.393469340\binom{30}{26}$	$2.70 \cdot 10^{-11}$	36	6	2	2	0.13	0
10^{-12}	$0.393469340287\binom{5}{3}$	$1.70 \cdot 10^{-13}$	49	7	2	2	0.14	0
10^{-15}	$0.39346934028736\binom{7}{6}$	$7.77 \cdot 10^{-16}$	162	9	3	4	0.51	0
10^{-16}	$0.39346934028736\binom{7}{6}$	$8.33 \cdot 10^{-16}$	81	9	2	2	0.26	1

Ohne Hilfe der maximalgenauen Intervallarithmetik ist es schwierig, den Rundungsfehler richtig zu behandeln. Für $\varepsilon_a = 10^{-15}$ benötigt Cubpack++ mehr als 10^8 Funktionsaufrufe, und die dabei erzielte Näherung ist nur von der absoluten Genauigkeit 10^{-11} .

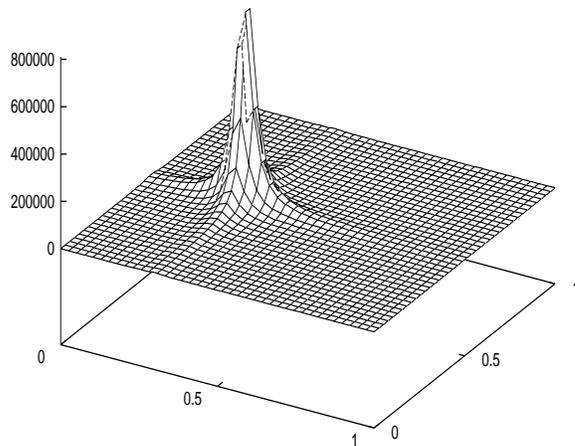
Cubpack++ :

ε_{abs}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	0.3934693402530517	$4.06 \cdot 10^{-9}$	36	0.01	in
10^{-3}	0.3934693402530517	$4.06 \cdot 10^{-9}$	36	0.01	in
10^{-6}	0.3934693402530517	$4.06 \cdot 10^{-9}$	36	0.01	in
10^{-9}	0.3934693402873665	$2.27 \cdot 10^{-14}$	220	0.01	in
10^{-12}	0.3934693402873665	$2.27 \cdot 10^{-14}$	220	0.02	in
10^{-15}	0.3934693402869063	$4.37 \cdot 10^{-15}$	100000009*	2022.90*	out

6.2.3 Funktionen mit Peak

Für diese Funktionsklasse benötigt das Programm VANI recht viele Funktionsauswertungen, und der Integrationsbereich Ω wird, insbesondere in der Nähe der Peaks, in sehr feine Teilbereiche zerlegt. Dies liegt daran, daß die Wertebereichseinschließungen der partiellen Ableitungen der zu integrierenden Funktionen, die mit Hilfe der Intervall-Taylorkoeffizienten berechnet werden, in der Nähe der Peaks i. a. recht grob sind. Dadurch wird eine längere Rechenzeit benötigt.

Beispiel 10



$$\frac{1}{((x \Leftrightarrow 0.3)^2 + a) \cdot ((y \Leftrightarrow 0.5)^2 + a)},$$

$$a = 0.001, \quad \Omega = [0, 1]^2$$

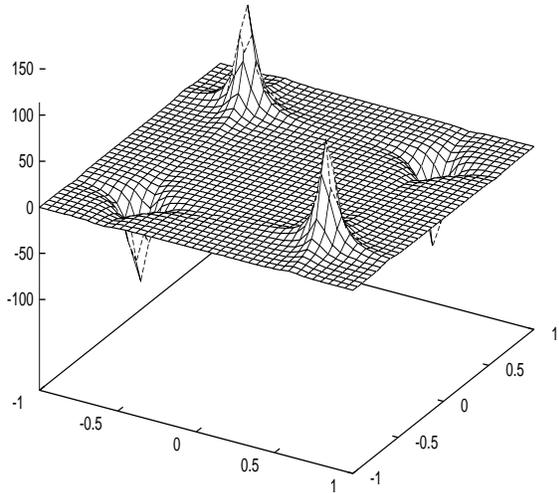
VANI — PGL :

ε_{rel}	$[I]$	$\frac{d([I])}{([I])}$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$\binom{9.5}{8.6} \cdot 10^3$	$9.85 \cdot 10^{-2}$	37662	386	771	20.62	0
10^{-3}	$9.0 \binom{25}{15} \cdot 10^3$	$9.87 \cdot 10^{-4}$	48193	470	939	25.80	0
10^{-6}	$9.0199 \binom{63}{53} \cdot 10^3$	$9.94 \cdot 10^{-7}$	77338	616	1231	38.33	0
10^{-9}	$9.0199580 \binom{43}{33} \cdot 10^3$	$9.94 \cdot 10^{-10}$	149212	981	1961	70.47	0
10^{-12}	$9.0199580383 \binom{73}{63} \cdot 10^3$	$9.99 \cdot 10^{-13}$	260672	1578	3155	119.36	0
10^{-15}	$9.0199580383677 \binom{6}{0} \cdot 10^3$	$5.65 \cdot 10^{-15}$	438349	2467	4933	196.59	1
10^{-16}	$9.0199580383677 \binom{5}{0} \cdot 10^3$	$4.84 \cdot 10^{-15}$	941414	6390	12779	459.11	1

In Cubpack++ wird nur die absolute Genauigkeit abgeschätzt, auch wenn der relative Fehler betrachtet ist.

Cubpack++ :

ε_{rel}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	$8.951260307816057 \cdot 10^3$	$7.68 \cdot 10^{+2}$	1073	0.03	in
10^{-3}	$9.019918759498198 \cdot 10^3$	$7.90 \cdot 10^{-0}$	2701	0.05	in
10^{-6}	$9.019958040323838 \cdot 10^3$	$8.44 \cdot 10^{-3}$	7585	0.10	in
10^{-9}	$9.019958038351622 \cdot 10^3$	$8.93 \cdot 10^{-6}$	20165	0.25	in
10^{-12}	$9.019958038367722 \cdot 10^3$	$9.01 \cdot 10^{-9}$	60347	0.76	in
10^{-15}	$9.019958038369139 \cdot 10^3$	$1.03 \cdot 10^{-10}$	100000085*	1228.08*	out

Beispiel 11

$$\begin{aligned} & \Leftrightarrow \frac{1}{((5x \Leftrightarrow 3)^2 + \frac{1}{10})(5y \Leftrightarrow 3)^2 + \frac{1}{10}} \\ & + \frac{1}{((5x + 3)^2 + \frac{1}{11})(5y \Leftrightarrow 3)^2 + \frac{1}{11}} \\ & \Leftrightarrow \frac{1}{((5x + 3)^2 + \frac{1}{10})(5y + 3)^2 + \frac{1}{10}} \\ & + \frac{1}{((5x \Leftrightarrow 3)^2 + \frac{1}{11})(5y + 3)^2 + \frac{1}{11}}, \\ & \Omega = [\Leftrightarrow 1, 1]^2 \end{aligned}$$

VANI — PGL :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$0.(\frac{80}{69})$	$9.98 \cdot 10^{-2}$	48767	1049	2097	104.36	0
10^{-3}	$0.74(\frac{2}{0})$	$9.90 \cdot 10^{-4}$	74630	1264	2527	136.62	0
10^{-6}	$0.74081(\frac{7}{5})$	$1.00 \cdot 10^{-6}$	137842	1682	3363	210.20	0
10^{-9}	$0.7408157(\frac{90}{88})$	$9.98 \cdot 10^{-10}$	231860	2848	5695	356.79	0
10^{-12}	$0.74081578940(\frac{5}{3})$	$9.97 \cdot 10^{-13}$	375320	4379	8757	559.00	0
10^{-15}	$0.740815789404(\frac{5}{4})$	$8.30 \cdot 10^{-14}$	604620	6437	12873	867.47	1
10^{-16}	$0.740815789404(\frac{5}{4})$	$8.29 \cdot 10^{-14}$	771750	8297	16593	1095.80	1

Für $\varepsilon_a = 10^{-15}$ liegt die von Cubpack++ erzielte Näherung außerhalb der entsprechenden Integraleinschließung. Diese unerreichbare Genauigkeit kann Cubpack++ nicht innerhalb vernünftiger Zeit ermitteln.

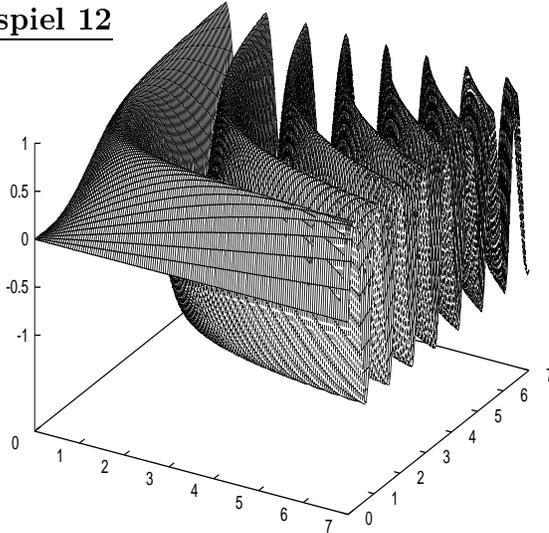
Cubpack++ :

ε_{abs}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	0.7404799805326308	$9.75 \cdot 10^{-2}$	3737	0.05	in
10^{-3}	0.7408154278349426	$9.93 \cdot 10^{-4}$	10693	0.15	in
10^{-6}	0.7408157894235132	$9.80 \cdot 10^{-7}$	28675	0.38	in
10^{-9}	0.7408157894045704	$9.94 \cdot 10^{-10}$	78551	1.06	in
10^{-12}	0.7408157894044466	$9.97 \cdot 10^{-13}$	204573	2.76	in
10^{-15}	0.7408157894043598	$1.45 \cdot 10^{-13}$	100000085*	1381.52	out

6.2.4 Oszillierende Funktionen

Im Vergleich zu Cubpack++ benötigt das Programm VANI für die betrachteten oszillierenden Funktionen relativ wenige Funktionsauswertungen, um die Integraleinschließungen zu berechnen, obwohl die Einschließungen mit Hilfe der Produktformeln errechnet sind, welche i. a. mehr Funktionsauswertungen benötigen als die vollsymmetrischen Kubaturformeln.

