Verifizierte Bestimmung singulärer Integrale - Quadratur und Kubatur

Dissertation

von

Stefan Wedner

Karlsruhe 2000

Verifizierte Bestimmung singulärer Integrale - Quadratur und Kubatur

Zur Erlangung des akademischen
Grades eines

DOKTORS DER
NATURWISSENSCHAFTEN

von der Fakultät für Mathematik der
Universität Karlsruhe (TH)

genehmigte

DISSERTATION

von Dipl.-Math. Stefan Wedner

aus Karlsruhe

Tag der mündlichen Prüfung:	11. Dezember 2000
Referent:	Prof. Dr. W. Krämer
Korreferent:	Prof. Dr. E. Kaucher

Danksagung

Mein besonderer Dank gilt meinem Referenten Herrn Prof. Dr. W. Krämer für seine konstruktiven Anregungen und Anmerkungen zu dieser Arbeit.
Ebenso möchte ich Herrn Prof. Dr. E. Kaucher für die freundliche Übernahme des Korreferats, Herrn Prof. Dr. U. Kulisch, der mir die Möglichkeit gab, an seinem Institut zu arbeiten und zu forschen, sowie Herrn Dr. R. Klatte für die hilfreichen Ratschläge bei der Anfertigung der Arbeit danken.
Herrn H.-Doz. Dr. R. Lohner und Frau Dr. U. Storck möchte ich auf diesem Weg für ihre Diskussionsbereitschaft und die Bereitstellung von Literatur danken.
Ganz herzlich möchte ich mich auch bei allen ehemaligen Kolleginnen und Kollegen des Instituts für Angewandte Mathematik bedanken, die es mir erlaubten, die Arbeit in einem freundlichen und harmonischen Arbeitsumfeld zu erstellen.
Auf keinen Fall unerwähnt bleiben dürfen meine Schwester Dr. S. Wedner-Bianzano, die sich der mühseligen Aufgabe des Korrekturlesens annahm, sowie meine Eltern und meine Familie, die mir immer mit Rat und Tat zur Seite standen.

Einleitung

Die numerische Integration ist eines der ältesten Anwendungsgebiete der Mathematik. Schon in der Antike, fast zweitausend Jahre vor der Erfindung der Differential- und Integralrechnung der modernen Analysis, beschäftigte sich Archimedes mit der Quadratur des Kreises durch ein- bzw. umbeschriebene Polygone. Hieraus leitet sich auch der Name Quadratur für die numerische Integration eindimensionaler Funktionen ab.

Der Einsatz moderner Rechenanlagen mit großer Rechenleistung ermöglicht heutzutage die Berechnung komplizierter Integrale, wie sie z. B. in natur- oder ingenieurswissenschaftlichen Anwendungen auftreten. Oft sind diese Integrale nur Zwischenergebnisse auf dem Weg zur Lösung eines komplexen Problems. Approximative Integrationsverfahren berechnen endliche Linearkombinationen von Punktfunktionalen als Näherungswert für das gesuchte Integral. Die Konvergenz eines Quadraturverfahrens (z. B. Gauß- oder Romberg-Verfahren) garantiert zwar, daß die Näherung durch Erhöhung der Stützstellenanzahl verbessert werden kann. Ohne praktikable Abschätzungsformeln für den Verfahrensfehler, der Differenz aus dem exakten Wert des Integrals und der Approximationssumme, lassen sich jedoch keine gesicherten Aussagen über die Qualität der Näherung machen. Doch selbst wenn solche Fehlerabschätzungen vorhanden sind, sind die Ergebnisse dieser Verfahren wegen der beim Berechnen des Approximationsterms und der Restgliedabschätzung auftretenden Rundungsfehler mit Unsicherheiten behaftet.

Numerische Integrationsverfahren mit automatischer Ergebnisverifikation berücksichtigen sowohl Verfahrensfehler als auch Rundungsfehler, um garantierte obere und untere Schranken für den Wert eines Integrals zu bestimmen. Eine wichtige Rolle bei der Einschließung des Verfahrensfehlers spielt die automatische Differentiation. Mit ihrer Hilfe können, wie schon in [11] und [71] gezeigt, die in der Restglieddarstellung auftretenden höheren Ableitungen eingeschlossen werden. Voraussetzung für eine enge Einschließung des Integrals ist eine gute Approximation des Integrals, da die Verfahrensfehlereinschließungen bei dieser Methode relativ grob sind. Alternativ können aber auch ableitungsfreie Abschätzungen (vgl. [17, 77, 28]) zur Einschließung des Restglieds verwendet werden.
Auf dem Rechner auftretende Rundungsfehler werden durch die Maschinenintervallarithmetik miterfaßt, die mit Hilfe einer präzise definierten Arithmetik (siehe [50]) z. B. in den Programmiersprachenerweiterungen PASCAL–XSC, FORTRAN–XSC und C–XSC implementiert ist.

Die vorliegende Arbeit beschäftigt sich mit der verifizierten Berechnung von singulären Integralen. Dabei werden sowohl schwach als auch stark singuläre Integrale behandelt. Die bekanntesten Vertreter stark singulärer Integrale sind sicherlich Cauchy-Hauptwerte, die u. a. in vielen Anwendungen der Aerodynamik (siehe [88]) auftreten. Außerdem wird die Einschließung mehrfach iterierter singulärer Integrale mit Hilfe von

Produktformeln untersucht. Die Arbeit unterteilt sich in sechs Kapitel.
Im ersten Kapitel werden zunächst die wichtigsten Grundbegriffe und Bezeichnungen der Intervall- und Rechnerarithmetik, soweit sie für die Arbeit von Interesse sind, vorgestellt. Danach folgt eine kurze Beschreibung des erweiterten Intervall-Newton-Verfahrens. Das erweiterte Intervall-Newton-Verfahren wird im weiteren Verlauf der Arbeit zum Einschließen von Nullstellen der Peano-Kernfunktionen verwendet. Die Bedeutung der automatischen Differentiation für die numerische Integration mit automatischer Ergebnisverifikation ist schon weiter oben erwähnt worden. Ein Abschnitt über die Bestimmung von Taylorkoeffizienten mit Hilfe automatischer Differentiation beendet das Kapitel Grundlagen und Bezeichnungen.

Kapitel zwei enthält eine Beschreibung allgemeiner Quadraturverfahren und speziell des Gauß-Quadraturverfahrens. Es wird ein Algorithmus zur verifizierten Berechnung der Stützstellen und Gewichte von Gauß-Quadraturformeln angegeben. Sind die Rekursionskoeffizienten der zur Gewichtsfunktion gehörigen Orthogonalpolynome analytisch bekannt und berechenbar, wie z. B. bei den Legendre-Polynomen, so können mit diesem Algorithmus die Knoten und Gewichte auch bei großer Knotenanzahl ($n > 1000$) sehr eng eingeschlossen werden. Im allgemeinen Fall können die Rekursionskoeffizienten aus den gewöhnlichen Momenten berechnet werden. Die hierbei auftretenden numerischen Instabilitäten können durch Verwendung modifizierter Momente (siehe [25]) umgangen werden.
Das Restglied einer Quadraturformel kann als Integral des Produkts einer Peano-Kernfunktion und der entsprechenden Ableitung der zu integrierenden Funktion dargestellt werden. Aus den Peano-Kernfunktionen werden Integrationskonstanten bestimmt, mit deren Hilfe verschiedene Einschließungen des Quadraturfehlers berechnet werden können. Dies ermöglicht die Angabe eines mehrfach adaptiven Verfahrens zur verifizierten Bestimmung von schwach singulären Integralen. Anhand numerischer Beispiele wird die Effizienz des Verfahrens demonstriert.

Das dritte Kapitel wendet sich den stark singulären Integralen zu. Cauchy-Hauptwerte können mit speziellen Quadraturformeln, sogenannten Hauptwertformeln, berechnet werden. Besonderes Augenmerk muß hierbei auf die Berechnung der Approximationssumme gelegt werden, da hier numerische Instabilitäten autreten können. Die Restglieder dieser Hauptwertformeln können ebenfalls mit Hilfe der Restdarstellung von Peano bestimmt werden. Die Berechnung der Integrationskonstanten gestaltet sich jedoch wegen der komplizierteren Gestalt der Kernfunktion deutlich aufwendiger. Für spezielle stark singuläre Integrale werden Quadraturformeln und Restglieddarstellung hergeleitet. Das in Kapitel zwei beschriebene Verfahren zur Berechnung schwach singulärer Integrale wird auf stark singuläre Integrale übertragen und anhand von Beispielen getestet.

Mehrfach iterierte singuläre Integrale sind das Thema des vierten Kapitels. Dabei werden die bei einer adaptiven Zerlegung auftretenden Integraltypen näher untersucht und

verwertbare Verfahrensfehlerdarstellungen für die aus Tensorbildung gewonnenen Produktformeln angegeben. Wiederum runden numerische Beispiele das Kapitel ab.

Im fünften und vorletzten Kapitel wird die in C++ geschriebene Klassenbibliothek CLAVIS (**Cla**sses for **V**erified **I**ntegration over **S**ingularities) beschrieben, die Implementierungen der zuvor beschriebenen Verfahren enthält. Die in CLAVIS enthaltenen Integrationsklassen bieten eine intuitive Benutzerschnittstelle und eine einfache Erweiterungsmöglichkeit.

Die Arbeit wird mit einer Zusammenfassung und einem kurzen Ausblick in Kapitel sechs abgeschlossen.

Inhaltsverzeichnis

Kapitel 1

Grundlagen und Bezeichnungen

1.1 Intervallarithmetik

Die Intervallarithmetik ist das zentrale Hilfsmittel zur Einschließung der Lösungen numerischer Probleme. Deswegen stellen wir hier die wichtigsten Grundbegriffe der Intervallarithmetik zusammen. Für weitere Details verweisen wir auf die ausführlicheren Darstellungen in [2], [51], [60] oder [64].

Als (reelles) Intervall bezeichnen wir die abgeschlossene Menge $[x]$ mit

$$[x] := [\underline{x}, \overline{x}] := \{x \in I\!R \mid \underline{x} \leq x \leq \overline{x}\}.$$

Die reellen Zahlen $\underline{x}$ und $\overline{x}$ werden Endpunkte oder auch Intervallgrenzen genannt. Ein Intervall $[x]$ mit $\underline{x} = \overline{x}$ heißt Punktintervall. Die Menge aller reellen Intervalle bezeichnen wir mit $I\!I\!R$. Für Intervalle $[x] = [\underline{x}, \overline{x}]$ und $[y] = [\underline{y}, \overline{y}]$ führen wir die Begriffe Mittelpunkt, Absolutwert, Abstand und Durchmesser wie folgt ein.

$$\begin{array}{rrcl}
\text{Mittelpunkt:} & mid([x]) & := & \dfrac{\underline{x} + \overline{x}}{2} \\
\text{kleinster Absolutwert:} & \langle [x] \rangle & := & \min\{|x| \mid x \in [x]\} \\
\text{größter Absolutwert:} & |[x]| & := & \max\{|x| \mid x \in [x]\} = \max\{|\underline{x}|, |\overline{x}|\} \\
\text{Abstand:} & q([x],[y]) & := & \max\{|\underline{x} - \underline{y}|, |\overline{x} - \overline{y}| \\
\text{Durchmesser:} & d([x]) & := & \overline{x} - \underline{x} \\
\text{relativer Durchmesser:} & d_{rel}([x]) & := & \begin{cases} \dfrac{d([x])}{\langle [x] \rangle} & \text{falls } 0 \notin [x] \\ d([x]) & \text{sonst} \end{cases}
\end{array}$$

Der Abstand zweier Intervalle ist eine Metrik, mit deren Hilfe die Konvergenz einer Folge von Intervallen beschrieben werden kann.

1.1.1 Intervallverknüpfungen

Die arithmetischen Grundoperationen werden folgendermaßen definiert

$$[x] \circ [y] := \{x \circ y \mid x \in [x], y \in [y]\}$$

mit $\circ \in \{+, -, \cdot, /\}$. Bei der Division setzen wir $0 \notin [\underline{y}, \overline{y}]$ voraus. Eine wichtige Eigenschaft der Intervallverknüpfungen ist die Inklusionsisotonie

$$[x] \in [X], [y] \in [Y] \quad \Rightarrow \quad [x] \circ [y] \in [X] \circ [Y] \ ,$$

für alle $\circ \in \{+, -, \cdot, /\}$ und $[x], [X], [y], [Y] \in I\mathbb{R}$, die sich direkt aus der Definition ergibt. Sie ist der Garant dafür, daß wir nicht exakt darstellbare Werte durch einschließende Intervalle ersetzen können, um eine Einschließung des exakten Ergebnisses zu erhalten. Bemerkenswert ist auch, daß Addition und Multiplikation zwar assoziativ und kommutativ sind, jedoch nicht das Distributivgesetz erfüllen. Es gilt lediglich die Subdistributivität

$$[x] \cdot ([y] + [z]) \subseteq [x] \cdot [y] + [x] \cdot [z].$$

Zur Berechnung von Intervallverknüpfungen können folgende Formeln verwendet werden:

$$\begin{aligned}
[x] \ + \ [y] &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \\
[x] \ - \ [y] &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}], \\
[x] \ \cdot \ [y] &= [\min\{\underline{x}\,\underline{y}, \underline{x}\,\overline{y}, \overline{x}\,\underline{y}, \overline{x}\,\overline{y}\}, \max\{\underline{x}\,\underline{y}, \underline{x}\,\overline{y}, \overline{x}\,\underline{y}, \overline{x}\,\overline{y}\}], \\
[x] \ / \ [y] &= [x] \cdot [1/\overline{y}, 1/\underline{y}], \qquad 0 \notin [y]
\end{aligned}$$

1.1.2 Intervallmäßige Auswertung einer reellen Funktion

Die Bestimmung enger Einschließungen des Wertebereichs $W_f([x]) := \{f(x) \mid x \in [x]\}$ einer reellen stetigen Funktion $f : D \subset \mathbb{R} \to \mathbb{R}$ ist von großer Bedeutung für Verifikationsalgorithmen. Setzt sich f nur aus den Grundoperationen zusammen, werten wir die Funktion einfach intervallmäßig aus. Dazu ersetzt man jedes Auftreten der Variablen x durch das Intervall $[x]$ und alle Konstanten durch entsprechende Punktintervalle. Interpretiert man die Operationen als Intervallverknüpfungen, so garantiert die Inklusionsisotonie, daß die intervallmäßige Auswertung $f([x])$ von f eine Obermenge des gesuchten Wertebereichs bildet. Die gewonnenen Einschließungen sind dabei jedoch stark abhängig von der gewählten Darstellung der Funktion (siehe z. B. Subdistributivität). Zur intervallmäßigen Auswertung von Funktionen, die aus Standardfunktionen wie z. B. $\log, \exp$ oder $\sin$ aufgebaut sind, benötigt man die entsprechenden Standardintervallfunktionen. Diese sind für beliebige Intervalle, die im Definitionsbereich der dazugehörigen reellen Standardfunktion φ liegen, durch

$$\varphi([x]) := \{\varphi(x) \mid x \in [x]\}$$

definiert. Ist φ eine monotone wachsende Funktion (wie z. B. ln oder exp), so kann die entsprechende Intervallfunktion mittels

$$\varphi([x]) = [\varphi(\underline{x}), \varphi(\overline{x})]$$

bestimmt werden. Im allgemeinen ist die Auswertung von Standardintervallfunktionen aber wesentlich komplizierter.

Spezielle zentrierte Form

Aus dem Mittelwertsatz lassen sich für stetig differenzierbare Funktionen f Funktionsaudrücke in zentrierter Form herleiten. Für $m, x \in [x]$ existiert ein $\xi \in [x]$ mit

$$\begin{aligned} f(x) \;=\; f(m) + f'(\xi)(x-m) \quad &\in \quad f(m) + f'([x])(x-m) \\ &\subseteq \quad \underbrace{f(m) + f'([x])([x]-m)}_{=:f_m([x])}. \end{aligned} \tag{1.1}$$

Unter bestimmten Voraussetzungen an die Funktionen f und f' (genauer an die gewählten Funktionsausdrücke siehe [2]), die häufig erfüllt sind, existieren positive Konstanten c_1 und c_2, so daß gilt

$$\begin{aligned} q(f([x]), f([x])) &\leq c_1\, d([x]) \\ q(f([x]), f_m([x])) &\leq c_2\, d^2([x]). \end{aligned}$$

Das heißt für Intervalle mit hinreichend engen Durchmessern erwartet man bei der zentrierten Form deutlich engere Einschließungen des Wertebereichs als bei der einfachen intervallmäßigen Auswertung.

1.2 Rechnerarithmetik

Auf dem Rechner ist nur eine endliche Teilmenge (Raster) R der reellen Zahlen $\mathbb{R}$ darstellbar. Deshalb müssen bei Problemstellungen, die das Rechnen mit reellen Zahlen erfordern, diese auf Gleitkommazahlen oder Maschinenzahlen abgebildet werden und die arithmetischen Grundoperationen durch schnell durchführbare Maschinenoperationen approximiert werden. Für ein zuverlässiges numerisches Rechnen auf Computern ist daher eine mathematisch exakte Definition der Rechnerarithmetik, wie sie in [50] oder [51] gegeben wird, unerläßlich.

Definition 1 *Eine Abbildung $\square : \mathbb{R} \to R$ nennen wir Rundung, wenn gilt*

(R1) $\square x = x, \quad$ *für* $x \in R$ *(Projektion)*

(R2) $x \leq y \Rightarrow \square x \leq \square y, \quad$ *für* $x, y \in \mathbb{R}$. *(Monotonie)*

Eine Rundung nennen wir antisymmetrisch, wenn sie

$$(R3) \quad \square(-x) = -(\square x), \quad \text{für } x \in \mathbb{R} \qquad \text{(Antisymmetrie)}$$

erfüllt. Eine Rundung heißt nach unten (oben) gerichtet, falls gilt

$$(R4) \quad \square x \leq x \, (\square x \geq x), \quad \text{für } x \in \mathbb{R}.$$

Die nach unten bzw. oben gerichtete Rundung zur nächstkleineren bzw. nächstgrößeren Maschinenzahl kürzen wir auch mit $\bigtriangledown$ bzw. $\bigtriangleup$ ab.

Die Maschinenoperationen $+, -, \cdot$ und / lassen sich über spezielle Rundungen, die Semimorphismen genannt werden, erklären.

Definition 2 *Eine antisymmetrische Rundung $\square : \mathbb{R} \to R$ heißt Semimorphismus, wenn die Verknüpfungen $+, -, \cdot, /$ in R definiert sind durch*

$$(RG) \quad x \boxdot y = \square(x \circ y), \quad \text{für } x, y \in R, \circ \in \{+, -, \cdot, /\}.$$

Verknüpfungen, die über einen Semimorphismus erklärt sind, liefern Ergebnisse maximaler Genauigkeit, d. h. zwischen dem in $\mathbb{R}$ berechneten Verknüpfungsergebnis $x \circ y$ und seiner Approximation $x \boxdot y$ in R liegt kein weiteres Element aus R.
Für die praktische Bestimmung von $x \boxdot y$ ist es nicht nötig, zumal oft auch gar nicht möglich, das exakte Ergebnis $x \circ y$ zu bestimmen, sondern es genügt ein Näherungsergebnis von $x \circ y$ zu berechnen, dessen Bild unter der Rundung mit dem des exakten Ergebnisses übereinstimmt.

1.2.1 Maschinenintervalle

Beim Rechnen mit Intervallen auf der Maschine müssen diese (analog zu den reeellen Zahlen) auf eine endliche Menge sogenannter Maschinenintervalle abgebildet werden. Ein Maschinenintervall $[x]$ ist definiert als

$$[x] = \{x \in \mathbb{R} \,|\, \underline{x} \leq x \leq \overline{x}, \quad \underline{x}, \overline{x} \in R\}.$$

Ein reelles Interval $[x]$ wird auf ein Maschinenintervall abgebildet durch

$$\diamondsuit[\underline{x}, \overline{x}] = [\bigtriangledown \underline{x}, \bigtriangleup \overline{x}].$$

Die Rundung $\diamondsuit$ ist antisymmetrisch und wird auch Rundung nach außen genannt.

Die inneren und äußeren Verknüpfungen für Maschinenintervalle werden mittels $\diamondsuit$ über das Prinzip des Semimorphismus definiert. Seien x, y Maschinenintervalle, dann gilt

$$x \diamondsuit y := \diamondsuit(x \circ y) \qquad \text{für } \circ \in \{+, -, \cdot, /\}.$$

Nach dem gleichen Prinzip können, wie in [50] und [51] gezeigt, Verknüpfungen von komplexen Zahlen, Vektoren und Matrizen approximiert werden.

1.3 Erweitertes Intervall-Newton-Verfahren

Zum Auffinden der Nullstellen einer stetig differenzierbaren Funktion $f : \mathbb{R} \to \mathbb{R}$ dient das erweiterte Intervall-Newton-Verfahren, das auf dem Newton-Operator

$$N([x]) := mid([x]) - \frac{f(mid([x]))}{f'([x])} \tag{1.2}$$

basiert. Ausgehend von einem Startintervall $[x^{(0)}]$ verwendet es die Iterationsvorschrift

$$[x]^{(k+1)} := [x]^{(k)} \cap N([x]^{(k)}), \quad k \geq 0,$$

um die in $[x^{(0)}]$ vorhandenen Nullstellen von f einzuschließen. Die Schnittbildung verhindert ein Divergieren des Verfahrens. Ist der Schnitt leer, so hat f keine Nullstelle in $[x]^{(0)}$. Besitzt f in $[x]$ mehrere Nullstellen, dann gilt $0 \in f'([x])$ und die Division in 1.2 ist in herkömmlicher Intervallarithmetik nicht durchführbar. Deswegen führt man die Menge $I\mathbb{R}^*$

$$I\mathbb{R}^* = I\mathbb{R} \cup \{[-\infty, r] \mid r \in \mathbb{R}\} \cup \{[l, \infty] \mid l \in \mathbb{R}\} \cup \{[-\infty, \infty]\}$$

der erweiterten reellen Intervalle ein. Dabei bezeichnet $[-\infty, r]$ die Menge $\{x \in \mathbb{R} \mid x \leq r\}$, $[r, \infty]$ die Menge $\{x \in \mathbb{R} \mid x \geq r\}$ und $[-\infty, +\infty]$ die gesamte reelle Achse. Mit Hilfe von erweiterten Intervallen wird das Ergebnis der Intervalldivision x/y für Intervalle y mit $0 \in y$ definiert (siehe [72]):

$$[\underline{x}, \overline{x}]/[\underline{y}, \overline{y}] := \begin{cases} [-\infty, +\infty] & \text{für } 0 \in x \text{ und } 0 \in b, \\ [\overline{x}/\underline{y}, +\infty] & \text{für } \overline{x} < 0 \text{ und } \underline{y} < \overline{y} = 0, \\ [-\infty, \overline{x}/\overline{y}] \cup [\overline{x}/\underline{y}, +\infty] & \text{für } \overline{x} < 0 \text{ und } \underline{y} < 0 < \overline{y}, \\ [-\infty, \overline{x}/\overline{y}] & \text{für } \overline{a} < 0 \text{ und } 0 = \underline{y} < \overline{y}, \\ [-\infty, \underline{x}/\underline{y}] & \text{für } 0 < \underline{x} \ \wedge \ \underline{y} < \overline{y} = 0, \\ [-\infty, \underline{x}/\underline{y}] \cup [\underline{x}/\overline{y}, +\infty] & \text{für } 0 < \underline{x} \text{ und } \underline{y} < 0 < \overline{y}, \\ [\underline{x}/\overline{y}, +\infty] & \text{für } 0 < \underline{x} \text{ und } 0 = \underline{y} < \overline{y}, \\ \emptyset & \text{für } 0 \notin x \text{ und } \underline{y} = \overline{y} = 0. \end{cases} \tag{1.3}$$

Setzt man $x - (+\infty) = -\infty$ und $x - (-\infty) = +\infty$, so kann man die Differenz einer reellen Zahl r und eines erweiterten Intervalls $[\underline{x}, \overline{x}]$ in gewohnter Weise definieren

$$r - [\underline{x}, \overline{x}] = [r - \overline{x}, r - \underline{x}].$$

Zusätzlich definiert man noch die Spezialfälle

$$\begin{aligned} r - \emptyset &= \emptyset \\ r - ([x] \cup [y]) &= (r - [x]) \cup (r - [y]). \end{aligned}$$

für erweiterte Intervalle $[x]$ und $[y]$. Die so definierten Verknüpfungen sind inklusionsisoton und ermöglichen den Newton-Operator auch für Funktionen mit mehreren Nullstellen in $[x]$ einzusetzen, denn es gilt der folgende Satz.

Satz 3 *Sei $f : D \subseteq \mathbb{R} \to \mathbb{R}$ eine stetig differenzierbare Funktion und $[x] \in I\mathbb{R}$ ein Intervall mit $[x] \subseteq D$, dann hat der Newtonoperator die folgenden Eigenschaften:*

1. *Für jede Nullstelle $x^* \in [x]$ von f gilt $x^* \in N([x])$.*
2. *Falls $N([x]) \cap [x] = \emptyset$ gilt, so existiert keine Nullstelle von f auf $[x]$.*
3. *Wenn $N([x]) \overset{\circ}{\subseteq} [x]$ gilt, dann existiert eine eindeutige Nullstelle von f auf $[x]$ und somit auch auf $N([x])$.*

Eine detailliertere Beschreibung des erweiterten Intervall-Newton-Verfahrens wird in [31] gegeben. Das Konvergenzverhalten des Intervall-Newton-Verfahrens wird in [2] genauer untersucht.

1.4 Bestimmung von Taylorkoeffizienten

Um gesicherte Aussagen über die Lösung eines numerischen Verfahrens machen zu können, ist es erforderlich, den Verfahrensfehler zu bestimmen. Bei den in dieser Arbeit beschriebenen Quadratur- und Kubaturverfahren ist dies gleichbedeutend mit der Einschließung eines höheren Ableitungsterms einer Funktion. In diesem Abschnitt beschreiben wir eine Möglichkeit, die Taylorkoeffizienten einer hinreichend glatten reellwertigen Funktion rekursiv zu berechnen. Die unter dem Namen Automatische Differentiation ([71]) bekannte Methode erlaubt bei intervallmäßiger Durchführung, Taylorkoeffizienten höherer Ordnung mit wenig Aufwand einzuschließen.

1.4.1 Funktionen einer reellen Veränderlichen

Gegeben sei eine reellwertige Funktion $f : \mathbb{R} \to \mathbb{R}$, die in einer offenen Umgebung U von x_0 hinreichend oft stetig differenzierbar sei. Für die Taylorkoeffizienten der Funktion f an der Stelle x_0 wählen wir die Schreibweise

$$(f)_k := \frac{1}{k!} f^{(k)}(x_0) = \frac{1}{k!} \frac{d^k f(x_0)}{dx^k}, \quad k \geq 0. \tag{1.4}$$

Ist die Funktion f analytisch für $x \in U$, so hat die dazugehörige Taylorreihe die Form

$$f(x) = \sum_{k=0}^{\infty} (f)_k (x - x_0)^k. \tag{1.5}$$

Die Taylorkoeffizienten konstanter Funktionen $f \equiv c$ und der Identität $f(x) = x$ (statt $(f)_k$ schreiben wir auch $(c)_k$ bzw. $(x)_k$) erfüllen trivialerweise die Beziehungen

$$(c)_k = \begin{cases} c & \text{für } k = 0 \\ 0 & \text{für } k > 0 \end{cases} \qquad (x)_k = \begin{cases} x_0 & \text{für } k = 0 \\ 1 & \text{für } k = 1 \\ 0 & \text{für } k > 1 \end{cases}. \tag{1.6}$$

Für $k = 0$ gilt $(f)_0 = f(x_0)$, weswegen wir uns jetzt ausschließlich auf den Fall $k \geq 1$ beschränken wollen.
Aus den Rechenregeln für Potenzreihen ergeben sich die Rekursionsformeln für Funktionen, die sich aus den vier Grundrechenarten zusammensetzen

$$\begin{aligned} (f \pm g)_k &= (f)_k \pm (g)_k \\ (f \cdot g)_k &= \sum_{i=0}^{k} (f)_i \, (g)_{k-i} \\ (f/g)_k &= \frac{1}{(g)_0} \left((f)_k - \sum_{i=1}^{k} (g)_i \, (f/g)_{k-i} \right). \end{aligned} \tag{1.7}$$

Mit Hilfe der Formeln (1.6) und (1.7) lassen sich die Taylorkoeffizienten beliebiger rationaler Funktionen, die unsere Voraussetzungen erfüllen, rekursiv berechnen.
Im Spezialfall $f \equiv g$ vereinfacht sich die Produktformel zu

$$(f^2)_k = \begin{cases} 2 \sum_{i=0}^{(k-1)/2} (f)_i (f)_{k-i}, & \text{für } k \text{ ungerade} \\ 2 \sum_{i=0}^{(k-2)/2} (f)_i (f)_{k-i} + (f)_{k/2}^2, & \text{für } k \text{ gerade} \end{cases}. \tag{1.8}$$

Für das Weitere setzen wir voraus, daß die auftretenden Funktionen analytisch sind und mit ihrer Taylorreihe übereinstimmen. Aus (1.4) erhalten wir durch Ableiten

$$f'(x) = \sum_{k=1}^{\infty} k (f)_k (x - x_0)^{k-1} = \sum_{k=0}^{\infty} (k+1)(f)_{k+1} (x - x_0)^k \tag{1.9}$$

und Koeffizientenvergleich eine Darstellung der Taylorkoeffizenten $(f')_k$ der Ableitung von f

$$(f')_k = (k+1)(f)_{k+1}. \tag{1.10}$$

Am Beispiel der Logarithmusfunktion soll erläutert werden, wie aus Gleichung (1.10) Rekursionsformeln für Standardfunktionen gewonnen werden können. Sei $h := \ln f$ bzw. $h(x) = \ln(f(x))$ mit $f > 0$. Aus $h' = f'/f$ folgt für $k > 0$

$$k(f)_k = (f')_{k-1} = (h'f)_{k-1} = \sum_{i=0}^{k-1} (h')_i (f)_{k-i-1} = \sum_{i=1}^{k} i (h)_i (f)_{k-i}.$$

Zusammengefaßt erhält man die Gleichung:

$$(\ln f)_k = \frac{1}{(f)_0} \left((f)_k - \frac{1}{k} \sum_{i=1}^{k-1} i (\ln f)_i (f)_{k-i} \right) \tag{1.11}$$

Dieser Ansatz kann allgemein (siehe [47]) auf alle Funktionen der Form $h = g \circ f$ angewandt werden, falls eine Funktion w mit $g' \circ f = 1/w$ existiert, deren Taylorkoeffizienten $(w)_k$ berechnet werden können, denn es gilt

$$k(f)_k = (f')_{k-1} = (h'w)_{k-1} = \sum_{i=0}^{k-1} (h')_i (w)_{k-i-1} = \sum_{i=1}^{k} i(h)_i (w)_{k-i}.$$

Auflösen nach $(h)_k$ ergibt

$$(h)_k = \frac{1}{(w)_0}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(h)_i (w)_{k-i}). \tag{1.12}$$

Für die Exponentiation erhält man mit dem Ansatz $(u^a)'u = au^a u'$ und Gleichung (1.10) folgende Rechenvorschrift

$$(f^a)_k \quad = \quad \frac{1}{k(f)_0}\sum_{i=0}^{k-1}(a(k-i)-i)(f)_{k-i}(f^a)_i. \tag{1.13}$$

Im Fall $a = 1/2$ verwendet man besser die effizientere Variante

$$(\sqrt{f})_k \quad = \quad \begin{cases} \dfrac{1}{2(\sqrt{f})_0}((f)_k - 2\displaystyle\sum_{i=1}^{(k-1)/2}(\sqrt{f})_i(\sqrt{f})_{k-i}), & k \text{ ungerade} \\ \dfrac{1}{2(\sqrt{f})_0}((f)_k - 2\displaystyle\sum_{i=1}^{(k-2)/2}(\sqrt{f})_i(\sqrt{f})_{k-i} - (\sqrt{f})_{k/2}^2), & k \text{ gerade} \end{cases},$$

die aus der Produktformel abgeleitet wird.
Enthält für $a \in I\!N$ $(f^a)_0$ die Null, so können mit Hilfe der Formel $(f^a)_k = (f)_0^{a-k} w_k$ und

$$w_k = \begin{cases} 1 & \text{für } k = 0 \\ \dfrac{1}{k}\displaystyle\sum_{i=0}^{k-1}(a(k-i)-i)(f)_0^{k-i-1}(f)_{k-i}w_i & \text{für } k \geq 1 \end{cases}$$

immerhin die Taylorkoeffizienten $(f)_k$ mit $k \leq a$ berechnet werden (siehe [64]). Weitere Formeln zur Berechnung von Taylorkoeffizienten können Tabelle 1.1 oder direkt aus [47] entnommen werden.

1.4.2 Funktionen zweier reeller Veränderlichen

Die Berechnung der Taylorkoeffizienten einer Funktion f zweier reeller Veränderlichen $f : I\!R^2 \to I\!R$, die in einer offenen Umgebung U von $(x_0, y_0) \in I\!R^2$ analytisch ist, verläuft ähnlich wie im eindimensionalen Fall. Für die Taylorkoeffizienten der Funktion f an der Stelle (x_0, y_0) schreiben wir

$$(f)_{k,l} := \frac{1}{k!l!} f^{(k,l)}(x_0, y_0) = \frac{1}{k!l!}\frac{\partial^{k+l} f(x_0, y_0)}{\partial x^k \partial y^l}, \quad k, l \geq 0. \tag{1.14}$$

$$
\begin{aligned}
(e^f)_k &= \sum_{i=0}^{k-1}(1-\frac{i}{k})(e^f)_i(f)_{k-i} \\
(\ln f)_k &= \frac{1}{(f)_0}\left((f)_k - \sum_{i=1}^{k-1}(k-\frac{1}{i})(f)_i(\ln f)_{k-i}\right) \\
(\sin f)_k &= \frac{1}{k}\sum_{i=0}^{k-1}(i+1)(\cos f)_{k-i-1}(f)_{i+1} \\
(\cos f)_k &= -\frac{1}{k}\sum_{i=0}^{k-1}(i+1)(\sin f)_{k-i-1}(f)_{i+1} \\
(\tan f)_k &= \frac{1}{\cos^2(f)_0}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\tan f)_i(\cos^2 f)_{k-i}) \\
(\cot f)_k &= \frac{-1}{\sin^2(f)_0}((f)_k + \frac{1}{k}\sum_{i=1}^{k-1} i(\cot f)_i(\sin^2 f)_{k-i}) \\
(\arcsin f)_k &= \frac{1}{\sqrt{1-(f)_0^2}}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\arcsin f)_i(\sqrt{1-f^2})_{k-i}) \\
(\arccos f)_k &= \frac{-1}{\sqrt{1-(f)_0^2}}((f)_k + \frac{1}{k}\sum_{i=1}^{k-1} i(\arccos f)_i(\sqrt{1-f^2})_{k-i}) \\
(\arctan f)_k &= \frac{1}{1+(f)_0^2}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\arctan f)_i(1+f^2)_{k-i}) \\
(\operatorname{arccot} f)_k &= \frac{-1}{1+(f)_0^2}((f)_k + \frac{1}{k}\sum_{i=1}^{k-1} i(\operatorname{arccot} f)_i(1+f^2)_{k-i}) \\
(\sinh f)_k &= \frac{1}{k}\sum_{i=0}^{k-1}(k-i)(\cosh f)_i(f)_{k-i} \\
(\cosh f)_k &= \frac{1}{k}\sum_{i=0}^{k-1}(k-i)(\sinh f)_i(f)_{k-i} \\
(\tanh f)_k &= \frac{1}{\cosh^2(f)_0}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\tanh f)_i(\cosh^2 f)_{k-i}) \\
(\coth)_k &= \frac{-1}{\sinh^2(f)_0}((f)_k + \frac{1}{k}\sum_{i=1}^{k-1} i(\coth f)_i(\sinh^2 f)_{k-i}) \\
(\operatorname{arsinh} f)_k &= \frac{1}{\sqrt{(f)_0^2+1}}((f)_k + \frac{1}{k}\sum_{i=1}^{k-1} i(\operatorname{arsinh} f)_i(\sqrt{f^2+1})_{k-i}) \\
(\operatorname{arcosh} f)_k &= \frac{1}{\sqrt{(f)_0^2-1}}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\operatorname{arcosh} f)_i(\sqrt{f^2-1})_{k-i}) \\
(\operatorname{artanh} f)_k &= \frac{1}{1-(f)_0^2}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\operatorname{artanh} f)_i(1-f^2)_{k-i}) \\
(\operatorname{arcoth} f)_k &= \frac{1}{1-(f)_0^2}((f)_k - \frac{1}{k}\sum_{i=1}^{k-1} i(\operatorname{arcoth} f)_i(1-f^2)_{k-i})
\end{aligned}
$$

Tabelle 1.1: Rekursionsformeln für Taylor-Koeffizenten

Die dazugehörige Taylorreihe hat die Form

$$f(x,y) = \sum_{j=0}^{\infty}\sum_{i=0}^{\infty}(f)_{i,j}(x-x_0)^i(y-y_0)^j. \tag{1.15}$$

Aus der Definition ergeben sich unmittelbar die folgenden Beziehungen für konstante Funktionen und die unabhängigen Variablen

$$(c)_{k,l} = \begin{cases} c & \text{für } k=l=0 \\ 0 & \text{sonst} \end{cases}$$

und

$$(x)_{k,l} = \begin{cases} x_0 & \text{für } k=l=0 \\ 1 & \text{für } k=1, l=0 \\ 0 & \text{sonst} \end{cases} \qquad (y)_{k,l} = \begin{cases} y_0 & \text{für } k=l=0 \\ 1 & \text{für } k=0, l=1 \\ 0 & \text{sonst} \end{cases}.$$

Die Regeln für die Grundrechenarten lauten:

$$\begin{aligned} (f \pm g)_{k,l} &= (f)_{k,l} \pm (g)_{k,l} \\ (f \cdot g)_{k,l} &= \sum_{j=0}^{l}\sum_{i=0}^{k}(f)_{i,j}\,(g)_{k-i,l-j} \\ (f/g)_{k,l} &= \frac{1}{(g)_0}\left((f)_{k,l} - \sum_{i=1}^{k}\sum_{j=0}^{l}(g)_{i,j}\,(f/g)_{k-i,l-j} - \sum_{j=1}^{l}(g)_{0,j}\,(f/g)_{i,l-j}\right). \end{aligned} \tag{1.16}$$

Obige Gleichungen können durch schrittweise Anwendung der eindimensionalen Regeln bewiesen werden (siehe z. B. [21]). Analog lassen sich Rechenvorschriften für die Taylorkoeffizienten von Standardfunktionen herleiten. Dies demonstrieren wir wiederum an der Logarithmusfunktion. Zunächst hält man die Komponente y fest und bildet die Taylorkoeffizienten $(\ln f)_{k,*}$ mit Hilfe der eindimensionalen Regel (1.11). Danach wendet man die Regeln (1.7) auf die Variable y an:

$$\begin{aligned} (\ln f)_{k,l} &= ((\ln f)_{k,*})_{*,l} = \left(\frac{1}{(f)_{0,*}}((f)_{k,*} - \frac{1}{k}\sum_{i=1}^{k-1} i(\ln f)_{i,*}(f)_{k-i,*})\right)_{*,l} \\ &= \frac{1}{(f)_{0,0}}\left((f)_{k,l} - \frac{1}{k}\sum_{j=0}^{l}\sum_{i=1}^{k-1} i(\ln f)_{i,j}(f)_{k-i,l-j} - \sum_{j=1}^{l}(f)_{0,j}(\ln f)_{k,l-j}\right) \end{aligned}$$

Für $k = 0$ oder $l = 0$ verwenden wir die eindimensionale Regel (1.11). Nach dem gleichen Schema werden die folgenden Formeln $(l, k > 0)$ berechnet.

$$\begin{aligned}
(f^a)_{k,l} &= \frac{1}{k(f)_{0,0}} \left(\sum_{i=0}^{k-1} (a(k-i)-i) \sum_{j=0}^{l} (f)_{k-i,l-j} (f^a)_{i,j} - k \sum_{j=1}^{l} (f)_{0,j} (f^a)_{k,l-j} \right) \\
(\sin f)_{k,l} &= \frac{1}{k} \sum_{i=0}^{k-1} (i+1) \sum_{j=0}^{l} (\cos f)_{k-i-1,j} (f)_{i+1,l-j} \\
(\cos f)_{k,l} &= -\frac{1}{k} \sum_{i=0}^{k-1} (i+1) \sum_{j=0}^{l} (\sin f)_{k-i-1,j} (f)_{i+1,l-j}.
\end{aligned}$$

Ist h die Verkettung zweier Funktionen $h = g \circ f$ mit einer Funktion g, deren Ableitung die Form $g' \circ f = 1/w$ hat, so können die Taylorkoeffizienten von h rekursiv aus den Taylorkoeffizienten $(w)_{k,l}$ berechnet werden

$$(h)_{k,l} = \frac{1}{(w)_{0,0}} \left((f)_{k,l} - \frac{1}{k} \sum_{j=0}^{l} \sum_{i=1}^{k-1} i (h)_{i,j} (w)_{k-i,l-j} - \sum_{j=1}^{l} (w)_{0,j} (h)_{k,l-j} \right). \tag{1.17}$$

Weitere Rekursionsformeln zur Bestimmung von Taylorkoeffizienten finden sich in [42].

Kapitel 2

Quadratur schwach singulärer Integrale

Die Integration einer Funktion ist eines der grundlegenden Probleme der Mathematik. Die Lösung dieses Problems mittels Stammfunktion ist in vielen Fällen praktisch nicht durchführbar, z. B. wenn die Stammfunktion nicht analytisch darstellbar, nur schwer berechenbar oder numerisch schlecht auswertbar ist. Gleiches gilt erst recht, wenn nur Näherungen der Funktion durch Meßwerte an bestimmten Stellen gegeben sind. In diesen Fällen verwendet man zur Bestimmung des gesuchten Integrals Quadraturverfahren. Das Gebiet der numerischen Integration ist viel zu umfangreich, um auch nur eine überblicksartige Darstellung der gängigen Verfahren zu präsentieren. Deswegen beschränken wir uns in diesem Kapitel auf ein paar allgemeinere Aussagen und die Beschreibung des Gauß-Quadraturverfahrens, soweit es für unsere Zwecke, nämlich die verifizierte Integration von Bedeutung ist. In Abschnitt 2.6.1 stellen wir schließlich ein Verfahren zur verifizierten Integration nicht singulärer und schwach singulärer Funktionen vor.

2.1 Quadraturverfahren

In diesem Abschnitt bezeichne f eine auf dem Intervall $[a, b]$ stetige Funktion und w eine zulässige Gewichtsfunktion ($0 < \int_a^b w(x)\,dx < \infty$), d. h. der Integrand $f\,w$ ist Lebesgue-integrierbar. Zur numerischen Bestimmung des Integrals

$$I(f) = \int_a^b f(x)w(x)\,dx \tag{2.1}$$

werten wir die Funktion f an den n Stützstellen $x_{1,n}, \ldots, x_{n,n}$ aus, multiplizieren die Funktionswerte $f(x_{i,n})$ mit den entsprechenden Gewichten $w_{i,n}$ und erhalten so eine Quadratursumme $Q_n(f)$

$$Q_n(f) = \sum_{i=1}^{n} w_{i,n} f(x_{i,n}), \tag{2.2}$$

die eine Näherung für das gesuchte Integral darstellt. Das auf $C[a,b]$ definierte lineare Funktional Q_n heißt Quadraturformel mit n Stützstellen. Eine Quadraturformel heißt symmetrisch, wenn gilt:

$$\left.\begin{array}{rcl} x_{i,n} - a & = & b - x_{n-i+1,n} \\ w_{i,n} & = & w_{n-i+1,n} \end{array}\right\} \; i = 1, \ldots, n$$

Eine Folge von Quadraturformeln mit wachsender Stützstellenanzahl nennen wir Quadraturverfahren.
Läßt man in der Quadratursumme auch Ableitungsterme zu, so erhält man Quadraturformeln der Gestalt:

$$Q_n(f) = \sum_{j=0}^{m} \sum_{i=1}^{n_j} w_{i,n_j}^{(j)} f^{(j)}(x_{i,n_j}^{(j)}) \tag{2.3}$$

In diesem Fall spricht man auch von allgemeinen Quadraturformeln. In diesem Kapitel werden wir uns jedoch ausschließlich auf Quadraturformeln der Form (2.2) beschränken und uns erst im nächsten Kapitel den allgemeinen Quadraturverfahren widmen.
Das lineare Restfunktional einer Quadraturformel Q_n, die Differenz zwischen Integral und Näherung

$$R_n(f) = I(f) - Q_n(f) \tag{2.4}$$

nennt man auch Quadraturfehler.
Eine Quadraturformel Q_n hat den Exaktheitsgrad s genau dann, wenn gilt $R_n(p) = 0$ für alle Polynome p, deren Grad nicht größer als s ist. Gilt zusätzlich noch $R_n(x^{s+1}) \neq 0$, so hat Q_n den maximalen Exaktheitsgrad s:

$$deg(Q_n) = s \text{ oder auch } deg(R_n) = s$$

Falls aus dem Zusammenhang keine Mißverständnisse zu befürchten sind, werden wir den Index n für die Anzahl der Stützstellen auch weglassen.

Betrachten wir den Funktionenraum $C[a,b]$ mit der Maximumsnorm $\|f\|_\infty$, dann sind die Funktionale I, Q_n und R_n stetig und für die Abbildungsnormen der Funktionale gilt:

$$\|I\|_\infty = \int_a^b w(x)\,dx \tag{2.5}$$

$$\|Q_n\|_\infty = \sum_{i=1}^{n} |w_{i,n}| \tag{2.6}$$

$$\|R_n\|_\infty = \|I\|_\infty + \|Q_n\|_\infty \tag{2.7}$$

Durch geeignete Wahl der Gewichte können wir unabhängig von der Lage der Stützstellen Quadraturformeln mit Exaktheitsgrad $n-1$ konstruieren.

Sei $\omega_n(x) = (x - x_{1,n}) \cdots (x - x_{n,n})$ das zu Q_n gehörige Knotenpolynom und $l_{i,n}$ die Lagrangefaktoren zu den Stützstellen $x_{1,n}, \ldots, x_{n,n}$

$$l_{i,n}(x) = \frac{\omega_n(x_{i,n})}{(x - x_{i,n})\omega_n'(x_{i,n})}. \tag{2.8}$$

Legt man die Gewichte $w_{i,n}$ durch

$$w_{i,n} = I(l_{i,n}) \tag{2.9}$$

fest, erhält man einen interpolatorischen Näherungsoperator Q_n, dessen maximaler Exaktheitsgrad mindestens $n - 1$ ist. Umgekehrt ist klar, daß alle Funktionale der Bauart (2.2), deren Exaktheitsgrad gößer als $n - 1$, ist interpolatorisch sind.

2.2 Gauß-Quadratur

Die Idee der Gauß-Verfahren besteht darin, durch geeignete Wahl der Stützstellen Quadraturformeln zu konstruieren, die für einen Polynomraum maximalen Grades exakt sind. Die folgenden zwei Sätze stellen den Zusammenhang zwischen Exaktheitsgrad und Stützstellen her.

Satz 4 *Gegeben sei ein Funktional J auf $C[a,b]$ und ein interpolatorischer Näherungsoperator Q_n der Form (2.2), dann gilt für alle $m \in \mathbb{N}$:*

$$deg(Q_n) \geq n + m \Leftrightarrow J(\omega_n q) = 0 \text{ für alle } q \in \mathbb{R}^m[x] \tag{2.10}$$

Beweis: $f \in \mathbb{R}^{n+m}[x]$ läßt sich zerlegen in $f = q\,\omega_n + r$ mit $r \in \mathbb{R}^{n-1}[x]$ und $q \in \mathbb{R}^m[x]$, d. h.

$$\begin{aligned} J(f) - Q_n(f) &= J(q\,\omega_n + r) - Q_n(q\,\omega_n + r) \\ &= J(q\,\omega_n) + J(r) - Q_n(r) = J(q\,\omega_n) \end{aligned}$$

■

Ein Funktional J auf $C[a,b]$ nennen wir stark isoton, wenn für jede von der Nullfunktion verschiedene Funktion f, deren Funktionswerte nicht kleiner als Null sind, $J(f) > 0$ gilt.

Der nächste Satz stammt aus [86], in dem allgemeine Gauß-Verfahren ausführlicher behandelt werden.

Satz 5 (Gauß) *Es sei J ein stark isotones Funktional auf $C[a,b]$, dann gilt:*

1. Zu jeder natürlichen Zahl n gibt es genau einen Operator Q_n der Gestalt (2.2) mit der Eigenschaft

$$Q_n(f) = J(f) \quad \text{für alle } f \in \mathbb{R}^{2n-1}[x] \tag{2.11}$$

2. Es gibt keinen Operator der Form (2.2), für den gilt:

$$Q_n(f) = J(f) \quad \text{für alle } f \in \mathbb{R}^{2n}[x]$$

3. Die Stützstellen $x_{i,n}$ sind die Nullstellen des (bis auf einen Faktor eindeutig bestimmten) Polynoms $\omega_n(x)$, das hinsichtlich des Skalarproduktes $\langle f,g\rangle = J(fg)$ orthogonal zu allen Polynomen aus $\mathbb{R}^{n-1}[x]$ ist. Sie sind reell und paarweise verschieden.

4. Die Gewichte $w_{k,n}$ berechnen sich mittels einer Lagrangebasis $l_{i,n}(x)$ zu den in 3. angegebenen Stützstellen gemäß $w_{i,n} = J(l_{i,n})$ für $i = 1 \ldots n$.

Für nicht stark isotone Funktionale existieren auch Näherungsoperatoren mit höherem Exaktheitsgrad. Ein numerisch relevantes Beispiel sind die Hauptwertformeln, die im nächsten Kapitel genauer betrachtet werden.

Wählt man für J in Satz 5 gerade I aus (2.1) mit einer positiven Gewichtsfunktion, so ist klar, daß alle Voraussetzungen des Satzes 5 erfüllt sind. Das Skalarprodukt aus Satz 5 hat jetzt die Form:

$$\langle f,g\rangle = \int_a^b f(x)\, g(x)\, w(x)\, dx \tag{2.12}$$

und der Näherungsoperator Q_n ist gerade die Gauß-Quadraturformel. Die Gauß-Quadraturformeln haben nicht nur maximalen Exaktheitsgrad, sondern sie sind auch numerisch stabil, denn wegen

$$w_{k,n} = Q_n(l_{k,n}^2) = \int_a^b l_{k,n}^2(x) w(x)\, dx > 0$$

sind alle Gewichte positiv und nach dem Satz von Steklov und Fejér (vgl. [5]) konvergiert das Quadraturverfahren für alle Riemann-integrierbaren Funktionen gegen $I(f)$. Aus der Abschätzung des Interpolationsfehlers bei der Hermiteinterpolation (siehe z. B. [32]) oder über die Peano-Kerne (siehe z. B. Abschnitt 2.4) erhält man die bekannte Fehlerdarstellung der Gauß-Quadratur.