Beispiel 12



$$\sin(x \cdot y), \quad \Omega = [0, 7]^2$$

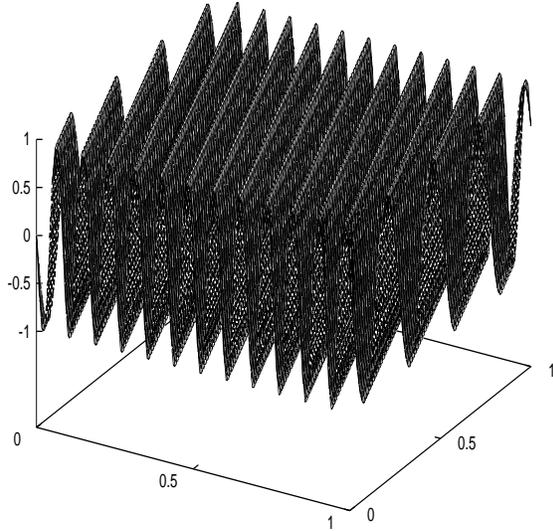
VANI — PGL :

ε_{abs}	$[I]$	$d([I])$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$4.(\frac{52}{45})$	$6.09 \cdot 10^{-2}$	426	7	13	3.44	0
10^{-3}	$4.48(\frac{9}{8})$	$6.13 \cdot 10^{-4}$	548	8	15	4.39	0
10^{-6}	$4.48860(\frac{93}{83})$	$3.83 \cdot 10^{-7}$	881	9	17	6.93	0
10^{-9}	$4.48860905(\frac{5}{3})$	$5.12 \cdot 10^{-10}$	1915	22	43	15.11	0
10^{-12}	$4.48860905381(\frac{6}{4})$	$8.15 \cdot 10^{-13}$	3005	31	61	23.63	0
10^{-15}	$4.48860905381(\frac{52}{49})$	$2.36 \cdot 10^{-13}$	6792	74	147	53.60	1
10^{-16}	$4.48860905381(\frac{52}{49})$	$2.39 \cdot 10^{-13}$	8620	90	179	68.07	1

Bzgl. der unerreichbaren Genauigkeit $\varepsilon_a = 10^{-15}$ besitzt die von dem Programm Cubpack++ erzielte Näherung einen absoluten Fehler, der größer als 10^{-12} ist.

Cubpack++ :

ε_{abs}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	4.488608608137418	$9.98 \cdot 10^{-2}$	3885	0.08	in
10^{-3}	4.488609055042244	$9.35 \cdot 10^{-4}$	9213	0.15	in
10^{-6}	4.488609053815131	$9.85 \cdot 10^{-7}$	22459	0.35	in
10^{-9}	4.488609053815022	$9.86 \cdot 10^{-10}$	66933	0.98	in
10^{-12}	4.488609053815053	$9.99 \cdot 10^{-13}$	190217	2.81	in
10^{-15}	4.488609053812540	$3.22 \cdot 10^{-13}$	100000085*	1456.80*	out

Beispiel 13

$$\cos(2\pi\beta + \alpha_1 x + \alpha_2 y),$$

$$\beta = 0.25, \alpha_1 = 75, \alpha_2 = 25,$$

$$\Omega = [0, 1]^2$$

Wegen des Rundungsfehlers ist bereits die relative Genauigkeitsanforderung $\varepsilon_r = 10^{-9}$ nicht erreichbar. Für die relative Fehlerbehandlung beinhaltet $\#f$ in VANI auch noch die vergeblichen Funktionsauswertungen. Trotzdem braucht VANI für diesen oszillierenden Integranden mindestens 10-mal weniger Funktionsauswertungen als Cubpack++.

VANI — PGL :

ε_{rel}	$[I]$	$\frac{d([I])}{([I])}$	$\#f$	$\#\Omega_j$	$\#\tilde{\Omega}_j$	sec	err
10^{-1}	$7 \cdot \binom{5}{2} \cdot 10^{-6}$	$2.19 \cdot 10^{-2}$	3004	16	31	23.46	0
10^{-3}	$7.34 \binom{30}{26} \cdot 10^{-6}$	$3.44 \cdot 10^{-5}$	6204	32	63	48.11	0
10^{-6}	$7.34279 \binom{72}{68} \cdot 10^{-6}$	$4.38 \cdot 10^{-8}$	12604	64	127	97.62	0
10^{-9}	$7.3427970 \binom{3}{0} \cdot 10^{-6}$	$3.13 \cdot 10^{-9}$	25204	127	253	195.62	1
10^{-12}	$7.3427970 \binom{3}{0} \cdot 10^{-6}$	$2.45 \cdot 10^{-9}$	156724	1023	2045	1242.54	1
10^{-15}	$7.3427970 \binom{3}{0} \cdot 10^{-6}$	$2.56 \cdot 10^{-9}$	438629	2866	5731	3688.27	1

Für dieses Beispiel hat Cubpack++ bereits mit $\varepsilon_r = 10^{-12}$ Schwierigkeit, den Rundungsfehler richtig zu berücksichtigen. Im Fall $\varepsilon_r = 10^{-15}$ braucht es sogar mehr als 10^9 Funktionsaufrufe, wobei keine Näherung geliefert werden konnte.

Cubpack++ :

ε_{rel}	I^*	$ I \Leftrightarrow I^* _{est}$	$\#f$	sec	in/out
10^{-1}	$7.342796914808474 \cdot 10^{-6}$	$7.12 \cdot 10^{-7}$	36297	0.51	in
10^{-3}	$7.342797011054294 \cdot 10^{-6}$	$7.30 \cdot 10^{-9}$	70929	0.99	in
10^{-6}	$7.342797011983157 \cdot 10^{-6}$	$7.34 \cdot 10^{-12}$	215081	3.02	in
10^{-9}	$7.342797011983646 \cdot 10^{-6}$	$7.34 \cdot 10^{-15}$	596625	8.51	in
10^{-12}	$7.342797011981981 \cdot 10^{-6}$	$7.19 \cdot 10^{-15}$	100000085*	1516.84	in
10^{-15}	$\Leftrightarrow \Leftrightarrow \Leftrightarrow$	$\Leftrightarrow \Leftrightarrow \Leftrightarrow$	1000000000*	1953.24	-

Kapitel 7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde das Programm VANI (s. Kapitel 5) für die ein- und zweidimensionale adaptive numerische Integration mit automatischer Ergebnisverifikation implementiert. Die Koppelung der ein- und zweidimensionalen automatischen Integratoren in einem Programm beruht darauf, daß für die zweidimensionalen Integrationsbereiche, deren Grenzen keine Gleitkommazahlen sind, das Integral über dem Bereichsrand mit Hilfe eines eindimensionalen adaptiven Integrationsverfahrens berechnet werden soll.

Mit Methoden der Verifikationsnumerik wurde bis jetzt noch kein zweidimensionaler automatischer Integrator zu allgemeinem Zweck implementiert. Diese Aufgabe besitzt eine wesentlich größere Komplexität als im Eindimensionalen, was daran liegt, daß es im Mehrdimensionalen eine große Zahl von Integrationsbereichen gibt, die sich durch affine Abbildungen nicht ineinandern überführen lassen. Wegen dieser Komplexität wird zur Zeit im Programm VANI die zweidimensionale adaptive numerische Integration auf Rechteckbereiche beschränkt. Integrale über Kreisen bzw. Dreiecken lassen sich durch nichtaffine Abbildungen in Integrale über Rechtecken transformieren. Diese Transformationen werden bei der Anwendung von VANI zur Zeit dem Benutzer überlassen.

Für die eindimensionale numerische Integration werden im Programm VANI die Gauß-Legendre-Formeln, die in [144] berechnet wurden, und die Clenshaw-Curtis-Formeln zur Verfügung gestellt. Die Clenshaw-Curtis-Formeln (s. Abschnitt 3.3.2) erwecken zwar seit langem das Interesse der Numeriker, wurden jedoch in der Verifikationsnumerik bisher noch nicht betrachtet. Für die zweidimensionale numerische Integration werden im Programm VANI außer den Produktformeln, die auf die vorhandenen Quadraturformeln zurückgreifen, noch die in [45] konstruierten vollsymmetrischen Kubaturformeln bereitgestellt.

Um die Approximationsfehler der Integrationsformeln zuverlässig abzuschätzen, werden im Programm VANI die von Peano 1913 aufgestellte und später von Sard 1952 verallgemeinerten Kerndarstellungen verwendet. Bei Anwendung der Kernmethode wird üblicherweise der Diskretisierungsfehler einer Integrationsformel anhand der L_1 -Norm der Kernfunktionen und der Maximumnorm der (partiellen) Ableitungen des Integranden

abgeschätzt. Die L_1 -Norm einer Kernfunktion wird auch als Fehlerkonstante bzw. Restgliedfaktor bezeichnet.

Im Rahmen dieser Arbeit wurden für die Berechnung der Fehlerkonstanten herangezogener Integrationsformeln verschiedene Überdeckungsmethoden, wie z. B. die Intervall-Riemann-Summe, die Bisektionsmethode und das komponentenweise Intervall-Newton-Verfahren usw., die die Nullstellen der Kernfunktionen explizit bzw. implizit berücksichtigen, untersucht. Für die eindimensionalen Kernfunktionen, welche auch in der Fehlerdarstellung einer Kubaturformel zu berücksichtigen sind, besitzt das erweiterte Intervall-Newton-Verfahren die beste Effektivität und für die zweidimensionalen Kernfunktionen leistet dies die von Hansen 1967 und Neumaier 1988 verallgemeinerte Methode (s. Abschnitte 2.3, 2.4, 3.4.3 und 4.3.4). Um den Rechenaufwand zu reduzieren wurden noch verschiedene Eigenschaften der Kernfunktionen, wie z. B. die Stetigkeit, die Differenzierbarkeit, die Vielfachheit der Nullstellen, die Symmetrie, die Vollsymmetrie, die lokale Definitheit und günstigere Darstellungsformeln untersucht bzw. hergeleitet (s. Abschnitte 3.4.1, 3.4.3, 4.3.3 und 4.3.4). Darüberhinaus wurde der Rechenaufwand durch Einsatz der Mittelwertform für die (partiellen) Ableitungen der Kernfunktionen drastisch verringert.

Die Maximumnorm der (partiellen) Ableitungen der zu integrierenden Funktionen, die bei der gewöhnlichen Numerik schwierig zu gewinnen ist, kann mit Hilfe der Differentiations- und Intervallarithmetik, nämlich mit den Intervall-Taylorkoeffizienten, zuverlässig abgeschätzt werden. Es werden die Wertebereichseinschließungen der (partiellen) Ableitungen berechnet. Für die Berechnung der eindimensionalen Taylorkoeffizienten steht das Modul **itaylor** zur Verfügung. Für die zweidimensionalen Funktionen wurden im Rahmen dieser Arbeit rekursive Formeln und Module zur Berechnung der zweidimensionalen Taylorkoeffizienten höherer Ordnungen hergeleitet bzw. implementiert (s. Abschnitt 2.5.2). Aus Effizienzgründen werden im Modul **ditaylor2** nur die Taylorkoeffizienten, die in dem linken oberen Dreieck der Matrix liegen, berechnet (s. Abschnitt 5.5.3).

Zur adaptiven Durchführung der numerischen Integration wird im Programm VANI die globale Strategie (s. Abschnitt 5.1.1) für die Bereichsverfeinerung angewendet und für die lokale Approximation werden adaptive Exaktheitsgrade zugelassen (s. Abschnitt 5.1.2). Für die Bereichsunterteilung kann zur Zeit bei Anwendung des Programms VANI ein zweidimensionales Rechteckgebiet wahlweise in 2 oder 4 Teilbereiche mit gleichem Volumen zerlegt werden. Dabei wird für die binäre Unterteilung im Zweidimensionalen die Unterteilungsrichtung mit Hilfe der ein- und zweidimensionalen Fehler festgelegt. Bei Anwendung von Nicht-Produktformeln werden gegebenenfalls noch die Intervall-Taylorkoeffizienten zur Festlegung der Unterteilungsrichtung eingesetzt (s. Abschnitt 5.2.1 und 5.2.2). Zur Registrierung der Informationen der Teilbereiche bietet VANI den partiell geordneten balancierten binären und ternären Baum zur Wahl an. (s. Abschnitt 5.5.2).

Um für eine reine absolute Genauigkeitsanforderung die numerische Integration effizient und zuverlässig durchzuführen, wird das im Programm VANI implementierte Integrationsverfahren in zwei Phasen durchgeführt. In der ersten Phase wird nur die Fehlereinschließung $[E]$ berücksichtigt. Erst nachdem $d([E]) \leq \varepsilon_a$ mit der absoluten Fehlerschranke ε_a erreicht ist, wird die Einschließung der Integralnäherung $[S]$ berechnet. Ist $d([S]) \geq \varepsilon_a$, so kann die geforderte absolute Genauigkeit wegen des Rundungsfehlers nicht erreicht werden. Ansonsten geht das Verfahren in die zweite Phase über. In dieser Phase wird stets

die Integraleinschließung $[I] = [S] + [E]$ betrachtet und das adaptive Integrationsverfahren wird fortgeführt, bis $d([I]) \leq \varepsilon_a$ gilt oder bis die maximale Anzahl der Teilbereiche bzw. der Funktionsaufrufe überschritten ist (s. Abschnitt 5.3.1). Bei einer zuverlässigen relativen Fehlerbehandlung soll jedoch stets die Bedingung $d([I]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ mit dem wahren Wert $[I]$ geprüft werden (s. Abschnitt 5.3.2). Die unerreichbare Genauigkeit kann durch $d([S]) \geq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ und $d([E]) \leq \max(\varepsilon_a, \varepsilon_r \cdot \langle [I] \rangle)$ erkannt werden.

Für Integrationsbereiche, deren Grenzen keine Gleitkommazahlen sind, ist die Verifikation über dem Bereichsrand erforderlich. Im Eindimensionalen werden die Teilintegrale über den Bereichsgrenzen direkt mit Hilfe der Intervall-Riemann-Summe verifiziert berechnet (s. Abschnitt 5.4.1). Hingegen lassen sich die Randbehandlung eines zweidimensionalen Rechteckgebiets in zwei Klassen einteilen (s. Abschnitt 5.4.2). Die Integrale über den Randbereichen vom Typ 0 werden analog zu dem eindimensionalen Fall mit Hilfe der Intervall-Riemann-Summe berechnet. Über den Randbereichen vom Typ 1 wird jedoch eine eindimensionale adaptive numerische Integration durchgeführt. Hierfür wird das Integral über dem inneren Integrationsbereich gemäß einer modifizierten Genauigkeit berechnet, damit die Genauigkeitsanforderungen möglichst gut erfüllt werden können.

Die Effizienz des in VANI implementierten adaptiven numerischen Integrationsverfahrens könnte noch erhöht werden, wenn z. B. die maximale Ordnung der Kernfunktionen der lokalen Integrationsformeln variiert würde, was mit hinreichend vielen Beispielen untersucht werden soll. Um die Effektivität des herangezogenen vollsymmetrischen Kubaturverfahren vollständig zu prüfen, sollen noch Integrationsformeln höherer Exaktheitsgrade konstruiert und ihre Fehlerkonstanten berechnet werden. Darüber hinaus läßt sich die Funktionalität des Programms durch weitergehende Arbeiten, wie z. B. die Behandlung von gewissen Singularitäten oder die automatische Durchführung der nichtaffinen Transformationen für eine breitere Klasse von Integrationsbereichen, erweitern. Für die zweidimensionale automatische Integration kann z. B. auch noch untersucht werden, wie die binäre und die 2^n -Unterteilung gemischt verwendet werden können.

Anhang A

Programm zur verifizierten Berechnung der Fehlerkonstante bzgl. der Peano-Kerne

```
PROGRAM Ks_XIN_MWF_sub;

USE mvi_ari;
USE i_ari;
USE x_real;

priority inn = =;
{*****}
{*} OPERATOR INN ( A : REAL; VAR B : INTERVAL ) INRIR : BOOLEAN;  {*}
{*****}
BEGIN                                     { operator --- in interior }
  INRIR:= ( B.INF < A ) AND ( A < B.SUP )
END;
{*****}
{*} FUNCTION power( x : INTERVAL; i : INTEGER ) : INTERVAL;  {*}
{*****}
VAR  r, s : INTERVAL;
BEGIN
  r := 1.0;
  s := x;
  WHILE i>0 DO
  BEGIN
    IF odd(i) THEN r := r*s;
    s := sqr(s);
    i := i div 2;
  END;
  power := r;
```

```

END; { power }
{***** E N D *****)
{-----}
{} CONST {}
{} n_max = 16; {}
{ n_max = 20; }
{} n_min = 2; {}
{} d_max = 31; {}
{} quadrature = 'G'; {}
{} in_data = 'GL_x_w_n2to24.dat'; {}
{} out_data = 'GL_ErrConst_n2to16.dat'; {}
{ in_data = 'CC_x_w_n2to20.dat'; }
{ out_data = 'CC_ErrConst_n2to20.dat2'; }
{} plus_infinity = maxreal; {}
{} minus_infinity = -maxreal; {}
{-----}
VAR w, y : IMATRIX[2..n_max, 1..n_max]; { weights and nodes }
rest : IVECTOR[2..n_max]; { to store error constant of }
{ highest order }
ErrP, ErrN : IMATRIX[2..n_max, 0..d_max]; { error constants }
n, s : INTEGER; { no. of nodes; order of kernel }
epsilon : REAL; { resolution --- diam. of a box }
first, last : REAL; { lb & ub of an integration-area }
cnt1, cnt2 : INTEGER; { no. of boxes; no. of integration }
{*****}
{*} FUNCTION positivDiff( x, y : INTERVAL; ) : INTERVAL; {*}
{*****}
VAR z : INTERVAL;
BEGIN
z := x - y; { z = x - y; if x >= y }
IF z.sup <= 0 THEN z := 0 { z = 0; if x < y }
ELSE IF z.inf < 0 THEN z.inf := 0;
positivDiff := z;
END;
{*****}
{*} FUNCTION dKernel(t : INTERVAL; lb : INTEGER) : INTERVAL; {*}
{*****}
VAR diff : IVECTOR[lb..n];
temp : INTERVAL;
i : INTEGER;
BEGIN { the first derivative of kernel }
FOR i := lb TO n DO { the functional part : S() }
IF s = 0 THEN
diff[i] := 0
ELSE BEGIN
diff[i] := positivDiff(y[n,i],t);

```

```

      IF (diff[i] <> 0) THEN
        IF (s=1) AND (y[n,i].inf < t.sup) THEN diff[i] := intval(-1,0)
        ELSE IF (s=1) THEN diff[i] := -1
        ELSE diff[i] := -s*power(diff[i], s-1)
      END;

temp := -power(1.0-t, s);           { the functional part : I() }

dKernel := ##( temp - FOR i := lb TO n SUM(w[n,i]*diff[i]) );
END;
{*****}
{*}  FUNCTION Kernel(t : INTERVAL; lb : INTEGER) : INTERVAL;  {*}
{*****}
VAR  diff      : IVECTOR[lb..n];
      temp, inv : INTERVAL;
      i         : INTEGER;
BEGIN                                     { function value of kernel }
  FOR i := lb TO n DO                   { the functional part : S() }
    BEGIN
      diff[i] := positivDiff(y[n,i],t);
      IF diff[i] <> 0 THEN
        IF (s=0) AND (y[n,i].inf < t.sup) THEN diff[i] := intval(0,1)
        ELSE IF (s=0) THEN diff[i] := 1
        ELSE diff[i] := power(diff[i], s)
      END;

      inv := 1.0/intval(s+1);           { the functional part : I() }
      temp := power(1.0-t, s+1);

      Kernel := ##( temp*inv - FOR i := lb TO n SUM(w[n,i]*diff[i]) );
    END;
  END;
{*****}
{*}  FUNCTION ddKernel(t : INTERVAL; lb : INTEGER) : INTERVAL;  {*}
{*****}
VAR  diff : IVECTOR[lb..n];
      temp : INTERVAL;
      i    : INTEGER;
BEGIN                                     { the second derivative of kernel }
  FOR i := lb TO n DO                   { the functional part : S() }
    IF (s = 0) OR (s = 1) THEN
      diff[i] := 0
    ELSE BEGIN
      diff[i] := positivDiff(y[n,i],t);
      IF (diff[i] <> 0) THEN
        IF (s=2) AND (y[n,i].inf < t.sup) THEN diff[i] := intval(0,1)
        ELSE IF (s=2) THEN diff[i] := 1
      END;
    END;
  END;

```

```

        ELSE diff[i] := s*(s-1)*power(diff[i], s-2)
    END;

    temp := s*power(1.0-t, s-1);           { the functional part : I() }

    ddKernel := ##( temp - FOR i := lb TO n SUM(w[n,i]*diff[i]) );
END;
{*****}
{*}   PROCEDURE IntegrateKernel( VAR PP, NN : IDOTPRECISION;  {*}
{*}                                     a, b : REAL;           {*}
{*}                                     lb : INTEGER );         {*}
{*****}
VAR xx      : IVECTOR[1b..n];
    diff, inv : INTERVAL;
    tmp1, tmp2 : INTERVAL;
    i         : INTEGER;
    Int       : IDOTPRECISION;           { subintegral }
BEGIN                                           { integrate kernel over [a,b] }
    cnt2 := cnt2 +1;
    inv := 1.0/intval(s+2);

    tmp1 := power(1.0-intval(a), s+2);       { the functional part : I() }
    tmp2 := power(1.0-intval(b), s+2);
    diff := positivDiff(tmp1,tmp2);

    FOR i := lb To n DO                       { the functional part : S() }
    BEGIN
        tmp1 := power(y[n,i]-a, s+1);
        tmp2 := power(y[n,i]-b, s+1);
        xx[i] := positivDiff(tmp1, tmp2);
    END;
                                           { subintegral over [a,b] }
    Int := #( diff*inv - FOR i := lb To n SUM(w[n,i]*xx[i]) );

    a := #<(Int.inf);  b := #>(Int.sup);     { add to the error constants }
    IF ( a >= 0) THEN PP := #( PP + Int )
    ELSE IF ( b <= 0) THEN NN := #( NN + Int )
    ELSE BEGIN
        PP := #( PP + intval(0, b) );
        NN := #( NN + intval(a, 0) );
    END;
END;
{*****}
{*} FUNCTION RangeOfdKernel( x : INTERVAL; lb : INTEGER ) : INTERVAL; {*}
{*****}
VAR m, dKm, ddK : INTERVAL;