Satz 6 *Für $f \in C^{(2n)}[a,b]$ gilt die Fehlerabschätzung:*

$$R_n(f) = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b \omega_n^2(x) w(x)\, dx, \quad \xi \in (a,b)$$

2.2.1 Spezielle Gewichtsfunktionen

Tschebyscheff-Gewichtsfunktion

Die Tschebyscheff-Polynome erster Art T_n mit

$$T_n(x) = \cos(\arccos(nx)), \quad -1 \le x \le 1$$

sind im Intervall $[-1,+1]$ bzgl. der Gewichtsfunktion $w(x) = (1-x^2)^{-1/2}$ orthogonal, d. h. die Knoten der Gauß-Tschebyscheff-Quadratur sind die Nullstellen

$$x_{k,n} = \cos\left(\frac{2k-(2n+1)}{2n}\,\pi\right), \quad k = 1, 2, \ldots, n$$

der Polynome T_n. Aus den Knoten berechnet man die entsprechenden Gewichte $w_{k,n}$ zu $w_{k,n} = \pi/n$. Unter Verwendung von Satz 6 erhält man schließlich die angegebene Quadraturformel für $f \in C^{2n}[-1,1]$

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx = \frac{\pi}{n}\sum_{k=1}^{n} \cos\left(\frac{2k-(2n+1)}{2n}\,\pi\right) + \frac{\pi}{(2n)!\,2^{2n-1}} f^{(2n)}(\xi), \tag{2.13}$$

mit einer unbekannten Zwischenstelle $\xi \in (-1,1)$.

Legendre-Gewichtsfunktion

Die Gauß-Legendre-Quadraturformeln (kurz Legendre-Formeln) sind durch die Gewichtsfunktion $w(x) \equiv 1$ und das Funktional

$$I(f) = \int_{-1}^{1} f(x)\,dx \tag{2.14}$$

festgelegt. Die Stützstellen $x_{1,n} < \ldots < x_{n,n}$ sind die Nullstellen der Legendre-Polynome.

Satz 7 *Die Gauß-Legendre-Quadraturformeln sind symmetrisch. Für die nach aufsteigender Größe geordneten Stützstellen $x_{i,n}$ und die dazugehörigen Gewichte $w_{1,n}, \ldots, w_{n,n}$ gilt (siehe [5])*

$$w_{1,n} < w_{2,n} < \ldots < w_{n/2,n}, \qquad \textit{für gerades } n$$

$$w_{1,n} < w_{2,n} < \ldots < w_{(n+1)/2,n}, \quad \textit{für ungerades } n$$

Fehlerdarstellung: Die Fehlerdarstellung aus Satz 6 geht für $f \in C^{(2n)}[-1,1]$ über in:

$$R_n(f) = \frac{2^{2n+1}(n!)^4}{(2n+1)((2n)!)^3} f^{(2n)}(\xi), \quad \xi \in (-1,1) \tag{2.15}$$

2.3 Bestimmung der Knoten und Gewichte der Gauß-Quadratur

Die Stützstellen der Gauß-Quadraturformeln sind gerade Nullstellen orthogonaler Polynome. Deswegen geben wir in den nächsten beiden Abschnitten ein Verfahren zur verifizierten Einschließung der Nullstellen orthogonaler Polynome an, das auf einem Bisektionsalgorithmus beruht, der auch zur Eigenwertbestimmung von symmetrischen Tridiagonalmatrixen benutzt werden kann.

2.3.1 Bisektionsalgorithmus

In diesem Abschnitt bezeichne $\mathbb{R}[x]$ die Menge aller reellen Polynome bzw. $\mathbb{R}^n[x]$ die Menge aller Polynome vom Grad n oder kleiner. Zu einer zulässigen und nicht negativen Gewichtsfunktion w führen wir auf $\mathbb{R}[x]$ das innere Produkt

$$\langle p, q\rangle := \int_a^b p(x)q(x)w(x)\,dx, \quad p, q \in \mathbb{R}[x] \tag{2.16}$$

ein. Eine (endliche oder unendliche) Folge von Polynomen (p_n) $(n = 0, 1, \dots)$ nennen wir orthogonales Polynomsystem, wenn p_i den genauen Grad i hat und orthogonal zu p_j $(\langle p_i, p_j\rangle = 0)$ für $i \neq j$ ist. Die Polynome p_i sind bis auf einen Faktor eindeutig bestimmt. Sie können durch die Vorgabe der Hauptkoeffizienten k_i eindeutig festgelegt und mit Hilfe des folgenden Satzes (siehe [86]) konstruiert werden.

Satz 8 *Zu vorgegebenen nicht verschwindenden Hauptkoeffizienten $k_0, k_1, \dots$ können die dazugehörigen orthogonalen Polynome $p_0, p_1, \dots$ rekursiv bestimmt werden durch*

$$\begin{aligned} &p_{-1}(x) := 0, \quad p_0(x) = k_0, \\ &p_n(x) = (\alpha_n x - \beta_n)p_{n-1}(x) - \gamma_n p_{n-2}(x), \quad n \geq 1 \end{aligned} \tag{2.17}$$

mit den Rekursionskoeffizienten α_n, β_n und γ_n definiert durch

$$\begin{aligned} \alpha_n &= \frac{k_n}{k_{n-1}}, \qquad \beta_n = \frac{\alpha_n\langle xp_{n-1}, p_{n-1}\rangle}{\langle p_{n-1}, p_{n-1}\rangle}, \quad n \geq 1 \\ \gamma_n &= \frac{\alpha_n\langle p_{n-1}, p_{n-1}\rangle}{\alpha_{n-1}\langle p_{n-2}, p_{n-2}\rangle}, \quad n \geq 2. \end{aligned} \tag{2.18}$$

Ist die Gewichtsfunktion w symmetrisch, d. h. $w(x) = w(-x)$ und $a + b = 0$, dann sind die Polynome p_n abwechselnd gerade bzw. ungerade Funktionen, und die Rekursionskoeffizienten β_i verschwinden alle.
Für $w \equiv 1$ und $[a, b] = [-1, 1]$ erhält man z. B. die Legendre-Polynome, die durch $p_n(1) = 1$ eindeutig bestimmt sind. Die dazugehörige Rekursionsgleichung lautet

$$p_n(x) = \frac{2n-1}{n}\,x\,p_{n-1}(x) - \frac{n-1}{n}p_{n-2}(x). \tag{2.19}$$

In Abschnitt 2.3.3 werden wir für orthonormierte Polynome $\pi_0, \pi_1, \ldots$ die Rekursionsgleichung

$$\begin{aligned} &\pi_{-1}(x) = 0, \quad \pi_0(x) = k_0, \\ &x\,\pi_n(x) = \alpha_n \pi_{n+1}(x) + \beta_n \pi_n(x) + \alpha_{n-1}\pi_{n-1}(x), \quad n \geq 0 \end{aligned} \tag{2.20}$$

mit

$$\alpha_n = \frac{k_n}{k_{n+1}}, \quad \beta_n = \langle x\pi_n, \pi_n \rangle, \quad n \geq 0 \tag{2.21}$$

verwenden, die sich unmittelbar aus dem vorherigen Satz ergibt.

Die Nullstellen orthogonaler Polynome sind alle einfach und reell. Jedoch beobachtet man z. B. bei den Legendre-Polynomen, daß mit wachsendem n die Nullstellen an den Endpunkten des Intervalls $[-1, 1]$ deutlich dichter zusammenliegen als in der Mitte. Dies kann bei der numerischen Bestimmung der Nullstellen zu Schwierigkeiten führen, da sich die Nullstellen an den Rändern ähnlich wie mehrfache Nullstellen verhalten. Mit Hilfe *Sturmscher Ketten* läßt sich ein einfaches Kriterium für die Existenz von Nullstellen orthogonaler Polynome in einem halboffenen Intervall angeben (siehe z. B. [39, 86]).

Satz 9 *Es sei $p_0, p_1, \ldots, p_n$ eine endliche Menge orthogonaler Polynome und $vw(\xi)$ bezeichne die Anzahl der Vorzeichenwechsel (nach Streichen aller Nullen) der Folge*

$$p_0(\xi), p_1(\xi), \ldots, p_n(\xi).$$

Gilt für die Hauptkoeffizenten $k_0, k_1, \ldots, k_n$ der Polynome $p_0, p_1, \ldots, p_n$

$$k_0 > 0 \text{ und } k_i\, k_{i-1} < 0 \text{ für } i = 1, 2, \ldots, n\,,$$

dann ist die Anzahl der Nullstellen von p_n im halboffenen Intervall [x,y) gerade die Differenz $vw(y) - vw(x)$.

Eine direkte Folgerung des Satzes ist es, daß die Vorzeichenwechselfunktion in den Nullstellen von p_n Sprungstellen hat, auf dem Rest des Intervalls $[a, b]$ konstant ist. Beim Durchlaufen einer Nullstelle erhöht sich der Wert der Funktion vw um Eins. Darauf aufbauend ist der Algorithmus 2.1 formuliert, der die nach Größe geordneten Nullstellen $x_1, x_2, \ldots, x_n$ des orthogonalen Polynoms p_n einschließt. Voraussetzung hierfür ist ein Startintervall $[a, b]$, das alle Nullstellen enthält und eine Prozedur zur Bestimmung der Vorzeichenwechsel der Polynomkette. Zum Berechnen der Polynomkette kann, falls die Rekursionskoeffizienten bekannt sind, die Rekursionsgleichung (2.17) benutzt werden. Ausgegeben wird eine Einschließung $[x_\nu]$ von x_ν, deren Durchmesser kleiner als ϵ ist. Sind die Einschließungen $[x_\nu]$ alle so eng, daß sie nur eine Nullstelle enthalten, so können die Zeilen $x := a$ bzw. $[x, y] := [x, b]$ durch $y := a$ bzw. $[x, y] := [y, b]$ ersetzt werden.

Bisektionsalgorithmus
Gesucht: Nullstellen $x_1 < x_2 < \ldots < x_n$ von p_n
Gegeben: Startintervall $[a, b] \ni x_1, \ldots, x_n$

$x := a$
for $\nu := 1$ **to** n **do**
  $[x, y] := [x, b]$
  while $(|y - x| > \epsilon)$ **do**
    $m := (x + y)/2$
    if $(vw(m) < \nu)$ **then** $x := m$
      else $y := m$
    fi
  od
  Ausgabe $[x_\nu] := [x, y]$
od

Abbildung 2.1: Bisektionsalgorithmus

2.3.2 Einschließung der Stützstellen

Implementiert man den Bisektionsalgorithmus mit gewöhnlicher Gleitkommaarithmetik, so entstehen beim Auswerten der Rekursionsgleichung Rundungsfehler, die die Werte der Polynomkette so stark verfälschen können, daß nicht einmal die Anzahl der Vorzeichenwechsel korrekt bestimmt werden kann. Auf diese Weise kann es passieren, daß als Lösung des Bisektionsalgorithmus ein Intervall geliefert wird, das die gesuchte Nullstelle überhaupt nicht enthält, ohne daß diese Tatsache dem Ergebnis anzusehen wäre[1]. Deswegen ist ein zusätzlicher Verifikationsschritt nötig, der sicherstellt, daß das Ergebnis $[x_\nu] = [\underline{x}_\nu, \overline{x}_\nu]$ des Bisektionsalgorithmus auch tatsächlich eine Einschließung der gesuchten Nullstelle x_ν ist. Dazu überprüfen wir die hinreichende Bedingung $vw(\underline{x}_\nu) < \nu$ und $vw(\overline{x}_\nu) \geq \nu$. Die Anzahl der Vorzeichenwechsel $vw(\underline{x}_\nu)$ (bzw. $vw(\overline{x}_\nu)$) bestimmen wir, indem wir Intervalleinschließungen der entsprechenden Polynomkette berechnen. Ein intervallmäßiges Auswerten der Rekursiongleichung (2.17) führt im allgemeinen nicht zu zufriedenstellenden Ergebnissen. Insbesondere für große n oder Auswertestellen $\underline{x}_\nu$, die nahe an der Nullstelle x_ν liegen, entstehen dabei Intervalle, die sowohl negative als auch positive Werte enthalten. Eine zuverlässige Aussage über die Anzahl der Vorzeichenwechsel ist also nicht möglich. Der Einsatz einer Langzahlintervallarithmetik ist in solchen Fällen nur dann hilfreich, wenn die benötigten Rekursionskoeffizienten hinreichend genau bestimmt werden können. Ist diese Bestimmung nicht möglich oder zu zeitaufwendig, führen wir die Auswertung der Rekursionsgleichung

[1] Ein Beispiel hierfür wird in [85] gegeben

(2.17) an der Stelle x auf eine Vektoriteration

$$y_{i+1} = A_i y_i + d_i, \quad i \geq 0 \tag{2.22}$$

mit

$$y_i = \begin{pmatrix} p_{i-1}(x) \\ p_i(x) \end{pmatrix}, \quad A_i = \begin{pmatrix} 0 & 1 \\ -\gamma_{i+1} & \alpha_{i+1}x - \beta_{i+1} \end{pmatrix}, \quad d_i = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{2.23}$$

zurück. Die einfachste Möglichkeit y_{i+1} auszuwerten ist, in jedem Iterationsschritt y_{i+1} in einen Intervallvektor $[y_{i+1}]$ einzuschließen. Diese Vorgehensweise entspricht der intervallmäßigen Auswertung der Rekursionsgleichung (2.17). Dabei tritt der sogenannte *wrapping*-Effekt (siehe [47, 57]) auf, der dafür sorgt, daß die Intervalldurchmesser der Einschließungen rapide anwachsen. Um diesen Effekt zu vermeiden, speichert man die Zwischenergebnisse y_i nicht in Intervallvektoren, sondern in Parallelepipeden der Form

$$y_i \in P(B_i, [z_i], \tilde{y}_i) := \{\tilde{y}_i + B_i z : z \in [z_i]\}$$

mit geeigneten Basismatrizen B_i ab. Dies führt mit den Startwerten $[z_0] = 0$ und $B_0 = E$ (E bezeichne die Einheitsmatrix) zu folgender Iterationsvorschrift (siehe [47]).

1. Bestimme Näherungslösung $\tilde{y}_{i+1}$ gemäß $\tilde{y}_{i+1} = A_i\,\tilde{y}_i + d_i$
2. Auswahl einer nicht singulären Matrix B_{i+1} (siehe unten)
3. Berechne $[z_{i+1}] = B_{i+1}^{-1} A_i B_i [z_i] + d_i + A_i\,\tilde{y}_i - \tilde{y}_{i+1}$

Für die (exakten) Zwischenergebnisse y_{i+1} in (2.22) gilt

$$y_{i+1} \in \tilde{y}_{i+1} + B_{i+1}[z_{i+1}].$$

Bei der Wahl der Basismatrizen hat es sich bewährt, die QR-Zerlegung der Matrix $\tilde{B}_{i+1} = mid(\bigdiamond A_i B_i) = Q_{i+1} R_{i+1}$ zu bilden und danach $B_{i+1} \approx Q_{i+1}$ zu setzen. Durch ein Sortieren der Spalten von $\tilde{B}_{i+1}$ vor der $QR-$Zerlegung kann die Berechnung noch verbessert werden. Ist y_0 auf der Maschine nicht darstellbar, verwenden wir eine enge Intervalleinschließung $[y_0]$ von y_0 und setzen $\tilde{y}_0 = mid([y_0])$ und $[z_0] = [y_0] - \tilde{y}_0$. Für weitere Einzelheiten zur Vermeidung des *wrapping*-Effekts verweisen wir auf die Ausführungen in [47] und [58]. In Tabelle 2.1 vergleichen wir die bei Auswertung der Legendre-Polynome p_n an der Stelle $x = 0.9$ gewonnenen Einschließungen mit denen, die durch direkte Auswertung der Rekursionsgleichung (2.17) erzielt werden. Gerechnet wurde mit der in C-XSC integrierten, mehrfach genauen reellen und intervallmäßigen Staggered-Correction Arithmetik. Eine mehrfachgenaue reelle Zahl im Staggered-Correction Format besteht aus einem Vektor reeller Zahlen und repräsentiert den Wert der Summe aller Komponenten des Vektors. C–XSC verwendet für die Komponenten des Vektors reelle Gleitkommazahlen nach dem IEEE 754-1985 Standard. Die Dimension des Vektors kann durch den Benutzer über die Variable `stagprec` vom Typ `int` verändert werden. Ein mehrfachgenaues Intervall besteht aus einer mehrfachgenauen reellen Zahl der Länge `stagprec` - 1 und einem Maschinenintervall. Nähere Einzelheiten zur Definition und Implementierung einer Langzahlarithmetik im Staggered-Correction Format findet man in [55, 44].

	Direkte Auswertung		QR- Zerlegung	
k	$[p_k(x)]$	$d([p_k(x)])$	$[p_k(x)]$	$d([p_k(x)])$
	stagprec = 1			
5	$-4.11412\frac{49999}{50001}E-02$	$7.38E-15$	$-4.11412\frac{49999}{50001}E-02$	$1.68E-15$
10	$-2.631456178\frac{5}{6}E-01$	$3.07E-13$	$-2.631456178\frac{5}{6}E-01$	$6.89E-15$
15	$3.0519754214\frac{4}{2}E-01$	$1.44E-11$	$3.0519754213\frac{3}{2}E-01$	$3.44E-15$
20	$-1.4930823\frac{49}{57}E-01$	$7.13E-10$	$-1.493082353\frac{0}{1}E-01$	$1.33E-14$
25	$-6.83699\frac{2}{7}E-02$	$3.65E-08$	$-6.8369945\frac{89}{90}E-02$	$2.92E-14$
30	$2.0181\frac{4}{1}E-01$	$1.90E-06$	$2.018123979\frac{6}{5}E-01$	$2.33E-14$
35	$-1.7\frac{91}{89}E-01$	$1.01E-04$	$-1.789990452\frac{1}{2}E-01$	$1.34E-14$
40	$3.\frac{43}{97}E-02$	$5.37E-03$	$3.698741842\frac{2}{1}E-02$	$5.25E-14$
45	$[-0.031, 0.26]$	$2.89E-01$	$1.139831317\frac{3}{2}E-01$	$6.42E-14$
	stagprec = 2		*stagprec* = 1	
45	$1.139831317\frac{3}{2}E-01$	$4.16E-17$	$1.139831317\frac{3}{2}E-01$	$6.42E-14$
50	$-1.700376599\frac{4}{5}E-01$	$2.97E-15$	$-1.700376599\frac{4}{5}E-01$	$2.12E-14$
55	$1.019919747\frac{8}{7}E-01$	$1.63E-13$	$1.019919747\frac{8}{7}E-01$	$5.75E-14$
60	$3.17895924\frac{22}{12}E-02$	$8.91E-12$	$3.178959241\frac{8}{7}E-02$	$1.09E-13$
65	$-1.32537258\frac{1}{6}E-01$	$4.89E-10$	$-1.325372583\frac{4}{5}E-01$	$8.13E-14$
70	$1.321319\frac{4}{0}E-01$	$2.69E-08$	$1.321319194\frac{8}{7}E-01$	$2.70E-14$
75	$-3.803\frac{0}{3}E-02$	$1.48E-06$	$-3.8031562\frac{799}{800}E-02$	$1.32E-13$
80	$-7.7\frac{0}{2}E-02$	$8.19E-05$	$-7.707716268\frac{8}{9}E-02$	$1.58E-13$
85	$1.\frac{33}{28}E-01$	$4.54E-03$	$1.303220254\frac{1}{0}E-01$	$5.92E-14$
90	$[-0.22, 0.04]$	$2.52E-01$	$-8.74931768\frac{79}{80}E-02$	$1.10E-13$
95	$[-7.1, 7.0]$	$1.40E+01$	$-1.560641424\frac{5}{7}E-02$	$2.18E-13$
100	$[-390, 390]$	$7.78E+02$	$1.022658205\frac{6}{5}E-01$	$1.72E-13$

Tabelle 2.1: Auswertung der Legendre-Polynome an der Stelle x = 0.9 (k Polynomgrad)

Verbesserung der Einschließung durch das Newton-Verfahren

Der Bisektionsalgorithmus 2.1 halbiert in jedem Schritt das Einschließungsintervall, d. h. das Verfahren konvergiert linear. Ausgehend von einer hinreichend engen Starteinschließung kann man das Intervall-Newton-Verfahren (siehe Abschnitt 1.3) verwenden, um eine engere Einschließung zu erhalten. Aus der Rekursionsgleichung (2.17) gewinnt man durch Differentiation eine Rekursionsformel für die Ableitungen. Diese führen wir wieder zurück auf eine Vektoriteration $y_{i+1} = A_i y_i + d_i, i \geq 0$ mit

$$y_i = \begin{pmatrix} p'_{i-1}(x) \\ p'_i(x) \end{pmatrix}, \quad A_i = \begin{pmatrix} 0 & 1 \\ -\gamma_{i+1} & \alpha_{i+1}x - \beta_{i+1} \end{pmatrix}, \quad d_i = \begin{pmatrix} 0 \\ \alpha_n p_n(x) \end{pmatrix}.$$

Da die Matrix A_i die gleiche Form wie in (2.23) hat, wird die Bestimmung von $p_n(x)$ und $p'_n(x)$ parallel mit der oben beschriebenen Methode durchgeführt.
Zusammenfassend erhalten wir also folgendes Verfahren:

1. Bestimme Einschließung $[x_\nu]$ der Nullstelle x_ν mit dem Bisektionsverfahren 2.1 (gerechnet wird in Gleitkommaarithmetik bzw. reeller Langzahlarithmetik).

2. Verifikationsschritt: Falls gesuchte Nullstelle x_ν nicht in $[x_\nu]$ liegt, führe ersten Schritt mit höherer Genauigkeit durch.

3. Verbesserung der Einschließung mit dem Intervall-Newton-Verfahren.

Das obige Verfahren kann auch zur verifizierten Bestimmung der Eigenwerte symmetrischer Tridiagonalmatrizen verwendet werden, da deren charakteristischen Polynome ebenfalls Rekursionsgleichungen der Form (2.17) genügen.

2.3.3 Berechnung der Rekursionskoeffizienten

Bisher haben wir zu Auswertung orthogonaler Polynome ausschließlich die dazugehörige Rekursionsgleichung herangezogen. In [56] werden die Legendre-Polynome und deren Ableitungen mit Hilfe der Formel von Rodriguez

$$P_n(x) = \frac{(-1)^n}{2^n\, n!} \frac{d^n}{dx^n}(1 - x^2)^n$$

und automatischer Differentiation ausgewertet. Die Nullstellen der Legendre-Polynome werden mit dem erweiterten Intervall-Newton-Verfahren bestimmt. Für beliebige Gewichtfunktionen sind aber weder geschlossene Darstellungen der Rekursionskoeffizienten noch die Formeln von Rodriguez verfügbar. Prinzipiell können die Rekursionskoeffizenten direkt aus den Gleichungen (2.18) und den einfachen Momenten μ_k

$$\mu_k = \int_a^b x^k w(x)\, dx, \quad k = 0, 1, \ldots, 2n \tag{2.24}$$

berechnet werden. Für die praktische Bestimmung ist dieser Ansatz jedoch wenig geeignet (siehe [85]). Stattdessen verwendet man besser einen der beiden folgenden Ansätze.

Methode von Gautschi

Die erste Methode, die Rekursionskoeffizienten zu bestimmen, stammt von Gautschi [29]. Ausgehend von den gewöhnlichen Momenten μ_k bildet man die Gramsche Matrix M der Monome $1, x, \ldots, x^n$

$$M_n = \begin{pmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_n \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{n+1} \\ \mu_2 & \mu_3 & \mu_4 & \cdots & \mu_{n+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mu_n & \mu_{n+1} & \mu_{n+2} & \cdots & \mu_{2n} \end{pmatrix}. \tag{2.25}$$

Da die Matrix M positiv definit ist, existiert eine obere Dreiecksmatrix $R = (r_{ij})$ mit $M = R^T R$ (Cholesky-Zerlegung). Die Inverse $\Pi = (\pi_{ij}) = R^{-1}$ von R ist ebenfalls eine obere Dreiecksmatrix. Aus der Gleichung $(R^{-1})^T M R^{-1} = E$ erkennt man, daß die Polynome π_j

$$\pi_j(x) := \sum_{k=0}^{j} \pi_{k,j}\, x^k, \quad j = 0, \ldots, n$$

ein orthonormiertes Polynomsystem bilden, das einer Rekursionsgleichung der Form (2.20) genügt. Ein Vergleich der Koeffizienten ergibt

$$\alpha_j = \frac{r_{j+1,j+1}}{r_{j,j}}, \quad \beta_j = \frac{r_{j,j+1}}{r_{j,j}} - \frac{r_{j-1,j}}{r_{j-1,j-1}}, \quad j = 0, \ldots, n-1 \tag{2.26}$$

mit $r_{-1,-1} := 1$ und $r_{-1,0} := 0$. In [23] und [26] zeigt Gautschi, daß die Berechnung der Knoten und Gewichte aus den gewöhnlichen Momenten mit wachsendem n zunehmend schlechter konditioniert ist. Deswegen empfiehlt er zur Berechnung modifizierte Momente zu verwenden und gibt hierfür eine Variante obiger Methode an.

Methode von Sack und Donovan

Diese Idee verbirgt sich auch hinter dem Algorithmus, der in [74] angegeben wird. Sei $(p_i)_i$ ein nicht unbedingt orthogonales Polynomsystem, das die Gleichung

$$x p_j(x) = \tilde{\alpha}_j p_{j+1}(x) + \tilde{\beta}_j p_j(x) + \tilde{\gamma}_{j-1} p_{j-1}(x), \quad j \geq 0 \tag{2.27}$$

mit $p_{-1} := 0$ erfüllt. Aus den modifizierten Momenten

$$\mu_k = \int_a^b p_k(x)\, w(x)\, dx$$

sollen die Rekursionsgleichung

$$x\pi_j(x) = \alpha_j \pi_{j+1}(x) + \beta_j \pi_j(x) + \alpha_{j-1} \pi_{j-1}(x)$$

der orthonormierten Polynome π_j bestimmt werden.
Es ist bekannt, daß die Nullstellen von π_n die Eigenwerte der symmetrischen Tridiagonalmatrix

$$A_n = \begin{pmatrix} \beta_0 & \alpha_0 & & & \\ \alpha_0 & \beta_1 & \alpha_1 & & 0 \\ & \alpha_1 & \ddots & \ddots & \\ & 0 & \ddots & \beta_{n-2} & \alpha_{n-2} \\ & & & \alpha_{n-2} & \beta_{n-1} \end{pmatrix} \tag{2.28}$$

sind. Die Elemente der Matrix $A_n - \lambda E$ sind die Integrale

$$\int_a^b \pi_i(x)\pi_j(x)(x-\lambda)w(x)\,dx, \quad i,j = 0,\ldots,n-1. \tag{2.29}$$

Jedes π_k läßt sich als Linearkombination der Polynome $p_0,\ldots,p_k$ darstellen. Es existiert also eine untere Dreiecksmatrix $Q_n = (q_{ij})$ mit nicht verschwindenden Diagonalelementen

$$(\pi_0(x),\ldots,\pi_{n-1}(x))^T = Q_n(p_0(x),\ldots,p_{n-1}(x))^T. \tag{2.30}$$

Definiert man die Matrizen $M_n = (\mu_{ij})$ und $\tilde{M}_n = (\tilde{\mu}_{ij})$ durch

$$\begin{aligned} \mu_{ij} &= \int_a^b p_i(x)\,p_j(x)\,w(x)\,dx \\ \tilde{\mu}_{ij} &= \int_a^b p_i(x)\,p_j(x)\,x\,w(x)\,dx \end{aligned}$$

für $0 \le i,j \le n-1$, so gilt

$$A_n - \lambda E = Q_n(\tilde{M}_n - \lambda M_n)Q_n^T. \tag{2.31}$$

Da Q_n und M_n nicht singulär sind, besitzen die Matrizen A_n und $M_n^{-1}\tilde{M}_n$ dieselben Eigenwerte. Ähnlich wie A_n definiert man

$$\tilde{A}_n = \begin{pmatrix} \tilde{\beta}_0 & \tilde{\gamma}_1 & & & \\ \tilde{\alpha}_0 & \tilde{\beta}_1 & \tilde{\gamma}_2 & & 0 \\ & \tilde{\alpha}_1 & \ddots & \ddots & \\ & 0 & \ddots & \tilde{\beta}_{n-2} & \tilde{\gamma}_{n-1} \\ & & & \tilde{\alpha}_{n-2} & \tilde{\beta}_{n-1} \end{pmatrix} \tag{2.32}$$

Wegen Gleichung (2.27) gilt $M_n^{-1}\tilde{M}_n = \tilde{A}_n + M_n^{-1}R_n$ mit einer Matrix R_n, die mit Ausnahme der letzten Spalte nur aus Nullen besteht. Unter Berücksichtigung von $Spur(A_n) = Spur(M_n^{-1}\tilde{M}_n)$ bzw. $Spur(A_n^2) = Spur((M_n^{-1}\tilde{M}_n)^2)$ erhält man durch

einfache Umformungen (siehe [74]) folgende Gleichungen für die Elemente $s_{i,j}$ der oberen Dreiecksmatrix $S := (\langle \pi_i, p_j \rangle / \langle \pi_i, p_i \rangle)$

$$\begin{aligned} s_{i+1,j} &= \frac{\tilde{\alpha}_i}{\alpha_i^2}((\tilde{\beta}_j - \sigma_i)s_{i,j} + \tilde{\alpha}_j s_{i,j+1} + \tilde{\gamma}_{j-1} s_{i,j-1} - \alpha_{i-1} s_{i-1,j}) \\ \beta_i &= \tilde{\alpha}_i s_{i,i+1} + \tilde{\beta}_i - \tilde{\alpha}_{i-1} s_{i-1,i} \\ \alpha_i^2 &= \tilde{\alpha}_i((\tilde{\beta}_{i+1} - \beta_i)s_{i,i+1} + \tilde{\alpha}_{i+1} s_{i,i+2} - \tilde{\alpha}_{i-1} s_{i-1,i+1} + \tilde{\gamma}_i). \end{aligned} \tag{2.33}$$

Ausgehend von den Startwerten $s_{-1,j} = 0$ und $s_{0,j} = \mu_j/\mu_0$ berechnet man mit Hilfe der Gleichungen (2.33) die Elemente $s_{i,j}$ für $j = i, \ldots, i-j-1$ zeilenweise.

$$\begin{array}{ccccccccccc} s_{0,0} & s_{0,1} & s_{0,2} & \cdots & \cdots & \cdots & \cdots & s_{0,2n-3} & s_{0,2n-2} & s_{0,2n-1} \\ & s_{1,1} & s_{1,2} & \cdots & \cdots & \cdots & \cdots & s_{1,2n-3} & s_{1,2n-2} & \\ & & s_{2,2} & \cdots & \cdots & \cdots & \cdots & s_{2,2n-3} & & \\ & & & \cdots & \cdots & \cdots & \cdots & & & \\ & & & & s_{n-1,n-1} & s_{n-1,n} & & & & \end{array}$$

Die Koeffizienten β_i $(i = 0, \ldots, n-1)$ und α_i^2 $(i = 0, \ldots, n-2)$ sind gerade die Rekursionskoeffizienten der orthogonalen Polynome q_j, deren Hauptkoeffizienten alle Eins sind.

$$q_j(x) = (x - \beta_{j-1})q_{j-1} - \alpha_{j-2}^2 q_{j-2}(x)$$

Wählt man für die Polynome p_j die Monome x^j, treten die gleichen Instabilitäten wie bei der Methode von Gautschi mit gewöhnlichen Momenten auf. Da bei der Berechnung der s_{ij} nur die letzten zwei Zeilen abgespeichert werden müssen, reduziert sich jedoch der Speicherbedarf im Vergleich zum Verfahren von Gautschi. Außerdem benötigt das Verfahren von Sack und Donovan weniger arithmetische Operationen, insbesonders entfällt die Wurzelbildung, ein Aspekt, der besonders bei der Rechnung mit Langzahlarithmetik günstig zu Buche schlägt.

Der entscheidende Vorteil bei der Verwendung modifizierter Gewichte ist, daß bei geeigneter Wahl der Polynome p_j sich die Berechnung der Matrix A_n wesentlich stabiler gestaltet, was wir an einem Beispiel verdeutlichen wollen. Wir bestimmen die Rekursionskoeffizienten orthogonaler Polyome zur logarithmischen Gewichtsfunktion $w(x) = \ln(1/x)$. Dazu berechnen wir die modifizierten Momente (siehe [7])

$$\mu_k = \int_0^1 p_k(x) \ln(x^{-1})\, dx = \begin{cases} 1 & \text{für } k = 0, \\ \frac{(-1)^k}{k(k+1)} & \text{für } k \geq 1 \end{cases},$$

der auf das Intervall $[0, 1]$ transformierten Legendre-Polynome p_k mit der Rekursionsgleichung

$$\begin{aligned} p_0(x) &= 1, \quad p_1 = 2x - 1 \\ (k+1)p_{k+1}(x) &= (2k+1)(2x-1)p_k(x) - k\, p_{k-1}(x), \quad k \geq 2. \end{aligned}$$

	Koeffizientenbestimmung mittels stagprec = 2			
	modifizierter Momente		gewöhnlicher Momente	
k	$d([\beta_{k-1}]$	$d([\alpha_{k-2}]$	$d([\beta_{k-1}]$	$d([\alpha_{k-2}]$
2	$6.78E-32$	$3.08E-33$	$8.63E-32$	$3.08E-33$
3	$4.41E-31$	$4.01E-32$	$1.04E-30$	$6.89E-32$
4	$2.15E-30$	$1.94E-31$	$1.32E-29$	$8.13E-31$
5	$1.03E-29$	$9.48E-31$	$2.06E-28$	$1.18E-29$
6	$4.85E-29$	$4.49E-30$	$3.91E-27$	$2.04E-28$
7	$2.29E-28$	$2.12E-29$	$9.55E-26$	$4.45E-27$
8	$1.08E-27$	$1.00E-28$	$2.93E-24$	$1.24E-25$
9	$5.14E-27$	$4.75E-28$	$1.03E-22$	$4.16E-24$
10	$2.44E-26$	$2.25E-27$	$7.94E-21$	$1.51E-22$
11	$1.16E-25$	$1.07E-26$	$7.66E-19$	$2.40E-20$
12	$5.53E-25$	$5.11E-26$	$8.33E-16$	$8.00E-17$
13	$2.64E-24$	$2.44E-25$	$8.72E-15$	$7.84E-16$
14	$1.26E-23$	$1.17E-24$	$8.37E-14$	$6.54E-15$
15	$6.03E-23$	$5.57E-24$	$1.12E-12$	$6.78E-14$
16	$2.89E-22$	$2.67E-23$	$2.47E-11$	$1.18E-12$
17	$1.38E-21$	$1.28E-22$	$7.37E-10$	$3.17E-11$
18	$6.64E-21$	$6.13E-22$	$2.48E-08$	$1.03E-09$
19	$3.19E-20$	$2.94E-21$	$8.75E-07$	$3.57E-08$
20	$1.53E-19$	$1.41E-20$	$3.15E-05$	$1.27E-06$

Tabelle 2.2: Bestimmung der Rekursionskoeffizienten

Die gewöhnlichen Momente berechnen sich gemäß

$$\int_0^1 x^n \ln(x^{-1})\,dx = \frac{1}{(1+k)^2}, \quad k \geq 0. \tag{2.34}$$

Die Ergebnisse der Tabelle 2.2 zeigen, daß die Berechnung der Rekursionskoeffizienten aus den modifizierten Momenten mit einer deutlich geringeren Mantissenlänge auskommt. Tabelle 2.3 enthält die aus den Rekursionskoeffizienten berechneten (stagprec = 2) Knoten und Gewichte.
Theoretisch ist es zwar möglich, die modifizierten Momente aus den gewöhnlichen Momenten zu berechnen, denn für $\nu_{ij} := \langle p_i, x^j \rangle$ gilt

$$\nu_{i+1,j} = \tilde{\alpha}_i^{-1}(\nu_{i,j+1} - \tilde{\beta}_i \nu_{i,j} - \tilde{\gamma}_i \nu_{i-1,j}). \tag{2.35}$$

Hat man jedoch durch die Verwendung eines geeigneten Polynomsystems $(p_i)_i$ eine Verbesserung der Konditionszahlen des Problems erreicht, stellt sich heraus, daß sich die

Berechnung der modifizierten Momente aus den gewöhnlichen Momenten sehr schwierig gestaltet. Deshalb ist man darauf angewiesen, einfachere Formeln zur Berechnung der modifizierten Momente zu kennen. In der Literatur werden solche z. B. in ([25, 22]) angegeben.

2.3.4 Einschließung der Gewichte

In Abschnitt 2.2 haben wir schon erwähnt, daß die Nullstellen x_i Eigenwerte der Matrix A_n sind. Setzt man $\Pi(x) := (\pi_0(x), \ldots, \pi_{n-1}(x))^T$ und $E_n := (0, \ldots, 0, 1)^T$, so sieht man dies an der Gleichung

$$x\,\Pi(x) = A_n\,\Pi(x) + \alpha_{n-1}\,\pi_n(x)E_n. \tag{2.36}$$

Für die Nullstelle $x_{i,n}$ und das entsprechende Gewicht $w_{i,n}$ von π_n $(i = 1, 2, \ldots, n)$ gilt

$$1 = w_{i,n}\,\Pi(x_{i,n})^T\Pi(x_{i,n}) = (\sqrt{w_{i,n}}\,\Pi(x_{i,n}))(\sqrt{w_{i,n}}\,\Pi(x_{i,n})^T). \tag{2.37}$$

Ist $y = (y_0, y_1, \ldots, y_{n-1})^T$ ein zum Eigenwert $x_{i,n}$ gehöriger Eigenvektor mit $y\,y^T = 1$, dann gilt

$$y_0 = \sqrt{w_{i,n}}\pi_0(x_{i,n}) = \sqrt{w_{i,n}}(\int_a^b w(x)dx)^{-0.5}. \tag{2.38}$$

Damit ist das zu $x_{i,n}$ gehörige Gewicht $w_{i,n}$ gegeben durch

$$w_{i,n} = y_0^2 \int_a^b w(x)dx \tag{2.39}$$

Prinzipiell kann also jeder Algorithmus, der die Eigenvektoren und Eigenwerte einer symmetrischen Tridiagonalmatrix bestimmt, verwendet werden. Liefert der Algorithmus nur Näherungen der Eigenwerte, so können um diese enge Intervalle so gelegt werden, daß die Vorzeichenwechselbedingung erfüllt sind. Für die verifizierte Bestimmung der Gewichte hat man dann verschiedene Möglichkeiten. Eine Möglichkeit besteht z. B. darin, die als bereits bekannt vorausgesetzte Quadraturformel Q_n mit n Stützstellen zu verwenden. Da diese für Polynome vom Grad $2n-1$ exakt ist, gilt

$$w_{i,2n} = \int_a^b l_{i,2n}(x)w(x)\,dx = \sum_{k=1}^{n} w_{k,n}l_{i,2n}(x_{i,n}). \tag{2.40}$$

Alternativ können wir auch auf eine der zwei Gleichungen

$$w_{i,n} = \frac{k_n}{k_{n-1}}\,\frac{\langle p_{n-1}, p_{n-1}\rangle}{p_n'(x_{i,n})p_{n-1}(x_{i,n})} \tag{2.41}$$

$$w_{i,n} = \left(\sum_{k=0}^{n-1}\frac{p_k(x_{i,n})p_k(x_{i,n})}{\langle p_k, p_k\rangle}\right)^{-1} \tag{2.42}$$

$w(x) = \ln(1/x)$				
j	$x_{j,20}$	$d(x_{j,20})$	$w_{j,20}$	$d(w_{j,20})$
1	$2.588327955921955428^{6}_{0}E-03$	$5.64E-22$	$4.31427521332080785^{9}_{7}E-02$	$9.80E-21$
2	$1.520966234956023^{24}_{19}E-02$	$3.10E-21$	$7.5383709908589359^{7}_{4}E-02$	$2.03E-19$
3	$3.853655037216532^{64}_{56}79E-02$	$7.56E-21$	$9.3053267451663051^{6}_{1}E-02$	$3.72E-19$
4	$7.218161381587390^{5}_{2}64E-02$	$1.49E-20$	$1.01456711849829754^{9}_{0}E-01$	$7.51E-19$
5	$1.154605264876331505^{7}_{5}E-01$	$1.48E-20$	$1.032017620560720^{95}_{86}6E-01$	$7.28E-19$
6	$1.674428562753296857^{3}_{0}E-01$	$2.11E-20$	$1.0002254980527316^{8}_{5}E-01$	$1.20E-18$
7	$2.269837872602025033^{8}_{5}E-01$	$1.81E-20$	$9.325979930029767^{9}_{7}E-02$	$1.06E-18$
8	$2.92754960941545^{301}_{298}83E-01$	$1.85E-20$	$8.402895287194105^{7}_{6}E-02$	$9.09E-19$
9	$3.6327742985785890^{5}_{2}45E-01$	$1.93E-20$	$7.328558913003074^{2}_{0}E-02$	$6.73E-19$
10	$4.369571400907683^{50}_{47}18E-01$	$2.11E-20$	$6.1850336913730^{903}_{896}2E-02$	$5.16E-19$
11	$5.121225946789673^{21}_{18}36E-01$	$2.07E-20$	$5.0416604438374677^{8}_{5}E-02$	$2.11E-19$
12	$5.870640449144099151^{5}_{2}E-01$	$2.45E-20$	$3.9551370005298385^{5}_{2}E-02$	$2.17E-19$
13	$6.60073413314909^{42}_{38}41E-01$	$2.88E-20$	$2.96940778958128^{50}_{46}4E-02$	$2.26E-19$
14	$7.2948408392968749^{9}_{5}88E-01$	$2.28E-20$	$2.1156315355427097^{8}_{6}E-02$	$1.35E-19$
15	$7.937096719870858177^{6}_{3}E-01$	$1.93E-20$	$1.4123732938964020^{5}_{3}E-02$	$9.50E-20$
16	$8.5128089278912572^{4}_{1}72E-01$	$1.59E-20$	$8.6609745043354986^{6}_{0}E-03$	$5.08E-20$
17	$9.0087968085441759^{3}_{1}42E-01$	$1.30E-20$	$4.7199401462036049^{8}_{3}E-03$	$3.14E-20$
18	$9.413697491290916^{31}_{29}76E-01$	$9.44E-21$	$2.1513974039652061^{3}_{0}E-03$	$1.83E-20$
19	$9.7182274107526319^{42}_{35}37E-01$	$5.42E-21$	$7.1972821465320264^{9}_{3}E-04$	$4.73E-21$
20	$9.9153808143871197^{4}_{1}265E-01$	$1.85E-21$	$1.204276763302167^{73}_{66}41E-04$	$5.48E-23$

Tabelle 2.3: Knoten und Gewichte bei logarithmischer Gewichtsfunktion

zurückgreifen, die sich aus der Christoffel-Darboux-Identität

$$\sum_{i=0}^{n} \frac{p_i(x)p_i(y)}{\langle p_i, p_i\rangle} = \frac{k_n}{k_{n+1}} \frac{p_{n+1}(x)p_n(y) - p_n(x)p_{n+1}(y)}{(x-y)\langle p_n, p_n\rangle} \tag{2.43}$$

und den Rekursionsformeln (2.17) herleiten lassen (vgl. [82]).
Die Gewichte der Tabelle 2.3 wurden aus Gleichung (2.42) mit stagprec = 2 berechnet.

2.4 Peanosche Restdarstellung

Um Aussagen über die Qualität eines Quadraturverfahrens treffen zu können, benötigen wir eine geeignete Darstellung oder zumindest eine Abschätzung des Restfunktionals R_n. Hierzu existiert in der Literatur eine Vielzahl von Ansätzen (vgl. [5], [15]). In diesem Abschnitt wollen wir uns auf die Fehlerdarstellung von Peano konzentrieren, die es erlaubt, den Fehler als Integral des Produkts des Peano-Kernes und eines Ableitungsterms darzustellen. Die in Abschnitt 2.4.1 beschriebene Einschließung der Integrationskonstanten beruht im wesentlichen auf der Vorgehensweise in [81].

Zur einfacheren Notation definieren wir zunächst die abgeschnittenen Polynome $(x-t)_+^m$ durch:

$$(x-t)_+^m = \begin{cases} (x-t)^m & \text{für} \quad x \geq t, \\ 0 & \text{für} \quad x < t. \end{cases} \tag{2.44}$$

Den Satz von Peano können wir nun folgendermaßen formulieren.

Satz 10 (Peano) *Es sei Q_n eine Quadraturformel mit $deg(R) \geq m-1$ und f eine reellwertige Funktion, deren m-te Ableitung stetig ist. Dann gilt für $s = 1, \ldots, m$*

$$R_n(f) = \int_a^b f^{(s)}(t) K_s(t)\, dt\,, \tag{2.45}$$

wobei K_s den $s-ten$ Peano-Kern[2]

$$K_s(t) = R_n\left(\frac{(\cdot - t)_+^{(s-1)}}{(s-1)!}\right)$$

von R_n bezeichnet.

Nachstehende Eigenschaften der Peano-Kerne finden sich für den Fall $w = 1$ in [5].

[2]Streng genommen müsste man $K_{s,n}$ schreiben, um die Abhängigkeit von der Stützstellenanzahl zu betonen. Solange aber keine Verwechslungsgefahr besteht, werden wir der Übersichtlichkeit zuliebe auf den Index n verzichten.

Satz 11 *Die Peano-Kerne $K_s\,(s > 1)$ einer Quadraturformel Q_n haben folgende Eigenschaften:*

1. $K_s(a) = K_s(b) = 0.$
2. $(s-1)!\,K_s(t) = \int_t^b (x-t)^{s-1}\,w(x)\,dx - \sum_{i=1}^{n} w_i(x_i - t)_+^{s-1}$
3. $(s-1)!\,K_s(t) = -\int_a^t (x-t)^{s-1}\,w(x)\,dx - (-1)^s \sum_{i=1}^{n} w_i(t - x_i)_+^{s-1}$
4. $K_s \in C^{(s-2)}[a,b]$
5. *Ist Q_n symmetrisch und $w(x) = w(a+b-x)$, so gilt $K_s(t) = (-1)^s K_s(b+a-t)$*
6. *Es ist $K_{s+1}(t) = -\int_a^t K_s(x)dx$*

Definition 12 *Ein Funktional J auf $C^{(k)}[a,b]$ heißt definit von der Ordnung k, wenn aus $f^{(k)} \geq 0$ folgt $J(f) \neq 0$. J heißt positiv (negativ) definit von der Ordnung k, falls aus $f^{(k)} \geq 0$ folgt $J(f) \geq 0$ $(J(f) \leq 0)$.*

Passend zu Definition 12 nennen wir eine Funktion f definit, falls sie keinen Vorzeichenwechsel hat und positiv (bzw. negativ) definit, wenn $f \geq 0$ (bzw. $f \leq 0$). Aus der Darstellung (2.45) sieht man, daß R genau dann definit von der Ordnung j ist, wenn der Peano-Kern K_j definit ist. Der Zusammenhang zwischen R und der entsprechenden Kernfunktion wird im folgenden Satz noch präzisiert.

Satz 13 *K_s ist auf dem Intervall $[a,b]$ genau dann definit, wenn gilt:*

$$R(f) = \frac{R(x^s)}{s!} f^{(s)}(\xi), \quad \xi \in [a,b] \tag{2.46}$$

Beweis: Die Notwendigkeit der Bedingung (2.46) folgt aus dem erweiterten Mittelwertsatz der Integralrechnung, die Umkehrung aus der Definitheit des Restfunktionals R. ■

Für definite Peano-Kerne K_s lässt sich nach Satz 13 der Fehler $R(f)$ darstellen als

$$R(f) = f^{(s)}(\xi) \int_a^b K_s(x)\,dx. \tag{2.47}$$

Wegen (2.46) ist klar, daß nur Peano-Kerne K_s maximaler Ordnung, $(s = deg(Q) + 1)$ positiv definit sein können. Um eine vergleichbare Darstellung des Fehlers für nicht definite Peano-Kerne K_s zu erhalten, zerlegen wir $[a,b]$ in zwei disjunkte Mengen K_s^+ und K_s^-, wobei K_s^+ den Teil des Definitionsbereiches $[a,b]$ von K_s bezeichnet, für den K_s positiv und K_s^- den Teil für den K_s negativ ist.

$$\begin{aligned} K_s^+ &= \{x \in [a,b] \,:\, K_s(x) > 0\} \\ K_s^- &= \{x \in [a,b] \,:\, K_s(x) < 0\} \end{aligned}$$

Da K_s nur endlich viele Nullstellen hat, sind K^+ und K^- gerade die Vereinigung endlich vieler offener oder halboffener disjunkter Intervalle I_i bzw. J_i

$$\begin{aligned} K_s^+ &= \bigcup_{i=1}^{n^+} I_i \\ K_s^- &= \bigcup_{i=1}^{n^-} J_i \end{aligned}$$

und es gilt

$$\begin{aligned} R(f) &= \int_a^b f^{(s)}(x) K_s(x)\,dx \\ &= \int_{K_s^+} f^{(s)}(x) K_s(x)\,dx + \int_{K_s^-} f^{(s)}(x) K_s(x)\,dx \\ &= \sum_{i=1}^{n^+} \int_{I_i} f^{(s)}(x) K_s(x)\,dx + \sum_{i=1}^{n^-} \int_{J_i} f^{(s)}(x) K_s(x)\,dx \\ &= \sum_{i=1}^{n^+} f^{(s)}(\xi_i) \int_{I_i} K_s(x)\,dx + \sum_{i=1}^{n^-} f^{(s)}(\zeta_i) \int_{J_i} K_s(x)\,dx \end{aligned}$$

mit geeigneten Zwischenstellen $\xi_i, \zeta_i \in [a,b]$. Ersetzt man ξ_i und ζ_i durch das Intervall $[a,b]$, so ergibt sich

$$\begin{aligned} R(f) &\in \sum_{i=1}^{n^+} f^{(s)}([a,b]) \int_{I_i} K_s(x)\,dx + \sum_{i=1}^{n^-} f^{(s)}([a,b]) \int_{J_i} K_s(x)\,dx \\ &= f^{(s)}([a,b]) \sum_{i=1}^{n^+} \int_{I_i} K_s(x)\,dx + f^{(s)}([a,b]) \sum_{i=1}^{n^-} \int_{J_i} K_s(x)\,dx \\ &= f^{(s)}([a,b]) \int_{K^+} K_s(x)\,dx + f^{(s)}([a,b]) \int_{K^-} K_s(x)\,dx. \end{aligned} \tag{2.48}$$

Die Integrale aus (2.48) hängen zwar von den Stützstellen, den Gewichten und der Gewichtsfunktion, jedoch nicht von der zu integrierenden Funktion ab. Deshalb bezeichnen wir sie als Integrationskonstanten[3] und kürzen sie mit c_s^+ bzw. c_s^- ab:

$$\begin{aligned} c_s^+ &:= \int_{K_s^+} K_s(x)\,dx \\ c_s^- &:= \int_{K_s^-} K_s(x)\,dx \end{aligned}$$

Die Fehlerdarstellung (2.48) hat jetzt die kurze und prägnante Form[4]

$$\begin{aligned} R(f) &= c_s^+ f^{(s)}(\xi) + c_s^- f^{(s)}(\zeta) \\ &\in c_s^+ f^{(s)}([a,b]) + c_s^- f^{(s)}([a,b]) \end{aligned} \tag{2.49}$$

mit geeigneten $\xi, \zeta \in [a,b]$.