```

```

EGIN                                     { (sub)range of derivative -- MWF }
  IF (diam(x) < 1) THEN
    BEGIN
      m := mid(x);   dKm := dKernel( m, lb );   m := x - m;
      ddK := ddKernel( x, lb );
      RangeOfdKernel := ##( dKm + ddK * m );
    END
  ELSE RangeOfdKernel := dKernel( x, lb );
END;
{*****}
{*} FUNCTION RangeOfKernel( x : INTERVAL; lb : INTEGER ) : INTERVAL; {*}
{*****}
VAR m, Km, dK : INTERVAL;
BEGIN                                     { (sub)range of kernel -- MWF }
  IF (diam(x) < 1) THEN
    BEGIN
      m := mid(x);   Km := Kernel( m, lb );   m := x - m;
      dK := RangeOfdKernel( x, lb );
      RangeOfKernel := ##( Km + dK * m );
    END
  ELSE RangeOfKernel := Kernel( x, lb );
END;
{*****}
{*}  PROCEDURE EncloseIntegral(VAR PP, NN : IDOTPRECISION;  {*}
{*}                               KV, x : INTERVAL);  {*}
{*****}
VAR d, KP, KN : INTERVAL;
BEGIN                                     { to enclose the subintgral }
  cnt1 := cnt1 +1;
  d := ##(x.sup-x.inf)*(s+1);             { base "area" }

  IF (0 INN KV) THEN
    BEGIN
      KP := intval( 0, KV.sup );           { positive range }
      KN := intval( KV.inf, 0 );           { negative range }
    END
  ELSE IF (KV.inf >= 0) THEN
    BEGIN
      KP := KV;   KN := 0;                 { positive only ! }
    END
  ELSE IF (KV.sup <= 0) THEN
    BEGIN
      KP := 0;   KN := KV;                 { negative only ! }
    END;

  PP := #( PP + KP * d );                  { add to error constants }

```

```

    NN := #( NN + KN * d );
END;
{*****}
{*}  PROCEDURE XINewton(VAR PP, NN : IDOTPRECISION;  {*}
{*}                                x : INTERVAL;      {*}
{*}                                lb : INTEGER );    {*}
{*****}
    forward;
{*****}
{*}  PROCEDURE ExtDivision( VAR  PP, NN : IDOTPRECISION; {*}
{*}                        infinity, dK : REAL;      {*}
{*}                        m, Km, x : INTERVAL;     {*}
{*}                        lb : INTEGER );          {*}
{*****}
VAR y : INTERVAL;
BEGIN                                { single step of extended division }
    y := infinity +* ( m - Km / dK );
    IF NOT ( y >> x ) THEN XINewton(PP, NN, y**x, lb);
END;
{*****}
{*}  PROCEDURE ExtensionPart( VAR  PP, NN : IDOTPRECISION; {*}
{*}                        m, Km, dK, x : INTERVAL;   {*}
{*}                        lb : INTEGER );           {*}
{*****}
BEGIN                                { extended division --- division by zero is allowed ! }
    IF Km.inf > 0 THEN                { case one }
    BEGIN
        IF dK.sup <> 0 THEN
            ExtDivision( PP, NN, minus_infinity, dK.sup, m, Km, x, lb );
        IF dK.inf <> 0 THEN
            ExtDivision( PP, NN, plus_infinity, dK.inf, m, Km, x, lb );
        END
    ELSE IF Km.sup < 0 THEN            { case two }
    BEGIN
        IF dK.inf <> 0 THEN
            ExtDivision( PP, NN, minus_infinity, dK.inf, m, Km, x, lb );
        IF dK.sup <> 0 THEN
            ExtDivision( PP, NN, plus_infinity, dK.sup, m, Km, x, lb );
        END
    ELSE BEGIN                        { result of division --- ( -infinity , +infinity ) }
        XINewton(PP, NN, intval(x.inf, m.sup), lb);{ bisection --- 1. part }
        XINewton(PP, NN, intval(m.inf, x.sup), lb);{ bisection --- 2. part }
    END;
END;
END;
{*****}
{*}  PROCEDURE XINewton(VAR PP, NN : IDOTPRECISION;  {*}

```



```

IW := RangeOfKernel( intval(a,b), lb );   { range over [a,b] -- MWF   }
IF NOT( 0 INN IW ) THEN                   { kernel is definit over [a,b] }
  IntegrateKernel(PP, NN, a, b, lb)
ELSE BEGIN                                { kernel has a changed sign }
  XINewton(PP, NN, intval(a,b), lb);
                                         { XIN step -- to find out all zeros }
  IF first <> last THEN                   { the last subarea over which }
    IntegrateKernel(PP, NN, first, last, lb); { kernel is definit }
  END;
END;
END;
{*****}
{*}  PROCEDURE FactorOverSubRegions( VAR PP, NN : IDOTPRECISION;  {*}
{*}                                     lb : INTEGER );           {*}
{*****}
VAR i      : INTEGER;
BEGIN      { error constants over [x_i.sup, x_(i+1).inf] }
  IF 0 < y[n,lb].inf THEN { over the first possible subinterval }
  BEGIN
    first := 0; last := y[n,lb].inf;
                                         { initial. bounds of integration-area }
    SubFactor(PP, NN, first, last, lb); { to find out all subareas for }
  END;                                       { integration of kernel }

  FOR i := lb TO n-1 DO { over [x_i.sup, x_(i+1).inf] }
  BEGIN
    first := y[n,i].sup; last := y[n,i+1].inf;
    SubFactor(PP, NN, first, last, i+1);
  END;

  IF y[n,n].sup < 1 THEN { over the last subinterval }
  BEGIN
    first := y[n,n].sup; last := 1;
    IntegrateKernel(PP, NN, first, last, n+1); { kernel is definit }
  END;
END;
END;
{*****}
{*} PROCEDURE EncloseSubIntegrals( VAR PP, NN : IDOTPRECISION; {*}
{*}                                     lb : INTEGER );           {*}
{*****}
VAR IW : INTERVAL;
      i : INTEGER;
BEGIN { to enclose the subintegrals over [x_i] }
  FOR i := lb TO n DO
  BEGIN
    IW := Kernel( y[n,i], i );
    EncloseIntegral( PP, NN, IW, y[n,i] );
  END;
END;

```

```

    END;
END;
{*****}
{*} PROCEDURE SymmetricSum( VAR PP, NN : IDOTPRECISION; ); {*}
{*****}
VAR temp : IDOTPRECISION;
BEGIN
    IF odd(s) THEN
        BEGIN
            PP := #( PP + PP );
            NN := #( NN + NN );
        END
    ELSE BEGIN
        temp := PP;
        PP := #( PP - NN );
        NN := #( NN - temp );
    END;
END;
END;
{*****}
{*} PROCEDURE WriteData; {*}
{*****}
VAR i, j, d : INTEGER;
    out_file : FILE OF INTERVAL;
BEGIN
    REWRITE(out_file, out_data);
    FOR i := n_min TO n_max DO
        BEGIN
            CASE quadrature OF
                'G' : d := 2*i-1;
                'C' : IF odd(i) THEN d := i ELSE d := i-1;
            END;
            FOR j := 0 TO d DO
                BEGIN
                    write(out_file, ErrP[i,j]);
                    write(out_file, ErrN[i,j]);
                END;
            END;
        END;
END;
END;
{*****}
{*} PROCEDURE CalculateFactor; {*}
{*****}
VAR PP, NN : IDOTPRECISION;
    lb : INTEGER;
BEGIN
    IF odd(n) THEN lb := trunc( (n+1)/2 )
    ELSE lb := trunc( n/2 ) + 1;

```

```

                                { here -- the positive nodes }
PP := #( intval(0) );   NN := PP;   { initialize the error constants }

EncloseSubIntegrals( PP, NN, lb );           { process over [x_i] }
FactorOverSubRegions( PP, NN, lb );
                                { process over [x_i.sup, x_(i+1).inf] }

SymmetricSum( PP, NN );           { concern the symmetry of kernel }
ErrP[n,s] := ##(PP);
ErrN[n,s] := ##(NN);
writeln('(',n,',',s,')', ErrP[n,s]);
writeln(' ', ErrN[n,s]); writeln;
writeln('PP-NN = ', ##(PP-NN)); writeln;
writeln('-----');
writeln('cnt1 = ',cnt1);
writeln('cnt2 = ',cnt2);
writeln;
END;
{*****}
{*} PROCEDURE ReadData;  {*}
{*****}
VAR i, j : INTEGER;
    in_file : FILE OF INTERVAL;
BEGIN
  RESET(in_file, in_data);           { to read the data }
  FOR i := n_min TO n_max DO
    CASE quadrature OF
      'G' :           { Gauss-Legendre rules }
      BEGIN
        FOR j := 1 TO i DO read(in_file, y[i,j]); { to read nodes }
        FOR j := 1 TO i DO read(in_file, w[i,j]); { to read weights }
        READ(in_file, rest[i]); { error constants of highest order }
      END;
      'C' :           { Clenshaw-Curtis rules }
      BEGIN
        FOR j := 1 TO i DO
          BEGIN
            read(in_file, y[i,j]);           { to read nodes }
            read(in_file, w[i,j]);           { to read weights }
          END;
        END;
      END;
    END;
  END;
END;
{***** Main Program *****}
VAR d : INTEGER;
BEGIN

```


Anhang B

Das Modul ditaylor2

```
MODULE ditaylor2;

USE GLOBAL i_ari;
USE GLOBAL mvi_ari;

{-----}
{ This module ditayor2 contains operators and functions for the }
{ recursive computation of Taylor coefficients for functions of }
{ two variables. }
{ ===== }
{ These operators and functions are: }
{ +,-(unary),+,-,*,/,sqr,exp,power,sin,cos }
{ }
{ There is a modification for the function "power". }
{-----}

{-----}
{ The data type of Taylor coefficients for functions of two variables }
{ is a 2-dimensional dynamic array of interval. }
{ A variable of type ditaylor2 represents the Taylor coefficients, i.e. }
{ u[0,0] = (u)_0,0, ..., u[i,j] = (u)_i,j }
{ All operands are assumed to have the same index ranges starting }
{ from [0,0]. This is never checked however ! }
{ }
{ In this Modul only the Taylor coefficients which lies in the upper }
{ left triangle of the array will be computed. (see the dissertation }
{ of Chin-Yun Chen) }
{-----}

FUNCTION UpperBound( a, b, i : INTEGER ) : INTEGER;
```

```

VAR m : INTEGER;
BEGIN
  IF (a<=b) THEN m := b - i
  ELSE BEGIN
    m := a - i;
    IF (b<m) THEN m := b;
  END;
  UpperBound := m;
END;

GLOBAL TYPE ditaylor2 = GLOBAL DYNAMIC ARRAY [*,*] OF INTERVAL;

{-----}
{ define a constant of type ditaylor2 by assigning a value }
{-----}

GLOBAL OPERATOR := ( VAR u : ditaylor2; c : INTERVAL );
VAR i,j,m : INTEGER;
BEGIN
  u[0,0]:= c;
  FOR i:= 1 TO ub(u,2) DO u[0,i]:= 0.0;
  FOR i:= 1 TO ub(u,1) DO
  BEGIN
    m := UpperBound( ub(u,1), ub(u,2), i );
    FOR j:= 0 TO m DO u[i,j]:= 0.0;
  END;
END;

GLOBAL OPERATOR := ( VAR u : ditaylor2; c : REAL );
VAR i,j,m : INTEGER;
BEGIN
  u[0,0]:= c;
  FOR i:= 1 TO ub(u,2) DO u[0,i]:= 0.0;
  FOR i:= 1 TO ub(u,1) DO
  BEGIN
    m := UpperBound( ub(u,1), ub(u,2), i );
    FOR j:= 0 TO m DO u[i,j]:= 0.0;
  END;
END;

GLOBAL OPERATOR := ( VAR u : ditaylor2; v : ditaylor2 );
VAR i,j,m : INTEGER;
BEGIN
  u[0,0]:= v[0,0];
  FOR i:= 1 TO ub(u,2) DO u[0,i]:= v[0,i];
  FOR i:= 1 TO ub(u,1) DO

```

```

    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO u[i,j]:= v[i,j];
    END;
END;

{-----}
{ define expansion point u0 and make u the independent variable      }
{-----}

GLOBAL PROCEDURE expand( VAR u : ditaylor2; u0 : INTERVAL; id : INTEGER );
BEGIN
  u:= u0;
  CASE id OF
    1 : IF ub(u,1) > 0 THEN u[1,0]:= 1.0;
    2 : IF ub(u,2) > 0 THEN u[0,1]:= 1.0;
  END;
END;

GLOBAL PROCEDURE expand( VAR u : ditaylor2; u0 : REAL; id : INTEGER );
BEGIN
  u:= u0;
  CASE id OF
    1 : IF ub(u,1) > 0 THEN u[1,0]:= 1.0;
    2 : IF ub(u,2) > 0 THEN u[0,1]:= 1.0;
  END;
END;

{-----}
{                               Unary +, - for ditaylor2              }
{-----}

GLOBAL OPERATOR + ( u : ditaylor2 )
                it2_plus : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2_plus[i,j]:= u[i,j];
    END;
  END;
END;

GLOBAL OPERATOR - ( u : ditaylor2 )
                it2_minus : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;

```

```

BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2_minus[i,j]:= -u[i,j];
    END;
  END;

  {-----}
  {                binary operators  ditaylor2 & ditaylor2                }
  {-----}

GLOBAL OPERATOR + ( u,v : ditaylor2 )
      it2it2_add : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2it2_add[i,j]:= u[i,j] + v[i,j];
    END;
  END;

GLOBAL OPERATOR - ( u,v : ditaylor2 )
      it2it2_sub : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2it2_sub[i,j]:= u[i,j] - v[i,j];
    END;
  END;

GLOBAL OPERATOR * ( u,v : ditaylor2 )
      it2it2_mul : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,k,l,m : INTEGER;
      tmp          : INTERVAL;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO
        BEGIN
          tmp:= 0;
          FOR k:= 0 TO i DO

```

```

        FOR l:= 0 TO j DO tmp:= tmp + u[k,l] * v[i-k,j-1];
        it2it2_mul[i,j]:= tmp;
    END;
END;
END;

```

```

GLOBAL OPERATOR / ( u,v : ditaylor2 )
        it2it2_div : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,k,l,m : INTEGER;
    w          : ditaylor2[0..ub(u,1), 0..ub(u,2)];
    tmp        : INTERVAL;
BEGIN
    FOR i:= 0 TO ub(u,1) DO
        BEGIN
            m := UpperBound( ub(u,1), ub(u,2), i );
            FOR j:= 0 TO m DO
                BEGIN
                    tmp:= 0;    w[i,j]:= 0.0;
                    FOR k:= 0 TO i DO
                        FOR l:= 0 TO j DO tmp:= tmp + w[k,l] * v[i-k,j-1];
                    END;
                    w[i,j]:= ( u[i,j] - tmp ) / v[0,0];
                    it2it2_div[i,j]:= w[i,j];
                END;
            END;
        END;
    END;
END;

```

```

{-----}
{          binary operators ditaylor2 & INTERVAL          }
{-----}

```

```

GLOBAL OPERATOR + ( u : ditaylor2; v : INTERVAL )
        it2i_add : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    it2i_add:= u;
    it2i_add[0,0]:= u[0,0] + v;
END;

```

```

GLOBAL OPERATOR - ( u : ditaylor2; v : INTERVAL )
        it2i_sub : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    it2i_sub:= u;
    it2i_sub[0,0]:= u[0,0] - v;
END;

```

```

GLOBAL OPERATOR * ( u : ditaylor2; v : INTERVAL )
        it2i_mul : ditaylor2[0..ub(u,1), 0..ub(u,2)];

```

```

VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2i_mul[i,j]:= u[i,j]*v;
    END;
  END;

GLOBAL OPERATOR / ( u : ditaylor2; v : INTERVAL )
  it2i_div : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2i_div[i,j]:= u[i,j] / v;
    END;
  END;

{-----}
{          binary operations  INTERVAL & ditaylor2          }
{-----}

GLOBAL OPERATOR + ( u : INTERVAL; v : ditaylor2 )
  iit2_add : ditaylor2[0..ub(v,1), 0..ub(v,2)];
BEGIN
  iit2_add:= v;
  iit2_add[0,0]:= u + v[0,0];
END;

GLOBAL OPERATOR - ( u : INTERVAL; v : ditaylor2 )
  iit2_sub : ditaylor2[0..ub(v,1), 0..ub(v,2)];
BEGIN
  iit2_sub:= -v;
  iit2_sub[0,0]:= u - v[0,0];
END;

GLOBAL OPERATOR * ( u : INTERVAL; v : ditaylor2 )
  iit2_mul : ditaylor2[0..ub(v,1), 0..ub(v,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(v,1) DO
    BEGIN
      m := ub(v,2)-i;
      IF m < 0 THEN m := 0;
    END;
  END;

```

```

        FOR j:= 0 TO m DO iit2_mul[i,j]:= u*v[i,j];
    END;
END;

GLOBAL OPERATOR / ( u : INTERVAL; v : ditaylor2 )
        iit2_div : ditaylor2[0..ub(v,1), 0..ub(v,2)];
VAR i,j,k,l,m : INTEGER;
    tmp      : INTERVAL;
    w        : ditaylor2[0..ub(v,1), 0..ub(v,2)];
BEGIN
    FOR i:= 0 TO ub(v,1) DO
        BEGIN
            m := ub(v,2)-i;
            IF m < 0 THEN m := 0;
            FOR j:= 0 TO m DO
                IF (i=0) AND (j=0) THEN
                    w[i,j]:= u / v[0,0]
                ELSE BEGIN
                    w[i,j]:= 0.0;    tmp:= 0;
                    FOR k:= 0 TO i DO
                        FOR l:= 0 TO j DO tmp:= tmp + w[k,l] * v[i-k,j-l];
                    END;
                    w[i,j]:= -tmp / v[0,0];
                END;
            END;
        END;
        iit2_div:= w;
    END;
END;

{-----}
{          binary operators  ditaylor2 & REAL          }
{-----}

GLOBAL OPERATOR + ( u : ditaylor2; v : REAL )
        it2r_add : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    it2r_add:= u;
    it2r_add[0,0]:= u[0,0] + v;
END;

GLOBAL OPERATOR - ( u : ditaylor2; v : REAL )
        it2r_sub : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    it2r_sub:= u;
    it2r_sub[0,0]:= u[0,0] - v;
END;

GLOBAL OPERATOR * ( u : ditaylor2; v : REAL )

```

```

        it2r_mul : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2r_mul[i,j]:= u[i,j]*v;
    END;
  END;

GLOBAL OPERATOR / ( u : ditaylor2; v : REAL )
        it2r_div : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO it2r_div[i,j]:= u[i,j] / v;
    END;
  END;

{-----}
{          binary operators      REAL & ditaylor2          }
{-----}

GLOBAL OPERATOR + ( u : REAL; v : ditaylor2 )
        rit2_add : ditaylor2[0..ub(v,1), 0..ub(v,2)];
BEGIN
  rit2_add:= v;
  rit2_add[0,0]:= u + v[0,0];
END;

GLOBAL OPERATOR - ( u : REAL; v : ditaylor2 )
        rit2_sub : ditaylor2[0..ub(v,1), 0..ub(v,2)];
BEGIN
  rit2_sub:= -v;
  rit2_sub[0,0]:= u - v[0,0];
END;

GLOBAL OPERATOR * ( u : REAL; v : ditaylor2 )
        rit2_mul : ditaylor2[0..ub(v,1), 0..ub(v,2)];
VAR i,j,m : INTEGER;
BEGIN
  FOR i:= 0 TO ub(v,1) DO
    BEGIN
      m := ub(v,2)-i;

```

```

        IF m < 0 THEN m := 0;
        FOR j:= 0 TO m DO rit2_mul[i,j]:= u*v[i,j];
    END;
END;

GLOBAL OPERATOR / ( u : REAL; v : ditaylor2 )
                rit2_div : ditaylor2[0..ub(v,1), 0..ub(v,2)];
VAR i,j,k,l,m : INTEGER;
    tmp       : INTERVAL;
    w         : ditaylor2[0..ub(v,1), 0..ub(v,2)];
BEGIN
    FOR i:= 0 TO ub(v,1) DO
        BEGIN
            m := ub(v,2)-i;
            IF m < 0 THEN m := 0;
            FOR j:= 0 TO m DO
                IF (i=0) AND (j=0) THEN
                    w[i,j]:= u / v[0,0]
                ELSE BEGIN
                    w[i,j]:= 0.0;    tmp:= 0;
                    FOR k:= 0 TO i DO
                        FOR l:= 0 TO j DO tmp:= tmp + w[k,l] * v[i-k,j-l];
                    END;
                    w[i,j]:= -tmp / v[0,0];
                END;
            END;
        END;
        rit2_div:= w;
    END;

{-----}
{               elementary functions for ditaylor2               }
{-----}

GLOBAL FUNCTION sqr( u : ditaylor2 )
                : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,k,l,m,n, up : INTEGER;
    tmp       : INTERVAL;
    w         : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    FOR i:= 0 TO ub(u,1) DO
        BEGIN
            up := UpperBound( ub(u,1), ub(u,2), i );
            FOR j:= 0 TO up DO
                IF (i=0) AND (j=0) THEN w[0,0]:=sqr(u[0,0])
                ELSE IF odd(i) THEN { case 1 : i odd }
                    BEGIN
                        tmp:= 0.0;