[3] In der Literatur (vgl. [16]) wird die Konstante $c_s^+ - c_s^-$ auch Peano-Konstante genannt.

[4] Eine ähnliche Herleitung dieser Fehlerdarstellung wird u. a. in [81] gegeben.

2.4.1 Einschließung der Integrationskonstanten

Zur Bestimmung der Integrationskonstanten c^+ und c^- zerlegen wir das Intervall $[a,b]$ in endlich viele Teilintervalle $Z_1, Z_2, \ldots, Z_m$ mit $\bigcup_{k=1}^{m} Z_k = [a,b]$. Der Schnitt zweier Intervalle Z_i und Z_j $(i \neq j)$ soll dabei leer sein oder höchstens einen Randpunkt enthalten. Dann bestimmen wir für jedes Teilintervall Z das Integral P_k des positiven Anteils und das Integral N_k des negativen Anteils der Kernfunktion über Z_k.

$$P_{Z_k} := \int_{Z_k \cap K^+} K(t)\,dt \tag{2.50}$$

$$N_{Z_k} := \int_{Z_k \cap K^-} K(t)\,dt \tag{2.51}$$

Zum Schluß addieren wir die entsprechenden Anteile auf und erhalten so die gesuchten Werte der Integrationskonstanten.

$$c^+ = \sum_{k=1}^{m} P_{Z_k} \qquad c^- = \sum_{k=1}^{m} N_{Z_k}$$

Ausgehend von den Einschließungen $[x_1] = [\underline{x}_1, \overline{x}_1], \ldots, [x_n]$ der Stützstellen $x_1, \ldots, x_n$ betrachten wir die Zerlegung $\{a, \underline{x}_1, \overline{x}_1, \ldots, \underline{x}_n, \overline{x}_n, b\}$. Hierbei setzen wir voraus, daß die Stützstellen der Größe nach sortiert, die Einschließungen der Stützstellen paarweise disjunkt und die zugrundeliegende Quadraturformel offen ist, d. h. $x_1 \neq a$ und $x_n \neq b$ sind. Bei vorliegender Zerlegung muß über folgende Teilintervalle integriert werden:

- **Integration über Einschließungen der Stützstellen**
  Sei $[x_j] = [\underline{x}_j, \overline{x}_j]$ das Integrationsintervall: Nach Satz 11 gilt:

$$K(x) = \begin{cases} K_l(x) & \text{für } x \in [\underline{x}_j, x_j] \\ K_r(x) & \text{für } x \in [x_j, \overline{x}_j] \end{cases}$$

  mit

$$K_l(x) = \int_t^b \frac{(x-t)^{s-1}}{(s-1)!}\,w(t)\,dt - \frac{1}{(s-1)!}\sum_{i=j}^{n} w_i (x_i - t)^{s-1}$$

$$K_r(x) = \int_t^b \frac{(x-t)^{s-1}}{(s-1)!}\,w(t)\,dt - \frac{1}{(s-1)!}\sum_{i=j+1}^{n} w_i (x_i - t)^{s-1}$$

  Falls K auf $[x_j]$ positiv ist, gilt nach dem erweiterten Mittelwertsatz der Integralrechnung:

$$P_{[x_j]} = \int_{[x_j] \cap K^+} K(t)\,dt = \int_{[x_j]} K(t)\,dt \in W_K([x_j])\,(\overline{x}_j - \underline{x}_j)$$

Falls K auf $[x_j]$ einen Vorzeichenwechsel hat, so gilt ebenfalls nach dem Mittelwertsatz der Integralrechnung:

$$\begin{aligned} P_{[x_j]} &= \int_{[x_j]\cap K^+} K(t)\,dt \quad \in \quad W_K([x_j])\,(\overline{x}_j - \underline{x}_j) \\ N_{[x_j]} &= \int_{[x_j]\cap K^-} K(t)\,dt \quad \in \quad W_K([x_j])\,(\overline{x}_j - \underline{x}_j) \end{aligned}$$

Zur Bestimmung von $W_K([x_j])$ verwenden wir:

$$W_K([x_j]) \subset K_l([\underline{x}_j, x_j]) \cup K_r([x_j, \overline{x}_j]) \subset K_l([x_j]) \cup K_r([x_j]) \subset K_l([x_j])$$

- **Integration über Bereiche zwischen zwei Stützstelleneinschließungen**
  Nun sei $[\overline{x}_j, \underline{x}_{j+1}]$ das Teilintervall, über das integriert werden soll. Falls wir $0 \in W_K([\overline{x}_j, \underline{x}_{j+1}])$ ausschließen könen (z. B. durch Bestätigen der Bedingung $0 \notin K([\overline{x}_j, \underline{x}_{j+1}])$, dann berechnen wir das Integral mit Hilfe der Stammfunktion. Ansonsten verwenden wir das erweiterte Intervall-Newton-Verfahren, das als Ergebnis eine Menge von Intervallen liefert, auf denen K möglicherweise eine Nullstelle hat. Diese Teilintervalle werden ganz analog zu den Einschließungen der Stützstellen verarbeitet. Für die verbleibenden Intervalle ist garantiert, daß K keinen Vorzeichenwechsel hat. Zur Berechnung des positiven (bzw. negativen) Anteils P_y (bzw. N_y) auf einem solchen Intervall $[\underline{y}, \overline{y}]$ können wir Satz 11 Punkt 6 heranziehen, es gilt nämlich für $1 \le s \le deg(Q)$ (o. B. d. A. K positiv auf $[\underline{y}, \overline{y}]$)

$$\begin{aligned} P_y &= \int_{[\underline{y},\overline{y}]\cap K^+} K_s(t)\,dt = \int_{\underline{y}}^{\overline{y}} K_s(t)\,dt = -\left[K_{s+1}(t)\right]_{\underline{y}}^{\overline{y}} \\ &= \frac{1}{s!}\int_{\underline{y}}^{b} (x-\underline{y})^s w(x)\,dx - \frac{1}{s!}\int_{\overline{y}}^{b} (x-\overline{y})^s w(x)\,dx + \\ &\quad \frac{1}{s!}\sum_{i=j+1}^{n} w_i((x_i-\overline{y})^s - (x_i-\underline{y})^s) \end{aligned} \tag{2.52}$$

- **Integration über die Randbereiche**
  Aus Satz 11 Punkt 2 (bzw. Punkt 3) folgt unmittelbar, daß K auf $[a, \underline{x}_1]$ und $[\overline{x}_n, b]$ keine Vorzeichenwechsel hat, d. h. wir können zur Berechnung wieder auf die Stammfunktion zurückgreifen. Für den linken Rand $[a, \underline{x}_1]$ gilt für $s \le deg(Q)$:

$$P_{[a,\underline{x}_1]} = \begin{cases} \dfrac{1}{s!}\displaystyle\int_a^{\underline{x}_1} (x-\underline{x}_1)^s w(x)\,dx & \text{für } s \text{ gerade} \\ 0 & \text{für } s \text{ ungerade} \end{cases}$$

$$N_{[a,\underline{x}_1]} = \begin{cases} \dfrac{1}{s!}\displaystyle\int_a^{\underline{x}_1} (x-\underline{x}_1)^s w(x)\,dx & \text{für } s \text{ ungerade} \\ 0 & \text{für } s \text{ gerade} \end{cases}$$

Und analog für den rechten Teilbereich $[\overline{x}_n, b]$:

$$P_{[\overline{x}_n,b]} = \frac{1}{s!}\int_{\overline{x}_n}^{b}(x-\overline{x}_n)^s w(x)\,dx \tag{2.53}$$

$$N_{[\overline{x}_n,b]} = 0$$

Für $1 < s \leq deg(Q)$ genügt es wegen

$$0 = K_{s+1}(b) = -\int_a^b K_s(x)\,dx$$

nur den positiven Teil c^+ zu berechnen, da $c^+ + c^- = 0$ gilt. Für $s = deg(Q)+1$ gilt dies nicht. Erfüllt die Quadraturformel und die Gewichtsfunktion die Voraussetzungen von Satz 11 Punkt 5 (Symmetrieeigenschaften), dann ist der Kern für gerade s symmetrisch und es ist wegen

$$2\int\limits_{[a,\frac{a+b}{2}]\cap K^+} K(x)\,dx = 2\int\limits_{[\frac{a+b}{2},b]\cap K^+} K(x)\,dx = \int\limits_{[a,b]\cap K^+} K(x)\,dx$$

ausreichend, die Kernfunktion nur auf einem der Teilintervalle $[a, \frac{a+b}{2}]$ oder $[\frac{a+b}{2}, b]$ zu untersuchen.
Im folgenden beschäftigen wir uns mit zwei speziellen Gewichtsfunktionen der Gauß-Quadratur, nämlich der Legendre-Gewichtsfunktion $w = 1$ und der nicht symmetrischen Gewichtsfunktion $w = x^{-\frac{1}{2}}$.

Legendre-Gewichtsfunktion

Zunächst betrachten wir die Legendre-Gewichtsfunktion $w \equiv 1$ auf dem Intervall $[-1, 1]$. Die Peano-Kerne K_s haben die Darstellung

$$K_s(t) = \frac{(1-t)^s}{s!} - \frac{1}{(s-1)!}\sum_{i=1}^{n} w_i (x_i - t)_+^{s-1} \tag{2.54}$$

mit den Nullstellen x_i der Legendre-Polynome. Da die zugrundeliegende Quadraturformel Q symmetrisch ist, gilt für gerade s: $K_s(t) = K_s(-t)$ bzw. für ungerade s: $K_s(t) = -K_s(-t)$. In den Abbildungen 2.2 und 2.3 sind die Funktionen $s!K_s(t)$ der Gauß-Legendre Quadratur für verschiedene Werte s graphisch dargestellt.
Für die intervallmäßige Auswertung der Kernfunktion beim erweiterten Intervall-Newton-Verfahren empfiehlt es sich, die Mittelwertform (1.1) (siehe [9]) zu verwenden, da hierdurch die Laufzeit zur Berechnung der Integrationskonstanten und die Qualität der Einschließungen deutlich verbessert werden. Da die Integrationskonstanten jedoch üblicherweise einmal vorab berechnet und dann tabelliert werden, spielt die Laufzeit nur eine untergeordnete Rolle. Aus dem Schaubild 2.4 wird ersichtlich, daß die Integrationskonstante c_s^+ mit wachsendem n (bei festem s) monoton fallen. Halten wir jedoch

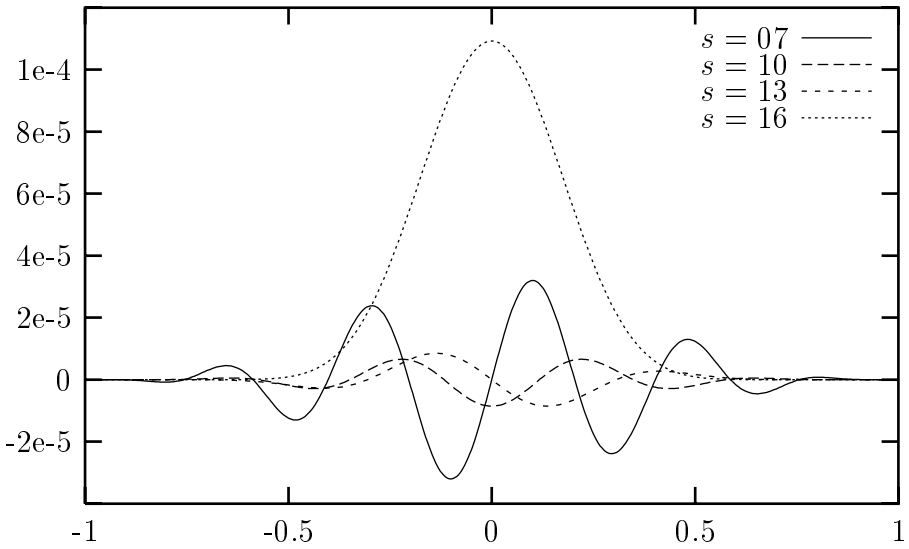


Abbildung 2.2: $s!\,K_s(t)$ (Gauß-Legendre $n = 8$)

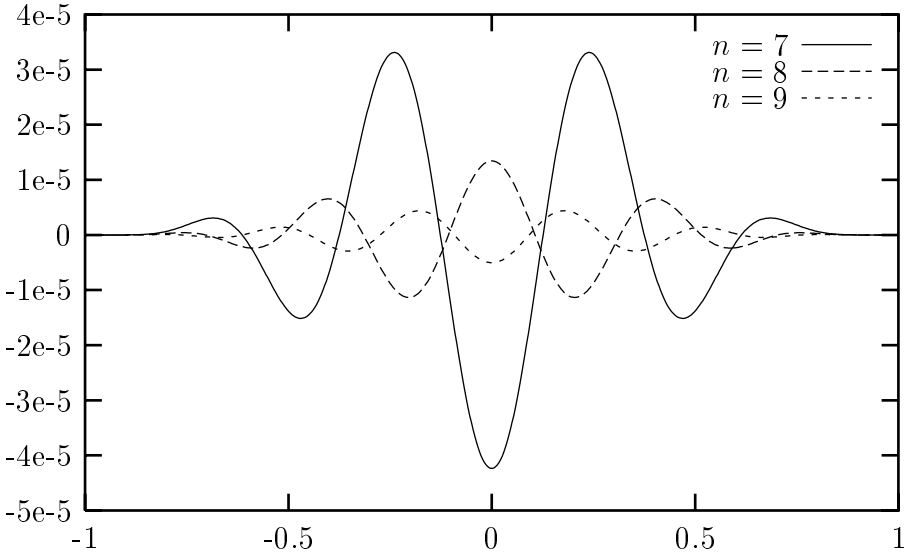


Abbildung 2.3: $6!\,K_6(t)$ (Gauß-Legendre)

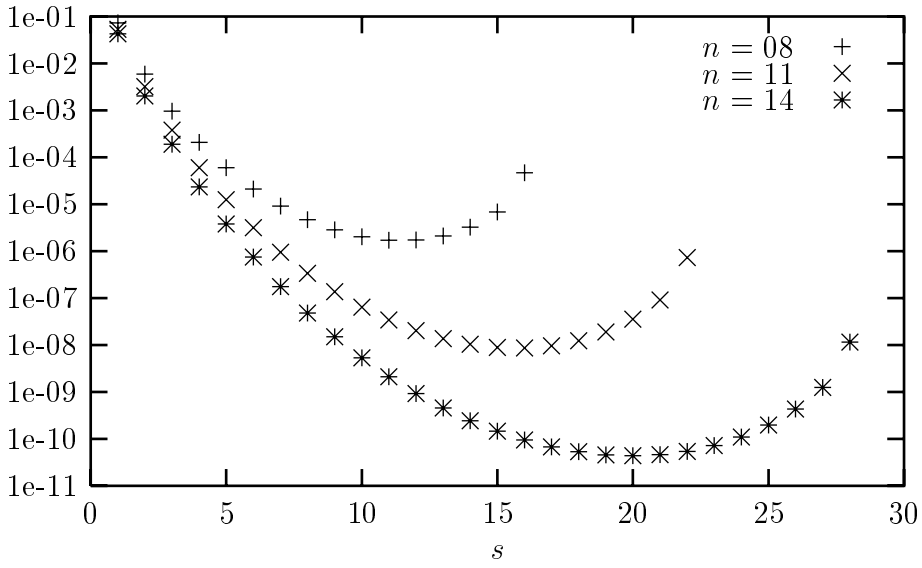


Abbildung 2.4: Integrationskonstanten $s!\,c_s^+$ (Gauß-Legendre)

n fest und variieren s, so sehen wir sofort, daß die Integrationskonstanten $s!c_s^+$ für größere s wieder ansteigen. Dies ist mit einer der Gründe, warum es nicht immer sinnvoll ist, für die Fehlerabschätzung den Peano-Kern höchster Ordnung zu verwenden. Die Integrationskonstanten c_s^+ und c_s^- sind für $s = 1, \ldots, 2n-1$ betragsmäßig gleich und für $s = 2n$ gilt $c_{2n}^- = 0$, da die entsprechende Kernfunktion definit ist (siehe [5]). Einschließungen der Integrationskonstanten c_s^+ und c_s^- finden sich in ([81, 59, 9]).
Die Peano-Konstanten c_s berechnen sich für $1 \le s \le deg(Q)$ aus

$$c_s = \int_{-1}^{1} |K_s(t)|\,dt = 2 \int\limits_{[-1,1]\cap K^+} K_s(t)\,dt = 2\,c_s^+ \tag{2.55}$$

und für $s = deg(Q) + 1$ aus

$$c_s = c_s^+ = \frac{2^{2n+1}(n!)^4}{(2n+1)((2n)!)^3}. \tag{2.56}$$

Die letzte Gleichung folgt unmittelbar aus (2.15).

Gewichtsfunktion $w = x^{-\frac{1}{2}}$

Nun betrachten wir die nicht symmetrische Gewichtsfunktion $w = x^{-\frac{1}{2}}$ auf dem Intervall $[0, 1]$. Zum Berechnen der Peano-Kerne verwenden wir folgendes Lemma.

Lemma 14 *Für $s \in I\!N$ gilt:*

$$\sum_{k=0}^{s-1} \left(\begin{array}{c} s-1 \\ k \end{array} \right) (-1)^k \frac{2}{2k+1} = \frac{\sqrt{\pi}\,\Gamma(s)}{\Gamma(s+0.5)} \tag{2.57}$$

Beweis: Für $s \in I\!N$ gilt:

$$\begin{aligned}
0 &= 2(1-1)^s = 2\sum_{k=0}^{s} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k \frac{2k+1}{2k+1} \\
&= 2\sum_{k=0}^{s} \left(\begin{array}{c} s \\ k \end{array} \right) k(-1)^k \frac{2}{2k+1} + \sum_{k=0}^{s} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k \frac{2}{2k+1} \\
&= 2\sum_{k=0}^{s-1} \left(\begin{array}{c} s \\ k \end{array} \right) k(-1)^k \frac{2}{2k+1} + \sum_{k=0}^{s} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k \frac{2}{2k+1} + 2s\,(-1)^s \frac{2}{2s+1} \\
&= -2\sum_{k=0}^{s-1} \left(\begin{array}{c} s \\ k \end{array} \right) (s-k)(-1)^k \frac{2}{2k+1} + (2s+1)\sum_{k=0}^{s} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k \frac{2}{2k+1}
\end{aligned}$$

Und aus

$$\begin{aligned}
2s\sum_{k=0}^{s-1} \left(\begin{array}{c} s-1 \\ k \end{array} \right) (-1)^k \frac{2}{2k+1} &= 2s\sum_{k=0}^{s-1} \left(\begin{array}{c} s \\ k \end{array} \right) \frac{s-k}{s} (-1)^k \frac{2}{2k+1} \\
&= (2s+1)\sum_{k=0}^{s} \left(\begin{array}{c} s \\ k \end{array} \right) (-1)^k \frac{2}{2k+1}
\end{aligned}$$

folgt die Behauptung durch vollständige Induktion nach s. ■

Mithilfe des vorhergehenden Lemmata und Satz 11 Punkt 3 erhalten wir folgende Formeln für die Peano-Kerne:

$$\begin{aligned}
(s-1)!\,K_s(t) &= (-1)^s(\alpha_s(t) - \beta_s(t)) \qquad (2.58) \\
\alpha_s(t) &= \int_0^1 \frac{(t-x)_+^{s-1}}{\sqrt{x}}\,dx \\
&= \sum_{k=0}^{s-1} \left(\begin{array}{c} s-1 \\ k \end{array} \right) (-1)^k\, t^{s-1-k} \left[\frac{2x^{k+\frac{1}{2}}}{2k+1} \right]_0^t \\
&= t^{s-\frac{1}{2}} \sum_{k=0}^{s-1} \left(\begin{array}{c} s-1 \\ k \end{array} \right) (-1)^k \frac{2}{2k+1} \\
&= t^{s-\frac{1}{2}} \frac{\sqrt{\pi}\,\Gamma(s)}{\Gamma(s+\frac{1}{2})} \\
\beta_s(t) &= \sum_{k=1}^{n} w_k (t-x_k)_+^{s-1}
\end{aligned}$$

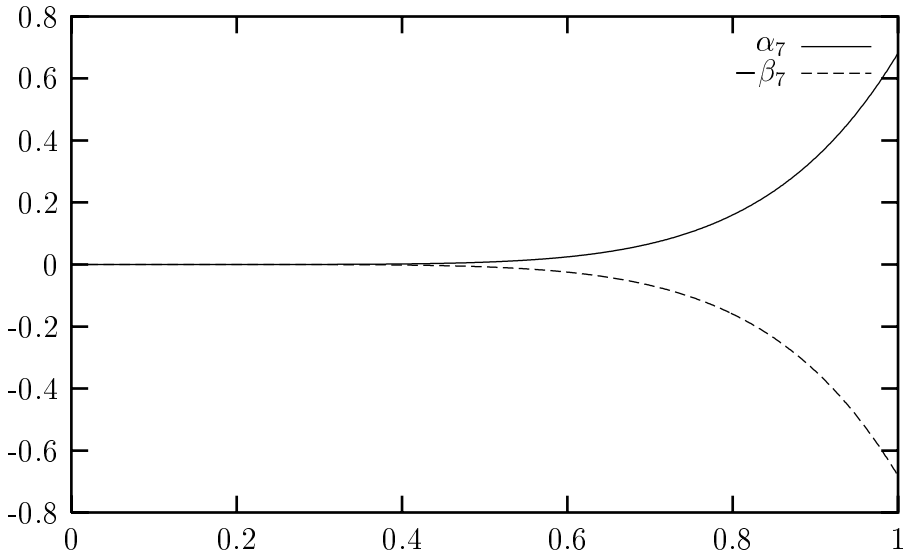


Abbildung 2.5: $\alpha_7(t)$ und $-\beta_7(t)$

Verwenden wir Punkt 2 aus Satz 11 erhalten wir die alternative Darstellung:

$$
\begin{aligned}
(s-1)!\,K_s(t) &= \gamma_s(t) - \delta_s(t) \qquad (2.59)\\
\gamma_s(t) &= \int_0^1 \frac{(x-t)_+^{s-1}}{\sqrt{x}}\,dx\\
&= \sum_{k=0}^{s-1} \binom{s-1}{k} (-t)^{s-k-1} \frac{2}{2k+1}\left(1 - t^{k+\frac{1}{2}}\right)\\
&= \sum_{k=0}^{s-1} \binom{s-1}{k} (-t)^{s-k-1} \frac{2}{2k+1} + (-1)^s\,\alpha_s(t)\\
\delta_s(t) &= \sum_{k=1}^{n} w_k (x_k - t)_+^{s-1}
\end{aligned}
$$

Zur intervallmäßigen Auswertung von K_s empfiehlt es sich zunächst Darstellung (2.58) zu verwenden, da die Auswertung von γ_s deutlich aufwendiger als die von α_s ist. Es stellt sich aber heraus, daß die Auswertung für Intervalle, die nahe am rechten Integrationsrand liegen, problematisch ist. Dies liegt daran, daß $\alpha_s(t)$, $\beta_s(t)$ und deren Ableitungen positiv und monoton wachsend auf dem Intervall $(0,1]$ sind (siehe auch Abbildung 2.5). Wertet man $\alpha_s - \beta_s$ intervallmäßig auf $[\underline{t},\overline{t}]$ aus, so bedeutet dies, daß wegen

$$\alpha_s([\underline{t},\overline{t}]) - \beta_s([\underline{t},\overline{t}]) \quad \supseteq \quad [\alpha_s(\underline{t}) - \beta_s(\overline{t}), \alpha_s(\overline{t}) - \beta_s(\underline{t})]$$

der Durchmesser der Intervallauswertung größer wird, wenn $[\underline{t},\overline{t}]$ näher an 1 liegt. Dem Schaubild 2.6 entnehmen wir aber, daß die Kernfunktionen K_s zum Rand hin immer

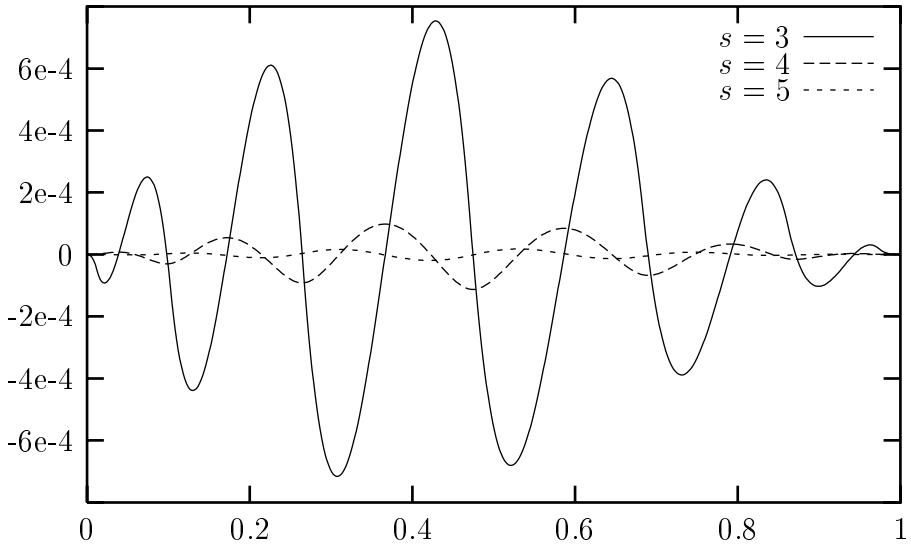


Abbildung 2.6: $s!\,K_s$ (Gewichtsfunktion $w = x^{-\frac{1}{2}}, n = 7$)

flacher werden. Hieraus resultieren numerische Schwierigkeiten bei der Bestimmung der Nullstellen des Peano-Kernes durch das erweiterte Intervall-Newton-Verfahren. Mit zunehmendem s wächst die Anzahl der potentiellen Intervalle, die eine Nullstelle von K_s enthalten, drastisch an. Auch der Einsatz einer Langzahlarithmetik bringt keine entscheidende Besserung, da das eigentliche Problem in der Überschätzung des Wertebereichs der Kernfunktion bei der intervallmäßigen Auswertung liegt. Um ein unkontrolliertes Anwachsen der Anzahl der Intervalle, in denen die Kernfunktion eventuell eine Nullstelle besitzt, beim erweiterten Intervall-Newton-Verfahren zu verhindern, darf der minimale Durchmesser der Suchintervalle nicht zu klein gewählt werden. Da alle Intervalle, für die nicht ausgeschlossen werden kann, daß die Kernfunktion in ihnen eine Nullstelle hat, wie die Einschließungsintervalle der Stützstellen behandelt werden, führt dies letztendlich zu groben Einschließungen der Integrationskonstanten. Eine bessere Lösung ist es, zur Auswertung zusätzlich Darstellung (2.59) heranzuziehen. Die Funktionen γ_s, δ_s sind nicht negativ und monoton fallend auf dem Intervall $[0, 1]$. Ähnliche Überlegungen wie oben verdeutlichen, daß Darstellung (2.59) zwar nicht zur intervallmäßigen Auswertung von K_s für Intervalle, die nahe am linken Rand liegen, geeignet ist, jedoch deutlich bessere Ergebnisse als Darstellung (2.58) für Intervalle am rechten Rand liefert. Voraussetzung hierfür ist jedoch, daß wir zur Einschließung des Wertebereichs von K_s folgende Relation

$$W_{K_s}([\underline{t}, \overline{t}]) \subseteq \frac{1}{(s-1)!}[\gamma_s(\overline{t}) - \delta_s(\underline{t}), \gamma_s(\underline{t}) - \delta_s(\overline{t})]$$

d. h. auf der Maschine den Ausdruck

$$\frac{1}{(s-1)!}[\inf(\gamma_s([\overline{t},\overline{t}])\diamondsuit\delta_s([\underline{t},\underline{t}])),\sup(\gamma_s([\underline{t},\underline{t}])\diamondsuit\delta_s([\overline{t},\overline{t}]))]$$

verwenden, um den Wertebereich von K_s über $[\underline{t},\overline{t}]$ einzuschließen. Es ist offensichtlich, daß gilt

$$\frac{1}{(s-1)!}[\gamma_s(\overline{t})-\delta_s(\underline{t}),\gamma_s(\underline{t})-\delta_s(\overline{t})]\subseteq\frac{1}{(s-1)!}(\gamma_s([\underline{t},\overline{t}])-\delta_s([\underline{t},\overline{t}])).$$

Daß diese Einschließung in der Regel tatsächlich deutlich engere Schranken liefert, sehen wir an der Tatsache, daß die Summe in γ_s Summanden mit alternierenden Vorzeichen enthält. Die Berechnung der Einschließung auf dem Rechner erfordert jetzt aber den doppelten Aufwand, da wir γ_s und δ_s zweimal intervallmäßig auswerten müssen. In Tabelle 2.4 sind die so berechneten Einschließungen für $n = 11$ aufgelistet. Die Berechnungen wurden unter Verwendung einer istaggered-Arithmetik (stagprec = 2) in C–XSC durchgeführt. Ein Vergleich der Integrationskonstanten $s!c_s^+$ für $n = 5, 8, 11$ in Schaubild 2.7 zeigt einen ähnlichen Verlauf wie bei der Gauß-Legendre Gewichtsfunktion. Wiederum zeigt sich, daß eine Erhöhung der Stützstellenzahl bei festem s zu kleineren Integrationskonstanten führt, was umgekehrt bei einer Erhöhung von s bei festem n nicht garantiert ist. Insbesondere sind die Integrationskonstanten maximaler Ordung ($s = 2n$) nicht minimal.

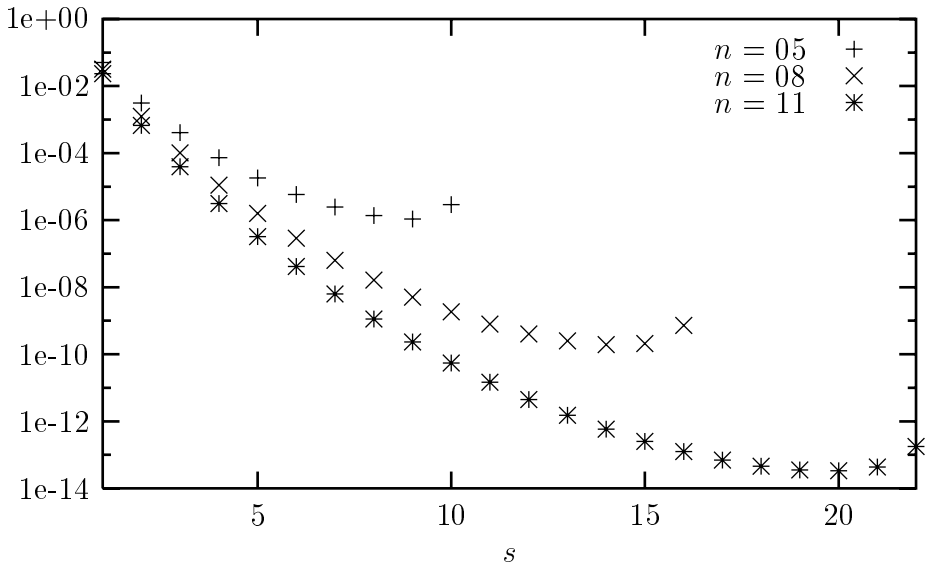


Abbildung 2.7: Integrationskonstanten $s!\,c_s^+$ $(w = x^{-\frac{1}{2}})$

s	$s![c_s^-]$	$d(s![c_s^-])$	$s![c_s^+]$	$d(s![c_s^+])$
1	$-2.33004712015^{19}_{27}e-02$	$7.5e-15$	$2.33004712015^{28}_{18}e-02$	$9.0e-15$
2	$-6.70110110^{57}_{60}e-04$	$2.9e-14$	$6.70110110^{61}_{56}e-04$	$4.0e-14$
3	$-3.936716950^{4}_{7}e-05$	$2.1e-15$	$3.936716950^{7}_{3}e-05$	$2.8e-15$
4	$-3.111809086^{17}_{23}e-06$	$4.7e-17$	$3.111809086^{23}_{17}e-06$	$4.4e-17$
5	$-3.23237575^{06}_{10}e-07$	$2.9e-17$	$3.23237575^{10}_{07}e-07$	$2.7e-17$
6	$-4.12167986^{5}_{7}e-08$	$1.5e-17$	$4.12167986^{8}_{5}e-08$	$1.9e-17$
7	$-6.2973803^{3}_{6}e-09$	$1.3e-17$	$6.2973803^{6}_{4}e-09$	$8.5e-18$
8	$-1.12713313^{0}_{7}e-09$	$5.9e-18$	$1.1271331^{5}_{2}e-09$	$1.3e-17$
9	$-2.330084^{2}_{4}e-10$	$1.2e-17$	$2.330084^{30}_{24}e-10$	$5.3e-18$
10	$-5.505067^{2}_{7}e-11$	$4.0e-18$	$5.50506^{9}_{7}e-11$	$1.3e-17$
11	$-1.47569^{5}_{7}e-11$	$1.2e-17$	$1.47569^{7}_{5}e-11$	$3.6e-18$
12	$-4.46^{6998}_{7003}e-12$	$3.5e-18$	$4.46^{701}_{699}e-12$	$9.6e-18$
13	$-1.52313^{0}_{9}e-12$	$8.6e-18$	$1.52313^{8}_{3}e-12$	$3.8e-18$
14	$-5.8482^{1}_{5}e-13$	$2.7e-18$	$5.848^{3}_{2}e-13$	$8.6e-18$
15	$-2.533^{62}_{70}e-13$	$7.4e-18$	$2.5336^{8}_{4}e-13$	$3.4e-18$
16	$-1.244^{49}_{53}e-13$	$3.0e-18$	$1.244^{55}_{46}e-13$	$7.2e-18$
17	$-6.9^{899}_{907}e-14$	$6.2e-18$	$6.990^{5}_{1}e-14$	$3.0e-18$
18	$-4.554^{0}_{5}e-14$	$3.7e-18$	$4.554^{6}_{0}e-14$	$5.1e-18$
19	$-3.52^{35}_{42}e-14$	$5.3e-18$	$3.52^{40}_{36}e-14$	$3.1e-18$
20	$-3.374^{1}_{6}e-14$	$4.3e-18$	$3.374^{5}_{0}e-14$	$3.7e-18$
21	$-4.32^{37}_{43}e-14$	$4.6e-18$	$4.32^{42}_{38}e-14$	$3.3e-18$
22	0	0	$1.766^{11}_{05}e-13$	$5.1e-18$

Tabelle 2.4: Integrationskonstanten $s!\,c_s^+$ für $w = x^{-\frac{1}{2}}$ und $n = 11$

2.5 Adaptive Quadratur

Wollen wir qualitativ hochwertige Aussagen, also enge Einschließungen oder zumindest gute Näherungswerte eines Integrals, gewinnnen, so sind die bisher besprochenen Quadraturverfahren nicht ausreichend. Zum einen ist die Konvergenz der Näherungsterme nur für positive Quadraturverfahren gewährleistet, und zum anderem ist die Berechnung der Restglieder für größere n sehr unhandlich. Hinzu kommt noch, daß die entsprechenden Stützstellen, Gewichte und Peano-Konstanten zur Laufzeit zur Verfügung stehen müssen. Lösung hierfür bieten adaptive Verfahren, bei denen das Integrationsintervall geeignet zerlegt und auf jedes Teilintervall eine passende Quadraturformel angewendet wird. Der entscheidende Vorteil adaptiver Verfahren besteht darin, daß sich die Zerlegung dem Verlauf der Funktion anpassen läßt, was zu einer deutlichen Reduzierung des Aufwands führt.

2.5.1 Affin transformierte Quadraturformel

Gegeben sei eine Quadraturformel der Form

$$\int_{\tilde{a}}^{\tilde{b}} f(x)\,dx = Q(f) + R(f) \tag{2.60}$$

für das Referenzintervall $[\tilde{a}, \tilde{b}]$. Durch die lineare Transformation $x = m_c\, t + c$ mit

$$m_c = \frac{b-a}{\tilde{b}-\tilde{a}} \quad \text{und} \quad c = \frac{\tilde{b}\,a - \tilde{a}\,b}{\tilde{b}-\tilde{a}} \quad , \tag{2.61}$$

die das Intervall $[\tilde{a}, \tilde{b}]$ auf $[a, b]$ abbildet, führt man Integrale mit beliebigen endlichen Grenzen a und b auf ein Integral mit den Grenzen $\tilde{a}$, $\tilde{b}$ und $\tilde{f}(x) = f(m_c\, x + c)$ über.

$$\int_a^b f(x)\,dx = m_c \int_{\tilde{a}}^{\tilde{b}} \tilde{f}(x)\,dx \tag{2.62}$$

Auf das rechtsstehende Integral können wir jetzt die Quadraturformel (2.60) anwenden.

Definition 15 *Es sei Q eine Quadraturformel auf dem Referenzintervall $[\tilde{a}, \tilde{b}]$*

$$Q(f) = \sum_{\nu=1}^{n} w_\nu f(x_\nu)$$

Die Quadraturformel $\tilde{Q}$ auf dem Intervall $[a, b]$

$$\tilde{Q}(f) = \frac{b-a}{\tilde{b}-\tilde{a}}\, Q(\tilde{f}) = \sum_{\nu=1}^{n} \tilde{w}_\nu f(\tilde{x}_\nu)$$

mit

$$\begin{aligned} \tilde{x}_\nu &= \frac{b-a}{\tilde{b}-\tilde{a}}\, x_\nu + \frac{a\tilde{b} - b\tilde{a}}{\tilde{b}-\tilde{a}} \\ \tilde{w}_\nu &= \frac{b-a}{\tilde{b}-\tilde{a}}\, w_\nu \end{aligned} \tag{2.63}$$

wird affin transformierte Quadraturformel von Q (auf $[a,b]$) genannt.

Hat das zu Q gehörige Restfunktional R die Darstellung

$$R(f) = c_m^+ f^{(m)}(\xi_1) + c_m^- f^{(m)}(\xi_2)$$

mit $\xi_1, \xi_2 \in [\tilde{a}, \tilde{b}]$, so ist $\tilde{R}$, das zu $\tilde{Q}$ gehörige Restfunktional, durch

$$\tilde{R}(f) = \frac{b-a}{\tilde{b}-\tilde{a}}\, R(\tilde{f}) = \left(\frac{b-a}{\tilde{b}-\tilde{a}}\right)^{m+1} (c_m^+ f^{(m)}(\tilde{\xi}_1) + c_m^- f^{(m)}(\tilde{\xi}_2))$$

mit $\tilde{\xi}_1, \tilde{\xi}_2 \in [a,b]$ gegeben. Kurz ausgedrückt haben wir eine Quadraturformel für ein kompaktes Referenzintervall, so können wir diese für jedes beliebige kompakte Integrationsintervall verwenden. Etwas anders sieht dies aus, wenn die Gewichtsfunktion nicht konstant ist. Für beliebige Gewichstfunktionen gilt nämlich

$$\int_a^b f(x)w(x)\,dx = m_c \int_{\tilde{a}}^{\tilde{b}} f(m_c\,x + c)\, w(m_c\,x + c)\,dx \tag{2.64}$$

und auf das rechte Integral ist unsere Ausgangsquadraturformel im allgemeinen nicht anwendbar, da sich die Gewichtsfunktion durch die Transformation verändert hat. In besonderen Fällen ist es jedoch möglich (z. B. durch einfache Umformungen), das rechtsstehende Integral auf ein Integral mit der ursprünglichen Gewichtsfunktion zurückzuführen. Wir wollen dies an zwei Beispielen zeigen.

Ausgehend von der Quadraturformel

$$\int_0^1 f(x)\, x^{-\frac{1}{2}}\,dx = Q(f) + c_m^+ f^{(m)}(\xi_1) + c_m^- f^{(m)}(\xi_2) \tag{2.65}$$

($\xi_1, \xi_2 \in [0,1]$) transformieren wir unser Ausgangsintervall durch die lineare Transformation $x = b\,t$ auf das Referenzintervall $[0,1]$

$$\int_0^b f(x) x^{-\frac{1}{x}}\,dx = \sqrt{b}\int_0^1 f(bx) x^{-\frac{1}{2}}\,dx$$

Wenden wir die Quadraturformel (2.65) auf das rechtsstehende Integral an, so erhalten wir die affin transformierte Quadraturformel $\tilde{Q}$ für das Intervall $[0,b]$

$$\begin{aligned}
\tilde{Q}(f) &= \sqrt{b}\,Q(\tilde{f}) = \sum_{k=1}^{n} \sqrt{b}\, w_k f(bx_k) \\
\tilde{R}(f) &= b^{m+\frac{1}{2}}\, c_m^+\, f^{(m)}(\tilde{\xi}_1) + b^{m+\frac{1}{2}}\, c_m^-\, f^{(m)}(\tilde{\xi}_1)
\end{aligned}$$

mit $\tilde{\xi}_1, \tilde{\xi}_2 \in [0,b]$.

Im zweiten Beispiel betrachten wir die Gewichtsfunktion $w = -\ln x$, genauer die Quadraturformel

$$\int_0^1 f(x) \ln \frac{1}{x}\, dx = Q(f) + c_m^+ f^{(m)}(\xi_1) + c_m^- f^{(m)}(\xi_2) \tag{2.66}$$

mit $\xi_1, \xi_2 \in [0,1]$. Wollen wir ein Integral der Form (2.66) mit beliebiger oberer Grenze b berechnen, wenden wir wieder die lineare Transformation $x = bt$ an.

$$\int_0^b f(x) \ln \frac{1}{x}\, dx = b \int_0^1 f(bx) \ln \frac{1}{x}\, dx - b \ln(b) \int_0^1 f(b\,x)\, dx$$

Auf das linksstehende Integral wenden wir die Quadraturformel (2.66) und auf das rechtsstehende eine Quadraturformel zur Gewichtsfunktion $w = 1$ an. Als Ergebnis erhalten wir eine Quadraturformel zur Gewichtsfunktion $w = -\ln x$ auf dem Intervall $[0, b]$.

2.5.2 Adaptive Zerlegung

Sei Q eine Quadraturformel zur Gewichtsfunktion $w \equiv 1$. Zerlegen wir unser Ausgangsintervall $[a, b]$ in l Teilintervalle $Z_1, Z_2, \ldots, Z_l$ mit gleichem Durchmesser und wenden auf jedes Teilintervall Z_i die affin transformierte Quadraturformel Q^{Z_i} an, so erhalten wir die zusammengesetzte Quadraturformel $\tilde{Q} := Q^{Z_1} + Q^{Z_2} + \ldots + Q^{Z_l}$. Die Quadraturformel $\tilde{Q}$ hat eine sehr einfache Form, sie enthält nur die transformierten Gewichte von Q und hat mindestens den gleichen Exaktheitsgrad wie die Ausgangsformel. Beispiele zusammengesetzter Quadraturverfahren sind das Trapezverfahren und das Simpson-Verfahren. Ein entscheidender Nachteil dieser zusammengesetzten Quadraturformel ist jedoch, daß sie sich nicht am Verlauf einer Funktion orientiert und so unnötig viel Aufwand erzeugt. Deswegen ist sie nur für Integranden geeignet, deren Verhalten sich auf verschiedenen Teilen des Integrationsbereichs nicht allzu sehr unterscheidet. Hat die Funktion in der Nähe des Integrationsintervalls eine Singularität, so ist es sinnvoll auch nicht äquidistante Zerlegungen des Ausgangsintervalls zuzulassen. Liegt eine Singularität im Integrationsintervall, dann werden zur Berechnung der Teilintegrale außerdem verschiedene Ausgansquadraturformeln benötigt. Zunächst wollen wir aber nur den Fall $w \equiv 1$ betrachten.
Zu einer vorgegebenen Zerlegung $\mathbf{Z} : a = z_0 < z_1 < \cdots < z_l = b$ der Feinheit $|\mathbf{Z}|$

$$|\mathbf{Z}| = \max_{i=0}^{l-1} \{z_{i+1} - z_i\}$$

berechnet sich die zusammengesetzte Legendre-Formel Q_n^Z, indem auf jedes Teilintervall $Z_i = [z_i, z_{i+1}]$ die affin transformierte Quadraturformel $Q^{[z_i, z_{i+1}]}$ angewendet wird.

$$Q_n^Z(f) = \sum_{i=0}^{l-1} \sum_{k=1}^{n} \frac{z_{i+1} - z_i}{2}\, w_{k,n} f\left(\frac{z_{i+1} - z_i}{2} x_{k,n} + \frac{z_{i+1} + z_i}{2}\right) \tag{2.67}$$

$$R_n^Z(f) = \sum_{i=0}^{l-1} \frac{(z_{i+1} - z_i)^{2n+1}}{2n+1} \left(c_{m,n}^+ f^{(m)}(\xi_{i,1}) + c_{m,n}^- f^{(m)}(\xi_{i,2})\right), \quad \xi_{i,1}, \xi_{i,2} \in (z_i, z_{i+1})$$

Lassen wir auf verschiedenen Teilintervallen Quadraturformeln mit unterschiedlicher Stützstellenanzahl zu, so erhalten wir

$$Q_N^Z(f) = \sum_{i=0}^{l-1} \sum_{k=1}^{n_i} \frac{z_{i+1}-z_i}{2} w_{k,n_i} f\left(\frac{z_{i+1}-z_i}{2} x_{k,n_i} + \frac{z_{i+1}+z_i}{2}\right), \tag{2.68}$$

wobei der Vektor $N = (n_0, n_2, \ldots, n_{l-1})$ die Anzahl der Stützstellen n_i der Quadraturformel auf dem Intervall $[z_i, z_{i+1}]$ angibt. Das Restglied läßt sich darstellen als

$$R^Z(f) = \sum_{i=0}^{l-1} \left(\frac{z_{i+1}-z_i}{2}\right)^{m_i+1} \left(c^+_{m_i,n_i} f^{(m_i)}(\xi_{i,1}) + c^-_{m_i,n_i} f^{(m_i)}(\xi_{i,2})\right) \tag{2.69}$$

mit $\xi_{i,1}, \xi_{i,2} \in (z_i, z_{i+1})$ und $m_i \in I\!N, 1 \leq m_i \leq 2n_i$.

Adaptive Strategien

Ziel adaptiver Algorithmen ist es, eine Zerlegung $\mathbf{Z}$ zu finden, aus der mit möglichst wenig Aufwand eine gute Näherung bzw. Einschließung des gesuchten Integrals berechnet werden kann. Die Abbildungen 2.8 und 2.9 enthalten eine allgemeinere Beschreibung adaptiver Verfahren passend sowohl für lokal adaptive als auch für global adaptive Verfahren. Neben den typischen Integrationsdaten Integrand f, Integrationsintervall $[a, b]$, den Daten der zur Verfügung stehenden Quadraturformeln und eines Abbruchkriteriums werden zusätzliche globalen Daten verwendet, nämlich die Menge A der aktiven Intervalle und die Menge D der nicht aktiven oder inaktiven Intervalle. Die Menge A enthält alle Teilintervalle, die eventuell weiter zerlegt werden, während die Menge D die Teilintervalle der Zerlegung enthält, die nicht mehr unterteilt werden sollen. Terminiert das Verfahren, d. h. das Genauigkeitskriterium wird erfüllt, dann ist die gesuchte Zerlegung durch die Vereinigung der Mengen A und D festgelegt.

Lokal adaptiver Algorithmus

In einem lokal adaptiven Algorithmus sind alle Intervalle aktiv, die ein lokales Genauigkeitskriterium nicht erfüllen. Das lokale Genauigkeitskriterium ist meistens eine geforderte Fehlertoleranz, die proportional zum Durchmesser des Intervalls ist. Erfüllt ein Intervall diese Forderung, dann wird es in die Menge der nicht aktiven Intervalle aufgenommen. Das Verfahren terminiert, wenn die Menge der aktiven Intervalle leer ist. Die gesuchte Zerlegung ist durch die Intervalle in D festgelegt.

Global adaptiver Algorithmus

In global adaptiven Algorithmen bleiben alle Intervalle aktiv, sofern sie nicht weiter unterteilt werden. Die Intervalle werden mithilfe einer Ordnungsfunktion μ der Größe nach geordnet. In jedem Zerlegungsschritt wird das Intervall mit dem größten Schlüssel ausgewählt und unterteilt. Das Abbruchkriterium ist abhängig von der Summe der Fehlerabschätzungen aller aktiven Intervalle. Terminiert das Verfahren, so enthält die Menge A die Intervalle der Zerlegung.

Abhängig von der Wahl der Ordnungsfunktion μ ergeben sich unterschiedliche globale Zerlegungsstrategien. Wählen wir für $\mu(J)$ den Betrag der Restgliedabschätzung auf J, so wird in jedem Zerlegungsschritt das Intervall mit der betragsmäßig größten Restgliedabschätzung unterteilt. Ebensogut kann $\mu(J)$ als Produkt des Durchmessers von J und dem Betrag der Fehlerabschätzung oder als Quotient des Betrags der Fehlerabschätzung und der Quadratursumme definiert werden. Für unsere Zwecke werden wir in Abschnitt 2.6.1 $\mu(J)$ als Durchmesser der Restgliedeinschließung auf dem Intervall J definieren.

Ein Vorteil der lokal adaptiven Verfahren ist, daß sie relativ einfach zu implementieren sind, wenn das lokale Genauigkeitskriterium im voraus feststeht. Die Datenverwaltung der Intervallzerlegung ist bei global adaptiven Verfahren deutlich aufwendiger, da die Teilintervalle in Listen, Bäumen oder Heaps sortiert werden. Das Ganze ändert sich schlagartig, wenn z. B. bei Vorgabe eines relativen Fehlers sich das lokale Genauigkeitskriterium in Abhängigkeit von der bisher berechneten Quadratursumme verändert und nicht aktive Intervalle wieder aktiv werden. Dies ist einer der Gründe, warum in neueren Softwarepaketen global adaptive Verfahren deutlich verbreiteter als lokal adaptive Verfahren sind (siehe [49]).