```

```

    IF (i>1) THEN m:= (i-1) DIV 2 ELSE m:=0;
    FOR k:= 0 TO m DO
    FOR l:= 0 TO j DO tmp:= tmp + u[k,l] * u[i-k,j-l];
    w[i,j]:= 2*tmp;
END
ELSE IF odd(j) THEN                                { case 2 : i even and j odd }
BEGIN
    tmp:= 0.0;
    IF (i>=2) THEN
    BEGIN
        IF (i>2) THEN m:= (i-2) DIV 2 ELSE m:=0;
        FOR k:= 0 TO m DO
        FOR l:= 0 TO j DO tmp:= tmp + u[k,l] * u[i-k,j-l];
        END;
    IF (i>0) THEN m:= i DIV 2 ELSE m:=0;
    IF (j>1) THEN n:= (j-1) DIV 2 ELSE n:=0;
    FOR l:= 0 TO n DO tmp:= tmp + u[m,l] * u[m,j-l];
    w[i,j]:= 2*tmp;
    END
ELSE                                                    { case 3 : i even and j even }
BEGIN
    tmp:= 0.0;
    IF (i>=2) THEN
    BEGIN
        IF (i>2) THEN m:= (i-2) DIV 2 ELSE m:=0;
        FOR k:= 0 TO m DO
        FOR l:= 0 TO j DO tmp:= tmp + u[k,l] * u[i-k,j-l];
        END;
    IF (j>=2) THEN
    BEGIN
        IF (i>0) THEN m:= i DIV 2 ELSE m:=0;
        IF (j>2) THEN n:= (j-2) DIV 2 ELSE n:=0;
        FOR l:= 0 TO n DO tmp:= tmp + u[m,l] * u[m,j-l];
        END;
    IF (j<2) THEN m:= i DIV 2;
    IF (j>0) THEN n:= j DIV 2 ELSE n:=0;
    w[i,j]:= 2*tmp + sqr(u[m,n]);
    END;
END;
END;
sqr:= w;
END;

GLOBAL FUNCTION power( u : ditaylor2; a : INTERVAL )
                        : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,k,l,m : INTEGER;
    uu        : INTERVAL;

```

```

    tmp      : INTERVAL;
    w        : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
  FOR i:= 0 TO ub(u,1) DO
  BEGIN
    m := UpperBound( ub(u,1), ub(u,2), i );
    FOR j:= 0 TO m DO
      IF (i=0) AND (j=0) THEN
        w[i,j]:= 1.0
      ELSE BEGIN
        w[i,j]:= 0.0;    tmp:= 0.0;
        FOR k:= 0 TO i DO
          FOR l:= 0 TO j DO
            BEGIN
              IF (i=k) AND (j=l) THEN uu := 0.0
              ELSE IF ((i+j-k-l)=1) THEN uu := (a*(i+j-k-l)-k-l) *
                u[i-k,j-l]
              ELSE uu:= (a*(i+j-k-l)-k-l) * u[i-k,j-l] *
                power(u[0,0], intval(i+j-k-l-1));
              tmp:= tmp + w[k,l] * uu;
            END;
          w[i,j]:= tmp / (i+j);
        END;
      END;
    END;

  FOR i:= 0 TO ub(u,1) DO
  BEGIN
    m := UpperBound( ub(u,1), ub(u,2), i );
    FOR j:= 0 TO m DO
      IF (a-i-j <> 0) THEN w[i,j]:= w[i,j] * power(u[0,0], a-i-j);
    END;
    power:= w;
  END;

GLOBAL FUNCTION power( u : ditaylor2; a : REAL )
                        : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
  power:= power( u, INTVAL(a) );
END;

GLOBAL FUNCTION exp( u : ditaylor2 )
                  : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR i,j,k,l,m : INTEGER;
    uu,tmp     : INTERVAL;
    w          : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN

```

```

FOR i:= 0 TO ub(u,1) DO
BEGIN
  m := UpperBound( ub(u,1), ub(u,2), i );
  FOR j:= 0 TO m DO
    IF (i=0) AND (j=0) THEN
      w[i,j]:= exp( u[0,0] )
    ELSE BEGIN
      w[i,j]:= 0.0;   tmp:= 0.0;
      FOR k:= 0 TO i DO
        FOR l:= 0 TO j DO
          BEGIN
            uu:= (i+j-k-l) * u[i-k,j-l];
            tmp:= tmp + w[k,l] * uu;
          END;
        w[i,j]:= tmp / (i+j);
      END;
    END;
  exp:= w;
END;

GLOBAL PROCEDURE sin_cos( u : ditaylor2; VAR w_sin,w_cos : ditaylor2 );
VAR i,j,k,l,m : INTEGER;
    uu        : INTERVAL;
    tmp_sin   : INTERVAL;
    tmp_cos   : INTERVAL;
BEGIN
  FOR i:= 0 TO ub(u,1) DO
    BEGIN
      m := UpperBound( ub(u,1), ub(u,2), i );
      FOR j:= 0 TO m DO
        IF (i=0) AND (j=0) THEN
          BEGIN
            w_sin[i,j]:= sin( u[0,0] );
            w_cos[i,j]:= cos( u[0,0] );
          END
        ELSE BEGIN
          w_sin[i,j]:= 0.0;   tmp_sin:= 0.0;
          w_cos[i,j]:= 0.0;   tmp_cos:= 0.0;
          FOR k:= 0 TO i DO
            FOR l:= 0 TO j DO
              BEGIN
                uu:= (i+j-k-l) * u[i-k,j-l];
                tmp_sin:= tmp_sin + w_cos[k,l] * uu;
                tmp_cos:= tmp_cos + w_sin[k,l] * uu;
              END;
            w_sin[i,j]:= tmp_sin / (i+j);
          END;
        END;
      END;
    END;
  END;

```

```
        w_cos[i,j]:= - tmp_cos / (i+j);
    END;
END;

GLOBAL FUNCTION sin( u : ditaylor2 )
                : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR w_sin,w_cos : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    sin_cos( u, w_sin, w_cos );
    sin:= w_sin;
END;

GLOBAL FUNCTION cos( u : ditaylor2 )
                : ditaylor2[0..ub(u,1), 0..ub(u,2)];
VAR w_sin,w_cos : ditaylor2[0..ub(u,1), 0..ub(u,2)];
BEGIN
    sin_cos( u, w_sin, w_cos );
    cos:= w_cos;
END;

END.
```

Literaturverzeichnis

- [1] Abramowski, S. und Müller, H. M. *Geometrisches Modellieren*. B. I. Wissenschaftsverlag, Mannheim, 1991.
- [2] Akrivis, G. *Fehlerabschätzungen bei der numerischen Integration in einer und mehreren Dimensionen*. Dissertation, Universität München, 1982.
- [3] Akrivis, G. und Förster, K.-J. On the Definiteness of Quadrature Formulae of Clenshaw-Curtis Type. *Computing*, 33:363–366, 1984.
- [4] Alefeld, G. und Grigoriff, R. D. *Fundamentals of Numerical Computation (Computer-Oriented Numerical Analysis)*. Springer, Wien, 1980.
- [5] Alefeld, G. und Herzberger, J. *Einführung in die Intervallrechnung*. Bibliographisches Institut, Mannheim / Wien / Zürich, 1974.
- [6] Alefeld, G. und Herzberger, J. *Introduction to Interval Computation*. Academic Press, New York, 1983.
- [7] Allgower, E. L. und Georg, K. *Numerical Continuation Methods*. Springer-Verlag, Berlin, Heidelberg, 1990.
- [8] Andrade, M. V. A., Comba, J. L. D. und Stolfi, J. Affine Arithmetic (draft). Submitted to Interval'94, 1994.
- [9] Aumann, G. und Spitzmüller, K. *Computerorientierte Geometrie*. B. I. Wissenschaftsverlag, Mannheim, Leipzig, Wien, Zürich, 1. Auflage, 1993.
- [10] Barnhill, R. E. An Error Analysis for Numerical Multiple Integration I. *Math. Comput.*, 22:98–109, 1968.
- [11] Barnhill, R. E. An Error Analysis for Numerical Multiple Integration II. *Math. Comput.*, 22:286–292, 1968.
- [12] Barnhill, R. E. und Little, F. F. Adaptive Triangular Cubatures. *Rocky Mt. J. Math.*, 14:53–75, 1984.
- [13] Barnhill, R. E. und Nielson, G. M. An Error Analysis for Numerical Multiple Integration III. *Math. Comput.*, 24(110):301–314, 1970.
- [14] Barnhill, R. E. und Pilcher, D. T. Sard Kernels for Certain Bivariate Cubatures. *Comm. ACM*, 16(9):567–570, 1973.

- [15] Barnhill, R. E. und Watson, S. H. Numerical Multiple Integration. *The mathematics of surfaces III, Proc. 3rd Conf. Oxford/UK 1998, Inst. Math. Appl. Conf. Ser., New Ser. 23, 1998*, S. 49–69, 1989.
- [16] Bauch, H., Jahn, K. U., Oelschlägel, D., Süsse, H. und Wiebigke, V. *Intervallmathematik*. BSB B. G. Teubner, Leipzig, 1. Auflage, 1987.
- [17] Berntsen, J., Espelid, T. O. und Genz, A. Algorithm 698: DCUHRE An Adaptive Multidimensional Integration Routine for a Vector of Integrals. *ACM Trans. Math. Software*, 17:452–456, 1991.
- [18] Berntsen, J., Espelid, T. O. und Genz, A. An Adaptive Algorithm for the Approximate Calculation of multiple Integrals. *ACM Trans. Math. Software*, 17:437–451, 1991.
- [19] Berntsen, J., Espelid, T. O. und Sørøvik, T. On the subdivision strategy in adaptive quadrature algorithms. *J. Comput. Appl. Math.*, 35(1-3):119–132, 1991.
- [20] Bojanov, B. D., Hakopian, H. A. und Sahakian, A. A. *Spline Functions and Multivariate Interpolations*. Kluwer Academic Publishers, Dordrecht, 1993.
- [21] Bonk, T. A new Algorithm for Multi-Dimensional Adaptive Numerical Quadrature. In W. Hackbusch, Hrsg., *Adaptive Methods - algorithms, theory and applications*, S. 54–68. 1993.
- [22] Braß, H. *Quadraturverfahren*. Vandenhoeck & Ruprecht, Göttingen, 1. Auflage, 1977.
- [23] Braß, H. Error Bounds based on Approximation Theory. In T. O. Espelid; A. Genz, Hrsg., *Numerical Integration*, S. 147–163. Kluwer Academic Publishers, 1992.
- [24] Braß, H. Error Estimation for the Clenshaw-Curtis Quadrature Method. In U. Wannagat, Hrsg., *Abh. Braunsch. Wiss. Ges.*, S. 45–53. Verlag Erich Goltze, 1992.
- [25] Braß, H. Bounds for peano kernels. In H. Braß; G. Hämmerlin, Hrsg., *Numerical Integration IV*, S. 39–55. Birkhäuser, 1993.
- [26] Braß, H. und Hämmerlin, G. *Numerical Integration III, ISNM 85*. Birkhäuser Verlag, Basel, 1988.
- [27] Braß, H. und Hämmerlin, G. *Numerical Integration IV, ISNM 112*. Birkhäuser Verlag, Basel, 1993.
- [28] Brass, H. und Förster, K.-J. On the estimation of linear functionals. *Analysis*, 7:237–258, 1987.
- [29] Bruckhaus, W. *Stoppregeln für Quadraturverfahren*. Dissertation, Fernuniversität, März 1982.
- [30] Burden, R. L., Faires, J. D. und Reynolds, A. C. *Numerical Analysis*. Prindle, Weber & Schmidt, Boston, 2. Auflage, 1981.

- [31] Cariño, R., Robinson, I. und de Doncker, E. Adaptive cubature over a collection of triangles using the d -transformation. *J. of Appl. Math.*, 50:171–183, 1994.
- [32] Cools, R. A survey of methods for constructing cubature formulae. In T. O. Espelid; A. Genz, Hrsg., *Numerical Intergration*, S. 1–24. Dordrecht, 1992.
- [33] Cools, R. und Haegemans, A. CUBPACK: PROGRESS REPORT. In T. O. Espelid; A. Genz, Hrsg., *Numerical Integration*, S. 305–315. Kluwer Academic Publisher, 1992.
- [34] Cools, R., Laurie, D. und Pluym, L. A User Manual for **cubpack++** Version 1.1. Technischer Bericht, Dept. of Computer Science, K. U. Leuven, Belgium, 1997.
- [35] Cools, R., Laurie, D. und Pluym, L. Algorithm 764: Cubpack++: A C++ package for Automatic Two-Dimensional Cubature. *ACM Trans. Math. Software*, 23(1):1–15, 1997.
- [36] Cools, R. und Maerten, B. A Hybrid Subdivision Strategy for Adaptive Integration Routines. *J. Univers. Comput. Science*, 4(5):486–500, 1998.
- [37] Corliss, G. F. Performance of self-validating adaptive quadrature. In P. Keast; G. Fairweather, Hrsg., *Numerical Integration*, S. 239–259. D. Reidel Publishing Company, 1987.
- [38] Corliss, G. F. und Rall, L. B. Adaptive, Self-Validating Numerical Quadrature. *SIAM J. Stat. Comput.*, 8(5):831–847, 1987.
- [39] Cranley, R. und Patterson, T. N. L. On the automatic numerical evaluation of definite integrals. *Comput. J.*, 14:189–198, 1971.
- [40] Davis, P. J. und Rabinowitz, P. *Methods of Numerical Integration*. Academic Press, New York, 2. Auflage, 1984.
- [41] de Boor, Carl. CADRE: An Algorithm for Numerical Quadrature. In J. R. Rice, Hrsg., *Mathematical Software*, S. 417–449. Academic Press, 1971.
- [42] de Boor, Carl. On Writing an Automatic Integration Algorithm. In J. R. Rice, Hrsg., *Mathematical Software*, S. 201–209. Academic Press, 1971.
- [43] de Doncker, E. und Robinson, I. An Algorithm for Automatic Integration Over a Triangle Using Nonlinear Extrapolation. *ACM. Trans. Math. Software*, 10:1–16, 1984.
- [44] Delves, L. M. On Adaptive Variable-Order Quadrature Routines. *Proc. 5th Manitoba Conf. numer. Math., Winnipeg, 1975*, S. 257–271, 1976.
- [45] Dietrich, S. Verifizierte Konstruktion vollsymmetrischer Kubaturformeln mit geringer Stützstellenanzahl. Diplomarbeit, Universität Karlsruhe, 1995.
- [46] Eiermann, M. C. Automatic, Guaranteed Integration of Analytic Functions. *BIT*, 29:270–282, 1989.

- [47] Engels, H. *Numerical Quadrature and Cubature*. Academic Press, New York, 1980.
- [48] Espelid, T. O. DQAIN: An algorithm for adaptive quadrature over a collection of finite intervals. In T. O. Espelid; A. Genz, Hrsg., *Numerical Integration*, S. 341–342. Kluwer Academic Publishers, 1991.
- [49] Espelid, T. O. und Genz, A. *Numerical Integration*. Kluwer Academic Publishers, Dordrecht, 1992.
- [50] Espelid, T. O. und Genz, A. DECUHR an algorithm for automatic integration of singular functions over a hyperrectangular region. *Numer. Algorithms*, 8(2-4):201–220, 1994.
- [51] Espelid, T. O. und Sørenvik, T. A Discussion of a New Error Estimate for Adaptive Quadrature. *BIT*, 29(2):283–294, 1989.
- [52] Favati, P., Lotti, G. und Romani, F. Peano kernel behaviour and error bounds for symmetric quadrature formulas. *J. Comput. Math. Appl.*, 29(6):27–34, 1995.
- [53] Fischer, H.-C. *Schnelle automatische Differentiation, Einschließungsmethode und Anwendungen*. Dissertation, Universität Karlsruhe, 1990.
- [54] Garloff, J. Convergent Bounds for the Range of Multivariate Polynomials. In K. Nickel, Hrsg., *Interval Mathematics 1985*, S. 37–56, Berlin, 1985.
- [55] Gautschi, W. Remainder Estimates for Analytic Functions. In T. O. Espelid; A. Genz, Hrsg., *Numerical Integration*, S. 133–145. Kluwer Academic Publishers, 1992.
- [56] Genz, A. C. und Malik, A. A. Remark on algorithm 006 : An adaptive algorithm for numerical integration over an N-dimensional rectangular region. *J. Comput. Appl. Math.*, 6(4):295–299, 1980.
- [57] Gerald, F. *Kurven und Flächen im Computer Aided Geometric Design*. Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden, 1994.
- [58] Golub, G. und Ortega, J. M., Hrsg. *Scientific Computing*. B. G. Teubner, Stuttgart, 1996.
- [59] Golub, G. H. and Welsch, J. H. Calculation of Gauss quadrature rules. *Math.-Comp.*, 23:221–230, 1969.
- [60] Hairer, E., Nørsett, S. P. und Wanner, G. *Solving Ordinary Differential Equations I*. Springer-Verlag, Berlin, 1. Auflage, 1987.
- [61] Hammer, R., Hocks, M., Kulisch, U. und Ratz, D. *Numerical Toolbox for Verified Computing I*. Springer, Berlin, 1993.
- [62] Hanke, W. Über ein zeitoptimales Suchverfahren mit Anwendung auf die numerische Integration. *ZAMM*, 60:T 294–T 295, 1980.
- [63] Hanke, W. Die optimale Sektion bei adaptiven Integrationsverfahren mit globaler Strategie. *ZAMM*, 62:T 327–T 329, 1982.

- [64] Hanke, W. und Kaussen, R.D. "RAKU ein Verfahren zur adaptiven Kubatur". Technischer Bericht 1622, Kernforschungsanlage Jülich GmbH, Zentralinstitut für Angewandte Mathematik, Jülich, BRD, 1979.
- [65] Hansen, E. On Solving Systems of Equations Using Interval Arithmetic. *Math. Comput.*, 22:374–384, 1968.
- [66] Hansen, E. *Topics in Interval Analysis*. Oxford University Press, London, 1969.
- [67] Hansen, E. Interval Forms of Newtons Method. *Computing*, 20:153–163, 1978.
- [68] Heindl, G. Inclusions of the range of functions and their derivatives. In Kaucher, E., Markov, S. M. und Mayer, G., Hrsg., *Computer Arithmetic, scientific computation and mathematical modelling*, S. 229–238. Scientific Publishing Co., 1991.
- [69] Heindl, G. Zur numerischen Einschließung der Werte von Peanofunktionalen. *ZAMM*, 75(Suppl. 2):637–638, 1995.
- [70] Hensen, R. *Über mehrdimensionale Interpolation und Kubatur*. Dissertation, Technische Hochschule Aachen, Februar 1988.
- [71] Herbort, S. Ein komponentenreduzierendes Branch-and-Prune-Verfahren zur verifizierten Lösung nichtlinearer Gleichungssysteme. Diplomarbeit, Universität Karlsruhe, 1995.
- [72] Herbort, S. und Ratz, D. "Improving the Efficiency of a Nonlinear-System-Solver Using a Componentwise Newton Method". Technischer Bericht 2/1997, Institut für Angewandte Mathematik, Universität Karlsruhe (TH), 1997.
- [73] Heuser, H. *Funktionalanalysis*. B. G. Teubner, Stuttgart, 1992.
- [74] Heuser, H. *Lehrbuch der Analysis*. B. G. Teubner, Stuttgart, 1992.
- [75] Hosaka, M. *Modeling of Curves and Surfaces in CAD/CAM*. B. G. Teubner, Berlin, Heidelberg, 1992.
- [76] Hoschek, J. und Lasser, D. *Grundlagen der geometrischen Datenverarbeitung*. B. G. Teubner, Stuttgart, 2. Auflage, 1992.
- [77] Ivanovs, L. *Über die Definitheit der Restfunktionale bestimmter Quadraturformeln p-ter Stufe Nach Euler-Maclaurin*. Dissertation, Hannover, Oktober 1984.
- [78] Kahaner, D. K. Comparison of Numerical Quadrature Formulas. In J. R. Rice, Hrsg., *Mathematical Software*, S. 229–259. Academic Press, 1971.
- [79] Kahaner, D. K. und Rechar, O. W. TWODQD an adaptive routine for two-dimensional integration. *J. Comput. Appl. Math.*, 17:215–234, 1987.
- [80] Kaucher, E., Kulisch, U. und Ullrich, Ch. *Computerarithmetic*. B. G. Teubner, Stuttgart, 1987.