Initialisierung
while $\neg$ Genauigkeitskriterium **do**
    Zerlegungsschritt
od

Abbildung 2.8: Adaptiver Algorithmus

Zerlegungsschritt:
  Auswahl aktives Intervall $J \in A$
  Zerlege J in k Teilintervalle $J_1, \ldots, J_k$
  Berechne Daten der neuen Teilintervalle
  Aktualisiere globale Daten (A, D)
    Entferne J aus der Menge der aktiven Intervalle
    Ergänze Menge der aktiven Intervalle $A = A \cup \{J_1, \ldots, J_k\}$

Abbildung 2.9: Intervallzerlegung

2.6 Verifizierte Quadratur

In diesem Abschnitt wenden wir uns der verifizierten Bestimmung nicht bzw. schwach singulärer Integrale eindimensionaler Funktionen zu. Dazu treffen wir folgende Voraussetzung. Die zu integrierende Funktion f sei auf der Maschine darstellbar, d. h. eine Komposition aus den arithmetischen Operatoren $+, -, *, /$ und Standardfunktionen wie z. B. $\sin, \cos, \ln$, oder $\exp$. Außerdem sei sie hinreichend oft stetig differenzierbar, und die benötigten Ableitungen mögen mit Hilfe der Rechenregeln der automatischen Differentiation (siehe Abschnitt 1.4) eingeschlossen werden können. Funktionen, deren Funktionswerte nur durch Messergebnisse an diskreten Werten vorgegeben sind, schließen wir hiermit aus.

Sei Q_n eine Quadraturformel mit

$$I(f) = Q_n(f) + R_n(f) = \sum_{i=1}^{n} w_{i,n} f(x_{i,n}) + c^+_{m,n}\, f^{(m)}(\xi_1) + c^-_{m,n}\, f^{(m)}(\xi_2). \tag{2.70}$$

Dann berechnen wir eine Einschließung $[Q_n(f)]$ der Quadratursumme $Q_n(f)$, indem wir die Stützstellen $x_{i,n}$ und Gewichte $w_{i,n}$ durch Einschließungen $[x_{i,n}]$ und $[w_{i,n}]$ derselben ersetzen. Die Funktionsauswertungen sind intervallmäßig durchzuführen und die Operatoren sind jetzt als Intervallverknüpfungen zu interpretieren. Eine Einschließung $[R_n(f)]$ des Restglieds erhalten wir, indem wir eine Einschließung des Taylorkoeffizienten der Ordnung m mit einer Einschließung der entsprechenden Integrationskonstanten multiplizieren

$$I(f) \in \sum_{i=1}^{n} [w_{i,n}] f([x_{i,n}]) + [m!\, c^+_{m,n}]\, f_m([a,b]) + [m!\, c^-_{m,n}]\, f_m([a,b]). \tag{2.71}$$

Berechnen wir die Einschließung der Quadratursumme in (2.71) mit Hilfe eines langen Akkumulators, so wird hierzu nur eine Rundung benötigt. Da die Faktoren $[w_{i,n}]$ und $f([x_{i,n}])$ im allgemeinen keine Punktintervalle sind, und sich die Durchmesser der Intervalle in der Summe addieren, können hier trotzdem numerische Schwierigkeiten entstehen, z. B. dann, wenn die Gewichte verschiedene Vorzeichen haben und Auslöschungseffekte auftreten. Für zusammengesetzte Quadraturformeln können Einschließungen ganz analog berechnet werden.

2.6.1 Ein global adaptiver Algorithmus

In diesem Abschnitt stellen wir einen global adaptiven Algorithmus (vgl. [45]) zur Berechnung von engen Einschließungen bestimmter Integrale vor. Gesucht ist das Integral

$$I(f) = \int_a^b f(x) w(x)\, dx \tag{2.72}$$

Für jedes Intervall $J \subset [a,b]$ existiere eine Quadraturformel so, daß wir eine Einschliessung $[Q^J(f)]$ für die Quadratursumme $Q^J(f)$ und eine Einschließung $[R^J(f)]$ für das Restglied $R^J(f)$ bestimmen können.

Zu einem vorgegeben absoluten Fehler ϵ_{abs} suchen wir nun eine Einschließung $[I(f)]$ von $I(f)$ mit $d([I(f)]) < \epsilon_{abs}$. Da normalerweise $d([R(f)]) \gg d([Q(f)])$ gilt, reduzieren wir unsere Forderung zu $d([R(f)]) < \epsilon_{abs}$, weisen aber darauf hin, daß es durchaus passieren kann, daß zwar $d([R(f)]) < \epsilon_{abs}$ gilt, aber die entsprechende Einschließung von $I(f)$ einen Durchmesser größer als ϵ_{abs} hat. Dies kann u. a. dann passieren, wenn der Durchmesser der Restgliedeinschließung sehr nahe an ϵ_{abs} liegt.
Die Menge A der aktiven Intervalle enthält zunächst nur das Integrationsintervall $[a, b]$ (siehe Abbildung 2.10). R_{inf} ist der Wert einer unteren und R_{sup} der Wert einer oberen Schranke für das Restglied auf dem gesamten Intervall $[a, b]$. In jedem Zerlegungsschritt wird das Intervall mit dem größten Fehlerglieddurchmesser $d([R^J(f)])$ in k Teilintervalle unterteilt. Übersteigt die Anzahl der aktiven Intervalle k_{max}, bricht das Verfahren ab und gibt eine Einschließung $[I]$ des gesuchten Integrals aus. Die geforderte Bedingung an den Durchmesser der Einschließung ist dann nicht mehr garantiert. Die Quadratursumme wird erst zum Schluß berechnet, wenn die Zerlegung schon feststeht. Verwenden wir für R_{inf} und R_{sup} lange Akkumulatoren (siehe [50]), so müssen die unteren und oberen Grenzen nicht nach jedem Zerlegungsschritt komplett neu berechnet werden, sondern können mit minimalem Aufwand aktualisiert werden. Zur Berechnung der Einschließung der Quadratursumme empfiehlt sich ebenfalls der Einsatz eines langen Akkumulators, es gelten jedoch die am Ende von Abschnitt 2.6 erwähnten Einschränkungen.

Wahl der Quadraturformeln

Prinzipiell können wir jede Quadraturformel Q^J auf ein Intervall J anwenden, vorausgesetzt es exisitiert eine berechenbare Fehlerdarstellung $R^J(f)$ mit

$$R^J(f) \to 0 \text{ für } d(J) \to 0$$

Für $w \equiv 1$ zeigt sich, daß die Gauß-Legendre Formeln wegen ihres hohen Exaktheitsgrades und ihrer betragsmäßig sehr niedrigen Integrationskonstanten und der damit verbundenen günstigen Restglieddarstellung sehr gut für unsere Zwecke geeignet sind. Ist w keine konstante Gewichtsfunktion, so verwenden wir entweder die entsprechende Gauß-Quadraturformel oder, falls w auf dem Intervall hinreichend oft stetig differenzierbar ist (d. h. die Singularität liegt nicht in J), die Gauß-Legendre Formel angewandt auf das Produkt wf. Dazu müssen die benötigten Stützstellen, Gewichte und Integrationskonstanten, sofern sie sich nicht durch einfache Transformationen berechnen lassen, im voraus berechnet und tabelliert werden, da eine Berechnung zur Laufzeit unter Umständen mehr Zeit in Anspruch nimmt als die eigentliche Berechnung des gesuchten Integrals. Auf die Wahl der Stützstellenzahl gehen wir im nächsten Abschnitt ein.

Bestimmung der Restgliedeinschließung

Ein wichtiger Punkt des Verfahrens ist die Auswahl einer geeigneten Restglieddarstellung. Ist $w \equiv 1$, dann können wir bei Verwendung der Gauß-Legendre Formeln

** Initialisierung **
$A := \{[a,b]\}$
Bestimme $r := [R^{[a,b]}(f)]$
$R_{inf} := \inf(r)$
$R_{sup} := \sup(r)$
$[Q] := [0,0]$

** Zerlegungsteil **
while $(\#A < k_{max} \wedge (R_{sup} - R_{inf}) > \epsilon)$ **do**
    Suche Intervall J aus A mit $d([R^J(f)])$ maximal
    Zerlege J in k Teilintervalle $J_1, J_2, \ldots, J_k$
    Entferne J aus A
    $R_{inf} := R_{inf} - \inf([R^J(f)])$
    $R_{sup} := R_{sup} - \sup([R^J(f)])$
    for $i := 1$ **to** k **do**
        Füge J_i in A ein
        Bestimme $r := [R^{J_i}(f)]$
        $R_{inf} := R_{inf} + \inf(r)$
        $R_{sup} := R_{sup} + \sup(r)$
    od
od

** Bestimme Quadratursumme **
for $J \in A$ **do**
    $[Q] := [Q] + [Q^J(f)]$
od

Ausgabe $[I] := [Q] + [R_{inf}, R_{sup}]$

Abbildung 2.10: Global adaptiver Algorithmus (GAA)

abhängig von der Anzahl n der Stützstellen aus $2n$ verschiedene Darstellungen des Restgliedes auswählen. Die Fehlereinschließung berechnen wir aus

$$[r_{m,n}] := [R^J_{m,n}(f)] = \left(\frac{d(J)}{2}\right)^{m+1} \left([m!\, c^+_{m,n}]\, f_m(J) + [m!\, c^-_{m,n}]\, f_m(J)\right). \tag{2.73}$$

Erhöht man in (2.73) die Anzahl der Stützstellen bei festem m, so verändert sich nur der Wert der Integrationskonstanten. In Abschnitt 2.4.1 haben wir schon erwähnt, daß mit wachsendem n die Integrationskonstanten $c^+_{m,n}$ fallen, d. h. für größere n erhalten wir engere Einschließungen, bei der Bildung der Quadratursummen benötigen wir dafür aber mehr Funktionsauswertungen. Erhöhen wir umgekehrt m (bei fester Stützstellenzahl), dann fällt zwar für Intervalle mit hinreichend kleinem Durchmesser der Vorfaktor der Fehlereinschließung, gleichzeitig muß aber auch ein Taylorkoeffizient $f_m(J)$ höherer Ordnung berechnet werden. Generell ist es also nicht möglich vorauszusagen, welche Kombination (m, n) auf einem Teilintervall J letztendlich zu einer optimalen Lösung unseres Ausgangsproblems führt. Die folgende Tabelle enthält die Durchmesser der aus (2.73) berechneten Restgliedeinschließungen der Funktion $f(x) = 10/(0.1 + x^2)$ für verschiedene Werte von J, m und n.

$J = [0, 0.5]$				
m	n			
	8	11	14	17
7	$2.32E-01$	$2.41E-02$	$4.48E-03$	$1.16E-03$
10	$1.05E+00$	$3.31E-02$	$2.74E-03$	$3.83E-04$
13	$2.21E+01$	$1.44E-01$	$4.76E-03$	$3.41E-04$
16	$9.92E+03$	$1.84E+00$	$2.02E-02$	$7.01E-04$
$J = [0, 0.25]$				
7	$1.91E-05$	$1.99E-06$	$3.70E-07$	$9.61E-08$
10	$2.32E-06$	$7.32E-08$	$6.06E-09$	$8.47E-10$
13	$1.31E-06$	$8.49E-09$	$2.82E-10$	$2.02E-11$
16	$1.57E-05$	$2.91E-09$	$3.19E-11$	$1.11E-12$

In der Praxis wird die Menge M aller möglichen Kombinationen (m, n) zusätzlich noch beschränkt, da sowohl die Stützstellen und Gewichte als auch die Integrationskonstanten nur für bestimmte Werte vorliegen.
In Abbildung 2.11 geben wir ein Verfahren zur Bestimmung einer Kombination (m, n) aus einer vorgegebenen Menge M mit

$$M = \{im' + m^* | i = 1, \ldots, i_{max}\} \times \{jn' + n^* | j = 1, \ldots, j_{max}\}$$

an. Die Konstanten n_{min} und n_{max} bezeichnen die minimale bzw. maximale Stützstellenanzahl der verfügbaren Quadraturformeln. Die von Algorithmus 2.11 ermittelte Ordnung m des Taylorglieds der Fehlerdarstellung wird bei einer Zerlegung des Intervalls als Startwert m_{start} an die Teilintervalle übergeben.

$\epsilon_{local} := c\,\epsilon_{abs}\frac{b-a}{d(J)}$
$m := m_{start}$
$d := d([r_{m,n_{max}}])$
if $(d < \epsilon_{local})$
  $k := n_{min}$
  while $((m,k) \in M)$ **do**
    if $(d([r_{m,k}]) \leq \epsilon_{local})$ **return** $[r_{m,k}]$ **fi**
    $k := k + n'$
  od
  return $[r_{m,n_{max}}]$
fi
if $((m - m', n_{max}) \in M \ \wedge \ d([r_{m-m',n_{max}}]) \leq d)$
  $m := m - m'$
  return $[r_{m,n_{max}}]$
fi
if $((m + m', n_{max}) \in M \ \wedge \ d([r_{m+m',n_{max}}]) \leq d)$
  $m := m + m'$
  return $[r_{m,n_{max}}]$
fi

Abbildung 2.11: Bestimmung der Restgliedeinschließung $[R^J_{m,n}(f)]$

Setzen wir für das lokale Fehlerkriterium $c = 1$, so können alle Intervalle J, die das Kriterium erfüllen, in die Menge der nicht aktiven Intervalle aufgenommen werden. In der Praxis empfiehlt es sich jedoch für c Werte größer als Eins zu wählen. Da die Auswahl von (m,n) abhängig vom Verhalten von f ist, erhalten wir zusammen mit dem global adaptiven Rahmen aus Abbildung 2.8 ein sogenanntes doppelt adaptives Verfahren. Für die Gewichtsfunktion $w = x^{-\frac{1}{2}}$ gelten ähnliche Aussagen, wobei die Fehlereinschließungen für ein Intervall J, das die singuläre Stelle enthält, die Form

$$[r_{m,n}] := [R^J_{m,n}(f)] = (d(J))^{m+\frac{1}{2}} \left([m!\, c^+_{m,n}] f_m(J) + [m!\, c^-_{m,n}] f_m(J) \right) \tag{2.74}$$

haben.

Aufteilung des Intervalls

In jedem Teilschritt wird das aktuelle Intervall in k Teilintervalle zerlegt. In Abbildung 2.12 werden typische Unterteilungen für eine Funktion mit einem Peak am rechten Rand skizziert. Schwarze Knoten repräsentieren Intervalle, die im weiteren Verlauf in kleinere Teilintervalle zerlegt werden. Weiße Knoten repräsentieren die Intervalle der Schlußzerlegung. Die Zerlegung auf der linken Seite entsteht durch Unterteilung in

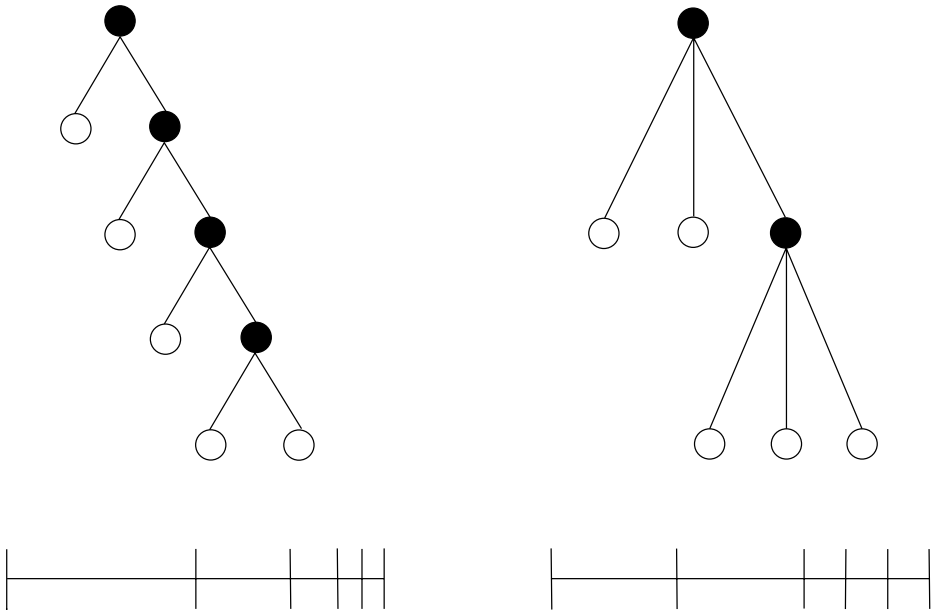


Abbildung 2.12: Vergleich: Bisektion und Trisektion

$k = 2$ Teilintervalle gleichen Durchmessers (Bisektion), die der rechten Seite durch Unterteilung in $k = 3$ Intervalle gleichen Durchmessers (Trisektion). Bei der Trisektion ist das Verhältnis der weißen zu den schwarzen Knoten deutlich gößer als bei der Bisektion. Allgemein müssen bei der k-Sektion zur Erzeugung von $(k-1)\,l+1$ aktiven Intervallen $k\,l+1$ Intervalle abgearbeitet werden, d. h. zur Bestimmung einer Zerlegung mit fester Anzahl aktiver Intervalle werden für höhere k insgesamt deutlich weniger Teilintervalle benötigt. Dies bedeutet einen Mehraufwand von etwa einem Drittel bei der Bisektion im Vergleich zur Trisektion. Auf der anderen Seite paßt sich für kleinere k die resultierende Zerlegung besser dem Verlauf der Funktion an, d. h. die resultierende Zerlegung enthält für größere k bei Funktionen mit unterschiedlichem Verlauf auf verschiedenen Teilbereichen des Integrationsbereichs mehr Intervalle. Dies ist der Grund, warum in der Praxis die Bisektion oft den Vorzug vor der Trisektion erhält.
Da sich die Durchmesser der Intervalle durch mehrmaliges Unterteilen automatisch dem Verlauf der Funktion anpassen, verzichten wir bei den einzelnen Zerlegungschritten auf Unterteilungen in Intervalle mit unterschiedlichen Durchmessern. Einzige Ausnahme hiervon wird die Zerlegung von Intervallen sein, auf denen der Integrand eine Singularität besitzt.

2.6.2 Relativer Fehler

Suchen wir eine Integraleinschließung, die ein relatives Fehlerkriterium mit vorgegebenem ϵ_{rel} erfüllt, so berechnen wir zunächst eine grobe Einschließung $[\underline{z}, \overline{z}]$ von $I(f)$. Enthält diese Einschließung nicht die Null, so definieren wir ϵ_{abs} durch

$$\epsilon_{abs} := \min\{|\underline{z}||\overline{z}|\}\,\epsilon_{rel}$$

Ausgehend von der bisherigen Zerlegung und den schon berechneten Restgliedern bestimmen wir jetzt eine Einschließung $[I(f)]$ von $I(f)$ mit $d([I(f)]) < \varepsilon_{abs}$. Für alle $q \in [I(f)]$ gilt dann (siehe auch [45])

$$\left|\frac{\int_a^b f(x)w(x)\,dx - q}{\int_a^b f(x)w(x)\,dx}\right| \le \left|\frac{d([I(f)])}{\int_a^b f(x)w(x)\,dx}\right| \le \frac{\epsilon_{abs}}{\min|\underline{z}||\overline{z}|} = \epsilon_{rel}.$$

Quadratursummen, die zur Bestimmung von $[\underline{z}, \overline{z}]$ berechnet wurden, können weiter verwendet werden, falls das dazugehörige Intervall nicht weiter unterteilt wird.

2.6.3 Integrationsbereiche mit nicht darstellbaren Randpunkten

Bisher haben wir immer stillschweigend vorausgesetzt, daß die Integrationsgrenzen a und b Gleitkommazahlen, also auf der Maschine darstellbar, sind. Sind a und b keine Rasterzahlen, dann bezeichnen wir mit $[\underline{a}, \overline{a}]$ eine Einschließung von a bzw. mit $[\underline{b}, \overline{b}]$ eine Einschließung von b mit jeweils benachbarten Gleitkommazahlen und es gilt

$$\begin{aligned}
\int_a^b f(x)\,dx &= \int_{\overline{a}}^{\underline{b}} f(x)\,dx + \int_a^{\overline{a}} f(x)\,dx + \int_{\underline{b}}^b f(x)\,dx \\
&= \int_{\overline{a}}^{\underline{b}} f(x)\,dx + (\overline{a} - a)\,f(\xi_1) + (b - \underline{b})\,f(\xi_2) \\
&\in \int_{\overline{a}}^{\underline{b}} f(x)\,dx + [0, \overline{a} - \underline{a}]\,f([\underline{a}, \overline{a}]) + [0, \overline{b} - \underline{b}]\,f([\underline{b}, \overline{b}]). \qquad (2.75)
\end{aligned}$$

Das Integral $\int_{\overline{a}}^{\underline{b}} f(x)\,dx$ berechnen wir mit dem Verfahren aus Abschnitt 2.6.1. Für nicht konstante Gewichtsfunktionen verfahren wir genauso. Es gilt z. B. für $w = \ln\left(\frac{1}{x}\right)$

$$\begin{aligned}
\int_0^b f(x)\,\ln\left(\frac{1}{x}\right)dx &= \int_0^{\underline{b}} f(x)\,\ln\left(\frac{1}{x}\right)dx + (b - \underline{b})\,f(\xi)\,\ln\left(\frac{1}{\xi}\right) \\
&\in \int_0^{\underline{b}} f(x)\,\ln\left(\frac{1}{x}\right)dx + [0, \overline{b} - \underline{b}]\,f([\underline{b}, \overline{b}])(-\ln([\underline{b}, \overline{b}]))
\end{aligned}$$

mit $\xi \in [\underline{b}, b]$.

2.6.4 Numerische Beispiele

In diesem Abschnitt verwenden wir den Algorithmus GAA aus Abschnitt 2.6.1 zur Bestimmung einiger Beispielintegrale. Dazu wurden, soweit nicht anderes angegeben, $8, 11, 14$ und 17-punktige Gauß-Legendre-Formeln bestimmt und jeweils die Integrationskonstanten der Ordnung $7, 10, 13$ und 16 berechnet. In der Notation des Algorithmus 2.10 heißt dies

$$M = \{7, 10, 13, 16\} \times \{8, 11, 14, 17\}. \tag{2.76}$$

Für die Darstellung der Ergebnisse verwenden wir die folgenden Abkürzungen:

ϵ_{abs}	Maximaler Durchmesser der Restgliedeinschließung
$[I]$	Einschließung des Integrals $I(f)$
$d([I])$	Durchmesser der Einschließung $[I]$
$\#f$	Summe der Funktionsauswertung zur Berechnung des Approximationsterms
$\#R$	Anzahl der aktiven Intervalle bei Terminierung
t	Gesamtrechenzeit (gemessen auf einem Intel-PC PII-400 MHZ in 1/100 s)
t_R	Rechenzeit zur Bestimmung der Zerlegung
t_Q	Rechenzeit zur Berechnung des Approximationsterms

Riemann-Integral mit Spitzen

Schwierigkeiten bei der Bestimmung des Integrals (siehe [42])

$$\begin{aligned} I(f_\alpha) &= \int_0^4 g_{\alpha,1}(x) - g_{\alpha,4}(x) + g_{\alpha,7}(x) - g_{\alpha,10}(x)\, dx, \\ g_{\alpha,k}(x) &= \frac{1}{\alpha^2 + (3x-k)^2} \end{aligned}$$

machen die vier "Spitzen" der Kurve, die abhängig von der Größe des Parameters α unterschiedlich stark ausfallen. Für kleinere Werte von α werden erwartungsgemäß deutlich mehr Teilintervalle und Funktionsauswertungen zur Berechnung benötigt, da in diesem Fall die Spitzen viel ausgeprägter sind. Numerische Auslöschungen bei der Berechnung des Approximationsterms verhindern für $\alpha = 0.001$ eine genauere Einschließung des Integrals, obwohl die Restglieder sehr eng eingeschlossen werden können. Ein Blick auf die letzte Spalten der nächsten beiden Tabellen zeigt, daß der Großteil der Rechenzeit zur Bestimmung der Zerlegung und der Restglieder benötigt wird. Darauf werden wir noch im nächsten Kapitel zu sprechen kommen.

Zum Vergleich berechnen wir das Integral $I(f_{0.001})$ mit dem global adaptiven Algorithmus 2.10, jedoch mit nur einer festen Quadraturformel und Restglieddarstellung. Die Werte für $M = \{(16, 8)\}$ und $M = \{(16, 17)\}$ sind in der letzten Tabelle dieses Abschnittes zusammengefaßt. Das Verfahren mit der 8-punktigen Gauß-Legendre-Quadraturformel und der klassischen Restglieddarstellung schneidet im direkten Vergleich am schlechtesten ab. Es braucht in allen Fällen die meiste Rechenzeit. Der Einsatz der 17-punktigen Legendre-Formel führt zwar – zumindest bei kleinerem ϵ – zu

mehr Funktionsauswertungen, bringt aber aufgrund der geringeren Anzahl an Zerlegungen einen erheblichen Geschwindigkeitsgewinn mit sich.

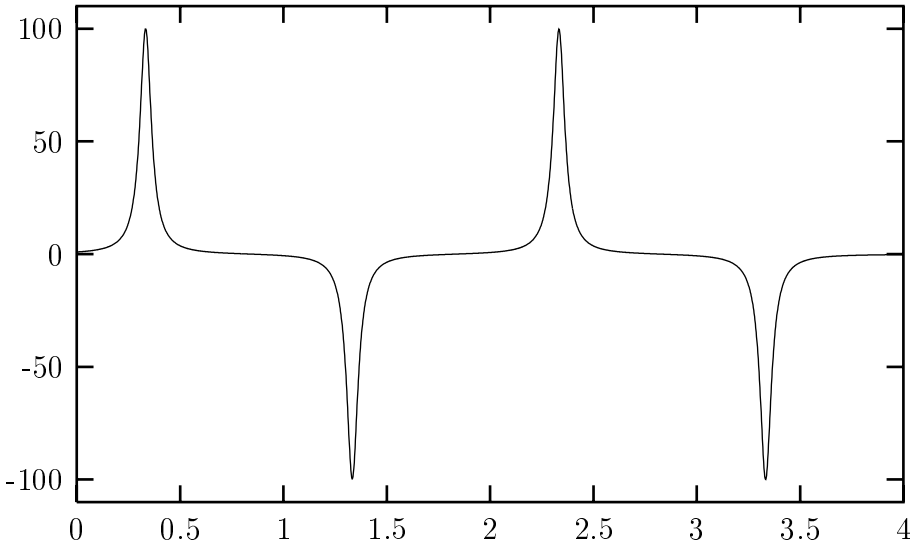


Abbildung 2.13: $f_{0.1}$

ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#f$	t_R/t_Q
$1.0E-02$	$-1.5^{48}_{58}E-01$	$9.7E-03$	76	848	4.3
$1.0E-04$	$-1.52^{87}_{96}E-01$	$7.9E-05$	88	1073	3.9
$1.0E-06$	$-1.5291^{1}_{3}E-01$	$9.6E-07$	99	1425	3.6
$1.0E-08$	$-1.529124^{2}_{4}E-01$	$9.7E-09$	138	1725	4.5
$1.0E-10$	$-1.52912433^{0}_{2}E-01$	$1.0E-10$	152	2134	4.1
$1.0E-12$	$-1.529124331^{1}_{6}E-01$	$3.7E-11$	182	2644	4.2
$\alpha = 0.01$					

ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#f$	t_R/t_Q
$1.0E-02$	$-1.\overset{48}{58}E-01$	$8.9E-03$	120	1380	4.2
$1.0E-04$	$-1.52\overset{89}{94}E-01$	$5.2E-05$	136	1781	3.9
$1.0E-06$	$-1.5292\overset{1}{3}E-01$	$9.8E-07$	149	2158	3.7
$1.0E-08$	$-1.529219\overset{7}{9}E-01$	$1.3E-08$	207	2682	4.4
$1.0E-10$	$-1.529219\overset{79}{84}E-01$	$3.7E-09$	231	3423	4.1
$1.0E-12$	$-1.5292198\overset{4}{0}E-01$	$3.3E-09$	263	4108	4.1
$\alpha = 0.001$					

	$M = \{(16,8)\}$			$M = \{(16,17)\}$			M nach (2.76)		
ϵ_{abs}	$\#R$	$\#f$	t	$\#R$	$\#f$	t	$\#R$	$\#f$	t
$1E-02$	178	1424	204	125	2125	157	120	1380	91
$1E-04$	218	1744	250	131	2227	165	136	1781	112
$1E-06$	252	2016	289	136	2312	171	149	2158	130
$1E-08$	304	2432	349	162	2574	204	207	2682	185
$1E-10$	387	4096	444	205	3485	260	231	3423	222
$1E-12$	479	3832	553	233	3961	292	263	4108	272
$1E-14$	591	4728	684	287	4879	362	322	4802	344
$\alpha = 0.001$									

Bestimmung von Fourierkoeffizienten

Die Funktion $f_r : I\!R \to I\!R$ mit

$$f_r(x) = \frac{1 - r\cos x}{1 - 2r\cos x + r^2}, \quad 0 < r < 1$$

ist analytisch, gerade und 2π-periodisch. Ihre Fourierreihenentwicklung hat die Darstellung

$$f_r(x) = \alpha_0 + 2\sum_{k=1}^{\infty} \alpha_k \cos(kx)$$

mit den Fourierkoeffizienten

$$\alpha_k \;=\; \frac{1}{2\pi}\int_0^{2\pi} f_r(x)\cos(kx)\,dx = \frac{1}{\pi}\int_0^{\pi} f_r(x)\cos(kx)\,dx.$$

Die Fourierkoeffizienten können in geschlossener Form angegeben werden

$$\alpha_k = \begin{cases} 1 & \text{für} \quad k = 0 \\ 0.5\,r^k & \text{für} \quad k \neq 0 \end{cases}.$$

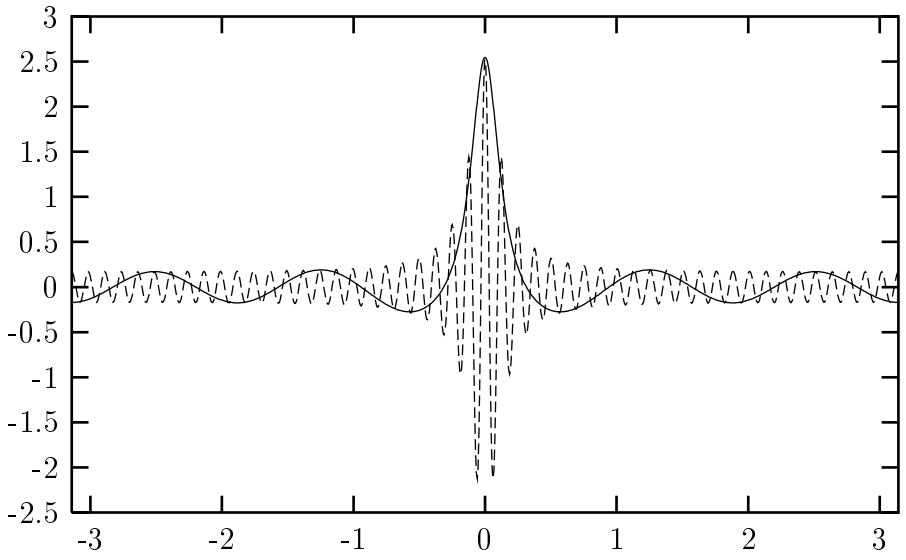


Abbildung 2.14: $f_{0.875}\cos(5x)$ und $f_{0.875}\cos(50x)$

Da der rechte Rand des Integrationsbereichs nicht exakt darstellbar ist, berechnen wir die Fourierkoeffizienten wie in Abschnitt 2.6.3 beschrieben. Das zusätzliche Teilintervall und die zusätzliche Funktionsauswertung sind in $\#R$ bzw. $\#f$ enthalten. Die Bestimmung des Fourierkoeffizienten α_{50} ergibt ähnlich Ergebnisse wie zuvor. Wiederum zeigt sich, daß die Verwendung verschiedener Quadraturformeln und verschiedener Restgliedarstellungen in Verbindung mit Algorithmus 2.11 die Berechnung erheblich beschleunigt. Tabelle 2.5 enthält Einschließungen der Fourierkoeffizienten α_k $(k = 5(5)100)$ für $r = 0.875$ mit einem relativem Fehlerdurchmesser von 10^{-7}.

Bestimmung α_{50} von $f_{0.875}$									
	$M = \{(16,8)\}$			$M = \{(16,17)\}$			M nach (2.76)		
ϵ_{abs}	$\#R$	$\#f$	t	$\#R$	$\#f$	t	$\#R$	$\#f$	t
$1E-02$	14	105	22	7	103	10	8	87	6
$1E-04$	19	145	32	8	120	12	10	118	11
$1E-06$	22	169	38	12	188	20	12	152	13
$1E-08$	35	273	63	13	205	21	13	196	17
$1E-10$	37	289	69	19	307	33	20	243	30
$1E-12$	55	433	105	20	324	39	23	300	36
$1E-14$	70	533	132	29	477	54	31	412	49

k	ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#f$
5	$2.56E-08$	$2.564544^{8}_{5}E-01$	$1.8E-08$	8	93
10	$1.32E-08$	$1.315377^{89}_{87}E-01$	$1.7E-09$	11	108
15	$6.75E-09$	$6.746690^{9}_{5}E-02$	$3.4E-09$	11	117
20	$3.46E-09$	$3.46043^{82}_{77}E-02$	$3.3E-09$	11	129
25	$1.77E-09$	$1.774889^{59}_{49}E-02$	$8.7E-10$	12	161
30	$9.10E-10$	$9.10356^{75}_{66}E-03$	$8.6E-10$	13	172
35	$4.67E-10$	$4.66930^{10}_{08}E-03$	$1.7E-10$	13	190
40	$2.39E-10$	$2.394926^{3}_{0}E-03$	$1.7E-10$	13	202
45	$1.23E-10$	$1.22837^{91}_{89}E-03$	$1.1E-10$	17	225
50	$6.30E-11$	$6.30046^{61}_{54}E-04$	$5.6E-11$	20	246
55	$3.23E-11$	$3.231565^{4}_{0}E-04$	$2.6E-11$	20	276
60	$1.66E-11$	$1.657498^{8}_{6}E-04$	$1.2E-11$	23	297
65	$8.50E-12$	$8.501458^{9}_{6}E-05$	$2.5E-12$	23	339
70	$4.36E-12$	$4.36047^{44}_{39}E-05$	$3.9E-12$	26	357
75	$2.24E-12$	$2.236526^{3}_{0}E-05$	$1.7E-12$	35	417
80	$1.15E-12$	$1.147134^{4}_{1}E-05$	$1.2E-12$	38	447
85	$5.88E-13$	$5.88375^{44}_{37}E-06$	$6.6E-13$	36	503
90	$3.02E-13$	$3.0178^{301}_{299}E-06$	$1.3E-13$	37	592
95	$1.55E-13$	$1.54787^{21}_{19}E-06$	$1.3E-13$	37	601
100	$7.94E-14$	$7.93917^{5}_{3}E-07$	$1.4E-13$	37	604

Tabelle 2.5: Fourierkoeffizienten α_k von $f_{0.875}$

Schwach singuläres Riemann-Integral

Im letzten Beispiel betrachten wir das uneigentliche Integral

$$I(f_\alpha) = \int_0^2 \frac{\sin(e^{\alpha(3.5-x)^3})}{\sqrt{x}}\, dx.$$

Der Integrand oszilliert stark am linken Rand und divergiert für x gegen Null gegen $-\infty$. Teilintegrale über Intervalle, die die Null enthalten, wurden mit $6, 9$ und 12-punktigen Gauß-Quadraturformeln berechnet. Für die Restgliedbestimmung standen die Integrationskonstanten der Ordnung $6, 9$ und 12 zur Verfügung. Die Ergebnisse sind in einer Tabelle weiter unten zusammengefaßt.

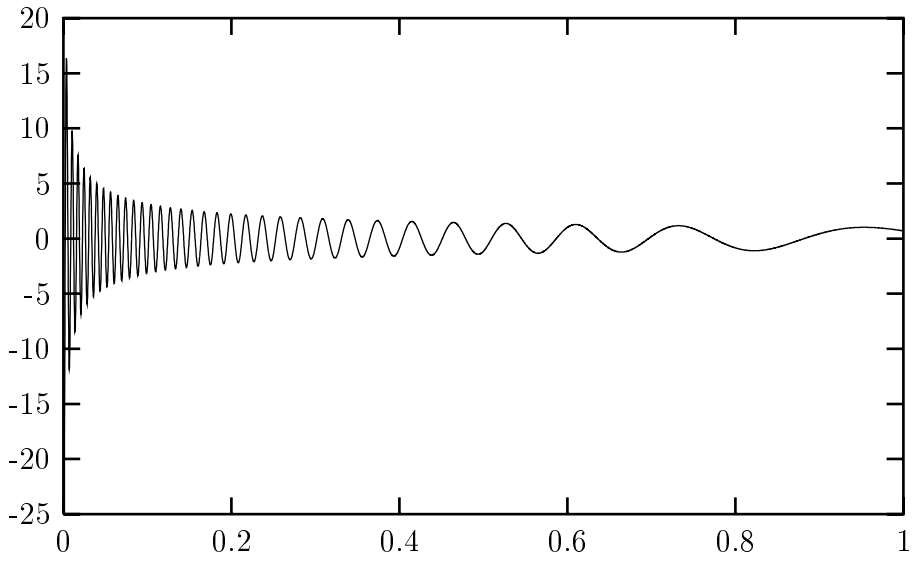


Abbildung 2.15: $f_{0.125}(x)/\sqrt{x}$

ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#f$
$1.0E-02$	$1.{}^{51}_{45}E-01$	$5.2E-03$	13	153
$1.0E-04$	$1.48{}^{10}_{05}E-01$	$4.1E-05$	19	204
$1.0E-06$	$1.4807{}^{55}_{47}E-01$	$7.4E-07$	23	278
$1.0E-08$	$1.480751{}^{1}_{0}E-01$	$9.4E-09$	30	355
$1.0E-10$	$1.48075105{}^{22}_{13}E-01$	$8.0E-11$	38	461
$1.0E-12$	$1.4807510517{}^{8}_{6}E-01$	$1.2E-12$	47	584
$1.0E-14$	$1.4807510517{}^{77}_{69}E-01$	$6.8E-13$	59	710

Zusammenfassung

Die Beispiele dieses Kapitels zeigen, daß mit dem Verfahren GAA Integralwerte bei kompliziertem Funktionsverlauf sehr eng eingeschlossen werden – vorausgesetzt die benötigten Ableitungen können mit automatischer Differentiation berechnet werden. Das Verfahren kann auch zur Bestimmung schwach singulärer Integrale eingesetzt werden, falls entsprechende Quadraturformeln zur Verfügung stehen. Stehen diese nicht zur Verfügung, kann man versuchen, mit Hilfe eines Taylorreihenansatzes (siehe [48]) über die kritischen Intervalle zu integrieren. Dieser Ansatz ist jedoch nicht sonderlich effizient und benötigt zumindest die Kenntnis der Werte der entsprechenden Momente. In [45] wird das Verfahren für $M = \{(8, 16)\}$ mit einem Intervall-Romberg-Verfahren verglichen, wobei das Gauß-Verfahren deutlich besser abschneidet. An Hand der Beispiele sieht man, daß das Verfahren durch den Einsatz verschiedener Quadraturformeln und Restglieddarstellungen in Verbindung mit Algorithmus 2.11 erheblich beschleunigt wird.
Die Ergebnisse der Berechnungen der Beispielintegrale zeigen auch, daß die bloße Angabe der zur Bestimmung des Integrals benötigten Funktionsauswertungen keine Rückschlüsse auf den tatsächlichen Aufwand zuläßt. Die zusätzliche Angabe der benötigten Teilintervalle der Zerlegung ist ebenfalls wenig aussagekräftig, solange man nicht die Ordnungen der zur Bestimmung der Restglieder benutzten Taylorkoeffizienten mit berücksichtigt.

Kapitel 3

Stark singuläre Integrale

Stark singuläre Integrale, auch bekannt unter dem Namen Hadamard-Integrale, finite-part-Integrale oder hypersinguläre Integrale, stellen eine Verallgemeinerung des Riemannschen Integrals für singuläre Funktionen oder unbeschränkte Integrationsbereiche dar. So ermöglichen sie es, Integralen, deren uneigentliches Riemann-Integral nicht definiert ist, einen endlichen Wert zuzuordnen. Stark singuläre Integrale wurden 1923 von Hadamard vorgestellt und treten unter anderem in der Aerodynamik in Form von Cauchy-Hauptwert-Integralen, in der Elastizitätstheorie oder bei der Lösung von bestimmten partiellen Differentialgleichungen vorwiegend in singulären Integralgleichungen auf. In diesem Kapitel stellen wir ein effizientes Verfahren zur verifizierten Bestimmung stark singulärer Integrale eindimensionaler Funktionen vor.

3.1 Definition Cauchy-Hauptwert-Integral

Um der Funktion $f(x) = \frac{1}{x-\lambda}$ über dem Intervall $[a,b]$ $(a < \lambda < b)$ ein Integral zuzuordnen, ist der Riemannsche Integralbegriff nicht ausreichend, da wegen

$$\int_a^\lambda \frac{1}{x-\lambda}\,dx = -\infty \;\text{ und }\; \int_\lambda^b \frac{1}{x-\lambda}\,dx = \infty$$

das (uneigentliche) Riemann-Integral $\int_a^b 1/(x-\lambda)\,dx$ nicht existiert. Bilden wir die beiden Grenzwerte jedoch simultan (siehe Abbildung 3.1), d. h. legen wir auf der x-Achse ein Intervall $[\lambda-\epsilon, \lambda+\epsilon] \subset [a,b]$ um λ, integrieren über die Teilintervalle $[a, \lambda-\epsilon]$ und $[\lambda+\epsilon, b]$ und lassen ϵ gegen Null gehen, dann erhalten wir aus

$$\lim_{\epsilon\to 0+}\left(\int_a^{\lambda-\epsilon} \frac{1}{x-\lambda}\,dx + \int_{\lambda+\epsilon}^b \frac{1}{x-\lambda}\,dx\right) = -\ln|\lambda-a| + \ln|b-\lambda| = \ln\left(\frac{b-\lambda}{\lambda-a}\right)$$

einen endlichen Wert, der Cauchy-Hauptwert genannt wird.

Definition 16 (Cauchy-Hauptwert-Integral) *Für eine auf dem Intervall $[a,b]$ definierte reelle Funktion f wird das Cauchy-Hauptwert-Integral an der Stelle $\lambda \in (a,b)$*

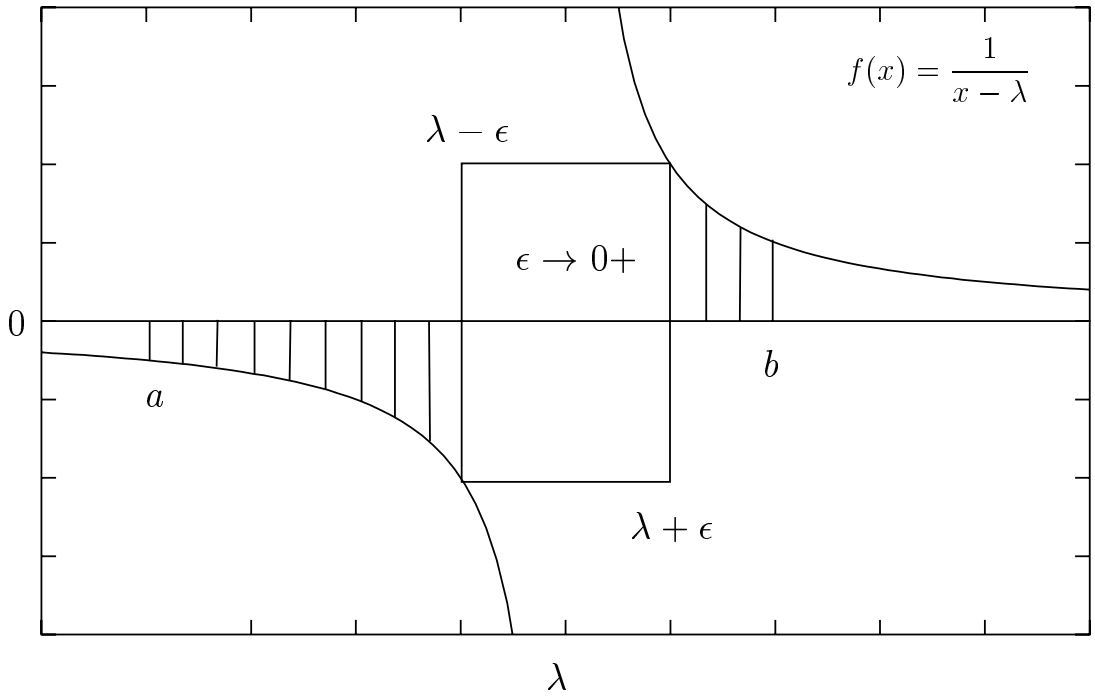


Abbildung 3.1: Grenzwertbildung bei Cauchy-Hauptwert-Integral

definiert durch

$$⨍_a^b f(x)\,dx = \lim_{\epsilon\to 0+}\left(\int_a^{\lambda-\epsilon} f(x)\,dx + \int_{\lambda+\epsilon}^b f(x)\,dx\right)\ , \tag{3.1}$$

vorausgesetzt der obige Grenzwert existiert.

Aus der Definition[1] wird unmittelbar klar, daß für Riemann-integrierbare Funktionen Cauchy-Hauptwert-Integral und Riemann-Integral übereinstimmen. Nach dem gleichen Prinzip werden Cauchy-Hauptwerte für unbeschränkte Integrationsbereiche definiert

$$⨍_{-\infty}^{\infty} f(x)\,dx = \lim_{r\to\infty}\left(\int_{-r}^{r} f(x)\,dx\right).$$

Durch geeignete Transformationen können diese jedoch wieder auf Cauchy-Hauptwert-Integrale mit endlichen Grenzen zurückgeführt werden, weswegen wir uns im weiteren ausschließlich auf Integrale mit endlichen Grenzen konzentrieren werden. Um eine hinreichende Bedingung für die Existenz des obigen Grenzwertes angeben zu können, benötigen wir den Begriff der Hölder-Stetigkeit,

Definition 17 *Eine auf dem Intervall $[a,b]$ definierte reelle Funktion f heißt Hölder-stetig mit der Ordnung q $(0 < q \leq 1)$ auf $[a,b]$, wenn gilt*

$$|f(x) - f(y)| \leq K|x-y|^q \tag{3.2}$$

[1]Eine allgemeinere Definition des Cauchy-Haupt-Wertes wird in [38] angegeben.

für $x, y \in [a,b]$ und einer geeigneten Konstanten $K > 0$.

Die Lipschitz-stetigen Funktionen sind gerade die Hölder-stetigen Funktionen der Ordnung Eins. Mit dem Begriff der Hölder-Stetigkeit können wir jetzt den folgenden Satz formulieren.

Satz 18 *Für jede Funktion $f : [a,b] \to \mathbb{R}$, die Hölder-stetig mit der Ordnung q $(0 < q \leq 1)$ ist, existiert der Cauchysche Hauptwert*

$$I(f;\ \lambda) := \fint_a^b \frac{f(x)}{x-\lambda}\,dx, \quad \lambda \in (a,b) \tag{3.3}$$

Beweis: Wegen der Hölder-Stetigkeit von f gilt

$$\left|\frac{f(x)-f(\lambda)}{x-\lambda}\right| \leq K \left|\frac{(x-\lambda)^q}{x-\lambda}\right| = K|x-\lambda|^{q-1},$$

$$\text{d. h. } \int_a^b \frac{f(x)-f(\lambda)}{x-\lambda}\,dx \text{ konvergiert absolut.}$$

Damit kann man (3.3) zerlegen in

$$\begin{aligned}\fint_a^b \frac{f(x)}{x-\lambda}\,dx &= \fint_a^b \frac{f(\lambda)}{x-\lambda}\,dx + \int_a^b \frac{f(x)-f(\lambda)}{x-\lambda}\,dx \\ &= f(\lambda)\ln\left(\frac{b-\lambda}{\lambda-a}\right) + \int_a^b \frac{f(x)-f(\lambda)}{x-\lambda}\,dx\end{aligned}$$

■

Aus dem Beweis des Satzes wird ersichtlich, daß die Stetigkeit von f keine hinreichende Bedingung für die Existenz des obigen Cauchy-Hauptwertes ist. Betrachten wir z. B. die Funktion $f(x) = \frac{x}{|x|\,\ln|x|}$. f ist im Nullpunkt stetig fortsetzbar und es gilt

$$\int_{-\frac{1}{2}}^{-\epsilon} \frac{f(x)}{x}\,dx + \int_{\epsilon}^{\frac{1}{2}} \frac{f(x)}{x}\,dx = 2\left[\ln(-\ln x)\right]_{\epsilon}^{\frac{1}{2}} \to -\infty \quad (\text{für } \epsilon \to 0+).$$

Umgekehrt ist die Hölder-Stetigkeit aber auch keine notwendige Bedingung für die Existenz, wie folgendes Beispiel zeigt. Die Funktion f

$$f(x) = \begin{cases} 0 & x = 0 \text{ oder } x = \frac{1}{n},\ n \in \mathbb{N} \\ 1 & \textit{sonst} \end{cases}$$

besitzt in $[-1,1]$ abzählbar viele Unstetigkeitsstellen, aber es gilt

$$\fint_{-1}^{1} \frac{f(x)}{x}\,dx = 0.$$

In der Literatur findet sich der Begriff Hilbert-Transformierte von f für die auf dem Intervall $[a,b]$ definierte Funktion $H_f(\lambda) := I(f;\ \lambda)$. Die Hilbert-Transformierte ist oft unbeschränkt, z. B. divergiert H_f für $f \equiv 1$ gegen -∞, wenn λ gegen b konvergiert, bzw. gegen ∞, wenn λ gegen a konvergiert.

Eine wichtige Rolle bei der Bestimmung von Quadraturformeln für Cauchy-Hauptwert-Integrale spielen die Funktionen zweiter Art, die folgendermaßen definiert sind.

Definition 19 *Sei $p_0, p_1, \ldots$ ein Orthogonalsystem reeller Polynome. Dann sind die Funktionen zweiter Art definiert durch*

$$q_n(z) := \fint_a^b \frac{p_n(x)}{x-z} w(x)\,dx, \quad z \in (a,b). \tag{3.4}$$

Funktionen zweiter Art können besonders einfach ausgewertet werden, da sie der Rekursiongleichung des dazugehörigen Polynomsystems genügen.

Satz 20 *Erfüllen die orthogonalen Polynome $p_0, p_1, \ldots$ die Rekursionsgleichung*

$$p_n(x) = (\alpha_n x - \beta_n)p_{n-1}(x) - \gamma_n p_{n-2}(x)\ ,\ n = 1, 2, \ldots$$

mit $p_0(x) = k_0$ und $p_{-1} = 0$, dann gilt für die gemäß (3.4) gebildeten (q_n) die gleiche Rekursionsgleichung mit den Startwerten

$$q_0(z) = \fint_a^b \frac{p_0(x)}{x-z} w(x)\,dx\ ,\ q_1(z) = p_1(z)q_0(z)/k_0 + \alpha_1 \int_a^b p_0(x)w(x)\,dx. \tag{3.5}$$

Beweis: Für $n = 1$

$$\begin{aligned} q_1(z) &= \fint_a^b \frac{(\alpha_1 x - \beta_1)p_0(x)}{x-z} w(x)\,dx \\ &= \fint_a^b \frac{\alpha_1(x-z+z)p_0(x)}{x-z} w(x)\,dx - \beta_1 q_0(z) \\ &= \alpha_1 \int_a^b p_0(x)w(x)\,dx + \alpha_1 z q_0(z) - \beta_1 q_0(z) \\ &= p_1(z)q_0(z)/k_0 + \alpha_1 \int_a^b p_0(x)w(x)\,dx \end{aligned}$$

Für $n > 1$

$$\begin{aligned} q_n(z) &= \fint_a^b \frac{(\alpha_n x - \beta_n)p_{n-1}(x) - \gamma_n p_{n-2}(x)}{x-z} w(x)\,dx \\ &= \fint_a^b \frac{\alpha_n(x-z+z)p_{n-1}(x)}{x-z} w(x)\,dx - \beta_n q_{n-1}(z) - \gamma_n q_{n-2}(z) \\ &= \alpha_n \int_a^b p_{n-1}(x)w(x)\,dx + \alpha_n z q_{n-1}(z) - \beta_n q_{n-1}(z) - \gamma_n q_{n-2}(z) \\ &= (\alpha_n z - \beta_n)q_{n-1}(z) - \gamma_n q_{n-2}(z) \end{aligned}$$

■

Eine direkte Folgerung aus dem vorangegangenen Satz ist, daß die Existenz von q_0 die Existenz der Funktionen q_n garantiert. Deshalb fordern wir von der Gewichtsfunktion w zusätzlich die Existenz von q_0. Zwischen den Nullstellen der Polynome p_n und den Nullstellen der dazugehörigen Funktionen zweiter Art besteht folgender Zusammenhang (vgl. [61]).

Satz 21 *Ist q_n die Funktion zweiter Art zum Orthogonalpolynom p_n, dann hat q_n mindestens n-1 Nullstellen, die mit den Nullstellen von p_n alternieren.*

3.2 Definition hypersinguläres Integral

Obwohl die Definition des Cauchy-Hauptwert-Integrals sehr anschaulich ist, so hat sie doch einen entscheidenden Nachteil. Sie ist z. B. nicht auf Integranden der Form $\frac{f(x)}{x^{p+1}}$ für $p > 0$ übertragbar. Deshalb benötigt man einen zusätzlichen Integralbegriff, der es erlaubt, auch solchen Integralen einen endlichen Wert zuzuweisen, ohne die typischen Integraleigenschaften wie Additivität und Linearität zu verlieren. Die Definition des hypersingulären Integrals nach Monegato [62] erfolgt in zwei Schritten. Im ersten Schritt wird es zunächst für Integranden der Form $\frac{1}{x^{p+1}}$ definiert. Dazu wird die Differenz der Ergebnisse, die durch Einsetzen der oberen bzw. der unteren Integrationsgrenze in die Stammfunktion enstehen, gebildet. Entsteht beim Einsetzen in die Stammfunktion ein divergenter Term, wird er einfach durch Null ersetzt. Das Ergebnis ist also der endliche Teil (finite part) der Differenz.

Definition 22 (Hypersinguläres Integral Teil 1) *Für $\lambda \in [a,b]$ mit $\lambda \neq a, b$ definieren wir*

$$\begin{aligned}
\oint_a^\lambda \frac{dx}{x-\lambda} &:= -\ln(\lambda - a) \\
\oint_\lambda^b \frac{dx}{x-\lambda} &:= \ln(b-\lambda) \\
\oint_a^b \frac{dx}{x-\lambda} &:= \oint_a^\lambda \frac{dx}{x-\lambda} + \oint_\lambda^b \frac{dx}{x-\lambda} = \ln\left(\frac{b-\lambda}{\lambda-a}\right).
\end{aligned}$$

und für positive reelle p

$$\oint_\lambda^b \frac{dx}{(x-\lambda)^{p+1}} := -\frac{1}{p(b-\lambda)^p}.$$

Ist $p \in \mathbb{N}$, so definieren wir zusätzlich noch

$$\begin{aligned}
\oint_a^\lambda \frac{dx}{(x-\lambda)^{p+1}} &:= \frac{1}{p(\lambda-a)^p} \\
\oint_a^b \frac{dx}{(x-\lambda)^{p+1}} &:= \oint_a^\lambda \frac{dx}{(x-\lambda)^{p+1}} + \oint_\lambda^b \frac{dx}{(x-\lambda)^{p+1}} = -\frac{1}{p}\left[\frac{1}{(x-\lambda)^p}\right]_a^b \\
&= \frac{1}{p(\lambda-a)^p} - \frac{1}{p(b-\lambda)^p}.
\end{aligned}$$

Im zweiten Teil wird das hypersinguläre Integral allgemeinerer Funktionen definiert. Hierzu wird der Integrand in ein (uneigentliches) Riemann-Integral und eine Summe von hypersingulären Integralen, wie sie in Teil 1 schon definiert worden sind, zerlegt.

Definition 23 (Hypersinguläres Integral Teil 2) *Sei $p > 0$ und $r = \lceil p \rceil^2$. Weiterhin sei f eine auf dem Intervall $[a,b]$ Riemann-integrierbare Funktion, die in $[a, a+\epsilon]$ ($\epsilon > 0$ hinreichend klein gewählt) r mal stetig differenzierbar ist. Dann wird das hypersinguläre Integral $\oint_a^b \frac{f(x)}{(x-a)^{p+1}}\,dx$ definiert durch*

$$\begin{aligned} \oint_a^b \frac{f(x)}{(x-a)^{p+1}}\,dx &:= \int_a^b \frac{f(x) - \sum_{k=0}^{r} f^{(k)}(a)(x-a)^k/k!}{(x-a)^{p+1}}\,dx \\ &\quad + \sum_{k=0}^{r} \frac{f^{(k)}(a)}{k!} \oint_a^b \frac{dx}{(x-a)^{p+1-k}}. \end{aligned}$$

Existiert für $p \in \mathbb{N} \cup \{0\}$ ein reelles $\epsilon > 0$ mit $f \in C^p[\lambda-\epsilon, \lambda+\epsilon]$, so daß die p-te Ableitung $f^{(p)}$ von f Hölder-stetig in $[\lambda-\epsilon, \lambda+\epsilon]$ ist, dann definieren wir

$$\begin{aligned} \oint_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx &:= \int_a^b \frac{f(x) - \sum_{k=0}^{p} f^{(k)}(\lambda)(x-\lambda)^k/k!}{(x-\lambda)^{p+1}}\,dx \\ &\quad + \sum_{k=0}^{p} \frac{f^{(k)}(\lambda)}{k!} \oint_a^b \frac{dx}{(x-\lambda)^{p+1-k}}. \end{aligned} \tag{3.6}$$

Die Voraussetzungen des Satzes garantieren, daß das erste Integral auf der rechten Seite vom Gleichheitszeichen ein (uneigentliches) Riemann-Integral ist. Die in der Summe auftretenden stark singulären Integrale sind im ersten Teil der Definition definiert worden. Für $p = 0$ und $\lambda \in (a,b)$ stimmt obige Definition mit der Definition des Cauchy-Hauptwert-Integrals überein. Wir haben es also mit einer weiteren Verallgemeinerung des Integral-Begriffs zu tun.

3.3 Eigenschaften stark singulärer Integrale

Die Definition 22 erscheint auf den ersten Blick etwas willkürlich gewählt, doch es zeigt sich, daß die so definierten hypersingulären Integrale viele typische Integraleigenschaften aufweisen. Setzen wir

$$I(f;\ \lambda) := \oint_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx, \tag{3.7}$$

so ist I ein lineares Funktional, es gilt also

$$I(\alpha f + \beta g, \lambda) = \alpha \oint_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx + \beta \oint_a^b \frac{g(x)}{(x-\lambda)^{p+1}}\,dx,$$

[2] $\lceil p \rceil$ bezeichne die kleinste ganze Zahl mit $\lceil p \rceil \geq p$.

vorausgesetzt die Integrale der rechten Seite existieren. Dies ist auf jeden Fall sichergestellt, wenn wir als Funktionenraum $C^{r+1}[a,b]$ mit $r = \lceil p \rceil$ wählen. I ist auf dem Banachraum $C^{r+1}[a,b]$ mit der Norm

$$\|f\| := \sum_{k=0}^{r+1} \|f^{(k)}\|_\infty = \sum_{k=0}^{r+1} \max_{a \le t \le b} |f^{(k)}(t)|$$

stetig. Die gleichmäßige Konvergenz einer Funktionenfolge (f_n) gegen f ist allerdings nicht ausreichend für die Konvergenz von $I(f_n, \lambda)$ gegen $I(f;\ \lambda)$ wie folgendes leicht modifizierte Beispiel aus [79] zeigt.

$$f_n(x) = \begin{cases} \dfrac{1}{n}\,(x+1)^{2^n} & -1 \le x < 0 \\ \dfrac{1}{n} & 0 \le x \le 1 \end{cases}$$

(f_n) konvergiert gleichmäßig auf $[-1,1]$ gegen die Nullfunktion, für die dazugehörigen Cauchyschen Hauptwerte $I(f_n;\ 0)$ gilt jedoch

$$\begin{aligned} I(f_n;\ 0) &= \ \ \Xint-_{-1}^{1} \frac{f_n(x) - f_n(0)}{x}\,dx = \frac{1}{n}\int_{-1}^{0} \frac{(x+1)^{2^n} - 1}{x}\,dx \\ &= \frac{1}{n}\sum_{k=1}^{2^n} \frac{1}{k} \ge \frac{n-1}{2n} \to \frac{1}{2}. \end{aligned}$$

Stark singuläre Integrale sind additiv, d. h. es gilt

$$\Xint-_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx = \Xint-_a^c \frac{f(x)}{(x-\lambda)^{p+1}}\,dx + \Xint-_c^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx.$$

Das Funktional I ist nicht (positiv) definit. Z. B. gilt für $p = 1$

$$\Xint-_{-1}^{1} \frac{1}{(x-\frac{1}{2})^2}\,dx = -\frac{4}{3}.$$

Die Substitutionsregel gilt für stark singuläre Integrale in einer eingeschränkten Form. Wir wollen sie aber nur für den Spezialfall einer linearen Transformation formulieren.

Satz 24 (Lineare Transformation) *Sei p eine natürliche Zahl, $a < \lambda < b$ und $c < d$, dann gilt:*

$$\Xint-_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx = \left(\frac{d-c}{b-a}\right)^p \Xint-_c^d \frac{f\left(\frac{b-a}{d-c}x + \frac{ad-bc}{d-c}\right)}{(x - \frac{bc-da+\lambda(d-c))}{b-a})^{p+1}}\,dx \tag{3.8}$$

Ersetzen wir λ durch a, so gilt Gleichung (3.8) nur noch, falls p keine natürliche Zahl ist. Für $p \in \mathbb{N}$ gilt jedoch:

$$\Xint-_\lambda^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx = \left(\frac{d-c}{b-\lambda}\right)^p \Xint-_c^d \frac{f\left(\frac{b-\lambda}{d-c}x + \frac{\lambda d-bc}{d-c}\right)}{(x-c)^{p+1}}\,dx + \frac{f^{(p)}(\lambda)}{p!}\ln\left(\frac{b-\lambda}{d-c}\right) \tag{3.9}$$

Beweis: Gleichung (3.8) erhalten wir, indem wir zunächst Definition 23 anwenden und dann die Transformation $x = \frac{b-a}{d-c}t + \frac{da-bc}{d-c}$ durchführen. Gleichung (3.9) gilt wegen

$$\oint_\lambda^b \frac{1}{x-\lambda}\,dx = \oint_c^d \frac{1}{x-c}\,dx + \ln\left(\frac{b-\lambda}{d-c}\right).$$

■

Die folgenden zwei Eigenschaften entnehmen wir [61]. In beiden Sätzen setzen wir voraus, daß f hinreichend oft stetig differenzierbar ist, so daß die auftretenden Integrale die Voraussetzungen von Definition 23 erfüllen.

Satz 25 (Vertauschung Differentiation und Integration) *Für $p > 0$, $p \notin \mathbb{N}$ gilt:*

$$\frac{d}{d\lambda}\oint_\lambda^b \frac{f(x)}{(x-\lambda)^p}\,dx = p\oint_\lambda^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx.$$

Sei $p \in \mathbb{N}$ und $a < \lambda < b$, dann gilt

$$\begin{aligned}\frac{d}{d\lambda}\oint_\lambda^b \frac{f(x)}{(x-\lambda)^p}\,dx &= p\oint_\lambda^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx - \frac{f^{(p)}(\lambda)}{p!}\\ \frac{d}{d\lambda}\oint_a^b \frac{f(x)}{(x-\lambda)^p}\,dx &= p\oint_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx\end{aligned}$$

Eine direkte Folgerung aus Satz 25 ist es, daß hypersinguläre Integrale der Form (3.6) gerade höhere Ableitungen eines Cauchy-Hauptwert-Integrals sind, denn für $p \in \mathbb{N}$ und $a < \lambda < b$ gilt

$$\frac{d^p}{d^p\lambda}\oint_a^b \frac{f(x)}{(x-\lambda)}\,dx = p!\oint_a^b \frac{f(x)}{(x-\lambda)^{p+1}}\,dx.$$

Satz 26 (Partielle Integration) *Für $a < \lambda < b$ und $p \in \mathbb{N}$ gilt*

$$\oint_a^b \frac{f(x)\,dx}{(x-\lambda)^{p+1}} = -\frac{1}{p}\left[\frac{f(x)}{(x-\lambda)^p}\right]_a^b + \oint_a^b \frac{f'(x)\,dx}{p(x-\lambda)^p}.$$

3.4 Hauptwertformeln

Zur numerischen Approximation von Cauchy-Hauptwert-Integralen der Form

$$I(f;\,\lambda) = \oint_a^b \frac{f(x)}{(x-\lambda)}\,w(x)\,dx \tag{3.10}$$

werden analog zu den Näherungsverfahren für Riemann-Integrale Quadraturformeln verwendet. In der Literatur werden diese Quadraturformeln auch Hauptwertformeln genannt. Wir wollen die zwei wichtigsten Methoden zur Konstruktion von Hauptwertformeln genauer betrachten.

3.4.1 Interpolationsquadraturverfahren

Ein naheliegender Ansatz wird von Elliott und Paget in ([18]) verwendet. Hierbei wird die zu integrierende Funktion f an von der Singularität unabhängigen Stützstellen $x_1, x_2, \ldots, x_n$ durch ein Polynom interpoliert. Den Näherungswert $Q_n(f;\ \lambda)$ erhalten wir, indem wir die Funktion f durch ihr Interpolationspolynom ersetzen und den entsprechenden Cauchy-Hauptwert bilden. Bezeichne L_n das Interpolationspolynom

$$L_n(x) = \sum_{i=1}^{n} f(x_i) l_i(x)$$

mit den Lagrangefaktoren l_i

$$l_i(x) = \frac{p_n(x)}{p_n'(x_i)(x - x_i)}, \quad p_n(x) = k_n \prod_{i=1}^{n}(x - x_i), \quad k_n \neq 0,$$

dann erhalten wir mit dem Ansatz

$$\begin{aligned} Q_n(f;\ \lambda) &= \sum_{i=1}^{n} w_i(\lambda) f(x_i) = I(L_n;\ \lambda) \\ &= \sum_{i=1}^{n} \barint_a^b \frac{p_n(x)}{p_n'(x_i)(x - x_i)} \frac{w(x)}{(x - \lambda)}\, dx\, f(x_i) \end{aligned}$$

und der Darstellung des Interpolationsfehlers

$$f(x) - L_n(x) = \frac{p_n(x) f^{(n)}(\xi(x))}{k_n n!} \tag{3.11}$$

folgenden Satz (Beweis siehe [18]).

Satz 27 *Gegeben sei die Hauptwertformel $Q_n(f;\ \lambda)$ mit den Stützstellen x_i und den Gewichten*[3] *$w_i(\lambda)$:*

$$\begin{aligned} Q_n(f;\ \lambda) &= \sum_{i=1}^{n} w_i(\lambda)\, f(x_i) \\ w_i(\lambda) &= \begin{cases} \dfrac{q_n(x_i) - q_n(\lambda)}{p_n'(x_i)(x_i - \lambda)} & \text{für} \quad x_i \neq \lambda \\ \dfrac{q_n'(\lambda)}{p_n'(\lambda)} & \text{falls} \quad x_i = \lambda \end{cases} \end{aligned} \tag{3.12}$$

Für $f \in C^n[a,b]$ hat das Restglied $R_n(f;\ \lambda)$ die Darstellung

$$R_n(f;\ \lambda) = \frac{1}{n!\, k_n} \barint_a^b \frac{p_n(x)}{x - \lambda} f^{(n)}(\xi(x)) w(x)\, dx, \quad \xi(x) \in (a,b) \tag{3.13}$$

[3] q_n bezeichne die in Definition 19 erklärten Funktionen zweiter Art.

	λ			
	0	0.25	0.5	0.75
$w_1(\lambda)$	$-1.05E-1$	$-4.58E-2$	$-1.08E-1$	$-2.79E-2$
$w_2(\lambda)$	$-2.79E-1$	$-3.53E-1$	$-3.26E-2$	$-2.55E-1$
$w_3(\lambda)$	$-5.97E-1$	$-8.74E-2$	$-5.99E-1$	$-2.06E-2$
$w_4(\lambda)$	$-1.98E+0$	$-1.54E+0$	$1.65E-2$	$-7.71E-1$
$w_5(\lambda)$	$1.98E+0$	$-8.59E-1$	$-2.33E+0$	$-9.77E-3$
$w_6(\lambda)$	$5.97E-1$	$2.03E+0$	$4.97E-1$	$-2.68E+0$
$w_7(\lambda)$	$2.79E-1$	$1.38E-1$	$1.36E+0$	$1.08E+0$
$w_8(\lambda)$	$1.05E-1$	$2.07E-1$	$9.81E-2$	$7.36E-1$

Tabelle 3.1: Gewichte der Hauptwertformel aus Satz 28 für $w \equiv 1$ und $n = 8$

Sind die Stützstellen x_i gerade die Knoten der entsprechenden Gauß-Quadraturformel

$$\int_a^b f(x)w(x)dx = \sum_{i=1}^{n} w_i f(x_i) + \frac{h_n}{k_n^2}\frac{f^{(2n)}(\xi)}{(2n)!} \quad \xi \in (a,b) \tag{3.14}$$

mit $h_n = \int_a^b p_n^2(x)w(x)\,dx$, dann folgt aus Satz 27:

Satz 28 *Sei $Q_n(f;\ \lambda)$ die Haupwertformel, die durch Interpolation an den Stützstellen der entsprechenden Gauß-Quadraturformel ensteht, so gilt*

$$w_i(\lambda) = w_i \sum_{k=0}^{n-1} \frac{1}{h_k} p_k(x_i)q_k(\lambda)\ ,\ i = 1, \ldots, n\ . \tag{3.15}$$

Für $f \in C^{(2n+1)}[a,b]$ hat das Restglied die Darstellung

$$R_n(f;\ \lambda) = \frac{q_n(\lambda)}{k_n n!} f^{(n)}(\xi_1) + \frac{h_n}{k_n^2(2n+1)!} f^{(2n+1)}(\xi_2), \quad \xi_1, \xi_2 \in (a,b). \tag{3.16}$$

Beweis siehe [18].

Der Exaktheitsgrad der Hauptwertformel aus Satz 28 ist, wie wir unschwer aus (3.16) ablesen können, mindestens n. Ist die Singularität λ eine Nullstelle der Funktion q_n, so hat die Hauptwertformel sogar den Exaktheitsgrad $2n$.
Tabelle 3.1 entnehmen wir, daß die nach obigem Satz für die Legendre-Gewichtsfunktion gebildeten Hauptwertformeln sowohl positive als auch negative Gewichte enthalten. Fordern wir $deg(Q_n) \geq 0$, so ist sofort klar, daß es sinnlos ist, nach Hauptwertformel mit ausschließlich positiven Gewichten zu suchen.

3.4.2 Hauptwertformel von Hunter

Ein ähnlicher Ansatz führt zu den Hauptwertformeln von Hunter [36]. Zusätzlich zu den Stützstellen $x_1, x_2, \ldots, x_n$ wird f noch an der Stelle λ interpoliert. Fällt λ mit einer Stützstelle x_j zusammen, so wird f in $x_1, x_2, \ldots, x_n$ und f' in x_j interpoliert. Das Ergebnis ist im folgenden Satz formuliert.

Satz 29 *Sind $x_1, \ldots, x_n$ die Stützstellen der Gauß-Quadraturformel (3.14), dann hat, falls λ mit keiner Stützstelle übereinstimmt, die Hunterformel Q^H_{n+1}*

$$Q^H_{n+1}(f;\ \lambda) = \sum_{i=1}^{n} w_i(\lambda) f(x_i) + w_0(\lambda) f(\lambda) \tag{3.17}$$

mit den Gewichten $w_i(\lambda)$

$$w_i(\lambda) = \begin{cases} \dfrac{w_i}{x_i - \lambda} & \text{für} \quad i = 1, \ldots, n \\ \dfrac{q_n(\lambda)}{p_n(\lambda)} & \text{für} \quad i = 0 \end{cases} \tag{3.18}$$

für $f \in C^{(2n+1)}[a,b]$ die Restglieddarstellung

$$R^H_{n+1}(f;\ \lambda) = \frac{h_n}{k_n^2 (2n+1)!} f^{(2n+1)}(\xi)\ , \xi \in (a,b). \tag{3.19}$$

Die Fehlerdarstellung (3.19) behält Ihre Gültigkeit, falls λ mit einer Stützstelle x_j zusammenfällt. Die Quadratursumme hat dann jedoch die Form

$$Q^H_{n+1}(f;\ \lambda) = \sum_{i=1}^{n} w_i(\lambda) f(x_i) + w_0(\lambda) f'(\lambda) \tag{3.20}$$

mit den Gewichten $w_i(\lambda)$

$$w_i(\lambda) = \begin{cases} \dfrac{q_n'(\lambda)}{p_n'(\lambda)} - \displaystyle\sum_{k=1, k\neq j}^{n} \frac{q_n(\lambda)}{p_n'(x_k)(x_k - \lambda)} & \text{für } i = j \\ w_j = \dfrac{q_n(\lambda)}{p_n'(\lambda)} & \text{für } i = 0 \\ \dfrac{w_i}{x_i - \lambda} & \text{sonst} \end{cases} \cdot \tag{3.21}$$

Obiger Satz wird in [18] bewiesen.

Die Hauptwertformel aus Satz 29 ist für alle Polynome exakt, deren Grad niedriger als $2n+1$ ist. Ist die Singularität λ eine Nullstelle der Funktion zweiter Art q_n, so sind die Hauptwertformeln aus Satz 28 und Satz 29 identisch.

	λ			
	0	0.25	0.5	0.75
$w_0(\lambda)$	$0.00E+0$	$4.79E+0$	$-1.21E+1$	$-4.32E+0$
$w_1(\lambda)$	$-1.05E-1$	$-8.36E-2$	$-6.93E-2$	$-5.92E-2$
$w_2(\lambda)$	$-2.79E-1$	$-2.12E-1$	$-1.72E-1$	$-1.44E-1$
$w_3(\lambda)$	$-5.97E-1$	$-4.05E-1$	$-3.06E-1$	$-2.46E-1$
$w_4(\lambda)$	$-1.98E+0$	$-8.37E-1$	$-5.31E-1$	$-3.89E-1$
$w_5(\lambda)$	$1.98E+0$	$-5.45E+0$	$-1.15E+0$	$-6.40E-1$
$w_6(\lambda)$	$5.97E-1$	$1.14E+0$	$1.23E+1$	$-1.40E+0$
$w_7(\lambda)$	$2.79E-1$	$4.07E-1$	$7.50E-1$	$4.77E+0$
$w_8(\lambda)$	$1.05E-1$	$1.43E-1$	$2.20E-1$	$4.81E-1$

Tabelle 3.2: Gewichte der Hunterformel (Legendre-Gewichtsfunktion, $n = 8$)

Ein Blick auf die Werte der Tabelle 3.2 zeigt, daß die Gewichte der Hunterformeln betragsmäßig größere Werte als die Gewichte der entsprechenden Hauptwertformeln der Tabelle 3.1 annehmen. Für $\lambda = 0$ stimmen die beiden Hauptwertformeln überein. Aus den Gleichungen (3.18) und (3.21) ist ersichtlich, daß bei festem n die Gewichte beliebig groß werden können, wenn die Singularität λ hinreichend nahe an einer Stützstelle liegt. Die Hunterformeln werden auch als modifizierte Quadraturformeln bezeichnet. Spalten wir nämlich vom Integranden in $I(f;\ \lambda)$ die Singularität ab

$$I(f;\ \lambda) = f(\lambda)\, q_0(\lambda) + \int_a^b \frac{f(x) - f(\lambda)}{x - \lambda} w(x)\, dx \tag{3.22}$$

und wenden auf das rechtsstehende Riemann-Integral die Gauß-Quadraturformel (3.14) an, so erhalten wir gerade die Hauptwertformel des Satzes 29.

3.5 Quadraturformeln für hypersinguläre Integrale

Ausgehend von der interpolatorischen Hauptwertformel ($deg(Q_n) \geq n - 1$)

$$⨍_a^b \frac{f(x)}{x - \lambda} w(x)\, dx = \sum_{i=1}^{n} w_i(\lambda) f(x_i) + \frac{1}{n!\, k_n} ⨍_a^b \frac{p_n(x)}{x - \lambda} f^{(n)}(\xi(x)) w(x)\, dx, \quad \xi(x) \in (a, b)$$

konstruieren wir nun eine Quadraturformel für hypersinguläre Integrale der Form

$$I(f;\ \lambda) = ⨎_a^b \frac{f(x)}{(x - \lambda)^2}\, w(x)\, dx, \tag{3.23}$$

indem wir den Ansatz aus Abschnitt 3.4.1 (siehe [18]) verwenden.

$$\begin{aligned} Q_n^{hyp}(f;\,\lambda) &= \sum_{i=1}^{n} w_i^{hyp}(\lambda) f(x_i) = \oint_a^b \frac{L_n(x)}{(x-\lambda)^2} w(x)\,dx \\ &= \oint_a^b \sum_{i=1}^{n} f(x_i) l_i(x) \frac{w(x)}{(x-\lambda)^2}\,dx \\ &= \sum_{i=1}^{n} f(x_i) \oint_a^b \frac{p_n(x)}{p_n'(x_i)(x-x_i)} \frac{w(x)}{(x-\lambda)^2}\,dx. \end{aligned} \tag{3.24}$$

Der Vergleich mit (3.24) ergibt

$$w_i^{hyp}(\lambda) = \frac{1}{p_n'(x_i)} \oint_a^b \frac{p_n(x)}{(x-x_i)(x-\lambda)^2} w(x)\,dx.$$

Für $x_i \neq \lambda$ gilt weiter

$$\begin{aligned} w_i^{hyp}(\lambda) &= \frac{1}{p_n'(x_i)} \oint_a^b \frac{1}{(x_i-\lambda)(x-\lambda)} \left(\frac{p_n(x)}{x-x_i} - \frac{p_n(x)}{x-\lambda} \right) w(x)\,dx \\ &= \frac{1}{p_n'(x_i)(x_i-\lambda)} \left(\sum_{k=1}^{n} w_k(\lambda) \frac{p_n(x_k)}{x_k-x_i} - q_n'(\lambda) \right) \\ &= \frac{1}{p_n'(x_i)(x_i-\lambda)} (w_i(\lambda) p_n'(x_i) - q_n'(\lambda)) \\ &= \frac{w_i(\lambda)}{x_i-\lambda} - \frac{q_n'(\lambda)}{p_n'(x_i)(x_i-\lambda)}. \end{aligned}$$

Dabei bezeichne q_n die Funktionen zweiter Art und q_n' die dazugehörigen Ableitungen

$$q_n'(\lambda) = \oint_a^b \frac{p_n(x)}{(x-\lambda)^2}\,w(x)\,dx.$$

Aus der Darstellung des Interpolationsfehlers (3.11) ergibt sich die Darstellung des Restglieds

$$\begin{aligned} R_n^{hyp}(f;\,\lambda) &= I(f;\,\lambda) - Q_n^{hyp}(f;\,\lambda) = I(f;\,\lambda) - I(L_n;\,\lambda) \\ &= \oint_a^b \frac{f(x)-L_n(x)}{(x-\lambda)^2}\,w(x)\,dx \\ &= \frac{1}{n!\,k_n} \oint_a^b \frac{p_n(x)}{(x-\lambda)^2} f^{(n)}(\xi(x)) w(x)\,dx. \end{aligned}$$

Satz 30 *Gegeben sei die Quadraturformel* $Q_n^{hyp}(f;\,\lambda) = \sum_{i=1}^{n} w_i^{hyp}(\lambda) f(x_i)$ *mit den Gewichten*

$$w_i^{hyp}(\lambda) = \begin{cases} \dfrac{w_i(\lambda) p_n'(x_i) - q_n'(\lambda)}{p_n'(x_i)(x_i-\lambda)} & \text{für } x_i \neq \lambda \\ \dfrac{q_n''(\lambda)}{2p_n'(\lambda)} & \text{für } x_i = \lambda \end{cases}.$$

Für $f \in C^n(a,b)$ hat das Fehlerglied $R_n^{hyp}(f;\ \lambda)$ die Darstellung

$$R_n^{hyp}(f;\ \lambda) = \frac{1}{n!\,k_n} \Xint=_a^b \frac{p_n(x)}{(x-\lambda)^2} f^{(n)}(\xi(x)) w(x)\,dx, \quad \xi(x) \in (a,b). \tag{3.25}$$

Wählen wir für x_i die Stützstellen der Gauß-Quadraturformel (3.14) , so erhalten wir das zu Satz 28 entsprechende Analogon.

Satz 31 *Sind die Stützstellen der Quadraturformel aus Satz 30 die Knoten der Gauß-Quadraturformel (3.14), dann gilt*

$$w_i^{hyp}(\lambda) = w_i \sum_{k=0}^{n-1} \frac{1}{h_k} p_k(x_i) q_k'(\lambda)\ ,\ i = 1, \ldots, n\ . \tag{3.26}$$

Für $f \in C^{(2n+2)}[a,b]$ hat das Restglied die Form

$$R_n^{hyp}(f;\ \lambda) = \frac{q_n'(\lambda)}{k_n n!} f^{(n)}(\xi_0) + \frac{q_n(\lambda)}{k_n (n+1)!} f^{(n+1)}(\xi_1) + \frac{h_n}{k_n^2 (2n+2)!} f^{(2n+2)}(\xi_2) \tag{3.27}$$

mit $\xi_0, \xi_1, \xi_2 \in (a,b)$.

Beweis: Der Beweis verläuft ähnlich zu dem Beweis von Satz 28 (siehe [18]).
$L_n(x)$ läßt sich als Linearkombination der Polynome $p_0, \ldots, p_{n-1}$ darstellen

$$L_n(x) = \sum_{k=0}^{n-1} a_k\, p_k(x)$$

mit den Koeffizienten

$$\begin{aligned} a_k &= \frac{1}{h_k} \int_a^b L_n(x) p_k(x) w(x)\,dx \\ &= \frac{1}{h_k} \sum_{i=1}^{n} w_i f(x_i) p_k(x_i). \end{aligned} \tag{3.28}$$

Damit können wir $Q_n^{hyp}(f;\ \lambda)$ gemäß

$$\begin{aligned} Q_n^{hyp}(f;\ \lambda) &= I(L_n(x), \lambda) = \Xint=_a^b \sum_{k=0}^{n-1} a_k \frac{p_k(x)}{(x-\lambda)^2} w(x)\,dx \\ &= \sum_{k=0}^{n-1} a_k\, q_k'(\lambda) \\ &= \sum_{i=1}^{n} \left(w_i \sum_{k=0}^{n-1} \frac{1}{h_k} p_k(x_i) q_k'(\lambda) \right) f(x_i) \end{aligned}$$

bestimmen. Nun definieren wir uns eine auf dem Intervall $[a,b]$ differenzierbare Funktion s_n durch

$$s_n(x) = \frac{f(x) - L_n(x)}{p_n(x)} = \sum_{i=1}^{n} \frac{f(x) - f(x_i)}{p_n'(x_i)(x - x_i)}. \tag{3.29}$$

Damit können wir das Restglied $R_n^{hyp}(f;\ \lambda)$ darstellen

$$R_n^{hyp}(f;\ \lambda) = I(f - L_n;\ \lambda) = \Xint= _a^b \frac{p_n(x)s_n(x)}{(x-\lambda)^2} w(x)\,dx \tag{3.30}$$

$$= s_n(\lambda)\, q_n'(\lambda) + \Xint= _a^b p_n(x)\frac{s_n(x) - s_n(\lambda)}{(x-\lambda)^2} w(x)\,dx. \tag{3.31}$$

Nun betrachten wir die zwei Summanden aus (3.31) getrennt. Wegen (3.11) existiert ein $\xi_0 \in (a,b)$, so daß gilt

$$s_n(\lambda) = f^{(n)}(\xi_0)/(k_n n!),$$

womit wir folgende Darstellung des ersten Summanden erhalten

$$R_n^{(1)}(f;\ \lambda) := s_n(\lambda)\, q_n'(\lambda) = q_n'(\lambda) f^{(n)}(\xi_0)/(k_n n!). \tag{3.32}$$

Wenden wir die Hauptwertformel (3.15) auf die Funktion h

$$h(x) := p_n(x)\frac{s_n(x) - s_n(\lambda)}{x - \lambda} = r(x) + \frac{f(x) - f(\lambda)}{x - \lambda}$$

mit

$$r(x) := \frac{f(\lambda) - L_n(x) - p_n(x)s_n(\lambda)}{x - \lambda} \in I\!R^{n-1}[x] \tag{3.33}$$

an, so erhalten wir

$$\begin{aligned}
R_n^{(2)}(f;\ \lambda) &:= \Xint= _a^b p_n(x)\frac{s_n(x) - s_n(\lambda)}{(x-\lambda)^2} w(x)\,dx \\
&= \frac{q_n(\lambda)}{k_n n!}\frac{d^n}{dx^n}\left(p_n(x)\frac{s_n(x) - s_n(\lambda)}{x - \lambda}\right)(\xi_1) \\
&\quad + \frac{h_n}{k_n^2(2n+1)!}\frac{d^{2n+1}}{dx^{2n+1}}\left(p_n(x)\frac{s_n(x) - s_n(\lambda)}{x - \lambda}\right)(\xi_2) \\
&= \frac{q_n(\lambda)}{k_n n!}\frac{d^n}{dx^n}\left(\frac{f(x) - f(\lambda)}{x - \lambda}\right)(\xi_1) + \frac{h_n}{k_n^2(2n+1)!}\frac{d^{2n+1}}{dx^{2n+1}}\left(\frac{f(x) - f(\lambda)}{x - \lambda}\right)(\xi_2) \\
&= \frac{q_n(\lambda)}{k_n n!}\frac{d^n}{dx^n}\left(\int_0^1 f'(\lambda + (x-\lambda)t)\,dt\right)(\xi_1) \\
&\quad + \frac{h_n}{k_n^2(2n+1)!}\frac{d^{2n+1}}{dx^{2n+1}}\left(\int_0^1 f'(\lambda + (x-\lambda)t)\,dt\right)(\xi_2).
\end{aligned}$$

Vertauschen der Reihenfolge der Differentiation und Integration ergibt schließlich

$$\begin{aligned}
R_n^{(2)}(f;\ \lambda) &= \frac{q_n(\lambda)}{k_n n!}\left(\int_0^1 \frac{d^n}{dx^n}(f'(\lambda + (x-\lambda)t))\,dt\right)(\xi_1) \\
&\quad + \frac{h_n}{k_n^2(2n+1)!}\left(\int_0^1 \frac{d^{2n+1}}{dx^{2n+1}}(f'(\lambda + (x-\lambda)t))\,dt\right)(\xi_2) \\
&= \frac{q_n(\lambda)}{k_n(n+1)!} f^{(n+1)}(\xi_1) + \frac{h_n}{k_n^2(2n+2)!} f^{(2n+2)}(\xi_2).
\end{aligned}$$

■

Nun verwenden wir den Ansatz aus Abschnitt 3.4.2 zur Konstruktion einer weiteren Quadraturformel Q_{n+1}^{hyp}. Dazu interpolieren wir f zusätzlich in λ, d. h. für $\lambda \neq x_i$ bilden wir das Interpolationspolynom $L_{n+1}(x,\lambda)$

$$L_{n+1}(x,\lambda) := \sum_{i=1}^{n} \frac{(x-\lambda)l_i(x)}{x_i-\lambda} f(x_i) + \frac{p_n(x)}{p_n(\lambda)} f(\lambda).$$

Aus $Q_{n+1}^{hyp}(f;\ \lambda) = I(L_{n+1}(\cdot,\lambda);\ \lambda)$ folgt

$$\begin{aligned} Q_{n+1}^{hyp}(f;\ \lambda) &= \ \ \Xint=_a^b \left(\sum_{i=1}^{n} \frac{l_i(x)}{(x_i-\lambda)(x-\lambda)} f(x_i) + \frac{p_n(x)}{p_n(\lambda)(x-\lambda)^2} f(\lambda) \right) w(x)\,dx \\ &= \sum_{i=1}^{n} \frac{f(x_i)}{x_i-\lambda} \Xint=_a^b \frac{l_i(x)}{x-\lambda} w(x)\,dx + f(\lambda) \Xint=_a^b \frac{p_n(x)}{p_n(\lambda)(x-\lambda)^2} w(x)\,dx \\ &= \sum_{i=1}^{n} \frac{w_i(\lambda)}{x_i-\lambda} f(x_i) + \frac{q_n'(\lambda)}{p_n(\lambda)} f(\lambda). \end{aligned}$$

Wenden wir die Quadraturformel aus Satz 30 auf $I(f;\ \lambda)$ an, so folgt unter Berücksichtigung der Gleichungen (3.30) und (3.29) für $\lambda \neq x_i$ $(i = 1,\ldots,n)$

$$I(f;\ \lambda) - Q_{n+1}^{hyp}(f;\ \lambda) \ = \ -q_n'(\lambda)s_n(\lambda) + \Xint=_a^b \frac{p_n(x)s_n(x)}{(x-\lambda)^2}\, w(x)\,dx. \qquad (3.34)$$

Nehmen wir für x_i die Knoten der entsprechenden Gauß-Quadraturformel, so erhalten wir den folgenden Satz.

Satz 32 *Bezeichne $x_1,\ldots,x_n$ die Stützstellen der Gauß-Quadraturformel (3.14) und $w_1(\lambda),\ldots,w_n(\lambda)$ die Gewichte der Hauptwerformel (3.15). Die Quadraturformel Q_{n+1}^{hyp} sei für $x_i \neq \lambda$ definiert durch $Q_{n+1}^{hyp}(f;\ \lambda) = \sum_{i=1}^{n} w_i^{hyp}(\lambda) f(x_i) + w_0^{hyp}(\lambda) f(\lambda)$ mit den Gewichten*

$$w_i^{hyp}(\lambda) = \begin{cases} \dfrac{w_i(\lambda)}{x_i-\lambda} & \text{für } i = 1,\ldots,n \\ \dfrac{q_n'(\lambda)}{p_n(\lambda)} & \text{für } i = 0 \end{cases} \qquad (3.35)$$

bzw. für $x_j = \lambda$ durch $Q_{n+1}^{hyp}(f;\ \lambda) = \sum_{i=1}^{n} w_i^{hyp}(\lambda) f(x_i) + w_0^{hyp}(\lambda) f'(\lambda)$ mit den Gewichten

$$w_i^{hyp}(\lambda) = \begin{cases} \dfrac{q_n'(\lambda)}{p_n'(\lambda)} & \text{für } i = 0 \\ \dfrac{q_n''(\lambda)}{2p_n'(\lambda)} - \displaystyle\sum_{k=1,k\neq j}^{n} \frac{q_n'(\lambda)}{p_n'(x_k)(x_k-\lambda)} & \text{für } i = j \\ \dfrac{w_i(\lambda)}{x_i-\lambda} & \text{sonst} \end{cases} . \qquad (3.36)$$

Für $f \in C^{2n+2}[a,b]$ gilt

$$R_{n+1}^{hyp}(f;\ \lambda) = \frac{q_n(\lambda)}{k_n(n+1)!} f^{(n+1)}(\xi_1) + \frac{h_n}{k_n^2(2n+2)!} f^{(2n+2)}(\xi_2), \tag{3.37}$$

mit $\xi_1, \xi_2 \in (a,b)$.

Beweis: Für $x_j = \lambda$ hat das Interpolationspolynom $L_{n+1}(x,\lambda)$ die Form

$$L_{n+1}(x,\lambda) = \sum_{i=1, i\neq j}^{n} l_i(x) \frac{(x-\lambda)f(x_i) - (x-x_i)f(\lambda)}{x_i - \lambda} + l_j(x) f(x_j) + \frac{p_n(x)}{p_n'(\lambda)} f'(\lambda).$$

Durch Nachrechnen folgt die Darstellung (3.36) der Gewichte und Gleichung (3.34). ■

Gilt $q_n(\lambda) = 0$, so stimmt obige Quadraturformel mit der modifizierten Quadraturformel überein, die entsteht, wenn wir das hypersinguläre Integral zerlegen in

$$\begin{aligned} \text{=}\!\!\!\!\!\!\int_a^b \frac{f(x)}{(x-\lambda)^2} w(x)\,dx &= \int_a^b \frac{f(x) - f(\lambda) - f'(\lambda)(x-\lambda)}{x-\lambda} w(x)\,dx \\ &\quad + f(\lambda) \text{=}\!\!\!\!\!\!\int_a^b \frac{w(x)}{(x-\lambda)^2}\,dx + f'(\lambda) \text{=}\!\!\!\!\!\!\int_a^b \frac{w(x)}{(x-\lambda)}\,dx \end{aligned}$$

und auf das Riemann-Integral die entsprechende Gauß-Quadraturformel anwenden. Das gleiche Ergebnis erhalten wir in diesem Fall auch, wenn wir L_{n+1} durch das verallgemeinerte Lagrange-Polynom vom Grad $n+1$ ersetzen, das f in λ und in den Stützstellen x_i und f' in λ interpoliert.

3.6 Einschließung stark singulärer Integrale

Die Sätze 29 und 32 enthalten Quadraturformeln mit geeigneten Restglieddarstellungen, die es ermöglichen, stark singuläre Integrale, auf die gleiche Art und Weise wie in Abschnitt 2.6 beschrieben, einzuschließen. Besonderes Augenmerk muß jedoch auf die Gewichte der Quadraturformeln gelegt werden. Wir beschränken uns hierbei auf den Legendre-Fall $w \equiv 1$ mit Integrationsbereich $[-1,1]$. Wie bereits erwähnt nehmen die Gewichte der Hunterformeln sowohl negative als auch positive Werte an. Liegt λ nahe an einer Stützstelle treten bei der Berechnung der Quadratursumme zusätzliche numerische Instabilitäten auf. Die Verstärkungsfaktoren K definiert durch

$$K = \sum_{k=0}^{n} |w_k(\lambda)|$$

sind für $n = 8$ und $n = 9$ für verschiedene λ in Abbildung 3.2 dargestellt. Elliott und Paget empfehlen in [18] zur Berechung der Quadratursumme, den Algorithmus von Clenshaw zu verwenden, dessen Aufwand aber $O(n^3)$ beträgt. Wir schlagen einen anderen Weg ein. Durch geeignete Unterteilung des Integrationsbereiches versuchen

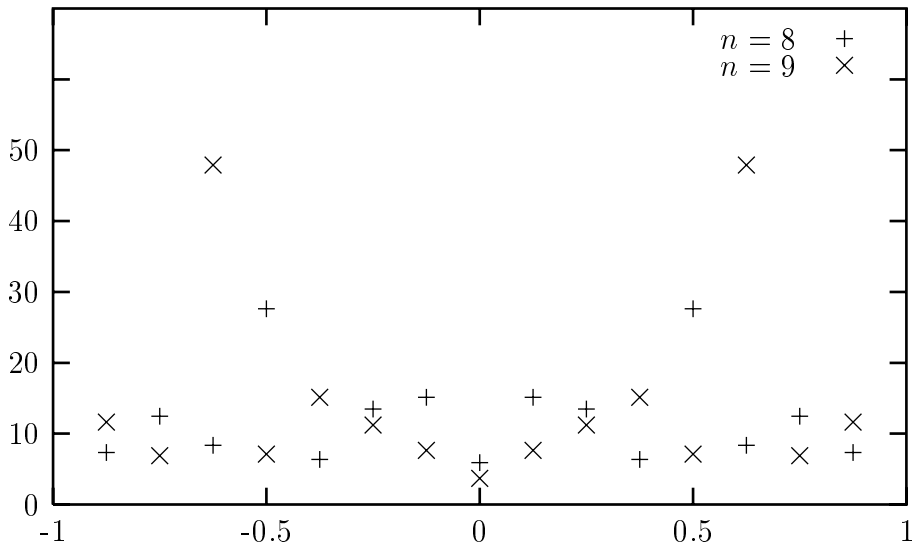
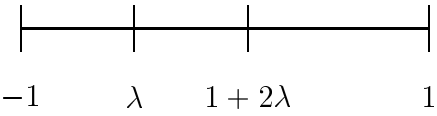


Abbildung 3.2: Verstärkungsfaktoren Hunterformeln $n = 8, 9$

wir die numerischen Probleme in den Griff zu bekommen (siehe auch [62]). Wiederum bezeichnen w_i und x_i die Stützstellen und Gewichte der Gauß-Quadraturformel (3.14). Die Legendre-Stützstellen liegen für größere n an den Rändern dichter zusammen als im Inneren des Intervalls (siehe Abbildung 3.3). In [5] werden untere und obere Abschätzungen für diese angeben. Es gilt die Faustregel, daß im Intervall $[-0.5, 0.5]$ ungefähr ein Drittel, in den beiden Randbereichen $[-1, -0.5]$ und $[0.5, 1]$ ebenfalls jeweils ein Drittel aller Stützstellen liegt. Aus diesem Grund zerlegen wir unser Ausgangsintervall $[-1, 1]$ im Falle $\lambda \leq 0$ in zwei Teilintervalle $I_1 = [-1, 1+2\lambda]$ und $I_2 = [1+2\lambda, 1]$.



Auf das linke Teilintervall wenden wir die Hunterformel (3.17) in der speziellen Form

$$⨍_{-1}^{2\lambda+1} \frac{f(x)}{x-\lambda}\,dx = \sum_{i=1}^{n} \frac{w_i}{x_i} f\left((\lambda+1)x_i + \lambda\right) + \frac{h_n(1+\lambda)^{2n+1}}{k_n^2(2n+1)!} f^{(2n+1)}(\xi) \tag{3.38}$$

an. Für ungerades n fällt die Singularität λ mit der Stützstelle x_j ($j = \frac{n+1}{2}$) zusammen, weswegen wir die Hunterformel (3.20) verwenden. Wegen der Symmetrie der Stützstellen x_i und der Gewichte w_i gilt

$$0 = ⨍_{-1}^{1} \frac{1}{x}\,dx = \sum_{i=1}^{n} w_i(0) = \sum_{i=1, i\neq j}^{n} \frac{w_i}{x_i} + w_j(0) = w_j(0).$$

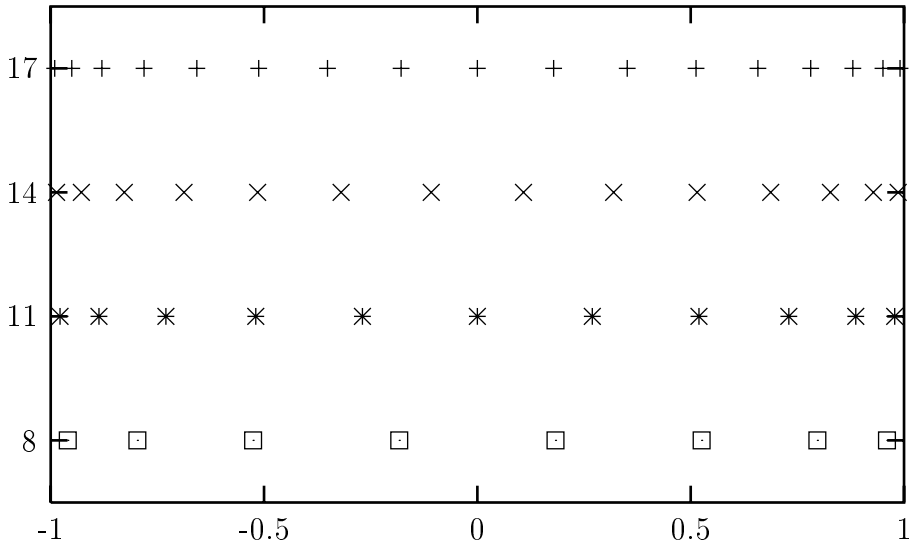


Abbildung 3.3: Legendre-Knoten für $n = 8, 11, 14, 17$

Die Hunterformel hat also die Form

$$\begin{aligned} \oint_{-1}^{1+2\lambda} \frac{f(x)}{x-\lambda}\,dx &= \sum_{i=1, i\neq j}^{n} \frac{w_i}{x_i} f\left((\lambda+1)x_i+\lambda\right) + (\lambda+1)\, w_j f'(\lambda) \\ &\quad + \frac{h_n(1+\lambda)^{2n+1}}{k_n^2(2n+1)!} f^{(2n+1)}(\xi), \quad \xi \in (-1, 1+2\lambda). \end{aligned} \tag{3.39}$$

Die Verstärkungsfaktoren K fallen vergleichsweise moderat aus. Für $\lambda = 0$ sind sie hier für verschiedene Werte von n aufgelistet. Man beachte, daß die Verstärkungsfaktoren nur für ungerade n von λ abhängig sind.

n	4	5	6	7	8	9	12	13	16	17	20
K	4.6	2.9	5.4	3.3	5.9	3.7	6.7	4.3	7.3	4.8	7.7

Für das rechte Teilintervall wenden wir die Gauß-Quadraturformeln (3.14) auf den Integranden $f(x)/(x-\lambda)$ an

$$\begin{aligned} \int_{1+2\lambda}^{1} \frac{f(x)}{x-\lambda}\,dx &= -\lambda \sum_{i=1}^{n} \frac{w_i}{-\lambda x_i + 1} f(-\lambda x_i + 1 + \lambda) \\ &\quad + \frac{d^{2n}}{dx^{2n}} \left(\frac{f(-\lambda x + 1 + \lambda)}{-\lambda x + 1} \right)(\xi), \quad \xi \in (-1, 1). \end{aligned} \tag{3.40}$$

Im Fall $\lambda > 0$ nehmen wir die Intervalle $I_1 = [-1, 2\lambda - 1]$ und $I_2 = [2\lambda - 1, 1]$ und verfahren wie oben.