- [81] Kaucher, E., Markov, S. M. und Mayer, G. *Computer Arithmetic, scientific computation and mathematical modelling*. Scientific Publishing Co., IMACS, 1991.
- [82] Keast, P. und Fairweather, G. *Numerical Integration*. D. Reidel Publishing Company, Dordrecht, 1987.
- [83] Kelch, R. *Ein adaptives Verfahren zur numerischen Quadratur mit automatischer Ergebnisverifikation*. Dissertation, Universität Karlsruhe, 1989.
- [84] Klatte, R., Kulisch, U., Neaga, M., Ratz, D. und Ullrich, Ch. *PASCAL-XSC*. Springer-Verlag, 1991.
- [85] Klein, W. *Zur Einschließung der Lösung von linearen und nichtlinearen Fredholm'schen Integralgleichungssystemen zweiter Art*. Dissertation, Universität Karlsruhe, 1990.
- [86] Knuth, D. E. *The Art of Computer Programming, Vol. 3*. Addison, Wesley, Mass., 1973.
- [87] Krämer, W., Kulisch, U. und Lohner, R. *Numerical Toolbox for Verified Computing II*. Springer, Berlin, Vorabdruck.
- [88] Krämer, W. und Wedner, S. Two adaptive Gauss-Legendre type algorithms for the verified computation of definite integrals. *Reliable Computing*, 2(3):241–253, 1996.
- [89] Kulisch, U. *Grundlagen des numerischen Rechnens*. Bibliographisches Institut, Mannheim, Wien, Zürich, 1976.
- [90] Kulisch, U., Hrsg.. *Scientific computation with automatic result verification*. Springer, Wien, 1988.
- [91] Kulisch, U. *Wissenschaftliches Rechnen mit Ergebnisverifikation*. Friedr. Vieweg & Sohn, Braunschweig, Wiesbaden, 1989.
- [92] Kulisch, U. und Miranker, W. L. *Computer arithmetic in theory and practice*. Academic Press, New York, 1981.
- [93] Kulisch, U. und Miranker, W. L. *A New Approach to Scientific Computation*. Academic Press, New York, 1983.
- [94] Kulisch, U. und Ullrich, Ch. *Wissenschaftliches Rechnen und Programmiersprachen*. B. G. Teubner, Stuttgart, 1982.
- [95] Lang, B. Verifizierte Lösung von Gleichungs- und Ungleichungssystemen. *ZAMM*, 75(Suppl. 2):541–542, 1995.
- [96] Laurie, D. P. Practical error estimation in numerical integration. *J. Comput. Appl. Math.*, 12&13:425–431, 1985.
- [97] Laurie, D. P. What is a good quadrature error estimate ? In H. Braß; G. Hämmerlin, Hrsg., *Numerical Intergration, ISNM 85*, S. 91–98. Birkhäuser, 1988.

- [98] Lenze, B. *Mehrdimensionale Einseitige Approximationsoperatoren*. (Habilitationsschrift), FernUniversität Hagen, 1989.
- [99] Lether, F. G. Cross-Product Cubature Error Bounds. *Math. Comput.*, 24(111):583–592, 1970.
- [100] Lether, F. G. An error representation for product cubature rules. *SIAM. J. Numer. Anal.*, 7(3):363–365, 1970.
- [101] Lether, F. G. Cubature error bounds for Gauss-Legendre product rules. *SIAM. J. Numer. Anal.*, 8(1):36–42, 1971.
- [102] Lether, F. G. Error Bounds for Fully Symmetric Cubature Rules. *SIAM. J. Numer. Anal.*, 8(1):49–60, 1971.
- [103] Lohner, R. *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*. Dissertation, Universität Karlsruhe, Juni 1988.
- [104] Luik, E. *Fehlerabschätzungen bei Quadratur und Kubatur auf der Grundlage von Approximationsgraden*. Dissertation, Universität Tübingen, März 1984.
- [105] Lyness, J. N. Symmetric Integration rules for Hypercubes III. Construction of Integration Rules Using Null Rules. *ZAMM*, 4(69):T 83–85, 1989.
- [106] Malcolm, M. A. und Simpson, R. B. Local versus Global Strategies for Adaptive Quadrature. *ACM. Trans. on Math. Softw.*, 1(2):129–146, 1975.
- [107] Mayer, G. Grundbegriffe der intervallrechnung. In Kulisch, U., Hrsg., *Wissenschaftliches Rechnen mit Ergebnisverifikation – Eine Einführung* –, S. 101–117. Friedr. Vieweg & Sohn, 1989.
- [108] McKeeman, W. M. Algorithm 145, Adaptive Numerical Integration by SIMPSON’s Rule. *Comm. ACM*, 5:604, 1962.
- [109] McKeeman, W. M. Certification of Algorithm 145, Adaptive Numerical Integration by SIMPSON’s Rule. *Comm. ACM*, 6:167–168, 1963.
- [110] McKeeman, W. M. und Tesler, L. Algorithm 182, Nonrecursive Adaptive Integration by SIMPSON’s Rule. *Comm. ACM*, 6:315, 1963.
- [111] Moore, R. E. *Interval Analysis*. Prentice Hall, Englewood Cliffs, New Jersey, 1969.
- [112] Moore, R. E. *Methods and Applications of Interval Analysis*. SIAM, Philadelphia, 1979.
- [113] Moore, R. E. *Reliability in Computing*. Acad. Pr., Boston, 1988.
- [114] Neumaier, A. The enclosure of solutions of parameter-dependent systems of equations. In R. E. Moore, Hrsg., *Reliability in computing*, S. 269–286. Academic Press, 1988.

- [115] Neumaier, A. *Interval Methods for systems of equations*. Cambridge University Press, New York, 1. Auflage, 1990.
- [116] K. Nickel, Hrsg.. *Interval Mathematics 1985*, Berlin Heidelberg, September 1985. Proceedings of the International Symposium Freiburg i. Br., Federal Republic of Germany, Springer-Verlag.
- [117] O'Hara, H. und Smith, F. J. The evaluation of definite integrals by interval subdivision. *Comput. J.*, 12:179–182, 1969.
- [118] Oliver, J. A Practical Strategy for the Clenshaw-Curtis Quadrature Method. *J. Inst. Maths. Appl.*, 8:53–56, 1971.
- [119] Oliver, J. A doubly-adaptive Clenshaw-Curtis quadrature method. *Computer. J.*, 15:141–147, 1972.
- [120] Piessens, R. An Algorithm for Automatic Integration. *Angewandte Informatik*, 15:399–401, 1973.
- [121] Rall, L. B. *Automatic Differentiation : Techniques and Applications*. Springer-Verlag, Berlin, 1981.
- [122] Rall, L. B. Representation of Interval und Optimal Error Bounds. *Math. Comp.*, 41(163):219–227, 1983.
- [123] Ratschek, H. Die Subdistributivität der Intervallarithmetik. *ZAMM*, 54:189–192, 1971.
- [124] Ratschek, H. und Rokne, J. *Computer Methods for the Range of Functions*. Ellis Horwood Limited, Chichester, 1. Auflage, 1984.
- [125] Ratz, D. *Automatische Ergebnisverifikation bei globalen Optimierungsproblemen*. Dissertation, Universität Karlsruhe, 1992.
- [126] Ratz, D. On the Selection of Subdivision Directions in Interval Branch-and-Bound Methods for Global Optimization. *J. of Global Optimization*, 7:183–207, 1995.
- [127] Ratz, D. "An Optimized Interval Slope Arithmetic and its Application". Technischer Bericht 4/1976, Institut für Angewandte Mathematik, Universität Karlsruhe (TH), 1996.
- [128] Ratz, D. "Inclusion Isotone Extended Interval Arithmetic — A Toolbox Update —". Technischer Bericht 5/1996, Institut für Angewandte Mathematik, Universität Karlsruhe (TH), 1996.
- [129] Rice, J. R., Hrsg.. *Numerical Software*. Academic Press, New York, 1971.
- [130] Rivlin, T. J. Bounds on a polynomial. *J. of Research of the National Bureau of Standards – B. Mathematical Sciences*, 74B(1):47–54, 1970.
- [131] Robinson, I. G. Adaptive Gaussian Integration. *Austral. Computer J.*, 3:126–129, 1971.

- [132] Rokne, J. G. Low Complexity k-dimensional Centered Forms. *Computing*, 37:247–253, 1986.
- [133] Sard, A. Remainders as Integrals of Partial Derivatives. *Proc. Amer. Math. Soc.*, 3:732–741, 1952.
- [134] Sard, A. *Linear Approximation*. American Mathematical Society, Providence, Rhode Island, 1963.
- [135] Schira, T. *Ableitungsfreie Fehlerabschätzungen bei numerischer Integration holomorpher Funktionen*. Dissertation, Universität Karlsruhe, Juni 1994.
- [136] Schneider, C. Zur numerischen Berechnung von Schranken für Quadraturfehler. *ZAMM*, 4(69):T 83–85, 1989.
- [137] Schramm, P. *Sichere Verschneidung von Kurven und Flächen im CAGD*. Dissertation, Universität Karlsruhe, Februar 1995.
- [138] Shiriaev, D. *Fast Automatic Differentiation for Vector Processors and Reduction of the Spatial Complexity in a Source Translation Environment*. Dissertation, Universität Karlsruhe, 1993.
- [139] Spaniol, O. Die Distributivität in der Intervallarithmetik. *Computing*, 5:6–16, 1970.
- [140] Stenger, F. L. *Numerical Methods Based on Sinc and Analytic Functions*. Springer-Verlag, New York, Berlin, Heidelberg, 1993.
- [141] Stoer, J. *Numerische Mathematik 1*. Springer-Verlag, Berlin, Heidelberg, New York, 7. Auflage, 1994.
- [142] Storck, U. Verified Calculation of the Nodes und Weights for Gaussian Quadrature Formulas. *Interval Computations*, (4):114–124, 1993.
- [143] Storck, U. Zweidimensionale Integration mit automatischer Ergebnisverifikation. *ZAMM*, 7/8(73):T 897–T 899, 1993.
- [144] Storck, U. *Verifizierte Berechnung mehrfach geschachtelter singulärer Integrale der Gaskinetik*. Dissertation, Universität Karlsruhe, 1995.
- [145] Stroud, A. H. Estimating quadrature errors for functions with low continuity. *SIAM J. Numer. Anal.*, 3:420–424, 1966.
- [146] Stroud, A. H. *Approximate Calculation of Multiple Integrals*. Prentice Hall, Englewood Cliffs, New Jersey, 1971.
- [147] Stroud, A. H. und Secrest, D. *Gaussian Quadrature Formulas*. Prentice Hall, Englewood Cliffs, New Jersey, 1966.
- [148] Szegő, G. *Orthogonal polynomials*. American Mathematical Society Colloquium Publications 23, Providence, 4 Auflage, 1975.

- [149] Überhuber, Ch. W. *Numerical computation; methods, software and analysis*. Springer, Berlin, 1997.
- [150] van Dooren, P. und de Ridder, L. An adaptive algorithm for numerical integration over an n-dimensional cube. *J. Comput. Appl. Math.*, 2(3):207–210, 1976.
- [151] Vandergraft, J. S. *Introduction to numerical computations*. Acad. Press, New York, 1978.
- [152] Waldvogel, J. Numerical quadrature in several dimensions. In H. Braß; G. Hämmerlin, Hrsg., *Numerical Intergration, ISNM 85*, S. 295–309. Birkhäuser, 1988.
- [153] Wedner, S. Numerische Quadratur mit automatischer Ergebnisverifikation. Diplomarbeit, Universität Karlsruhe, 1994.
- [154] Witte, W. *Ansätze zur intervallanalytischen Approximationstheorie*. Dissertation, Universität Düsseldorf, 1981.
- [155] Yazici, A. und Altas, I. Some computer tests on two dimensional adaptive quadrature. *J. Pure. Appl. Sci.*, 17:1–9, 1984.