Für hypersinguläre Integrale der Form (3.23) verwenden wir anstatt der Hunterformel die Quadraturformel (3.35) mit geradem n. Im Fall $\lambda \leq 0$ erhalten wir für das Integrationsintervall $I_1 = [-1, 1+2\lambda]$

$$\begin{aligned} \oint_{-1}^{1+2\lambda} \frac{f(x)}{(x-\lambda)^2}\,dx &= \frac{1}{(1+\lambda)} \sum_{i=1}^{n} \frac{w_i}{x_i^2} f((\lambda+1)x_i + \lambda) + \frac{q_n'(0)}{p_n(0)} f(\lambda) \qquad (3.41) \\ &\quad + \frac{h_n (1+\lambda)^{2n+2}}{k_n^2 (2n+2)!} f^{(2n+2)}(\xi), \quad \xi \in (-1, 1+2\lambda). \end{aligned}$$

Die dazugehörigen Verstärkungsfaktoren sind allerdings deutlich größer als die der entsprechenden Hunterformeln.

n	6	8	12	16	20	24	32
K	39	51	77	102	127	152	202

3.6.1 Einfach adaptiver Algorithmus

Mit der im letzten Abschnitt beschriebenen Zerlegung sind wir in der Lage, hypersinguläre Integrale einfacher Funktionen einzuschließen. Falls die Funktion allerdings etwas komplizierter verläuft, so ist eine einmalige Zerlegung im Allgemeinen nicht ausreichend, um eine vorgegebene Genauigkeit zu erreichen. In diesem Falle verwenden wir die Zerlegung als Startzerlegung und modifizieren den global adaptiven Algorithmus aus Abschnitt 2.6.1 für diesen Zweck. Hierzu ersetzen wir den Initialisierungsteil in Abbildung 2.10 durch die in Abbildung 3.4 beschriebene Startzerlegung.
Wird mit der Startzerlegung die gewünschte Genauigkeit nicht erreicht, so wird das Intervall I mit dem größten Fehlergliedddurchmesser weiter unterteilt. Bei dieser Unterteilung muß berücksichtigt werden, ob sich die Singularität im Intervall I oder außerhalb befindet. Liegt λ in I, dann unterteilen wir es in die drei Teilintervalle I_1, I_2 und I_3 (siehe Abbildung 3.5), ansonsten zerlegen wir es in zwei gleichgroße Teilintervalle I_1 und I_2 .
Eine Einschließung $[R^J(f)]$ des Restglieds bestimmen wir abhängig von der Ordnung und Lage der Singularität durch eine der Quadraturformeln (3.38), (3.39), (3.41) oder (3.40) transformiert auf J. Ist λ keine $p+1$-fache ($p \geq 1$) Nullstelle von f, dann gilt (siehe [62])

$$\oint_{\lambda-h}^{\lambda+h} \frac{f(x)}{(x-\lambda)^{p+1}}\,dx = \begin{cases} O(h^{-p+1}), & \text{falls } p \text{ gerade} \\ O(h^{-p}), & \text{falls } p \text{ ungerade} \end{cases}, \qquad (3.42)$$

d. h. das Integral divergiert für $h \to 0$ gegen ∞. Gleiches gilt damit aber automatisch für die Summe der Riemann-Integrale

$$\int_a^{\lambda-h} \frac{f(x)}{(x-\lambda)^{p+1}}\,dx + \int_{\lambda+h}^{b} \frac{f(x)}{(x-\lambda)^{p+1}}\,dx.$$

** Initialisierung **
$A := \emptyset$
$J_1 := [a, b]$
$R_{inf} := 0, \quad R_{sup} := 0$
$l := 2\lambda - a - b$
if $(l \neq 0)$
  if $(l < 0)$
    $J_1 := [a, 2\lambda - a], \quad J_2 := [2\lambda - a, b]$
  elsif
      $J_1 := [a, 2\lambda - b], \quad J_2 := [2\lambda - b, b]$
  fi
  Füge J_2 in A ein
  Bestimme $r := [R^{J_2}(f)]$
  $R_{inf} := \inf(r), \quad R_{sup} := \sup(r)$
fi
Füge J_1 in A ein
Bestimme $r := [R^{J_1}(f)]$
$R_{inf} := R_{inf} + \inf(r)$
$R_{sup} := R_{sup} + \sup(r)$

Abbildung 3.4: Startzerlegung

Zerlegungsschritt:

if $(\lambda \in [a, b])$
  $I_1 := [a, 0.5(a + \lambda)]$
  $I_2 := [0.5(a + \lambda), 0.5(b + \lambda)]$
  $I_3 := [0.5(b + \lambda), b]$
  return I_1, I_2, I_3
elsif $I_1 := [a, 0.5(a + b)]$
    $I_2 := [0.5(a + b), b]$
    return I_1, I_2
fi

Abbildung 3.5: Intervallzerlegung

Für $p = 1$ bedeutet dies, daß die Quadratursumme der zusammengesetzten Quadraturformel bei feiner Unterteilung um λ eine Differenz zweier großer Zahlen ist, was zu numerischen Schwierigkeiten bei der Berechnung der Quadratursumme führen kann.

3.6.2 Integrationskonstanten von Hauptwertformeln

Die Restglieder von Hauptwertformeln können genauso wie die Restglieder klassischer Quadraturformeln mit Hilfe von Peano-Kernen dargestellt werden. Zur Herleitung dieser Darstellungen benötigt man den Satz von Peano in einer etwas allgemeineren Form, die von Sard [75] stammt.

Satz 33 *(Peano/Sard) $A^s(a,b)$ bezeichne den Raum aller auf dem Intervall $[a,b]$ reellwertiger Funktionen, deren $(s-1)$-te Ableitung einer Lipschitzbedingung der Ordnung 1 genügt. Sei R ein stetiges lineares Funktional auf dem Raum $A^{s-1}(a,b)$ mit $deg(R) \geq m-1$, dann gilt für $s = 1, \ldots, m$ und $f \in A^s(a,b)$:*

$$R(f) = \int_a^b f^{(s)}(x)\, K_s(x)\, dx \tag{3.43}$$

mit

$$K_s(x) = R\left(\frac{(\cdot - x)_+^{s-1}}{(s-1)!}\right). \tag{3.44}$$

Wenden wir den vorhergehenden Satz auf die Quadraturformel Q

$$Q(f;\ \lambda) = \sum_{\nu=1}^{n} w_\nu f(x_\nu) + w_0(\lambda) f'(\lambda) \tag{3.45}$$

mit dem linearen Restfunktional R

$$R(f;\ \lambda) = \int\!\!\!\!\!\!-_{-1}^{1} \frac{f(x)}{x-\lambda}\, dx - Q(f;\ \lambda)$$

an, so erhalten wir folgenden Satz.

Satz 34 *Sei Q eine Quadraturformel mit der Darstellung (3.45) und $deg(R) = m > 0$, dann gilt für $2 \leq s \leq m+1$ und $f \in C^s[-1,1]$:*

$$R(f;\ \lambda) = \int_{-1}^{1} f^{(s)}(x) K_s(x)\, dx \tag{3.46}$$

mit

$$\begin{aligned}(s-1)!\, K_s(x) = &\sum_{i=1}^{s-1} \frac{1}{i}(1-x)^i(\lambda - x)^{s-1-i} + (\lambda - x)^{s-1} \ln\left|\frac{1-\lambda}{x-\lambda}\right| \\ &- \sum_{i=1}^{n} w_i(\lambda)(x_i - x)_+^{s-1} - w_0(\lambda)(s-1)(\lambda - x)_+^{s-2}.\end{aligned} \tag{3.47}$$

Beweis: Wegen der Stetigkeit von R gilt:

$$\begin{aligned}(s-1)!\,K_s(x) &= \fint_{-1}^{1} \frac{(t-x)_+^{s-1}}{t-\lambda}\,dt - \sum_{i=1}^{n} w_i(\lambda)(x_i-x)_+^{s-1} - w_0(\lambda)(s-1)(\lambda-x)_+^{s-2} \\ &= \fint_{x}^{1} \frac{\sum_{i=0}^{s-2}(t-x)^i(\lambda-x)^{s-2-i}(t-\lambda)+(\lambda-x)^{s-1}}{t-\lambda}\,dt \\ &\quad - \sum_{i=1}^{n} w_i(\lambda)(x_i-x)_+^{s-1} - w_0(\lambda)(s-1)(\lambda-x)_+^{s-2} \\ &= \sum_{i=1}^{s-1}\int_{x}^{1}(t-x)^{i-1}(\lambda-x)^{s-1-i}\,dt + \fint_{x}^{1}\frac{(\lambda-x)^{s-1}}{t-\lambda}\,dt \\ &\quad - \sum_{i=1}^{n} w_i(\lambda)(x_i-x)_+^{s-1} - w_0(\lambda)(s-1)(\lambda-x)_+^{s-2}.\end{aligned}$$

Daraus folgt unmittelbar die Behauptung. ■

In [16] wird die äquivalente Darstellung

$$\begin{aligned}(s-1)!\,K_s(x) &= \sum_{i=1}^{s-1}\frac{1}{i}\binom{s-1}{i}(1-\lambda)^i(\lambda-x)^{s-1-i} \\ &\quad + (\lambda-x)^{s-1}\left(\sum_{i=1}^{s-1}\frac{1}{i}+\ln\left|\frac{1-\lambda}{x-\lambda}\right|\right) - \\ &\quad \sum_{i=1}^{n} w_i(\lambda)(x_i-x)_+^{s-1} - w_0(\lambda)(s-1)(\lambda-x)_+^{s-2} \\ &= \sum_{i=1}^{s-1}\frac{1}{i}\binom{s-1}{i}(-1-\lambda)^i(\lambda-x)^{s-1-i} \\ &\quad + (\lambda-x)^{s-1}\left(\sum_{i=1}^{s-1}\frac{1}{i}+\ln\left|\frac{1-\lambda}{x-\lambda}\right|\right) \\ &\quad - (-1)^s\sum_{i=1}^{n} w_i(\lambda)(x-x_i)_+^{s-1} + (-1)^s w_0(\lambda)(s-1)(x-\lambda)_+^{s-2}\end{aligned} \tag{3.48}$$

sowie eine Darstellung für den Peano-Kern erster Ordnung K_1 angegeben. Der folgende Satz überträgt die Eigenschaften der Peano-Kerne klassischer Quadraturverfahren auf die Peano-Kerne von Hauptwertformeln. In [16] werden diese bewiesen.

Satz 35 *Die Peano-Kerne $K_s\,(s>1)$ der Quadraturformel (3.45) haben folgende Eigenschaften:*

1. $K_s(-1) = K_s(1) = 0.$
2. $K_s \in A^{(s-2)}[-1,1]$
3. *Für $w_0(\lambda) = 0$ gilt $K_s \in C^{(s-2)}[-1,1]$*

4. $K_{s+1}(x) = -\int_{-1}^{x} K_s(t)dt$

5. *Für* $-1 < x_1 < \ldots < x_n < 1$ *besitzt* K_s *höchstens* $2n+1-s$ *Vorzeichenwechsel.*

Analog zu Abschnitt 2.4 verwenden wir die Bezeichnungen c_s^{H+}, c_s^{H-} für die Integrationskonstanten der Hunterformel und c_s^H für die Peano-Konstanten

$$c_s^H := \int_{-1}^{1} |K_s^H(x)|\, dx = c_s^{H+} - c_s^{H-}. \tag{3.49}$$

Um Verwechslungen zu vermeiden, kürzen wir die Peano-Kerne der Hunterformeln mit K_s^H, die Peano-Kerne und Peano-Konstanten der zugrundeliegenden Gauß-Quadraturformel (3.14) mit K_s^G und c_s^G ab.
Zwischen den Peano-Kernen K_s^G und den Peano-Kernen der dazugehörigen Hunterformel besteht der folgende Zusammenhang (siehe [16]):

$$K_s^H(x) = \begin{cases} -(x-\lambda)^{s-1} \displaystyle\int_{-1}^{x} \frac{K_{s-1}^G(t)}{(t-\lambda)^s}\, dt, & \text{falls } -1 \le x < \lambda \\ K_{s-1}^G(\lambda)/(s-\lambda), & \text{falls } x = \lambda \\ (x-\lambda)^{s-1} \displaystyle\int_{x}^{1} \frac{K_{s-1}^G(t)}{(t-\lambda)^s}\, dt, & \text{falls } \lambda < x \le 1 \end{cases} \tag{3.50}$$

Obige Gleichung drängt die Frage auf, ob die Peano-Konstanten der Hunterformeln nicht einfach aus den Peano-Konstanten der zugrundeliegenden Quadraturformeln berechnet werden können. Für definite Kernfunktionen ist dies tatsächlich möglich, denn es gilt

$$c_{2n}^G = \frac{h_n}{k_n^2(2n)!} = (2n+1)\, c_{2n+1}^H.$$

Für $s = 2, \ldots, 2n$ gilt immerhin noch

$$c_{s+1}^H \le \frac{c_s^G}{s+1}. \tag{3.51}$$

Weitere Abschätzungen für die Peano-Konstanten der Hunterformeln werden im nachstehenden Satz gegeben. Beide Ungleichungen werden ebenfalls in [16] bewiesen.

Satz 36 *Für die Peano-Konstanten der Hunterformeln gelten die Abschätzungen*

$$c_{n+1}^H \le \frac{(n/2)^{\frac{1}{4}}}{(n+1)!(3\sqrt{3})^{n-1}} \tag{3.52}$$

und für $s > 2$, $n \ge 4$ *und* $n \ge s$

$$c_s^H \le \frac{\pi^2}{3n!}\left(\frac{3(n-s+1)!}{n(s-1)} + \frac{2(n-s)!}{s} + 4(n-s)!\ln(n)\right). \tag{3.53}$$

3.6.3 Einschließung der Integrationskonstanten

Zur Einschließung der Integrationskonstanten der Hunterformeln verwenden wir die in Abschnitt 2.4.1 beschriebene Methode. Im Fall $\lambda = 0$ vereinfacht sich die Darstellung der Peano-Kernfunktionen zu

$$\begin{aligned}
(s-1)!\,K_s(x) &= \alpha_s(x) + \beta_s(x) - \gamma_s(x) \\
\alpha_s(x) &:= \sum_{i=1}^{s-1} \frac{1}{i}\binom{s-1}{i}(-x)^{s-1-i} \\
\beta_s(x) &:= (-x)^{s-1}\left(\sum_{i=1}^{s-1}\frac{1}{i} - \ln|x|\right) \\
\gamma_s(x) &:= \sum_{i=1}^{n} w_i(0)(x_i - x)_+^{s-1} - w_0(0)(s-1)(-x)_+^{s-2}.
\end{aligned}$$

Für $\lambda = 0$ ist der Peano-Kern der Hunterformel symmetrisch, denn wegen (3.50) und $K_s^G(t) = (-1)^s K_s^G(-t)$ gilt für $x > 0$

$$\begin{aligned}
K_s^H(-x) &= -(-x)^{s-1}\int_{-1}^{x} \frac{K_{s-1}^G(t)}{t^s}\,dt \\
&= -(-x)^{s-1}\int_{x}^{1} \frac{-K_{s-1}^G(t)}{t^s}\,dt \\
&= (-1)^{s+1}K_s^H(x).
\end{aligned}$$

Das bedeutet, daß wir zur Berechnung der Integrationskonstanten die Kernfunktion nur auf dem Intervall $[0,1]$ untersuchen müssen, denn für $s = 2, \ldots, 2n$ gilt

$$c_s^{H_+} = -c_s^{H_-} = \begin{cases} 2\displaystyle\int_{K^+\cap[0,1]} K_s^H(t)\,dt, & \text{falls } s \text{ ungerade} \\ \displaystyle\int_{K^+\cap[0,1]} K_s^H(t)\,dt - \int_{K^-\cap[0,1]} K_s^H(t)\,dt, & \text{falls } s \text{ gerade} \end{cases}.$$

Die Integrationskonstanten für $s = 2n+1$ können wir mit

$$\begin{aligned}
c_{2n+1}^{H_+} &= \frac{h_n}{k_n^2(2n+1)!} \\
c_{2n+1}^{H_-} &= 0
\end{aligned}$$

direkt berechnen.
Zur Auswertung von K_s^H nutzen wir die Monotonieeigenschaften der Funktionen $\alpha_s + \beta_s$ und γ_s. Auf dem Intervall $[0,1]$ sind beide Funktionen monoton fallend, d. h. wir können den Wertebereich $W_{K_s^H}([\underline{x}, \overline{x}])$ mit $[\underline{x}, \overline{x}] \subset [0,1]$ einschließen durch

$$W_{K_s^H}([\underline{x}, \overline{x}]) \in \frac{1}{(s-1)!}[\alpha_s(\overline{x}) + \beta_s(\overline{x}) - \gamma_s(\underline{x}), \alpha_s(\underline{x}) + \beta_s(\underline{x}) - \gamma_s(\overline{x})].$$

Auf der Maschine verwenden wir

$$W_{K_s^H}([\underline{x},\overline{x}]) \quad \in \tfrac{1}{(s-1)!}[\inf(\alpha_s([\overline{x}])) \mathbin{\overline{\nabla}} \inf(\beta_s([\overline{x}])) \mathbin{\overline{\nabla}} \sup(\gamma_s([\underline{x}])),$$
$$\sup(\alpha_s([\underline{x}])) \mathbin{\overline{\Delta}} \sup(\beta_s([\underline{x}])) \mathbin{\overline{\Delta}} \inf(\gamma_s([\overline{x}]))].$$

Die Ableitungen der Kernfunktionen ($s = 3, \ldots, 2n+1$) können mit Hilfe der Gleichung

$$(s-1)!K_s^{H'} = \alpha_s'(x) + \beta_s'(x) - \gamma_s'(x) = -(s-1)(\alpha_{s-1}(x) + \beta_{s-1}(x) - \gamma_{s-1}(x))$$

berechnet werden. Gleiches gilt für die Stammfunktionen. In Abbildung 3.6 sind die

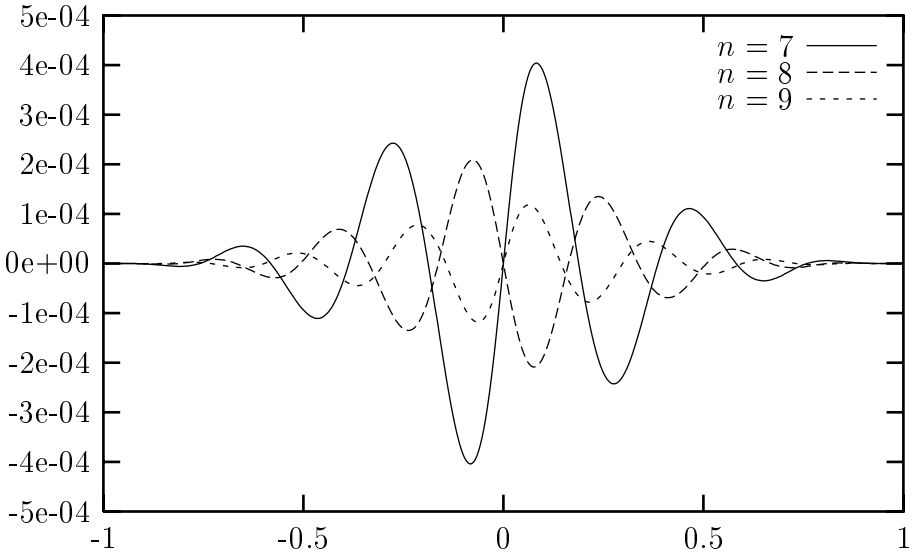


Abbildung 3.6: Kernfunktionen $6!\,K_6^H (n = 7, 8, 9)$

Kernfunktionen $6!K_6^H$ der Hunterformeln mit sieben, acht und neun Stützstellen graphisch dargestellt. Die Kernfunktionen sind punktsymmetrisch zum Ursprung und werden für größere Werte für n flacher. Außerdem verflachen sie zu den Rändern hin, was zu Problemen bei der Nullstellensuche in der Nähe des Randes führt. Da γ_s auf $[x_n, 1]$ verschwindet, $\alpha_s + \beta_s$ monton fallend und $K_s^H(1) = 0$ ist, nimmt K_s^H in $[x_n, 1]$ keine negativen Werte an, weswegen das Integral $P_{[x_n,1]}$ des positiven Anteils des Kerns auf $[x_n, 1]$ direkt mit

$$\int_{x_n}^{1} K_s^H(t)\,dt = K_{s+1}^H(x_n)$$

bestimmt werden kann. Zur Berechnung der Einschließungen der Peano-Konstanten für $n = 11$ und $s = 4, \ldots, 23$ in Tabelle 3.3 wurde die Klassenbibliothek C–XSC mit einer istaggered-Arithmetik (stagprec = 2) verwendet. Dabei wurde die Langzahlarithmetik nur für die Auswertung der Kernfunktionen (und deren Ableitungen) bei der Suche

nach Nullstellen eingesetzt. Alle anderen Berechnungen wurden mit der Intervallarithmetik von C-XSC durchgeführt. Eine Erhöhung der Präzision (stagprec > 2) bringt für $n = 11$ keine nennenswerten Verbesserung der Einschließungen, die benötigte Rechenzeit kann jedoch reduziert werden, da die Bestimmung der Nullstellen der Kernfunktion in diesem Fall weniger Schwierigkeiten bereitet. Schaubild 3.7[4] zeigt den typischen Verlauf der Integrationskonstanten $s!c_s^{H+}$. Bei festem n fallen sie zunächst mit zunehmenden s, um dann ab einem gewissen s wieder monoton zu wachsen. Bei festem s und wachsendem n fallen die Integrationskonstanten monton. Deshalb ist es sinnvoll, auch Restglieddarstellungen niedriger Ordnung ($s < 2n + 1$) zu berücksichtigen. In Tabelle 3.4 werden die Einschließungen der Peano-Konstanten mit den Abschätzungen (3.51) und (3.52) verglichen. Man sieht, daß beide Abschätzungen brauchbare Werte liefern. Abschätzung (3.51) liefert zwar deutlich genauere obere Schranken als Abschätzung (3.52), ist dafür aber auch deutlich schwerer zu berechnen. Die Abschätzung (3.53) liefert für diese Werte keine brauchbaren Ergebnisse. Ihre Bedeutung liegt auch mehr in der Gewinnung asymptotischer Aussagen über die Peano-Konstanten. So läßt sich mit ihr z. B. zeigen, daß die Peano-Konstanten der Hunterformeln die bestmögliche Größenordnung haben (siehe [16]). Für die Darstellung des Quadraturfehlers sind wir allerdings mehr an den konkreten Werten der Integrationskonstanten interessiert. Da diese einmal vorab berechnet und tabelliert werden können, ist es durchaus sinnvoll, sie genauer zu bestimmen.

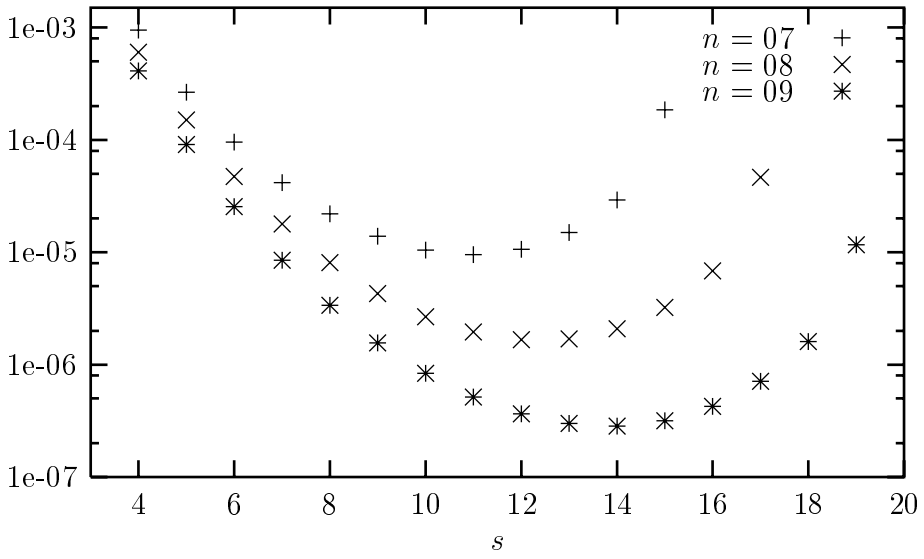


Abbildung 3.7: Integrationskonstanten $s!\, c_s^{H+}, n = 7, 8, 9$

[4]Man beachte den logarithmischen Maßstab der y-Achse

s	$s!\,[c_s^{H-}]$	$d(s!\,[c_s^{H-}])$	$s!\,[c_s^{H+}]$	$d(s!\,[c_s^{H+}])$
4	$-2.08196150^{89}_{92}E-04$	$2.0E-014$	$2.08196150^{92}_{89}E-04$	$2.0E-014$
5	$-3.83254158^{28}_{30}E-05$	$1.2E-015$	$3.832541582^{92}_{84}E-05$	$8.1E-016$
6	$-8.81578749^{6}_{7}E-06$	$8.4E-016$	$8.81578749^{70}_{61}E-06$	$8.4E-016$
7	$-2.4125048^{297}_{304}E-06$	$6.1E-016$	$2.4125048^{306}_{295}E-06$	$1.0E-015$
8	$-7.7495015^{37}_{44}E-07$	$6.2E-016$	$7.7495015^{44}_{37}E-07$	$6.2E-016$
9	$-2.8708623^{77}_{86}E-07$	$7.4E-016$	$2.8708623^{84}_{79}E-07$	$4.1E-016$
10	$-1.21516862^{3}_{8}E-07$	$4.4E-016$	$1.21516862^{8}_{3}E-07$	$4.4E-016$
11	$-5.829884^{88}_{92}E-08$	$3.2E-016$	$5.829884^{93}_{87}E-08$	$5.2E-016$
12	$-3.1542663^{4}_{9}E-08$	$4.6E-016$	$3.1542663^{9}_{4}E-08$	$4.6E-016$
13	$-1.918292^{5}_{7}E-08$	$1.1E-015$	$1.9182926^{7}_{1}E-08$	$5.7E-016$
14	$-1.3095713^{0}_{5}E-08$	$4.0E-016$	$1.3095713^{5}_{0}E-08$	$4.0E-016$
15	$-1.004051^{82}_{90}E-08$	$7.1E-016$	$1.0040518^{8}_{3}E-08$	$4.0E-016$
16	$-8.66860^{28}_{38}E-09$	$8.6E-016$	$8.66860^{38}_{28}E-09$	$8.6E-016$
17	$-8.472359^{4}_{9}E-09$	$4.3E-016$	$8.472359^{7}_{5}E-09$	$1.3E-016$
18	$-9.45827^{24}_{33}E-09$	$7.9E-016$	$9.45827^{33}_{24}E-09$	$7.9E-016$
19	$-1.2236928^{2}_{4}E-08$	$1.4E-016$	$1.2236928^{5}_{1}E-08$	$3.8E-016$
20	$-1.87916^{899}_{903}E-08$	$2.7E-016$	$1.87916^{903}_{899}E-08$	$2.7E-016$
21	$-3.5700401^{4}_{8}E-08$	$3.4E-016$	$3.57004015^{8}_{4}E-08$	$2.2E-017$
22	$-9.071760^{09}_{13}E-08$	$2.6E-016$	$9.071760^{13}_{09}E-08$	$2.6E-016$
23	$0.0E-00$	$0.0E-000$	$7.3291186^{6}_{2}E-07$	$3.0E-015$

Tabelle 3.3: Integrationskonstanten $s!\,c_s^{H+}$ und $s!\,c_s^{H-}$ für $n = 11$

n	$(n+1)!\,[c_{n+1}^{H+}]$	$n!\,[c_n^{G+}]$	$\frac{\left(\frac{n}{2}\right)^{0.25}}{2(3\sqrt{3})^{n-1}}$
7	$2.201821706^{53}_{47}E-05$	$2.417372631^{83}_{76}E-05$	$3.47E-05$
8	$4.294398856^{8}_{1}E-06$	$4.69148100^{92}_{84}E-06$	$6.91E-06$
9	$8.3609940^{73}_{68}E-07$	$9.0898058^{24}_{18}E-07$	$1.37E-06$
10	$1.6249991^{6}_{4}E-07$	$1.75900490^{8}_{0}E-07$	$2.71E-07$
11	$3.1542663^{9}_{4}E-08$	$3.400823^{63}_{57}E-08$	$5.34E-08$
12	$6.11611^{54}_{48}E-09$	$6.57044^{76}_{69}E-09$	$1.05E-08$
13	$1.18488^{5}_{3}E-09$	$1.26871^{8}_{5}E-09$	$2.06E-09$
14	$2.2938^{4}_{0}E-10$	$2.4487^{5}_{1}E-10$	$4.04E-10$

Tabelle 3.4: Vergleich der Abschätzungen (3.51) und (3.52)

3.6.4 Doppelt adaptiver Algorithmus

In Abschnitt 3.6.1 haben wir einen einfach adaptiven Algorithmus zur Bestimmung von Cauchy-Hauptwerten vorgestellt. Dabei haben wir für die Einschließungen der Restglieder $R^J(f)$

$$R^J(f) = \fint_a^b \frac{f(x)}{x-\lambda}\,dx - Q^H(f\left(\frac{b-a}{2}x+\frac{a+b}{2}\right))$$

jeweils die Peanokerne höchster Ordnung verwendet. Wie wir in den Abschnitten 2.4.1 und 3.6.3 schon erwähnt haben, sind die Integrationskonstanten höchster Ordnung jedoch nicht minimal. Deswegen erweitern wir den einfach adaptiven Algorithmus, indem wir aus der Menge der verfügbaren Restgliedeinschließungen $\{[r_{m,n}]|(m,n)\in M^H\}$

$$[r_{m,n}] = \left(\frac{b-a}{2}\right)^m ([c_m^{H+}]\,f^{(m)}([a,b]) + [c_m^{H-}]\,f^{(m)}([a,b])) \tag{3.54}$$

eine geeignete auszuwählen. Dazu gehen wir, wie in Abbildung 2.11 beschrieben, vor. Liegt λ außerhalb des Intervalls $[a,b]$, dann bezeichnet $R^J(f)$ das Restglied der Legendre-Formel

$$\begin{aligned} R^J(f) &= \int_a^b \frac{f(x)}{x-\lambda}\,dx - Q(g\left(\frac{b-a}{2}x+\frac{a+b}{2}\right)) \\ g(x) &:= \frac{f(x)}{x-\lambda} \end{aligned}$$

** Initialisierung **
Bilde Startzerlegung gemäß Abbildung 3.4

while $(\#A < k_{max} \wedge (R_{sup} - R_{inf}) > \epsilon_{abs})$ **do**
  Suche Intervall J aus A mit $d([R^J(f)])$ maximal
  Zerlege J in k Teilintervalle $J_1, \ldots, J_k$ gemäß Abbildung 3.5
  Entferne J aus A
  $R_{inf} := R_{inf} - \inf([R^J(f)])$
  $R_{sup} := R_{sup} - \sup([R^J(f)])$
  for $i := 1$ **to** k **do**
    Füge J_i in A ein
    Bestimme $r := [R^{J_i}(f)]$ gemäß Abbildung 2.11
    aus den Formeln (3.54) bzw. (3.55)
    $R_{inf} := R_{inf} + \inf(r)$
    $R_{sup} := R_{sup} + \sup(r)$
  od
od

** Bestimme Quadratursumme **
$[Q] := 0$
for $J \in A$ **do**
  $[Q] := [Q] + [Q^J(f)]$
od

Ausgabe $[I] := [Q] + [R_{inf}, R_{sup}]$

Abbildung 3.8: Erweiterter Algorithmus (DAA)

und die Menge der verfügbaren Restgliedeinschließungen ist $\{[r_{m,n}] | (m,n) \in M^G\}$ mit

$$[r_{m,n}] = \left(\frac{b-a}{2}\right)^{m+1} ([c_m^{G+}]\, g^{(m)}([a,b]) + [c_m^{G-}]\, g^{(m)}([a,b])). \tag{3.55}$$

In Abbildung 3.8 ist der erweiterte Algorithmus zusammengefasst. Bei vorgegebenem absoluten Restglieddurchmesser ϵ_{abs}, wird eine Einschließung $[I] = [Q] + [R]$ des gesuchten Integrals mit $d([R]) < \epsilon_{abs}$ berechnet. Zum Aufaddieren der Quadratursummen Q^J muß für jedes aktive Intervall aus A bekannt sein, welche Stützstellenzahl für die Restgliedeinschließung auf J verwendet worden ist. Liegt λ im Intervall J wird die entsprechende Hunterformel auf f, ansonsten die entsprechende Gauß-Legendre-Quadraturformel auf $f/(x-\lambda)$ angewandt. Um ein unnötig häufiges Überprüfen der Bedingung $\lambda \in J$ zu verhindern, ist es sinnvoll, diese Information ebenfalls mit dem

Intervall zusammen abzuspeichern. Eine Datenstruktur hierfür werden wir im Kapitel 5 vorstellen.

Stark singuläre Integrale der Form

$$I(f;\,\lambda) = \oint_a^b \frac{f(x)}{(x-\lambda)^2}\,dx$$

können ebenfalls mit dem Algorithmus berechnet werden. Eine Einschließung des Restglieds auf einem Intervall $[a,b]$, das λ enthält, berechnen wir mittels

$$[r_{2n+2,n}] := \left(\frac{b-a}{2}\right)^{2n+1} \frac{c_{2n}^{G+}}{(2n+2)(2n+1)} f^{(2n+2)}([a,b]), \tag{3.56}$$

für Teilintervalle $[a,b]$, die λ nicht enthalten, verwenden wir

$$[r_{m,n}] := \left(\frac{b-a}{2}\right)^{m+1} ([c_m^{G+}]\,g^{(m)}([a,b]) + [c_m^{G-}]\,g^{(m)}([a,b]) \tag{3.57}$$

mit $g(x) = f(x)/(x-\lambda)^2$.

3.6.5 Numerische Beispiele

In diesem Abschnitt untersuchen wir die in den Abschnitt 3.6.1 und 3.6.4 vorgestellten Algorithmen. Für den einfach adaptiven Algorithmus (kurz EAA) wurden die Gauß-Legendre-Quadraturformel mit acht Stützstellen und die dazugehörige Hunterformel verwendet. Für den verbesserten doppelt adaptiven Algorithmus (kurz DAA) wurden folgende Quadraturformeln bzw. Fehlerdarstellungen benutzt

$$\begin{aligned} M^G &= \{7,10,13,16\} \times \{8,11,14,17\} \\ M^H &= \{8,11,14,17\} \times \{8,11,14,17\}, \end{aligned}$$

d. h. die Quadraturformeln haben mindestens $n_{min} = 8$ und höchstens $n_{max} = 17$ Stützstellen und die Ordnungen der zu berechnenden Taylorkoeffizienten liegen zwischen 7 und 16 bei den Legendre-Formeln bzw. zwischen 8 und 17 bei den Hunterformeln. Die Wahl $m' = 3$ garantiert, daß $(m+m',n)$ in M^G (bzw. M^H) liegt, falls $(m,n) \in M^G$ (bzw. M^H) und $m+m' \leq 16$ (bzw. $m+m' \leq 17$) gilt. n' erhält ebenfalls den Wert drei.

Funktion mit hebbarer Singularität

Im ersten Beispiel betrachten wir das eigentliche Riemann-Integral

$$I(f;\,0) = \int_{-x}^{x} \frac{\sin t}{t}\,dt,$$

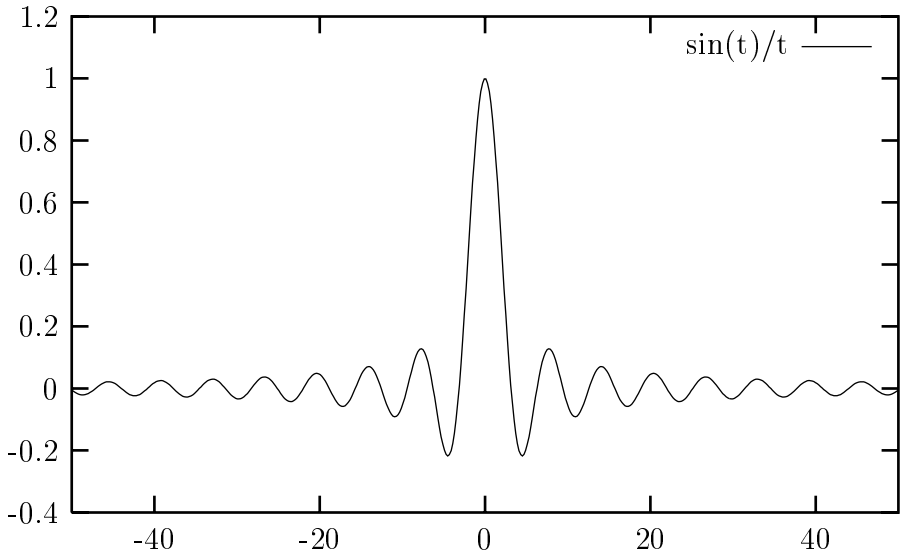


Abbildung 3.9: $\sin(t)/t$ auf $[-50, 50]$

das nicht durch elementare Funktionen dargestellt werden kann. Für $x \to \infty$ konvergiert $I(f;\ 0)$ gegen π. Das Integral kann problemlos mit den oben beschriebenen Algorithmen[5] auch für große Integrationsbereiche berechnet werden. Für $x = 1000$, d. h. für das Integral $\int_{-1000}^{1000} \frac{\sin t}{t}\, dt$, sind die Werte unten tabelliert. Der erweiterte Algorithmus DAA kommt in allen Fällen mit deutlich weniger Funktionsauswertungen ($\#f$) aus. Betrachten wir die Anzahl der benötigten Teilintervalle ($\#R$), so fällt der Vergleich noch deutlicher zu Gunsten des doppelt adaptiven Algorithmus aus.

EAA					
ϵ_{abs}	$[I(f;\ 0)]$	$d([I(f;\ 0)])$	$\#R$	$\#f$	t
$1.0E-02$	$3.1\overset{45}{36}E-00$	$7.42E-03$	130	1040	71
$1.0E-04$	$3.140\overset{6}{4}E-00$	$9.79E-05$	244	1952	134
$1.0E-06$	$3.140466\overset{4}{1}E-00$	$1.81E-07$	259	2072	143
$1.0E-08$	$3.1404662\overset{5}{3}E-00$	$9.91E-09$	384	3072	212
$1.0E-10$	$3.14046624\overset{9}{0}E-00$	$7.99E-11$	516	4128	288
$1.0E-12$	$3.140466243\overset{9}{6}E-00$	$1.22E-12$	527	4216	292

[5]Der Algorithmus aus Abschnitt 2.6.1 ist hierzu nicht geeignet, da die erforderlichen Taylokoeffizienten nicht mit den Formeln aus Abschnitt 1.4 berechnet werden können.

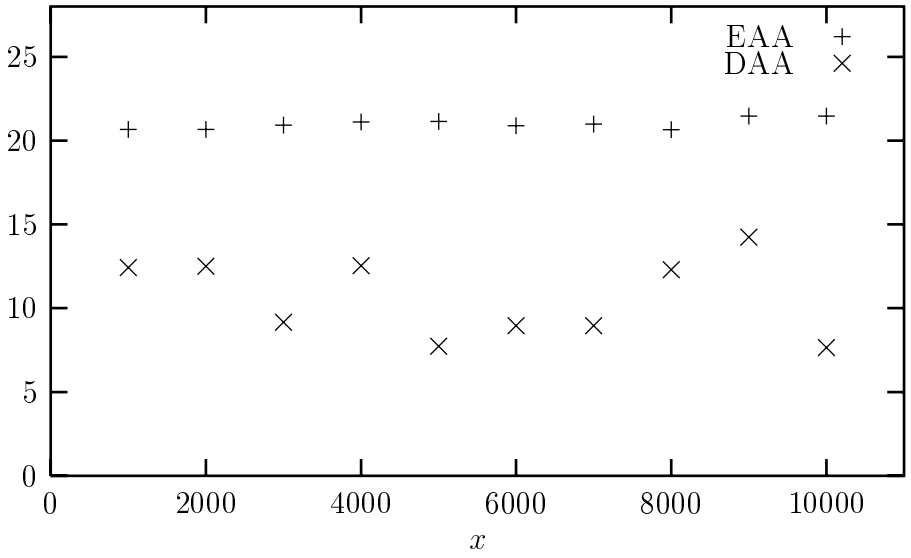


Abbildung 3.10: Quotient t_R/t_Q für EAA und DAA

DAA					
ϵ_{abs}	$[I(f;\,0)]$	$d([I(f;\,0)])$	$\#R$	$\#f$	t
$1.0E-02$	$3.1\overset{6}{3}E-00$	$9.39E-03$	65	955	22
$1.0E-04$	$3.140\overset{6}{4}E-00$	$9.62E-05$	85	1232	35
$1.0E-06$	$3.140466\overset{5}{0}E-00$	$3.82E-07$	129	1833	58
$1.0E-08$	$3.1404662\overset{5}{3}E-00$	$9.77E-09$	134	2236	66
$1.0E-10$	$3.14046624\overset{40}{38}E-00$	$9.97E-11$	194	2605	105
$1.0E-12$	$3.14046624393\overset{9}{7}E-00$	$9.21E-13$	257	3649	147

Abbildung 3.10 gibt die Quotienten t_R/t_Q von EAA und DAA in Abhängigkeit vom Integrationsbereich für festes $\epsilon_{abs} = 1E-10$ an. Dabei bezeichnet t_R die benötigte Rechenzeit zur Bestimmung der Zerlegung und des Verfahrenfehlers und t_Q die zur Berechnung der Quadratursumme benötigte Zeit. Es zeigt sich, daß der Aufwand zur Bestimmung der Zerlegung und des Restglieds beim einfach adaptiven Algorithmus mehr als 95 Prozent und beim doppelt adaptiven Algorithmus über 85 Prozent der Gesamtzeit beansprucht. Da der einfach adaptive Algorithmus im Schnitt mehr als doppelt soviele Teilintervalle bei der Berechnung der Einschließungen erzeugt, und die Berechnung eines einzelnen Restglieds auf einem Teilintervall mindestens genausoviel Zeit wie beim DAA beansprucht, können wir davon ausgehen, daß der Quotient $Q_t = t^{EAA}/t^{DAA}$ aus den Laufzeiten von EAA und DAA in erster Linie vom Verhältnis

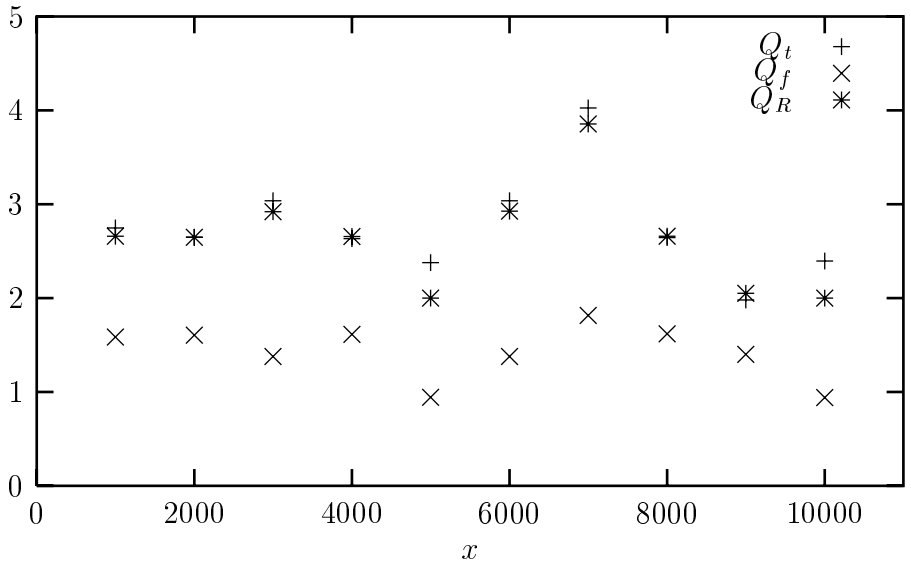


Abbildung 3.11: Vergleich Aufwand EAA und DAA

der benötigten Teilintervalle bestimmt wird. Ein Vergleich der benötigten Rechenzeit in Abbildung 3.11 von EAA und DAA bestätigt dies auch. Der Quotient $Q_t = t^{EAA}/t^{DAA}$ stimmt im wesentlichen mit dem Quotienten $Q_R = \#R^{EAA}/\#R^{DAA}$ der benötigten Teilintervalle überein. Daß Q_t für manche Werte von x (z. B. $x = 5000$) größer als Q_R und $Q_f = \#f^{EAA}/\#f^{DAA}$ ausfällt, läßt sich damit erklären, daß die Berechnung einzelner Restglieder beim DAA weniger Zeit beansprucht als beim EAA, da beim einfach adaptiven Algorithmus nur Taylorkoeffizienten der Ordnung 16 (Gauß-Legendre) bzw. 17 (Hunterformel), beim DAA hingegen auch Taylorkoeffizienten niedrigerer Ordnung auftreten können.

Funktion mit komplexer Polstelle

Die Funktion $f(x) = ((x-\alpha)^2 + \rho^2)^{-1}$ hat die komplexen Polstellen $x = \alpha \pm i\rho$. Auf $\mathbb{R}$ nimmt f in (α, ρ^{-2}) ein globales Maximum an. Der Cauchy-Hauptwert

$$I(f;\ \lambda) = \fint_a^b \frac{1}{((x-\alpha)^2 + \rho^2)(x-\lambda)}\, dx$$

kann direkt mit nachstehender Formel berechnet werden

$$I_0(f;\ \lambda) = \frac{1}{(\lambda-\alpha)^2 + \rho^2}\left[\frac{\alpha-\lambda}{\rho}\arctan\left(\frac{x-\alpha}{\rho}\right) + \ln(|\lambda - x|) - \frac{1}{2}\ln\left(\rho^2 + (x-\alpha)^2\right)\right]_a^b$$

Für $\lambda = 0.25, \alpha = 0.75, \rho = 0.01$, und $[a,b] = [0,1]$ sind die Ergebnisse für verschiedene Vorgaben an den Durchmesser ϵ_{abs} der Restgliedeinschließung unten zusammengefasst.

Der Algorithmus DAA benötigt wiederum deutlich weniger Teilintervalle und auch die Anzahl der Funktionsauswertungen ist im Vergleich zum EAA etwas geringer. Insgesamt fallen die Unterschiede bei diesem Beispiel nicht ganz so drastisch aus wie bei dem vorhergehenden. In Abbildung 3.12 sind die entstehenden Zerlegungen des Integrationsintervalls in einzelne Teilintervalle (EAA oben, DAA unten) skizziert. Man erkennt deutlich, daß die Teilintervalle sich in erster Linie um die Extremalstelle $x = \alpha$ häufen, die Berechnung des singulären Teils ist dagegen unproblematisch. Der doppelt adaptive Algorithmus kommt sowohl bei der singulären Stelle $x = \lambda$ als auch bei der lokalen Maximalstelle $x = \alpha$ mit breiteren und damit insgesamt mit weniger Teilintervallen als der einfach adaptive aus.

EAA				
ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$
$1.0E-02$	$6.261^{96}_{88}E+02$	$7.16E-03$	23	184
$1.0E-04$	$6.26191^{91}_{82}E+02$	$8.50E-05$	29	232
$1.0E-06$	$6.26191864^{6}_{0}E+02$	$5.30E-07$	36	288
$1.0E-08$	$6.261918642^{82}_{73}E+02$	$8.25E-09$	42	336
$1.0E-10$	$6.26191864279^{4}_{2}E+02$	$9.01E-11$	56	448
$1.0E-12$	$6.261918642793^{35}_{28}E+02$	$6.14E-12$	69	552
$\lambda = 0.25, \alpha = 0.75, \rho = 0.01, a = 0, b = 1$				

DAA				
ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$
$1.0E-02$	$6.26^{20}_{18}E+02$	$8.90E-03$	14	172
$1.0E-04$	$6.26191^{91}_{82}E+02$	$8.72E-05$	18	207
$1.0E-06$	$6.26191864^{5}_{1}E+02$	$2.95E-07$	23	265
$1.0E-08$	$6.261918642^{83}_{75}E+02$	$6.75E-09$	25	344
$1.0E-10$	$6.26191864279^{4}_{2}E+02$	$8.92E-11$	31	419
$1.0E-12$	$6.261918642793^{35}_{28}E+02$	$5.34E-12$	38	520
$\lambda = 0.25, \alpha = 0.75, \rho = 0.01, a = 0, b = 1$				

Setzen wir $\alpha = \lambda = 0.25$, ergibt sich ein ähnliches Bild. Der doppelt adaptive Algorithmus ist in der Anzahl der erforderlichen Teilintervalle dem einfach adaptiven klar überlegen. Beide Verfahren benötigen jetzt etwas mehr Teilintervalle und Funktionsauswertungen als zuvor. Dies liegt in erster Linie daran, daß zur Berechnung von

hinreichend engen Einschließungen der Restglieder in der Nähe von $x = \alpha$ wegen der komplizierteren Form des Integranden engere Teilintervalle benötigt werden (siehe Abbildung 3.13). Das Teilintervall, in dem λ liegt, hat dabei einen wesentlich größeren Durchmesser als die benachbarten Teilintervalle. Die Ergebnisse im Detail finden sich in den folgenden Tabellen.

EAA				
ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$
$1.0E-02$	$7.10^{52}_{44}E+00$	$6.75E-04$	26	208
$1.0E-04$	$7.104^{9}_{7}E+00$	$9.82E-05$	33	264
$1.0E-06$	$7.10479^{71}_{68}E+00$	$1.44E-07$	42	336
$1.0E-08$	$7.1047969^{4}_{2}E+00$	$9.38E-09$	50	400
$1.0E-10$	$7.1047969 32^{9}_{2}E+00$	$6.13E-10$	63	504
$1.0E-12$	$7.10479693^{4}_{2}E+00$	$9.52E-10$	81	648
$\lambda = 0.25, \alpha = 0.25, \rho = 0.01, a = 0, b = 1$				

DAA				
ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$
$1.0E-02$	$7.10^{9}_{1}E+00$	$6.90E-03$	14	205
$1.0E-04$	$7.104^{9}_{7}E+00$	$9.76E-05$	21	249
$1.0E-06$	$7.10479^{73}_{66}E+00$	$5.65E-07$	26	322
$1.0E-08$	$7.1047969^{36}_{29}E+00$	$6.84E-09$	26	388
$1.0E-10$	$7.1047969 32^{9}_{3}E+00$	$5.33E-10$	34	482
$1.0E-12$	$7.1047969 32^{9}_{3}E+00$	$4.78E-10$	46	611
$\lambda = 0.25, \alpha = 0.25, \rho = 0.01, a = 0, b = 1$				

Nun versehen wir die Funktion f mit einer doppelten Polstelle, d. h. wir betrachten das stark singuläre (hypersinguläre) Integral

$$I(f;\ \lambda) = \oint\!\!\!\!\!\!= \int_a^b \frac{1}{((x-\alpha)^2+\rho^2)(x-\lambda)^2}\,dx.$$

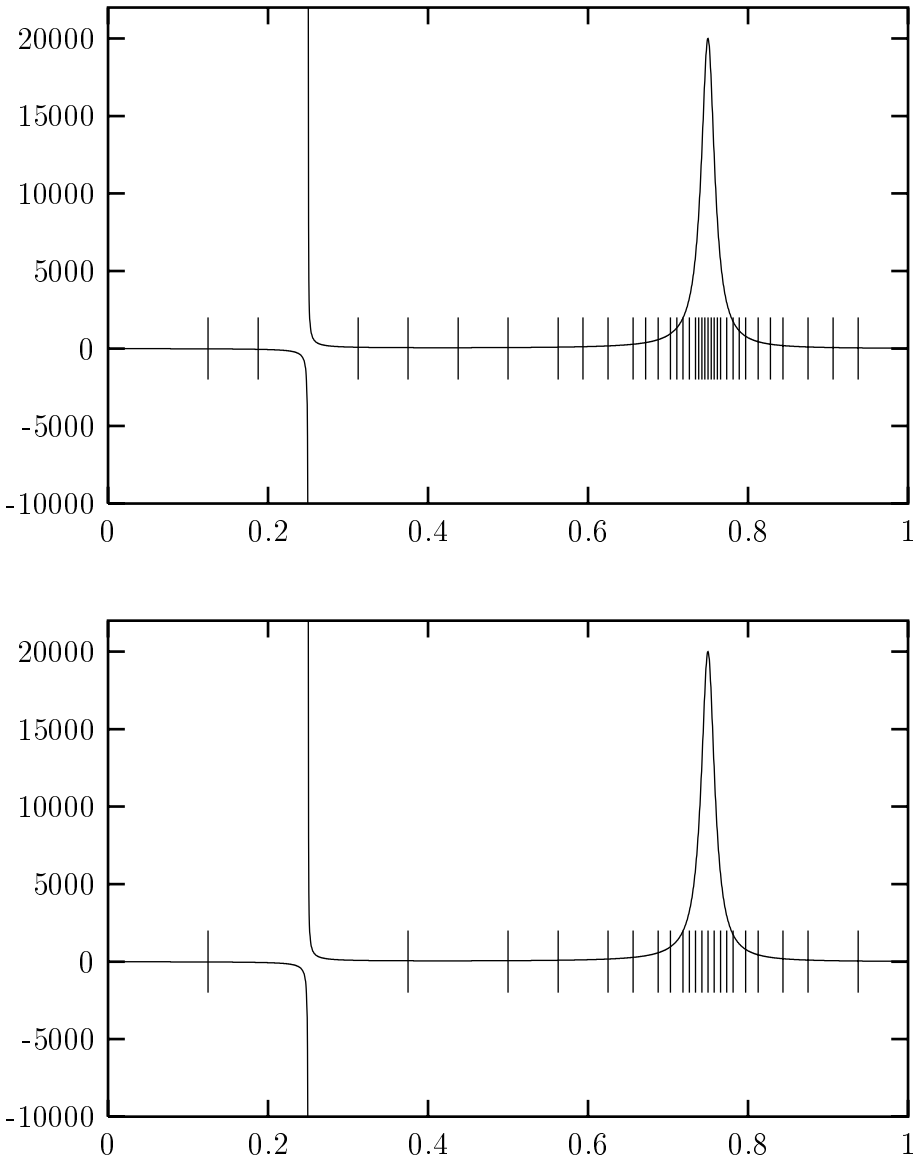


Abbildung 3.12: Zerlegung EAA (oben) und DAA ($\alpha = 0.75, \varepsilon_{abs} = 10^{-6}$)

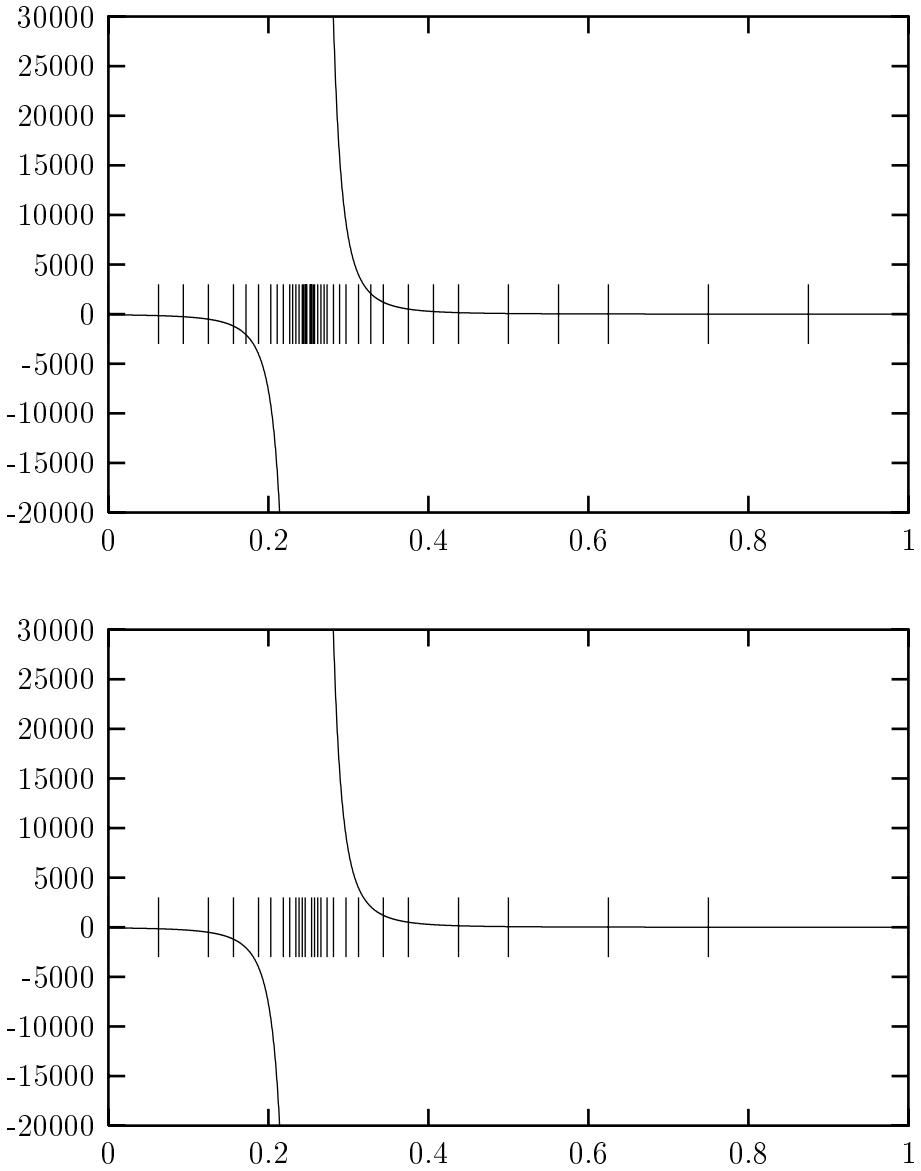


Abbildung 3.13: Zerlegung EAA (oben) und DAA ($\alpha = 0.25, \varepsilon_{abs} = 10^{-6}$)

Das Integral kann in geschlossener Form dargestellt werden, es gilt nämlich

$$I(f;\ \lambda) = \left[\left((\lambda-\alpha)\ln(\rho^2+(x-\alpha)^2)+2(\alpha-\lambda)\ln(|x-\lambda|)+\frac{(\alpha-\lambda)^2-\rho^2}{\rho}\right.\right.$$
$$\left.\left.\arctan\left(\frac{x-\alpha}{\rho}\right)\right)\frac{1}{(\rho^2+(\alpha-\lambda)^2)^2}+\frac{1}{((\alpha-\lambda)^2+\rho^2)(\lambda-x)}\right]_a^b.$$

Zur Berechnung der Einschließung verwenden wir beim DAA die Restglieddarstellungen $M^H = \{(18,8),(30,14)\}\}$, beim EAA die Formel (3.37) mit $n = 8$. Wegen der komplizierteren Form des Integranden benötigen beide Verfahren mehr Teilintervalle als bei der Berechnung des entsprechenden Cauchy-Haupwertes. Die Ergebnisse können den beiden folgenden Tabellen entnommen werden.

Doppelte Polstelle EAA					
ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$	t
$1.0E-02$	$1.2476\overset{4}{2}E+03$	$1.30E-03$	29	232	13
$1.0E-04$	$1.247622\overset{6}{4}E+03$	$8.67E-05$	37	296	16
$1.0E-06$	$1.247622505\overset{8}{2}E+03$	$5.59E-07$	44	352	18
$1.0E-08$	$1.2476225055\overset{17}{08}E+03$	$7.94E-09$	52	416	23
$1.0E-10$	$1.24762250551\overset{30}{27}E+03$	$1.14E-10$	74	592	34
$1.0E-12$	$1.2476225055128\overset{6}{3}E+03$	$2.52E-11$	92	736	40
$\lambda=0.25, \alpha=0.75, \rho=0.01, a=0, b=1$					

Doppelte Polstelle DAA					
ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$	t
$1.0E-02$	$1.2476\overset{28}{17}E+03$	$9.89E-03$	17	232	5
$1.0E-04$	$1.247622\overset{54}{47}E+03$	$6.04E-05$	26	271	8
$1.0E-06$	$1.247622505\overset{9}{2}E+03$	$5.87E-07$	29	337	10
$1.0E-08$	$1.2476225055\overset{17}{09}E+03$	$7.49E-09$	32	424	12
$1.0E-10$	$1.24762250551\overset{30}{27}E+03$	$1.18E-10$	50	592	19
$1.0E-12$	$1.2476225055128\overset{7}{2}E+03$	$3.61E-11$	57	720	23
$\lambda=0.25, \alpha=0.75, \rho=0.01, a=0, b=1$					

Stark oszillierende Funktion

Im letzten Beispiel betrachten wir die stark oszillierende Funktion $f(x) = e^{x^2}\sin(e^{x^2})$. Die Funktion schwingt mit zunehmendem x immer stärker, die Amplituden und Frequenzen wachsen rapide an. Die Ergebnisse der Berechnung der Cauchy-Hauptwerte

$$I(f;\,\lambda) = \fint_1^{2.5} \frac{e^{x^2}\sin(e^{x^2})}{x-\lambda}\,dx$$

für verschiedene Werte von λ sind unten aufgelistet.

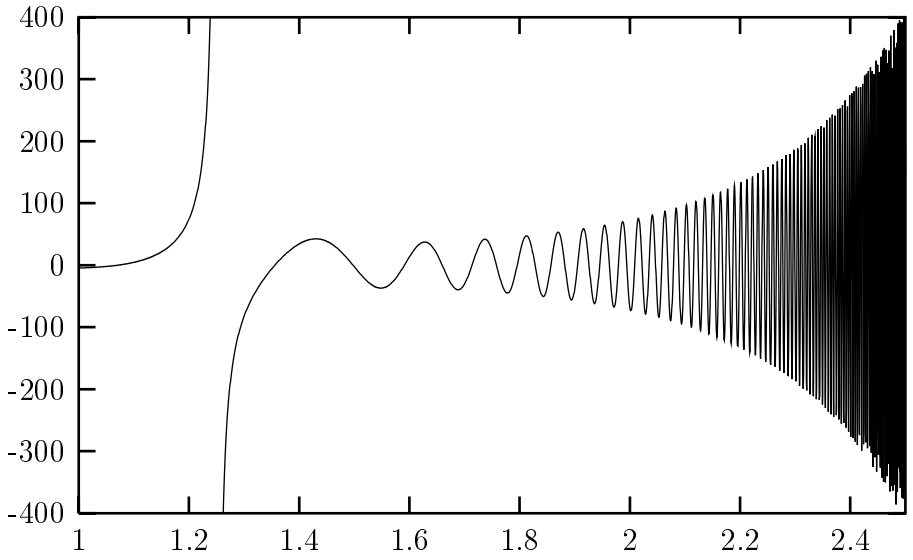


Abbildung 3.14: $I(f;\,1.25)$

EAA				
λ	$[I(f;\,\lambda)]$	$d([I(f;\,\lambda)])$	$\#R$	$\#f$
1.25	$2.687084785\frac{8}{5}E-00$	$1.87E-10$	159	1272
1.50	$-2.863181328\frac{08}{11}E+01$	$2.15E-10$	164	1312
1.75	$-5.4183857773\frac{2}{5}E+01$	$2.70E-10$	176	1408
2.00	$-6.275171576\frac{77}{82}E+01$	$4.55E-10$	177	1416
2.25	$3.0753958666\frac{3}{0}E+02$	$1.82E-09$	191	1528
$\epsilon_{abs} = 1e-10, a = 1, b = 2.5$				

DAA				
λ	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#f$
1.25	$2.687084785^{8}_{5}E+00$	$1.75E-10$	64	983
1.50	$-2.863181328^{08}_{11}E+01$	$1.93E-10$	67	1007
1.75	$-5.4183857773^{2}_{5}E+01$	$2.54E-10$	75	1110
2.00	$-6.275171576^{77}_{82}E+01$	$4.27E-10$	72	1104
2.25	$3.0753958666^{3}_{0}E+02$	$1.75E-09$	75	1140
$\epsilon_{abs} = 1e-10, a = 1, b = 2.5$				

In den Abbildungen 3.15, 3.16 und 3.17 werden die Ergebnisse noch einmal grafisch verdeutlicht. Beim DAA werden bei allen Werten von λ weniger Funktionsauswertungen benötigt (Abbildung 3.15). Der Unterschied bei der Anzahl der aktiven Teilintervalle der Zerlegungen, die das Genauigkeitkriterium erfüllen, ist noch gravierender (Abbildung 3.16). Am deutlichsten wird die Überlegenheit des doppelt adaptiven Algorithmus, wenn wir das Verhältnis t^{EAA}/t^{DAA} betrachten (Abbildung 3.17). Der einfach adaptive Algorithmus benötigt für alle untersuchten λ-Werte etwa drei mal soviel Zeit wie der doppelt adaptive.

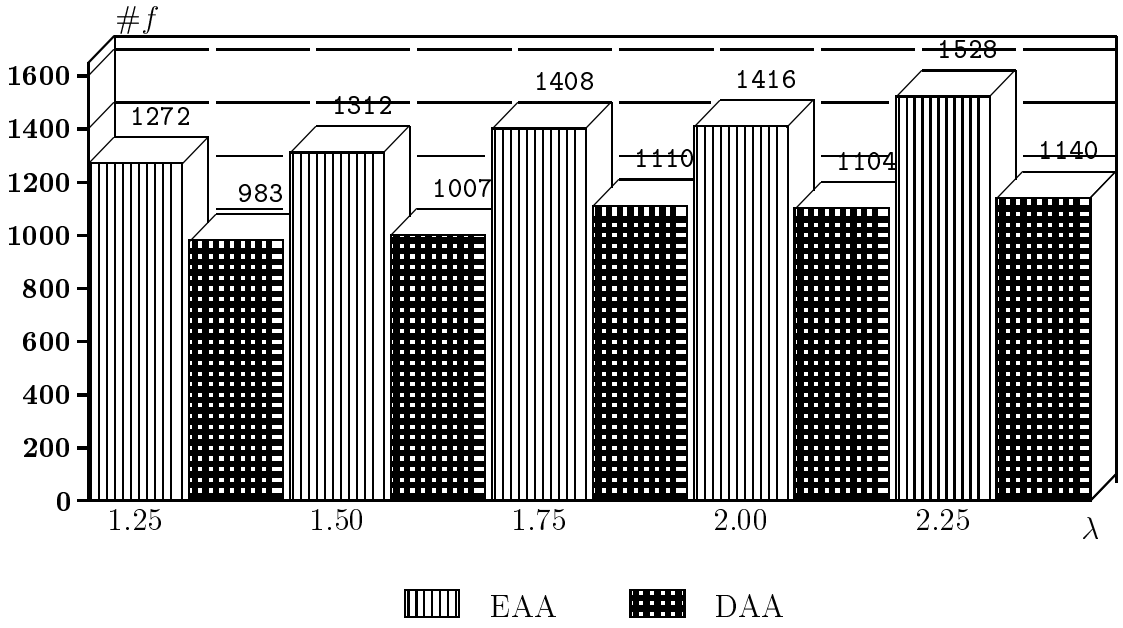


Abbildung 3.15: Vergleich $\#f$ von EAA und DAA

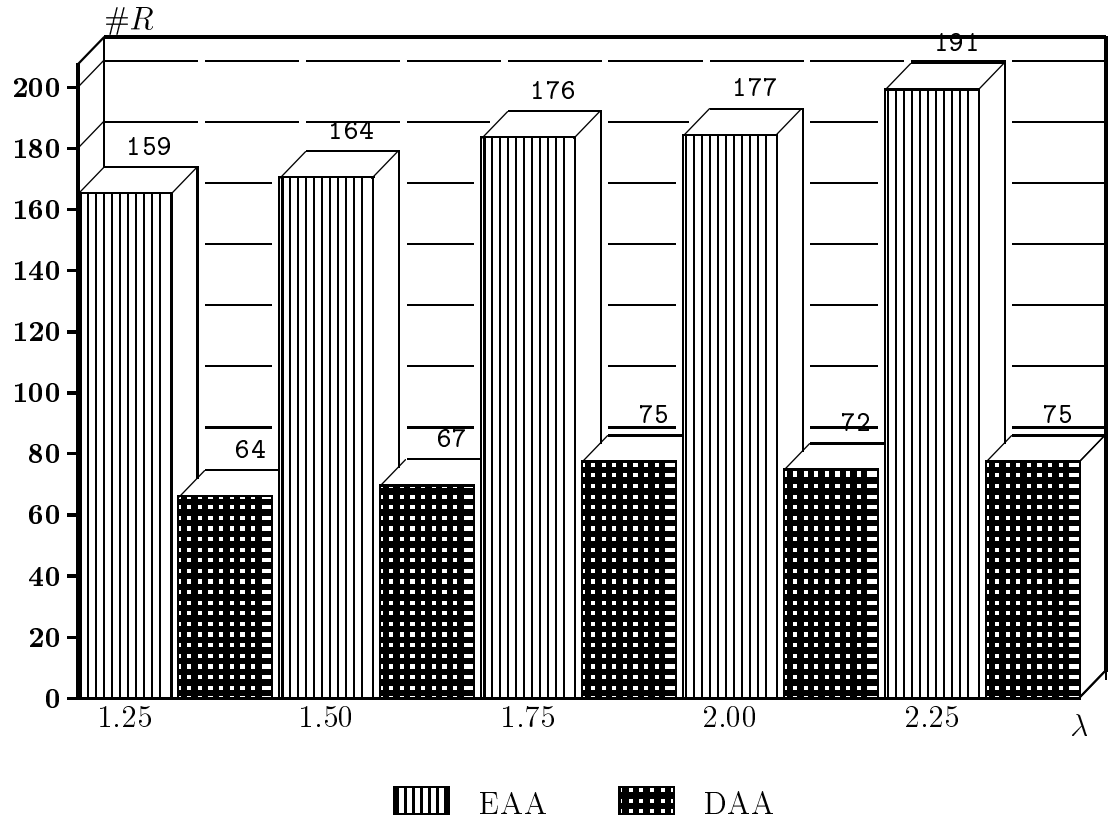


Abbildung 3.16: Vergleich $\#R$ von EAA und DAA

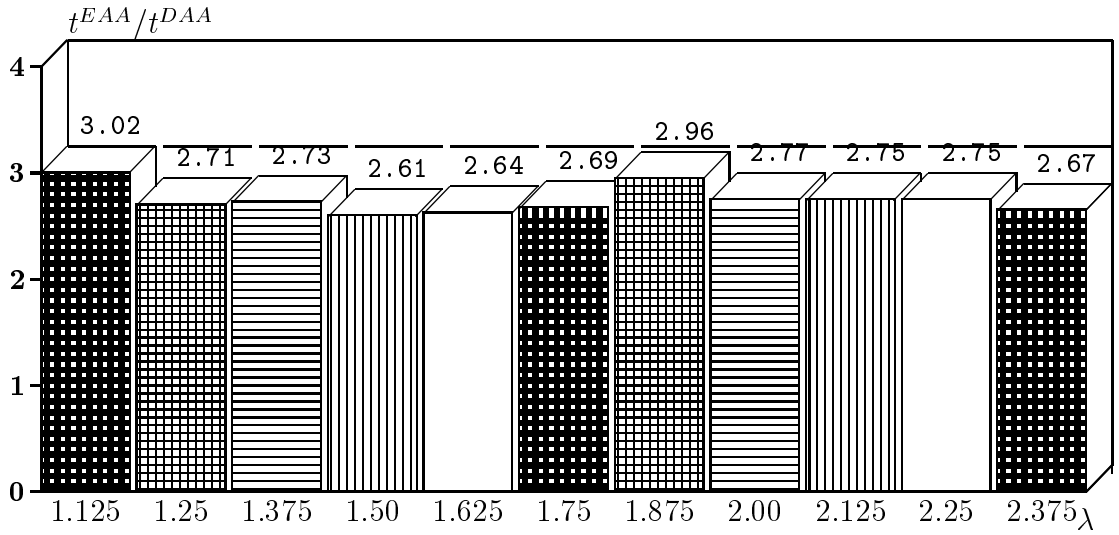


Abbildung 3.17: Quotient t^{EAA}/t^{DAA}

Zusammenfassung

Bei allen Beispielen zeigt sich, daß die doppelt adaptive Strategie der einfach adaptiven deutlich überlegen ist. Dies liegt einerseits an dem schon in Abschnitt 2.6.4 beschriebenen Geschwindigkeitsgewinn bei der Berechnung der Riemann-Integrale, andererseits daran, daß bei der doppelt adaptiven Strategie das aktive Intervall, das die singuläre Stelle enthält, größer als bei der einfach adaptiven ausfällt. Aber gerade die Berechung der Teilintegrale über Intervalle, die sich in unmittelbarer Nähe der singulären Stelle befinden, benötigen oft den meisten Aufwand.
Die Berechnung von Cauchy-Hauptwerten, deren singuläre Stelle auf der Maschine nicht exakt darstellbar ist, kann auf ähnliche Weise erfolgen. Dabei entstehen bei der Zerlegung zunächst Teilintervalle mit nicht darstellbaren Rändern. Diese werden im weiteren Verlauf durch Transformation auf den Referenzbereich bestimmt. Bei der Auswertung der transformierten Funktion wird die singuläre Stelle λ einfach durch ein entsprechendes Maschinenintervall ersetzt. Da diese Rechnungen auf der Maschine sowieso in Intervallarithmetik durchgeführt werden, ergeben sich hierdurch in den entsprechenden Programmen keine großen Änderungen.

Kapitel 4

Mehrfach iterierte singuläre Integrale

In diesem Kapitel untersuchen wir spezielle zweidimensionale Cauchy-Hauptwert-Integrale, die durch Hintereinanderschalten von eindimensionalen Cauchy-Haupwert-Integralen entstehen. Die numerische Bestimmung solcher Integrale wird u. a. von Monegato in [63] behandelt. Während Monegato sich mit der Verallgemeinerung von Konvergenzaussagen aus dem eindimensionalen Fall auf iterierte Produktformeln beschäftigt, geben wir hingegen ein Verfahren zur Einschließung solcher Integrale an.

4.1 Existenz und Eigenschaften

Das mehrfach iterierte singuläre Integral $I(f;\ \lambda, \mu)$ sei gegeben durch

$$I(f;\ \lambda, \mu) = \fint_a^b \fint_c^d \frac{f(x,y)}{(x-\lambda)(y-\mu)}\, dy\, dx. \tag{4.1}$$

Es ist klar, daß bloße Stetigkeit von f keine Garantie für die Existenz des Integrals (4.1) ist. Der Begriff der Hölder-Stetigkeit kann jedoch problemlos auf Funktionen von zwei reellen Argumenten übertragen werden. Die folgende Definition stammt aus [65].

Definition 37 *Die Funktion f erfüllt die Hölder-Bedingung $H(q_1, q_2)$ auf dem zweidimensionalen Intervall $B = [a,b] \times [c,d]$, wenn für beliebige Paare $(x,y), (x',y') \in B$ gilt*

$$|f(x,y) - f(x',y')| \leq p_1 |x - x'|^{q_1} + p_2 |y - y'|^{q_2} \tag{4.2}$$

mit geeigneten Konstanten q_1, q_2 $(0 < q_1, q_2 \leq 1)$, p_1 und p_2.

Erfüllt f die Hölder-Bedingung $H(q_1, q_2)$, dann ist die Funktion $g(x) := f(x,y)$ für beliebiges y Hölder-stetig von der Ordnung q_1. Die Hölder-Bedingung ist hinreichend für die Existenz des Integrals (4.1), es gilt nämlich der folgende in [63] bewiesene Satz.

Satz 38 *Das doppelt iterierte singuläre Integral $I(f;\ \lambda,\mu)$ existiert, falls f einer Hölder-Bedingung $H(q_1,q_2)$ genügt.*

Es ist offensichtlich, daß stetig differenzierbare Funktionen der Hölder-Bedingung $H(1,1)$ genügen. Damit ist die Existenz von (4.1) für alle auf einem kompakten Intervall stetig differenzierbaren Funktionen sichergestellt. Bevor wir uns der numerischen Bestimmung von (4.1) zuwenden, geben wir noch ein paar einfache Eigenschaften mehrfach iterierter Cauchy-Hauptwerte an. Ausgangspunkt ist dabei folgendes Lemma.

Lemma 39 *Die auf dem kompakten Intervall $B=[a,b]\times[c,d]$ $(c\le\mu\le d)$ durch*

$$g(x,y):=\begin{cases}\dfrac{f(x,y)-f(x,\mu)}{y-\mu} & \text{für } y\neq\mu\\[2ex] \dfrac{\partial f}{\partial y}(x,\mu) & \text{für } y=\mu\end{cases} \tag{4.3}$$

definierte Funktion g ist auf B stetig, falls f und $\frac{\partial f}{\partial y}$ auf B stetig sind.

Beweis: Sei $x^*\in[a,b]$ beliebig. Da $\frac{\partial f}{\partial y}$ auf B stetig ist, existiert nach dem Mittelwertsatz für alle Paare $(x,y)\in B, y\neq\mu$ ein ξ_{xy} mit $|\xi_{xy}-\mu|<|y-\mu|$ und

$$\left|\frac{f(x,y)-f(x,\mu)}{y-\mu}-\frac{\partial f}{\partial y}(x^*,\mu)\right|=\left|\frac{\partial f}{\partial y}(x,\xi_{xy})-\frac{\partial f}{\partial y}(x^*,\mu)\right|.$$

Aus der Stetigkeit von $\frac{\partial f}{\partial y}$ und obiger Gleichung folgt die Stetigkeit von g in (x^*,μ). ■

Mithilfe obigen Lemmas lassen sich Aussagen über die Vertauschbarkeit der Integrationsreihenfolge bzw. der Integration und Differentiation bei singulären Integralen formulieren. Die an f gemachten Voraussetzungen sind dabei einschränkender als nötig. Für das Weitere ist das Lemma in dieser Form aber völlig ausreichend.

Satz 40 *Ist die Funktion f und deren partielle Ableitung $\frac{\partial f}{\partial y}$ auf dem kompakten Intervall $B=[a,b]\times[c,d]$ stetig, so gilt für $c\le\mu\le d$*

$$\int_a^b \fint_c^d \frac{f(x,y)}{y-\mu}\,dy\,dx=\fint_c^d\int_a^b\frac{f(x,y)}{y-\mu}\,dx\,dy. \tag{4.4}$$

Ist $f\in C^2(I)$, so gilt für $c\le\mu\le d$ und $a\le\lambda\le b$

$$\fint_a^b\fint_c^d\frac{f(x,y)}{(x-\lambda)(y-\mu)}\,dy\,dx = \fint_c^d\fint_a^b\frac{f(x,y)}{(x-\lambda)(y-\mu)}\,dx\,dy \tag{4.5}$$

$$\frac{d}{dx}\fint_a^b\frac{f(x,y)}{y-\mu}\,dy = \fint_a^b\frac{\partial f}{\partial x}(x,y)/(y-\mu)\,dy. \tag{4.6}$$

Beweis: Sei g definiert nach (4.3) und $c < \mu < d$. Wegen der Stetigkeit von g gilt

$$\begin{aligned}\int_a^b \not\int_c^d \frac{f(x,y)}{y-\mu}\,dy\,dx &= \int_a^b \int_c^d g(x,y)\,dy + f(x,\mu)\ln\left(\left|\frac{d-\mu}{c-\mu}\right|\right)\,dx \\ &= \not\int_c^d \int_a^b g(x,y) + \frac{f(x,\mu)}{y-\mu}\,dx\,dy = \not\int_c^d \int_a^b \frac{f(x,y)}{y-\mu}\,dx\,dy.\end{aligned}$$

Unter der zusätzlichen Voraussetzung $f \in C^2(B)$ ist die Funktion $\frac{\partial g}{\partial x}$ nach Lemma 39 ebenfalls stetig und wegen der zuvor bewiesenen Gleichung (4.4) gilt

$$\begin{aligned}\not\int_a^b \not\int_c^d \frac{f(x,y)}{(x-\lambda)(y-\mu)}\,dy\,dx &= \not\int_a^b \int_c^d \frac{g(x,y)}{x-\lambda}\,dy\,dx + \not\int_a^b \not\int_c^d \frac{f(x,\mu)}{(x-\lambda)(y-\mu)}\,dy\,dx \\ &= \int_c^d \not\int_a^b \frac{g(x,y)}{x-\lambda}\,dx\,dy + \not\int_c^d \not\int_a^b \frac{f(x,\mu)}{(x-\lambda)(y-\mu)}\,dx\,dy \\ &= \not\int_c^d \not\int_a^b \frac{f(x,y)}{(x-\lambda)(y-\mu)}\,dx\,dy.\end{aligned}$$

Die Gleichung (4.6) ergibt sich aus

$$\begin{aligned}\frac{d}{dx}\not\int_a^b \frac{f(x,y)}{y-\mu}\,dy &= \frac{d}{dx}\left(\int_a^b g(x,y)\,dy + f(x,\mu)\not\int_a^b \frac{1}{y-\mu}\,dy\right) \\ &= \int_a^b \frac{\partial g}{\partial x}(x,y)\,dx + \frac{\partial f}{\partial x}(x,\mu)\not\int_a^b \frac{1}{y-\mu}\,dy \\ &= \not\int_a^b \frac{\partial f}{\partial x}(x,y)/(y-\mu)\,dx,\end{aligned}$$

wobei für die zweite Gleichung die Stetigkeit von $\frac{\partial g}{\partial x}$ und für die letzte Gleichung die Stetigkeit von $\frac{\partial^2 f}{\partial y \partial x}$ benutzt wurde. Die Fälle $\mu = c$ und $\mu = d$ beweist man analog. ■

Satz 40 läßt sich folgendermaßen zusammenfassen. Ist f bzw. sind die partiellen Ableitungen von f stetig differenzierbar, so gelten die gleichen Aussagen über Vertauschbarkeit von Integration bzw. Integration und Differentiation, wie sie von Riemann- oder Lebesgue- Integralen bekannt sind.

4.2 Produktformeln

Kartesische Produktformeln werden zur Berechnung von mehrdimensionalen Integralen (z. B. Flächenintegralen), die sich aus der Hintereinanderschaltung mehrerer eindimensionaler Integrale ergeben, verwendet. Gegeben seien die Hauptwertformeln

$$\begin{aligned}I_x(g;\ \lambda) &= \not\int_a^b \frac{g(x)}{x-\lambda}\,dx = Q_x(g) + R_x(g) \qquad (4.7)\\ I_y(g;\ \lambda) &= \not\int_c^d \frac{g(y)}{y-\mu}\,dy = Q_y(g) + R_y(g),\end{aligned}$$

wobei g eine Hölder-stetige Funktion bezeichne. Der Produktoperator I_xI_y ist für Funktionen f, die auf einem kompakten Intervall $B = [a,b] \times [c,d]$ einer Hölder-Bedingung der Form (4.2) genügen, folgendermaßen definiert. Die eindimensionale Funktion $f_x(y) = f(x,y)$ ist für beliebiges aber festes x aus dem Intervall $[a,b]$ Hölder-stetig, d. h. durch $g(x) := I_y(f_x)$ wird die eindimensionale Funktion g punktweise definiert. Da g ebenfalls Hölder-stetig ist (siehe [63]), existiert der Cauchy-Hauptwert $I_x(g)$ und dies soll genau der Wert von $I_xI_y(f)$ sein. $I_y(f)$ werden wir als Kurzschreibweise für die Funktion g mit $g(x) = I_y(f_x)$ verwenden. Der kartesische Produktoperator $Q_{x\times y} := Q_xQ_y$ ist ein Näherungsoperator für das Funktional $I_{x\times y}$

$$I_{x\times y}(f) := I_xI_y(f) = \fint_a^b \fint_c^d \frac{f(x)}{(x-\lambda)(y-\mu)}\,dy\,dx.$$

Das Fehlerfunktional $R_{x\times y} := I_{x\times y} - Q_{x\times y}$ kann aus den Fehleroperatoren der zugrundeliegenden Quadraturformeln berechnet werden. Durch einfache Umformungen (siehe [53]) ergibt sich nämlich

$$I_xI_y = (Q_x + R_x)I_y = Q_xQ_y + Q_xR_y + R_xI_y \tag{4.8}$$

und analog

$$I_xI_y = Q_xQ_y + R_xQ_y + I_xR_y. \tag{4.9}$$

Sind die Operatoren I_x und R_y vertauschbar, d. h. $I_xR_y = R_yI_x$, so hat $R_{x\times y}$ die äquivalenten Darstellungen

$$\begin{aligned} R_{x\times y} &= Q_xR_y + R_xI_y & (4.10)\\ &= Q_yR_x + R_yI_x. & (4.11)\end{aligned}$$

Die Produktformel $Q_{x\times y}$ ist für alle Polynome $p(x,y) = x^iy^j$ mit $0 \le i \le deg(Q_x)$ und $0 \le j \le deg(Q_y)$ exakt.

Ist eines der Integrale $I_x(f)$ oder $I_y(f)$ ein Riemann-Integral, werden Produktformeln genauso konstruiert. Nicht rechteckige Integrationsbereiche, wie z. B. Dreiecke oder Kreisscheiben, können durch nicht affine Transformationen auf rechteckige Bereiche zurückgeführt werden.

4.3 Adaptive Zerlegung

Nun wenden wir uns dem eigentlichen Problem der Berechnung einer Einschließung $[I(f;\ \lambda,\mu)]$ des Integrals (4.1) mit $d([R(f;\ \lambda,\mu)]) < \epsilon_{abs}$ bei einer vorgegebenen absoluten Schranke ϵ_{abs} zu. Hierbei setzen wir voraus, daß die Funktion f hinreichend oft stetig differenzierbar sei. Zur Bestimmung der Fehlereinschließung $[R(f;\ \lambda,\mu)]$

zerlegen wir den Integrationsbereich $B_{start} = [a,b] \times [c,d]$ in rechteckige Teilbereiche und berechnen die Integrale auf den entstandenen Teilbereichen mit Produktformeln. Je nach Lage der Singularitäten λ und μ treten dabei verschiedene Typen von Integralen auf. Liegt z. B. der Punkt (λ, μ) im neu enstandenen Integrationsbereich $B = B_x \times B_y (B_x, B_y \in \boldsymbol{I}R)$, so haben wir es mit einem doppelt iterierten Cauchy-Hauptwert zu tun. Liegt λ außerhalb von B_x und μ außerhalb von B_y, so liegt ein doppelt iteriertes Riemann-Integral vor. Der Einfachheit halber führen wir deshalb folgende Bezeichnungen für die verschiedenen auftretenden Integraltypen ein:

Typ Cauchy x Cauchy	falls	$\lambda \in B_x$ und $\mu \in B_y$
Typ Cauchy x Riemann	falls	$\lambda \in B_x$ und $\mu \notin B_y$
Typ Riemann x Cauchy	falls	$\lambda \notin B_x$ und $\mu \in B_y$
Typ Riemann x Riemann	falls	$\lambda \notin B_X$ und $\mu \notin B_y$

Die Zerlegung des Ausgangsbereichs B_{start} bestimmen wir mit Hilfe des global adaptiven Algorithmus 3.8. Dazu benötigen wir für jeden Integraltyp geeignete Fehlerdarstellungen. Für einen festen Integrationsbereich B hängen die Fehlereinschließungen von der Anzahl der Stützstellen n_x und n_y der Quadraturformeln Q_x und Q_y, sowie von den gewählten Ordnungen m_x und m_y der Taylorkoeffizienten der eindimensionalen Restglieder R_x und R_y ab. Wie diese im einzelnen aussehen, wird in den nächsten Abschnitten noch genauer angegeben. Um aus der Menge der berechenbaren Restgliedeinschließungen mit wenig Rechenaufwand eine möglichst günstige auszuwählen, zerlegen wir den Fehler $R_{x\times y}(f)$ in einen Anteil E_x in x-Richtung und einen Anteil E_y in y- Richtung mit

$$\begin{array}{rcl} E_x &=& R_x Q_y(f) \\ E_y &=& R_y I_x(f) \end{array} \quad \text{bzw.} \quad \begin{array}{rcl} E_x &=& R_x I_y(f) \\ E_y &=& R_y Q_x(f) \end{array}$$

Die Namen sind insofern gerechtfertigt, als E_x in erster Linie von Größen der Quadraturformel Q_x und E_y in erster Linie von Größen der Quadraturformel Q_y abhängt. Zur Bestimmung der Einschließungen $[E_x]$ und $[E_y]$ verwenden wir den Algorithmus 2.11, passen das lokale Genauigkeitskriterium aber der veränderten Ausgangssituation an. Außerdem muß für jeden auftretenden Integrationstyp festgelegt werden, wie im

$\epsilon_{local} := 0.5\,\eta\,\epsilon_{abs}\,\dfrac{|B|}{|B_{start}|}$
Bestimme $[E_x]$ gemäß Algorithmus 2.11
Bestimme $[E_y]$ gemäß Algorithmus 2.11
$[R_{x\times y}] := [E_x] + [E_y]$

Abbildung 4.1: Bestimmung der Restgliedeinschließung $[R_{x\times y}]$

Falle einer notwendigen Unterteilung verfahren werden soll. Darauf werden wir in den

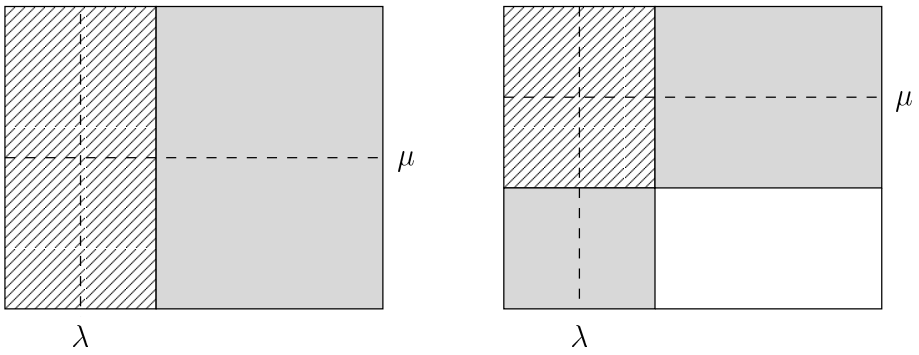


Abbildung 4.2: Startzerlegung mit zwei bzw. vier Teilbereichen

nächsten Abschnitten genauer eingehen.
Zu Beginn zerlegen wir unser Ausgangsintervall entsprechend dem eindimensionalen Fall (siehe Abschnitt 3.6.1) einmal entlang der x-Achse [1] und einmal entlang der y-Achse. Je nach Lage der Singularitäten erhalten wir ein, zwei oder vier Teilbereiche (siehe Abbildung 4.2). Der Punkt (λ, μ) ist Mittelpunkt eines Rechtecks. Außer dem iterierten Cauchy-Hauptwert (schraffierte Flächen) treten noch Integrale vom Typ Cauchy x Riemann bzw. Riemann x Cauchy (graue Flächen) und iterierte Riemann Integrale (weiße Flächen) auf.

4.3.1 Integral Typ Cauchy x Cauchy

Ausgehend von den Operatoren

$$\begin{aligned} I_x(f;\,0) &= \oint_{-1}^{1} \frac{f(x)}{x}\,dx = Q_x^H(f) + R_x^H(f) \\ I_y(f;\,0) &= \oint_{-1}^{1} \frac{f(y)}{y}\,dy = Q_y^H(f) + R_y^H(f) \end{aligned} \tag{4.12}$$

bilden wir den Produktoperator $I_{x\times y}$. Dabei sei Q_x^H die Hunterformel aus Satz 29 mit den Stützstellen x_i und den Gewichten v_i^H, Q_y^H die Hunterformel mit den Stützstellen y_i und den Gewichten w_i^H. Für $\lambda = mid(B_x)$ und $\mu = mid(B_y)$ gilt

$$I(f;\,\lambda,\mu) = \oint_{B_x}\oint_{B_y} \frac{f(x,y)}{(x-\lambda)(y-\mu)}\,dy\,dx = I_{x\times y}(g) = Q_{x\times y}(g) + R_{x\times y}(g) \tag{4.13}$$

mit $g(x) := f(\frac{1}{2}\,d_x\,x + mid(B_x), \frac{1}{2}\,d_y\,y + mid(B_y))$, $d_x := d(B_x)$ und $d_y := d(B_y)$.

[1] Diese Zerlegung ermöglicht uns den Einsatz der speziellen Hunterformeln (3.38) und (3.39).

Kubatursumme

Die Kubatursumme $Q(f;\ \lambda,\mu) := Q_{x\times y}(g)$ hat für gerades n_y die Darstellung

$$Q(f;\ \lambda,\mu) = \begin{cases} \displaystyle\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} v_i^H\ w_j^H\ g(x_i,y_j) & \text{für } n_x \text{ gerade} \\ \displaystyle\sum_{j=1}^{n_y} w_j^H \left(\sum_{i=1}^{n_x} v_i^H\ g(x_i,y_j) + v_0^H\ \frac{\partial g}{\partial x}(0,y_j)\right) & \text{für } n_x \text{ ungerade} \end{cases}$$

Für den Fall n_x gerade und n_y ungerade erhält man die Kubatursumme aus dem Fall n_x ungerade und n_y gerade durch Vertauschen der Bezeichnungen. Sind n_y und n_x ungerade, gilt

$$\begin{aligned} Q(f;\ \lambda,\mu) &= \sum_{i=1}^{n_x}\left(\sum_{j=1}^{n_y} v_i^H\ w_j^H\ g(x_i,y_j) + v_i^H\ w_0^H\ \frac{\partial g}{\partial y}(x_i,0)\right) \\ &\quad + \sum_{j=1}^{n_y} v_0^H\ w_j^H\ \frac{\partial g}{\partial x}(0,y_j) + v_0^H\ w_0^H\ \frac{\partial^2 g}{\partial x\partial y}(0,0). \end{aligned}$$

Restglied

Mit der Bezeichnung dieses Abschnitts gilt folgende Aussage über das Restglied $R(f;\ \lambda,\mu) = I(f;\ \lambda,\mu) - Q(f;\ \lambda,\mu)$ des iterierten Cauchy-Hauptwertes.

Satz 41 *Sei v^+ die Summe aller positiven Gewichte der Hunterformel Q_x^H. Das Restglied $R(f;\ \lambda,\mu)$ hat die Form $R(f;\ \lambda,\mu) = R_x^H I_y(g) + R_y^H Q_x^H(g)$ mit*

$$\begin{aligned} R_y^H Q_x^H(g) &\in v^+ \left(\frac{d_y}{2}\right)^{m_y} (c^{H+}_{m_y,n_y} - c^{H-}_{m_y,n_y})\,(f^{(0,m_y)}(B) - f^{(0,m_y)}(B)) + \delta_{n_x} \\ R_x^H I_y(g) &\in d_y \left(\frac{d_x}{2}\right)^{m_x} (c^{H+}_{m_x,n_x}\, f^{(m_x,1)}(B) + c^{H-}_{m_x,n_x}\, f^{(m_x,1)}(B)) \end{aligned} \tag{4.14}$$

und

$$\delta_{n_x} = \begin{cases} 0 & n_x \text{ gerade} \\ \left(\frac{d_x}{2}\right)\left(\frac{d_y}{2}\right)^{m_y} v_0^H\ (c^{H+}_{m_y,n_y}\, f^{(1,m_y)}(B) + c^{H-}_{m_y,n_y}\, f^{(1,m_y)}(B)) & n_x \text{ ungerade} \end{cases}$$

Beweis: Wegen der Symmetrie der Gewichte v_i^H gilt $\sum_{i=1}^{n_x} v_i^H = 2\,v^+$ und damit erhalten wir für gerade n_x ($\xi^+, \xi^- \in [-1,1]$)

$$\begin{aligned} R_y^H Q_x^H(g) &= Q_x^H(c^{H+}_{m_y,n_y}\, g^{(0,m_y)}(x,\xi^+) + c^{H-}_{m_y,n_y}\, g^{(0,m_y)}(x,\xi^-)) \\ &\in \sum_{i=1}^{n_x} v_i^H\ c^{H+}_{m_y,n_y}\, g^{(0,m_y)}(B) + \sum_{i=1}^{n_x} v_i^H\ c^{H-}_{m_y,n_y}\, g^{(0,m_y)}(B) \\ &= v^+\, c^{H+}_{m_y,n_y}\, g^{(0,m_y)}(B) + v^+\, c^{H+}_{m_y,n_y}\,(-g^{(0,m_y)}(B)) + v^+(-c^{H-}_{m_y,n_y})\, g^{(0,m_y)}(B) \\ &\quad + v^+(-c^{H-}_{m_y,n_y})\,(-g^{(0,m_y)}(B)). \end{aligned}$$

Aus der letzten Gleichung folgt die Behauptung. Für ungerade n_x ist die Umformung ähnlich.

$$\begin{aligned} R_x^H I_y(g) &= I_y(c^{H+}_{m_x,n_x} g^{(m_x,0)}(\xi^+, y) + c^{H+}_{m_x,n_x} g^{(m_x,0)}(\xi^-, y)) \\ &= c^{H+}_{m_x,n_x}\, g^{(m_x,1)}(\xi^+, \eta^+) + c^{H+}_{m_x,n_x}\, g^{(m_x,1)}(\xi^-, \eta^-) \end{aligned}$$

Die letzte Gleichung folgt aus dem Mittelwertsatz angewandt auf die Funktionen

$$\frac{g^{(m_x,0)}(\xi^+, y) - g^{(m_x,0)}(\xi^+, 0)}{y} \text{ bzw. } \frac{g^{(m_x,0)}(\xi^-, y) - g^{(m_x,0)}(\xi^-, 0)}{y}$$

und dem Mittelwertsatz der Integralrechnung. ■

Zerlegung

Der Teilbereich $B = B_x \times B_y$ mit $B_x = [a', b']$ und $B_y = [c', d']$ wird in fünf neue Teilbereiche unterteilt. Die zusätzlichen $x-$ und $y-$Koordinaten der Eckpunkte ergeben sich aus den Zerlegungen der Intervalle B_x und B_y gemäß Algorithmus 3.4.

$$\begin{aligned} \alpha &= \tfrac{\lambda + a'}{2}, \quad \beta = \tfrac{\lambda + b'}{2}, \\ \gamma &= \tfrac{\mu + c'}{2}, \quad \delta = \tfrac{\mu + c'}{2}. \end{aligned}$$

Mit diesen Eckpunkten stehen mehrere mögliche Zerlegungen zur Auswahl. Wir beschränken uns auf die beiden untenstehenden mit jeweils fünf Teilbereichen.

$x-$ Zerlegung			$y-$ Zerlegung		
I_1	=	$[\alpha, \beta] \times [\gamma, \delta]$	I_1	=	$[\alpha, \beta] \times [\gamma, \delta]$
I_2	=	$[\alpha, \beta] \times [c', \gamma]$	I_2	=	$[a', b'] \times [c', \gamma]$
I_3	=	$[\alpha, \beta] \times [\delta, d']$	I_3	=	$[a', b'] \times [\delta, d']$
I_4	=	$[a', \alpha] \times [c', d']$	I_4	=	$[a', \alpha] \times [\gamma, \delta]$
I_5	=	$[\beta, b'] \times [c', d']$	I_5	=	$[\beta, b'] \times [\gamma, \delta]$

Welche der beiden Zerlegungen gewählt wird, machen wir von den Fehleranteilen $E_x = R_x^H I_y(g)$ und $E_y = R_y^H Q_x^H(g)$ abhängig. Überwiegt die Komponente in $x-$Richtung ($d([E_x]) > d([E_y])$), dann wählen wir die $x-$Zerlegung, ansonsten die $y-$Zerlegung. Außerdem weisen wir darauf hin, daß bei der Bestimmung der Fehleranteile die Reihenfolge der Berechnung von Bedeutung ist, da der Anteil E_y in y-Richtung auch abhängig von der Anzahl der Stützstellen n_x der Quadraturformel Q_x^H ist. Deswegen muß zuerst der Anteil in $x-$Richtung und dann der in $y-$Richtung bestimmt werden. Der Integrationsbereich des iterierten Cauchy-Hauptwertes ist sowohl in $x-$ als auch in $y-$Richtung halbiert. Dies garantiert eine Reduzierung der Restgliedeinschließung, da einerseits die Vorfaktoren d_x und d_y halbiert werden und andererseits die entsprechenden Taylorkoeffizienten über einem deutlich kleineren Teilbereich bestimmt werden.

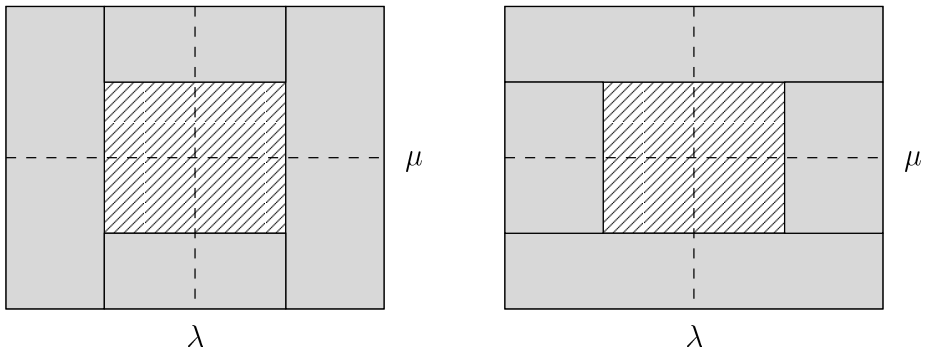


Abbildung 4.3: Zerlegung für Integrationstyp Cauchy x Cauchy

4.3.2 Integral Typ Cauchy x Riemann

Die Operatoren I_x und Q_x^H übernehmen wir unverändert aus dem vorhergehenden Abschnitt. $I_y(f)$ bezeichne jetzt das Riemann-Integral

$$I_y(f) = \int_{-1}^{1} f(y)\,dy = Q_y(f) + R_y(f)$$

und Q_y die Gauß-Legendre-Formel mit den Stützstellen y_i und den Gewichten w_i, R_y das dazugehörige Restfunktional. Für $\lambda = mid(B_x)$ gilt

$$I(f;\,\lambda) := \fint_{B_x} \int_{B_y} \frac{f(x,y)}{x-\lambda}\,dy\,dx = \frac{d_y}{2}\,I_{x\times y}(g) = \frac{d_y}{2}\,(Q_{x\times y}(g) + R_{x\times y}(g)). \qquad (4.15)$$

Kubatursumme

Die Kubatursumme $Q(f;\,\lambda) := \frac{d_y}{2}\,Q_{x\times y}(g)$ besitzt die Form

$$Q(f;\,\lambda) = \begin{cases} \dfrac{d_y}{2}\displaystyle\sum_{i=1}^{n_x}\sum_{j=1}^{n_y} v_i^H\,w_j\,g(x_i,y_j), & \text{für } n_x \text{ gerade} \\ \dfrac{d_y}{2}\displaystyle\sum_{j=1}^{n_y} w_j \left(\sum_{i=1}^{n_x} v_i^H\,g(x_i,y_j) + v_0^H\,\frac{\partial g}{\partial x}(0,y_j)\right), & \text{für } n_x \text{ ungerade} \end{cases}$$

Restglied

Analog zum vorherigen Satz läßt sich folgende Aussage über das Restglied $R(f;\,\lambda)$ des Integrals (4.15) herleiten.

Satz 42 *Sei v^+ die Summe aller positiven Gewichte der Hunterformel Q_x^H. Das Restglied hat die Darstellung*

$$R(f;\,\lambda) = \frac{d_y}{2}\,(R_x^H Q_y(g) + R_y I_x(g)) = \frac{d_y}{2}\,(R_y Q_x^H(g) + R_x^H I_y(g))$$

mit

$$R_x^H Q_y(g) \in 2\left(\frac{d_x}{2}\right)^{m_x} (c^{H+}_{m_x,n_x} f^{(m_x,0)}(B) + c^{H-}_{m_x,n_x} f^{(m_x,0)}(B)) \tag{4.16}$$

$$R_y I_x(g) \in d_x \left(\frac{d_y}{2}\right)^{m_y} (c^{+}_{m_y,n_y} f^{(1,m_y)}(B) + c^{-}_{m_y,n_y} f^{(1,m_y)}(B))$$

$$R_x^H I_y(g) \in 2\left(\frac{d_x}{2}\right)^{m_x} (c^{H+}_{m_x,n_x} f^{(m_x,0)}(B) + c^{H-}_{m_x,n_x} f^{(m_x,0)}(B)) \tag{4.17}$$

$$R_y Q_x^H(g) \in v^+ \left(\frac{d_y}{2}\right)^{m_y} (c^{+}_{m_y,n_y} - c^{-}_{m_y,n_y}) (f^{(0,m_y)}(B) - f^{(0,m_y)}(B)) + \delta_{n_x}$$

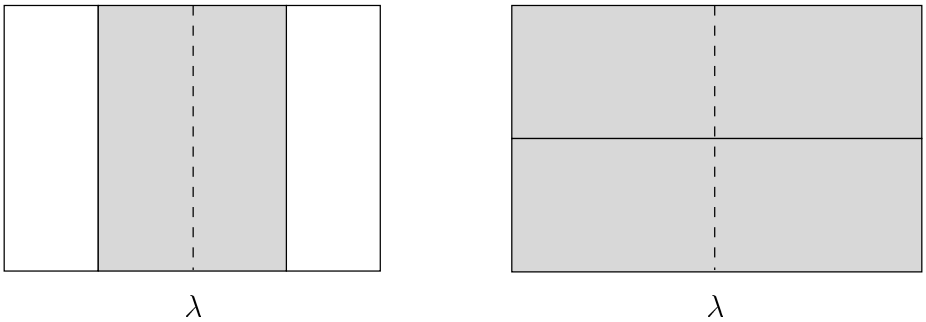
und

$$\delta_{n_x} = \begin{cases} 0, & n_x \text{ gerade} \\ \left(\frac{d_x}{2}\right)\left(\frac{d_y}{2}\right)^{m_y} v_0^H (c^{+}_{m_y,n_y} f^{(1,m_y)}(B) + c^{-}_{m_y,n_y} f^{(1,m_y)}(B)), & n_x \text{ ungerade} \end{cases}$$

Satz 42 bietet zwei Möglichkeiten zur Berechnung der Fehlergliedeinschließung an. Die erste Möglichkeit mit $E_x = R_x^H Q_y(g)$ und $E_y = R_y I_x(g)$ hat den Vorteil, daß der Fehler in y-Richtung unabhängig von der Anzahl der Stützstellen n_x ist. Außerdem fällt der Vorfaktor normalerweise günstiger als im Falle $E_y = R_y Q_x^H(g)$ aus. Dafür muß für gerade n_x ein Taylorkoeffizient höherer Ordnung berechnet werden. Für die Fehler in x-Richtung sind die beiden Möglichkeiten äquivalent.

Zerlegung

Das Intervall $B = [a', b'] \times [c', d']$ zerlegen wir abhängig von den Fehlern E_x und E_y in x- und/oder y-Richtung (siehe Abbildung 4.4) . Die Wahl $\eta = 1$ garantiert, daß der Durchmesser der Restgliedeinschließung für den Teilbereich B kleiner als $\epsilon_{abs} \frac{|B|}{|B_{start}|}$ ist, wenn das Intervall nicht mehr weiter zerlegt wird. Erfüllen alle Teilbereiche der Zerlegung diese Bedingung, dann terminiert das Verfahren. In der Praxis erweist es sich jedoch als vorteilhaft, η etwas größer zu wählen (z. B. 2). Um in diesem Fall eine Terminierung des Verfahrens sicherzustellen, müssen zusätzliche Maßnahmen getroffen werden.



λ λ

$\epsilon_{local} := \epsilon_{abs} \frac{\eta}{2} \frac{B}{B_{start}}$
$Z_x := \{[a', b']\}$
$Z_y := \{[c', d']\}$
if $(d([E_x]) > \epsilon_{local})$
  $\alpha := 0.5\,(a' + \lambda)$
  $\beta := 0.5\,(b' + \lambda)$
  $Z_x := \{[a', \alpha], [\alpha, \beta], [\beta, b']\}$
fi
if $(d([E_y]) > \epsilon_{local})$
  $\gamma := \frac{1}{2}(c' + d')$
  $Z_y := \{[c', \gamma], [\gamma, d']\}$
fi
$Z := Z_x \times Z_y$

Abbildung 4.4: Zerlegung Typ Cauchy x Riemann

4.3.3 Integral Typ Riemann x Riemann

In diesem Abschnitt übernehmen wir die Bezeichnungen für I_y, Q_y unverändert aus dem vorigen Abschnitt. I_x bezeichne jetzt ebenfalls ein Riemann-Integral, Q_x die Gauß-Legendre-Formel mit den n_x Knoten x_i und Gewichten v_i und R_x das entsprechende Fehlerfunktional

$$I_x(f) = \int_{-1}^{1} f(x)\,dx = Q_x(f) + R_x(f).$$

Der Produktoperator ist das Flächenintegral

$$\begin{aligned} I(f) &:= \int_B f(x,y)\,d(x,y) = \frac{d_x\,d_y}{4}\,I_{x\times y}(g) \\ &= \frac{d_x\,d_y}{4}\,(Q_{x\times y}(g) + R_{x\times y}(g)). \end{aligned} \tag{4.18}$$

Kubatursumme und Restglied

Die Kubatursumme hat die Form

$$Q(f) = \frac{d_x\,d_y}{4} \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} v_i\,w_j\,g(x_i, y_j)$$

und das dazugehörige Fehlerglied

$$R(f) = \frac{d_x\,d_y}{4}\,(R_x Q_y(g) + R_y I_x(g))$$

mit

$$\begin{aligned} R_x Q_y(g) &\in 2\left(\frac{d_x}{2}\right)^{m_x} (c^+_{m_x,n_x} f^{(m_x,0)}(B) + c^-_{m_x,n_x} f^{(m_x,0)}(B)), \\ R_y I_x(g) &\in 2\left(\frac{d_y}{2}\right)^{m_y} (c^+_{m_y,n_y} f^{(0,m_y)}(B) + c^-_{m_y,n_y} f^{(0,m_y)}(B)). \end{aligned}$$

Zerlegung

Die Zerlegung des Teilbereichs $B = [a', b'] \times [c', d']$ gestaltet sich ähnlich wie bei den Integralen vom Typ Cauchy x Riemann. Allerdings wird das Intervall B_x im Falle einer Zerlegung in y-Richtung lediglich halbiert.

$\epsilon_{local} := \epsilon_{abs} \frac{\eta}{2} \frac{B}{B_{start}}$
$Z_x := \{[a', b']\}$
$Z_y := \{[c', d']\}$
if $(d([E_x]) > \epsilon_{local})$ $Z_x := \{[a', 0.5\,(a' + b')], [0.5\,(a' + b'), b']\}$ **fi**
if $(d([E_y]) > \epsilon_{local})$ $Z_y := \{[c', 0.5\,(c' + d')], [0.5\,(c' + d'), d']\}$ **fi**
$Z := Z_x \times Z_y$

Abbildung 4.5: Zerlegung Typ Riemann x Riemann

4.4 Numerische Beispiele

Die Notation übernehmen wir unverändert aus Abschnitt 2.6.4. $\#\tilde{R}$ bezeichnet die Anzahl der Intervalle, die im Laufe der Zerlegung unterteilt werden.

4.4.1 Mehrfach iterierte Riemann-Integrale

Zunächst bestimmen wir zwei doppelt iterierte Riemann-Integrale, deren Integranden zwar stetig differenzierbar sind, deren partielle Ableitungen aber wegen hebbarer Singularitäten nicht mit automatischer Differentiation berechnet werden könen. Die Integrale können problemlos mit den zuvor beschriebenen Methoden verifiziert bestimmt werden.

Beispiel 1

Die Funktion h mit $h(x) = \ln(x+1)\cos(kxy)/x$ besitzt in $x = 0$ eine hebbare Singularität. Wir berechnen das Integral

$$I(f;\, 0) = \int_{-0.5}^{1} \int_{0}^{1} \ln(x+1)\cos(kxy)/x \, dy \, dx,$$

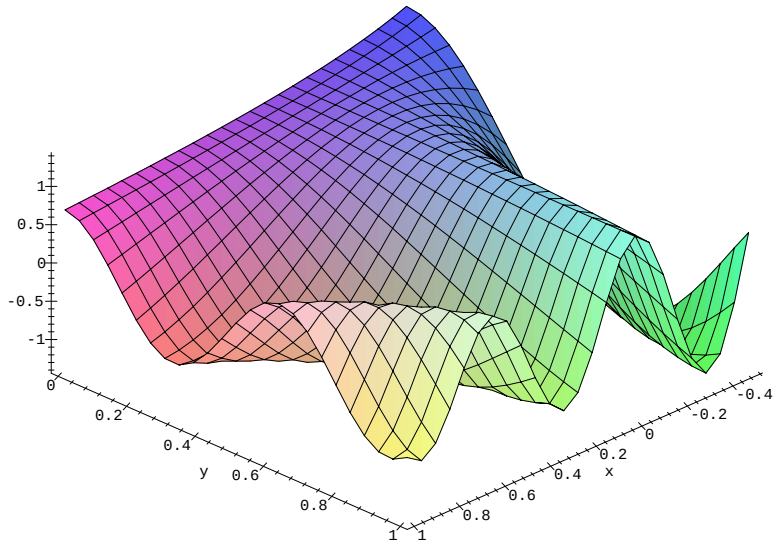


Abbildung 4.6: $h(x) = \ln(x+1)\cos(10xy)/x$

indem wir es als Integral vom Typ Cauchy x Riemann auffassen. Die Ergebnisse sind für $k = 30$ und $k = 100$ weiter unten zusammengefaßt. Die Berechnung der bei der Zerlegung erzeugten Integrale vom Typ Cauchy x Riemann wurden mit Formel (4.17) durchgeführt. Verwendet man dazu Formel (4.16) ergeben sich die gleichen Ergebnisse, es erhöht sich lediglich die Laufzeit geringfügig.

ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$1.0\frac{9}{5}E-01$	$3.1E-03$	2	0	224
$1.0E-04$	$1.06\frac{90}{87}E-01$	$1.5E-05$	2	0	374
$1.0E-06$	$1.0688\frac{52}{45}E-01$	$6.0E-07$	5	2	572
$1.0E-08$	$1.068848\frac{96}{88}E-01$	$7.2E-09$	5	2	812
$1.0E-10$	$1.06884892\frac{22}{16}E-01$	$5.8E-11$	10	2	1363
$1.0E-12$	$1.068848921\frac{90}{88}E-01$	$8.9E-13$	14	5	2225
$1.0E-14$	$1.06884892189\frac{12}{09}E-01$	$1.8E-014$	19	10	3577
$k = 30$					

ϵ_{abs}	$[I(f;\ \lambda)]$	$d([I(f;\ \lambda)])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-04$	$3.1\overset{13}{05}E-02$	$7.2E-05$	14	5	2048
$1.0E-06$	$3.10\overset{904}{897}E-02$	$5.9E-07$	16	7	3067
$1.0E-08$	$3.10900\overset{7}{5}E-02$	$9.8E-09$	42	20	5232
$1.0E-10$	$3.1090061\overset{5}{3}E-02$	$9.6E-11$	53	31	8354
$1.0E-12$	$3.109006144\overset{4}{2}E-02$	$1.0E-12$	89	42	13745
$1.0E-14$	$3.10900614429\overset{5}{2}E-02$	$2.6E-14$	148	79	22717
$k=100$					

Beispiel 2

Die Funktion h mit $h(x)=\sin(xy)/(xy)$ besitzt in $x=0$ und $y=0$ hebbare Singularitäten. Das Riemann-Integral

$$I(f;\ 0,0)=\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}\frac{\sin(xy)}{xy}\,dy\,dx$$

berechnen wir, indem wir es wie ein doppelt iteriertes Cauchy-Integral behandeln. Die auf der Maschine nicht exakt darstellbaren Integrationsgrenzen werden durch enge Einschließungen mit $\overline{a}<0<\underline{b}$ bzw. $\overline{c}<0<\underline{d}$ ersetzt, und das Integral folgendermaßen zerlegt

$$\begin{aligned}
&\oint_{[a]}^{[b]}\oint_{[c]}^{[d]}\frac{f(x,y)}{xy}\,dy\,dx \quad = \qquad (4.19)\\
&\int_{[a]}^{\overline{a}}\int_{[c]}^{\overline{c}}\frac{f(x,y)}{xy}\,dy\,dx+\int_{[a]}^{\overline{a}}\oint_{\overline{c}}^{\underline{d}}\frac{f(x,y)}{xy}\,dy\,dx+\int_{[a]}^{\overline{a}}\int_{\underline{d}}^{[d]}\frac{f(x,y)}{xy}\,dy\,dx+\\
&\oint_{\overline{a}}^{\underline{b}}\int_{[c]}^{\overline{c}}\frac{f(x,y)}{xy}\,dy\,dx+\oint_{\overline{a}}^{\underline{b}}\oint_{\overline{c}}^{\underline{d}}\frac{f(x,y)}{xy}\,dy\,dx+\oint_{\overline{a}}^{\underline{b}}\int_{\underline{d}}^{[d]}\frac{f(x,y)}{xy}\,dy\,dx+\\
&\int_{\underline{b}}^{[b]}\int_{[c]}^{\overline{c}}\frac{f(x,y)}{xy}\,dy\,dx+\int_{\underline{b}}^{[b]}\oint_{\overline{c}}^{\underline{d}}\frac{f(x,y)}{xy}\,dy\,dx+\int_{\underline{b}}^{[b]}\int_{\underline{d}}^{[d]}\frac{f(x,y)}{xy}\,dy\,dx.
\end{aligned}$$

Die doppelt iterierten Riemann-Integrale lassen sich mit Hilfe des Mittelwertsatzes der Integralrechnung bestimmen, so gilt z. B.

$$\int_{[a]}^{\overline{a}}\int_{\underline{d}}^{[d]}\frac{f(x,y)}{xy}\,dy\,dx\in[0,\overline{a}-\underline{a}]\,[0,\overline{d}-\underline{d}]\,\frac{f([a],[d])}{[a]\,[d]}. \qquad (4.20)$$

Da die Maschinenintervalle $[a]$ und $[b]$ Einschließungen reeller Punktgrößen sind, erhält man in der Regel auf diese Weise sehr enge Einschließungen. Die in (4.19) auftretenden

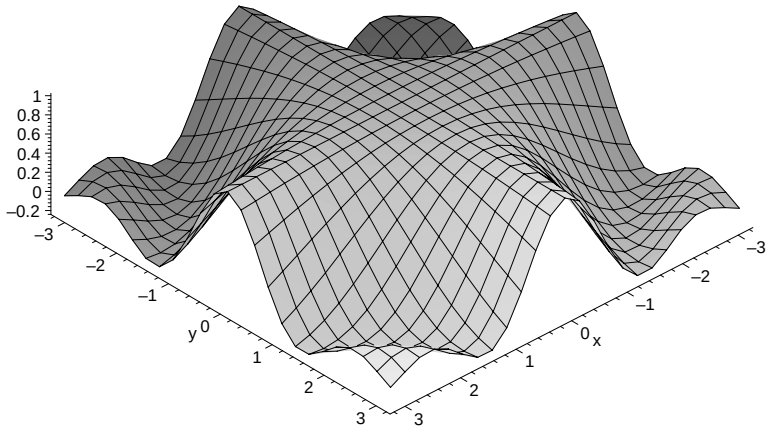


Abbildung 4.7: $h(x) = \sin(xy)/(xy)$

Integrale vom Typ Cauchy x Riemann bzw. Riemann x Cauchy können mit eindimensionalen Hunterformeln bestimmt werden, denn es gilt

$$\begin{aligned}
\int_{[a]}^{\overline{a}} \oint_{\overline{c}}^{\underline{d}} \frac{f(x,y)}{xy}\,dy\,dx &\in [0, \overline{a} - \underline{a}]\ \oint_{\overline{c}}^{\underline{d}} \frac{f([a],y)}{[a]\,y}\,dy \\
&:= \{\xi_1 \oint_{\overline{c}}^{\underline{d}} \frac{f(\xi_2,y)}{\xi_2\,y}\,dy | 0 \le \xi_1 \le \overline{a} - \underline{a}, \xi_2 \in [a]\} \\
\oint_{\overline{a}}^{\underline{b}} \int_{\underline{d}}^{[d]} \frac{f(x,y)}{xy}\,dy\,dx &\in [0, \overline{d} - \underline{d}]\ \oint_{\overline{a}}^{\underline{b}} \frac{f(x,[d])}{x\,[d]}\,dx.
\end{aligned}$$

Bei Vorgabe eines geeigneten lokalen Genauigkeitskriteriums können diese Integrationsbereiche mit in die global adaptive Strategie zur Berechnung des doppelt iterierten Cauchy-Hauptwertes eingebunden werden.

Die Ergebnisse der Berechnungen unter Verwendung von Formel (4.17) bzw. Formel (4.16) sind in den nächsten beiden Tabellen zusammengefaßt. Die acht Randbereiche der Zerlegung (4.19) sind in der Anzahl der aktiven Teilbereiche mitenthalten.

Formel (4.16)					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$1.801^{84}_{78}E+01$	$4.5E-04$	9	0	169
$1.0E-04$	$1.8018^{13}_{06}E+01$	$5.8E-05$	9	0	202
$1.0E-06$	$1.801809^{45}_{38}E+01$	$6.4E-07$	9	0	298
$1.0E-08$	$1.80180941^{48}_{39}E+01$	$8.0E-09$	17	2	1104
$1.0E-10$	$1.8018094143^{55}_{46}E+01$	$7.8E-11$	33	8	2716
$1.0E-12$	$1.80180941435061^{6}_{1}E+01$	$4.3E-13$	33	8	3817
$1.0E-14$	$1.801809414350638^{8}_{4}E+01$	$7.5E-14$	63	30	7984

Formel (4.17)					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$1.801^{84}_{78}E+01$	$4.5E-04$	9	0	169
$1.0E-04$	$1.8018^{13}_{06}E+01$	$5.8E-05$	9	0	202
$1.0E-06$	$1.801809^{45}_{38}E+01$	$6.4E-07$	9	0	298
$1.0E-08$	$1.801809414^{7}_{0}E+01$	$6.1E-09$	17	2	1104
$1.0E-10$	$1.8018094143^{55}_{47}E+01$	$7.0E-11$	37	9	2984
$1.0E-12$	$1.801809414350^{68}_{60}E+01$	$6.9E-13$	33	8	3715
$1.0E-14$	$1.801809414350638^{8}_{5}E+01$	$1.8E-14$	65	32	8016

4.4.2 Integral vom Typ Cauchy x Riemann

Als nächstes betrachten wir das singuläre Integral

$$I(f;\,0) = \fint_{-0.5\pi}^{0.5\pi} \int_{-0.5\pi}^{0.5\pi} \frac{\sinh(x^2+y^2)(x+2)(y+0.25)}{x}\,dy\,dx.$$

Um die auf der Maschine nicht exakt darstellbaren Randbereiche in den Griff zu bekommen, zerlegt man den Integrationsbereich ähnlich wie im vorherigen Beispiel. Die zusätzlich entstehenden doppelt iterierten Riemann-Integrale über Randbereiche werden mit (eindimensionalen) Quadraturformeln bestimmt. Die insgesamt acht Randbereiche sind wiederum in $\#R$ enthalten. Die unten aufgelisteten Ergebnisse wurden

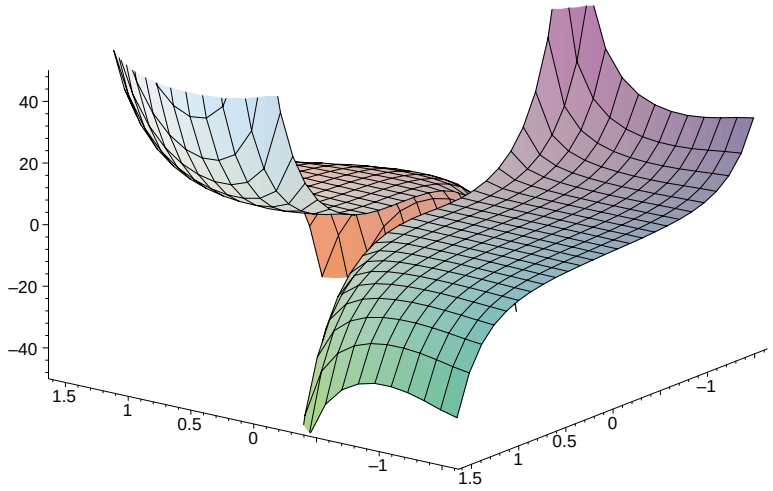


Abbildung 4.8: $h(x) = \sinh(x^2 + y^2)(x + 2)(y + 0.25)/x$

unter Verwendung von Formel (4.17) erzielt. Der Einsatz von Formel (4.16) bringt keine nennenswerte Unterschiede.

ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$1.12^{11}_{07}E+01$	$2.6E-03$	9	0	202
$1.0E-04$	$1.1209^{20}_{16}E+01$	$3.3E-05$	9	0	298
$1.0E-06$	$1.1209177^{2}_{0}E+01$	$1.9E-07$	14	1	732
$1.0E-08$	$1.1209177^{11}_{09}E+01$	$9.6E-09$	15	2	985
$1.0E-10$	$1.1209177097^{51}_{41}E+01$	$8.1E-11$	21	5	1936
$1.0E-12$	$1.12091770974^{66}_{57}E+01$	$7.7E-12$	28	7	2378

4.4.3 Mehrfach iterierte Cauchy-Hauptwerte

Funktion mit Peak

Die Funktion f mit $f(x) = (0.01+(x-0.25)^2+(y-0.75)^2)^{-1}$ nimmt in $P = (0.25, 0.75)$ ihr globales Maximum ein. Wir berechnen das doppelt iterierte Cauchy-Integral

$$I(f;\ \lambda,\mu) = \fint_0^1 \fint_0^1 \frac{1}{(0.01 + (x-0.25)^2 + (y-0.75)^2)(x-\lambda)(y-\mu)}\, dy\, dx$$

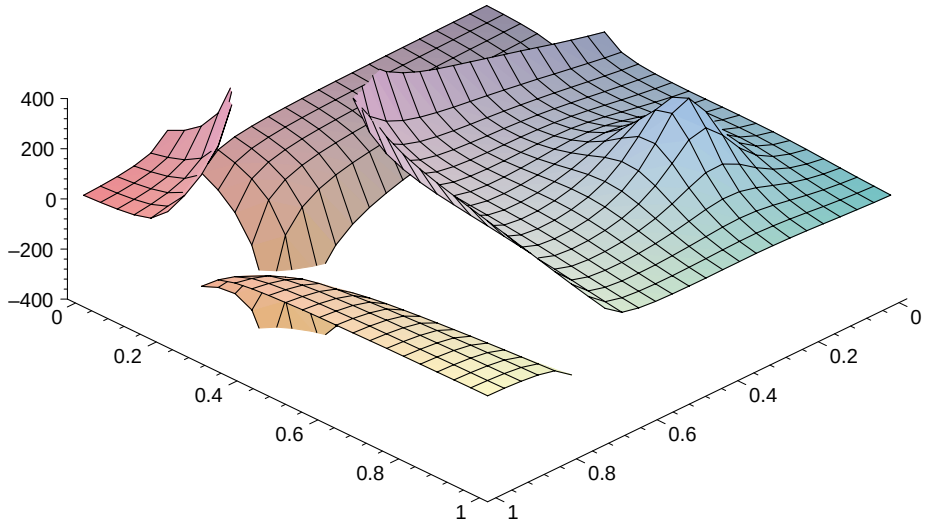


Abbildung 4.9: $h(x) = ((0.01 + (x - 0.25)^2 + (y - 0.75)^2)(x - 0.75)(y - 0.25))^{-1}$

für $(\lambda, \mu) = (0.75, 0.25)$ und $(\lambda, \mu) = P$. Fällt (λ, μ) mit P zusammen, vergrößert sich erwartungsmäß der zur Berechnung des Integrals benötigte Aufwand. Der Vergleich des Aufwands unter Verwendung von Formel (4.16) bzw. von Formel (4.17) ergibt kein eindeutiges Ergebnis, welche der beiden zu bevorzugen ist. Während Formel (4.17) bei kleinem ϵ_{abs} vorteilhaft zu sein scheint, benötigt das Verfahren unter Verwendung von (4.16) zumindest im ersten Fall für größere ϵ_{abs} weniger Teilbereiche und weniger Funktionsauswertungen, was sich allerdings nur in geringem Geschwindigkeitszuwachs äußert.

Formel (4.16)					
$\lambda = 0.75, \mu = 0.25$					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$-5.07\frac{77}{84}E+01$	$5.6E-03$	37	9	2668
$1.0E-04$	$-5.0780\frac{5}{7}E+01$	$9.8E-05$	39	10	4113
$1.0E-06$	$-5.078064\frac{2}{4}E+01$	$9.8E-07$	51	14	6723
$1.0E-08$	$-5.07806427\frac{3}{4}E+01$	$9.4E-09$	81	29	10617
$1.0E-10$	$-5.078064273\frac{48}{50}E+01$	$9.6E-11$	132	39	16656
$1.0E-12$	$-5.078064273493\frac{1}{4}E+01$	$1.6E-12$	157	52	24751

Formel (4.17)					
$\lambda = 0.75, \mu = 0.25$					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$-5.07^{77}_{84}E+01$	$5.8E-03$	31	9	2758
$1.0E-04$	$-5.0780^{5}_{7}E+01$	$9.6E-05$	39	10	4113
$1.0E-06$	$-5.078064^{2}_{4}E+01$	$9.5E-07$	57	17	7095
$1.0E-08$	$-5.07806427^{2}_{4}E+01$	$9.9E-09$	86	30	10466
$1.0E-10$	$-5.078064273^{48}_{50}E+01$	$9.3E-11$	135	41	16803
$1.0E-12$	$-5.0780642734931^{}_{4}E+01$	$1.6E-12$	161	56	24854

Formel (4.16)					
$\lambda = 0.25, \mu = 0.75$					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$-3.1^{61}_{70}E+00$	$8.2E-03$	34	13	3466
$1.0E-04$	$-3.165^{3}_{5}E+00$	$9.2E-05$	44	11	5672
$1.0E-06$	$-3.1653^{89}_{91}E+00$	$9.6E-07$	69	17	8667
$1.0E-08$	$-3.1653899^{2}_{4}E+00$	$1.0E-08$	110	36	16226
$1.0E-10$	$-3.165389927^{7}_{9}E+00$	$1.2E-10$	146	48	25205

Formel (4.17)					
$\lambda = 0.25, \mu = 0.75$					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$-3.1^{61}_{70}E+00$	$8.5E-03$	29	10	3194
$1.0E-04$	$-3.165^{3}_{5}E+00$	$8.7E-05$	43	10	5554
$1.0E-06$	$-3.1653^{89}_{91}E+00$	$9.9E-07$	63	17	8937
$1.0E-08$	$-3.1653899^{2}_{4}E+00$	$1.0E-08$	110	36	16442
$1.0E-10$	$-3.165389927^{7}_{9}E+00$	$1.2E-10$	143	51	25337

Stark oszillierende Funktion

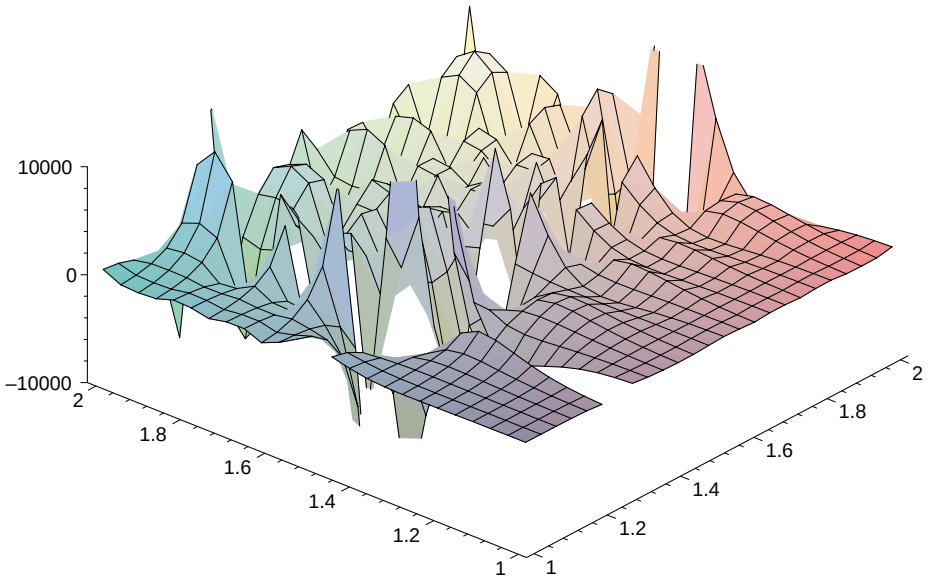


Abbildung 4.10: $h(x) = \frac{\sin(e^{x^2})\sin(e^{y^2})\,e^{x^2+y^2}}{(x-1.25)(y-1.5)}$

$$I(f;\ \lambda,\mu) = \oint_1^2 \oint_1^2 \frac{\sin(e^{x^2})\sin(e^{y^2})e^{x^2+y^2}}{(x-\lambda)(y-\mu)}\,dy\,dx$$

Der Integrand f ist sehr stark oszillierend. Die partiellen Ableitungen wachsen für große $x-$ und $y-$Werte rasant an. Dies führt zu Schwierigkeiten bei der Bestimmung der Taylorkoeffizienten von f bzw. h. Für $\lambda = 1.25$ und $\mu = 1.5$ sind die Werte unten tabelliert. Es zeigt sich, daß das Verfahren bei der Verwendung von Formel (4.17) weniger Teilbereiche und Funktionsauswertungen benötigt. Die Aufteilung des Integrationsbereiches für ϵ_{abs} ist in Abbildung 4.11 skizziert.

Formel (4.16)					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$-7.62\underset{5}{2}E+01$	$9.0E-03$	56	22	6620
$1.0E-04$	$-7.6237\underset{1}{0}E+01$	$9.0E-05$	81	33	10926
$1.0E-06$	$-7.623705\underset{6}{4}E+01$	$1.0E-06$	114	47	18117
$1.0E-08$	$-7.62370546\underset{8}{6}E+01$	$1.0E-08$	192	83	29568
$1.0E-10$	$-7.623705467\underset{12}{07}E+01$	$4.5E-10$	270	118	44667

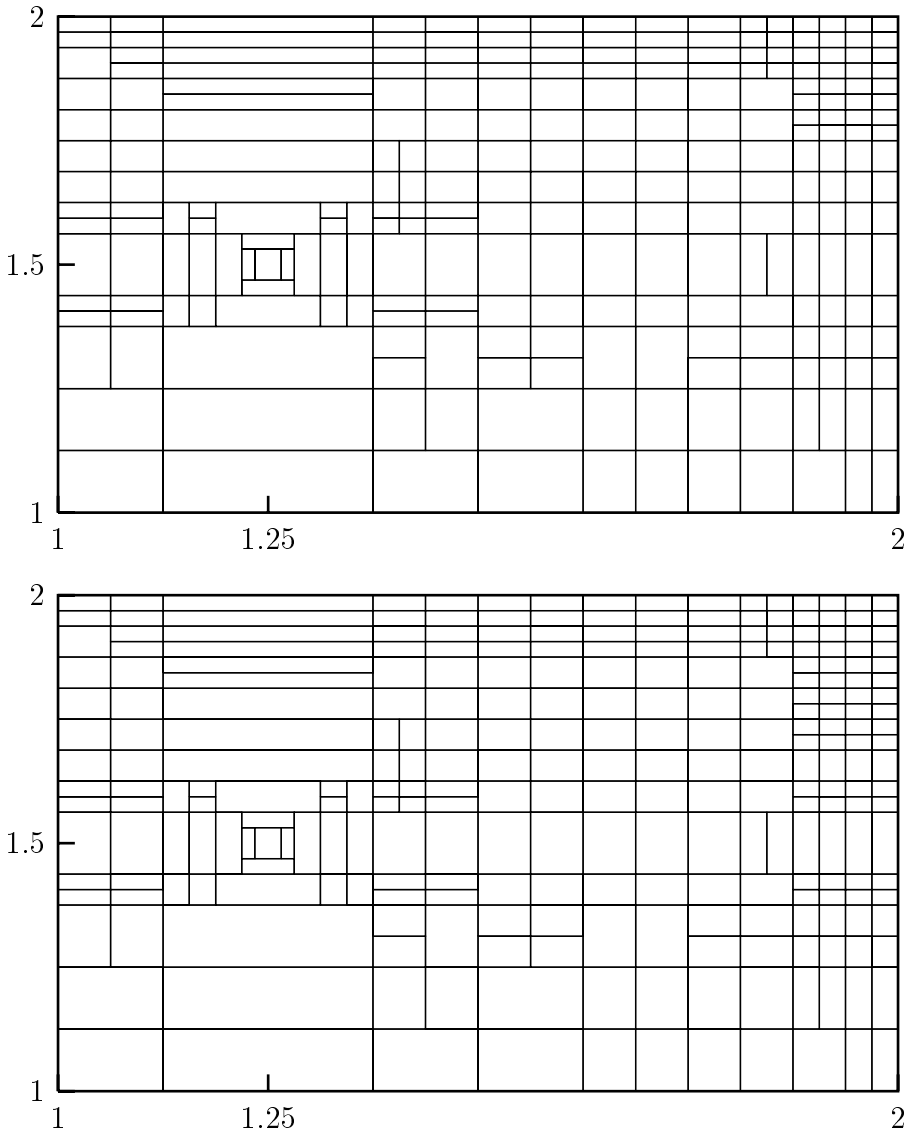


Abbildung 4.11: Zerlegung für $\epsilon_{abs} = 1E-10$ unter Verwendung von Formel (4.17) (oben) bzw. Formel (4.16) (unten)

Formel (4.17)					
ϵ_{abs}	$[I]$	$d([I])$	$\#R$	$\#\tilde{R}$	$\#f$
$1.0E-02$	$-7.62\overset{2}{5}E+01$	$9.5E-03$	55	21	6298
$1.0E-04$	$-7.6237\overset{0}{2}E+01$	$9.8E-05$	75	31	10353
$1.0E-06$	$-7.623705\overset{4}{6}E+01$	$9.9E-07$	109	44	17233
$1.0E-08$	$-7.62370546\overset{6}{8}E+01$	$9.9E-09$	187	82	28360
$1.0E-10$	$-7.623705467\overset{07}{12}E+01$	$4.5E-10$	258	118	43893

Kapitel 5

Die Klassenbibliothek CLAVIS

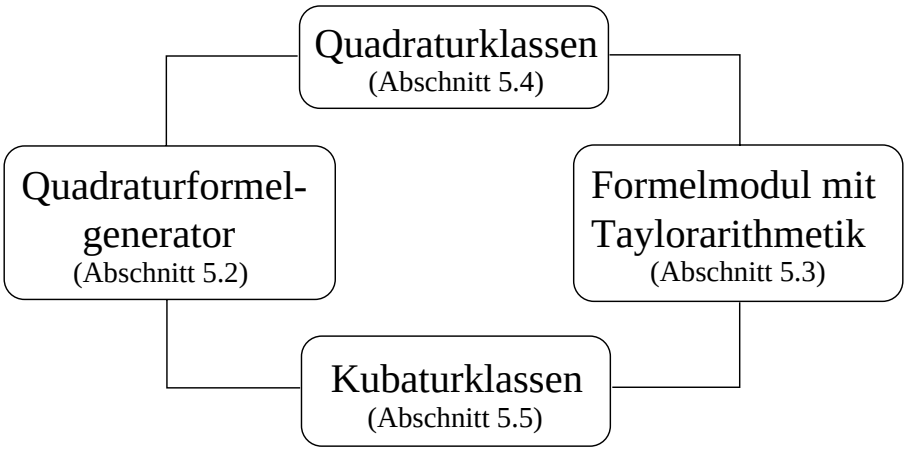
5.1 Überblick

CLAVIS (**Cla**sses for **V**erified **I**ntegration over **S**ingularities) ist eine in C++ unter Verwendung von C–XSC entwickelte numerische Klassenbibliothek zur verifizierten Integration nicht singulärer, schwach singulärer und stark singulärer Integrale. Sie wurde auf einem Intel PC mit Linux Betriebssystem und dem Gnu C++-Compiler egcs-2.91.66 (egcs-1.1.2 release) entwickelt, ist aber prinzipiell auf jede Plattform portierbar, vorausgesetzt es steht ein standardkonformer C++-Compiler und eine vorübersetzte Version der am hiesigen Institut erstellten C++-Bibliothek C–XSC zur Verfügung.

Die Programmentwicklungsumgebung C–XSC stellt ein Vielzahl vordefinierter abstrakter Datentypen, Operatoren und Funktionen zur Verfügung, die zur Entwicklung von numerischen Algorithmen mit hoher Genauigkeit und automatischer Ergebnisverifikation unverzichtbar sind. Die Datentypen sind als C++-Klassen implementiert und können direkt in C++-Programme eingebunden werden. Eine detaillierte Beschreibung von C–XSC wird in [44, 31] gegeben. Nach der Verabschiedung des endgültigen ANSI/ISO C++ Standards [40] wurde C-XSC neu überarbeitet und ist jetzt als freie Software unter den Bedingungen der GNU Lesser General Public License erhältlich.

CLAVIS besitzt eine einfache intuitive Benutzerschnittstelle, die es auch einem nicht mit einer objektorientierten Sprache vertrauten Anwender erlaubt, die zur Verfügung gestellten Verfahren zur Bestimmung singulärer Integrale einzusetzen. Die bereitgestellten Integrationsverfahren sind in Form von Klassen in einer getrennt übersetzbaren Bibliothek implementiert und können ohne großen Aufwand in ein bestehendes C++-Programm mit eingebunden werden. Damit stehen dem Benutzer von CLAVIS alle Möglichkeiten einer mächtigen objektorientierten Programmiersprache wie C++ zur Seite. Außerdem enthält die Klassenbibliothek noch Werkzeuge, die es dem Benutzer ermöglichen, Quadraturformel zu bestimmen und Verfahren für eigene Integrationsprobleme zu erstellen.
CLAVIS läßt sich grob in folgende vier Bestandteile zerlegen, auf die wir in den nächsten

Abschnitten im einzelnen eingehen wollen.



- Einen Quadraturformelgenerator zur Berechnung der Knoten und Gewichte von Gauß-Quadraturformeln für allgemeine Gewichtsfunktionen. Außerdem können mit dem Generator die Peanokonstanten für Gauß-Quadraturformeln und Hauptwertformeln bestimmt werden.

- Ein Formelmodul, das die nötigen Operatoren und Funktionen zum Aufbau der zu integrierenden Funktion enthält. Es können sowohl Integranden einer reellen als auch zweier reeller Veränderlichen gebildet werden. Mit Hilfe der integrierten Taylorarithmetik können die partiellen Ableitungen des Integranden berechnet werden.

- Einen Quadraturteil, der die zur verifizierten Bestimmung von Riemann-Integralen, Cauchy-Hauptwerten und Hadamard-Integralen benötigten Integrations-Klassen enthält.

- Einen Kubaturteil, der die zur verifizierten Bestimmung von doppelt iterierten Riemann-Integralen, doppelt iterierten Cauchy-Hauptwerten und von Integralen vom Typ Cauchy x Riemann benötigten Klassen enthält.

5.2 Der Quadraturformelgenerator

Der Quadraturformelgenerator stellt Hilfsmittel zur Berechnung und Ausgabe von Einschließungen der Knoten und Gewichte von Gauß-Quadraturformeln bereit. Außerdem können mit ihm Integrationskonstanten für Gauß-Quadraturformeln und Hauptwertformeln verifiziert bestimmt werden. Dazu werden die in Kapitel 2 und 3 besprochenen Verfahren eingesetzt. Die erforderlichen Berechnungen werden mit Hilfe einer Langzahl- bzw. Langzahlintervallarithmetik durchgeführt. An einem einfachen Beispiel wollen wir zunächst die Benutzung des Quadraturformelgenerators demonstrieren, bevor wir eine

Aufstellung der implementierten Klassen präsentieren. Das folgende Programm berechnet Einschließungen der Knoten und Gewichte der zehn-punktigen Gauß-Legendre Quadraturformel und gibt sie anschließend im Dezimalformat auf dem Bildschirm aus.

```
#include<iostream>
#include"node.h"

// --------------- legendre polynomials-----------------

int main(){
  legendre example1(10);
  example1.calc_nodes_and_weights();
  cout << Decimal << Nodes << example1;
  return 0;
}
```

Der Konstruktoraufruf `example1(10)` erzeugt ein Objekt der Klasse `legendre`, dessen Knoten und Gewichte in der folgenden Zeile durch den Aufruf einer Elementfunktion berechnet werden. Die Manipulatoren `Decimal` und `Nodes` bewirken, daß auf dem Bildschirm die Knoten im Dezimalformat ausgegeben werden. Das Programm zur Berechnung der Knoten und Gewichte zur logarithmischen Gewichtsfunktion (siehe Abschnitt 2.3.3) sieht ähnlich aus.

```
#include<iostream>
#include<fstream>
#include"node.h"

// --------------- weight ln(1/x) [0,1] ------------------

l_interval moments(int i){
// ordinary moments int_0^1 x^k ln(1/x) dx see (2.3.4)
  l_interval l_i(i);
  return 1 / sqr(1 + l_i);
}

int main(){
  ofstream output_file("ln10.hex");
  gautschi example2(10, 0, 1, moments);
  example2.calc_nodes_and_weights(3,3,30);
  output_file << Nodes << example2  << Weights << example2;
  return 0;
}
```

Zunächst wird ein Objekt der Klasse `gautschi` konstruiert, das mit der Anzahl der Stützstellen, dem Referenzintervall der Gewichtsfunktion (hier das Intervall $[0, 1]$) und einer Funktion `moments` vom Typ `init_function`,

```
typedef   l_interval  (*init_function)(int);
```

die die Momente berechnet, initialisiert wird. Die Argumente der Elementfunktion `calc_nodes_and_weights` legen die Anzahl der Bisektionsschritte und die Präzisionen der reellen und der intervallmäßig durchgeführten Berechnungen fest. Die Ausgabe der Stützstellen und Gewichte erfolgt standardmäßig im Hexadezimalformat in eine Datei. Außer diesen beiden Klassen stehen noch weitere zur Verfügung, die sich in der Berechnung der Rekursionskoeffizienten unterscheiden. Objekte der Klasse `legendre` bestimmen diese direkt über die bekannten Formeln, Objekte der Klasse `coeff` verwenden Funktionen vom Typ `init_function`, die als Argumente des Konstruktors übergeben werden. Die Objekte der Klassen `gautschi` und `sado` verwenden die einfachen bzw. modifizierten Momente, wie in Abschnitt 2.3.3 beschrieben. Für symmetrische Gewichtsfunktionen stehen separate Klassen bereit.

Klassen Qf-Generator	
Name	Argumente Konstruktor
legendre	int
coeff_symm	int, real, init_function, init_funcion, l_interval
coeff	int, real, real,init_function, init_funcion, init_funcion, l_interval
gautschi_symm	int, real, init_function
gautschi	int, real, real, init_function
sado_symm	int, real, init_function, init_function, init_function, bool = true
sado	int, real, real, init_function, init_function, init_function, init_function, bool = false

Treten bei der Berechnung der Stützstellen oder Gewichte Fehler auf, die auf eine zu gering gewählte Präzision bei vorausgegangenen Teilberechnungen zurückzuführen sind, so werden Ausnahmen (exceptions) geworfen, die von einem Fehlerhandler aufgefangen werden können, der die Teilberechnungen noch einmal von vorne mit einer größeren Präzision ausführt. Dadurch ist das Programm in der Lage, die Parameter für die Berechnung der Größen dynamisch zu erhöhen, um auch bei zu klein gewählten Startwerten für die Präzisionen oder die Anzahl der Bisektionsschritte noch zu einem Ergebnis zu kommen.

Ausnahmen Qf-Generator	
Name	Bedeutung
`newton_error`	Division durch Null im Newtonverfahren
`weight_error`	Division durch Null bei der Gewichtsberechnung
`verify_error`	0: Nullstelle nicht im Intervall (falsifiziert) 1: Mehrere Nullstellen im Intervall 2: Nicht entscheidbar
`qr_error`	Division durch Null bei der QR Zerlegung
`invert_error`	Matrix konnte nicht invertiert werden
`incomplete_error`	Benutzerfehler
`fatal_error`	Tritt nicht auf

Standardmäßig werden die Knoten im Hexadezimalformat ausgegeben. Die Ausgabe kann jedoch durch folgende Manipulatoren beeinflußt werden.

Ausgabe: Manipulatoren	
Name	Bedeutung
Decimal	Ausgabe in Dezimalform
Hexa	Ausgabe im Hexadezimalformat
Long	Ausgabe im Langzahlformat
Nodes	Ausgabe der Knoten
Weights	Ausgabe der Gewichte

Zur Berechnung der Integrationskonstanten stehen die Klassen `peano`, `peano_weight` und `peano_cauchy` zur Verfügung, die in ähnlicher Weise wie die zuvor besprochenen Klassen verwendet werden können, was folgendes Beispiel verdeutlichen soll.

```
#include"peano.h"
#include"peano.tpl"

int main(){
  peano_weight<l_interval> example3("sqrt5.long");
  example3.calculate(10);
  cout << Decimal << example3 << endl;
  return 0;
}
```

Die Knoten und Gewichte der 5-punktigen Quadraturformel (2.65) werden dem Konstruktor der template-Klasse `peano_weight` in einer Datei mit dem Namen `sqrt5.long` übergeben. Da in diesem Beispiel die Auswertung der Kernfunktion numerische Schwierigkeiten bereitet (siehe Abschnitt 2.4.1), wird sie mit Langzahlarithmetik durchgeführt. Die Knoten und Gewichte können deshalb abhängig vom template-Parameter entweder im Hexadezimal- oder im Langzahlformat importiert werden.
Der Aufruf `example3.calculate(10)` berechnet die Integrationskonstanten der Ordnung 10. Die Ausgabe der Integrationskonstanten kann, wie in den Beispielen zuvor, mit den Manipulatoren `Hexa` und `Decimal` beeinflußt werden.

5.3 Das Taylormodul f_ari

Das Taylormodul `f_ari` bietet dem Benutzer die Möglichkeit, eine zu integrierende Funktion einfach und intuitiv in den Programmtext einzugeben. Realisiert wird dies durch die zwei Klassen `operand` und `integrand`. Die Aufgabe der Klasse `operand` ist es, zur Ausführungszeit eine Baumstruktur aufzubauen, die eine schnelle Berechnung der Funktionswerte und partiellen Ableitungen der repräsentierten Funktion ermöglicht. Die Funktion darf sich aus dem unären Minus, den binären Operatoren `+,-,*,/` und den Standardfunktionen der folgenden Tabelle zusammensetzen.

Operandenfunktionen			
sqr	sqrt	pow	power
exp	ln		
sin	cos	tan	cot
asin	acos	atan	acot
sinh	cosh	tanh	coth
asinh	acosh	atanh	acoth

Die binären Operatoren dürfen auch in gemischter Form mit einem Operanden des Typs `real` oder `interval` verwendet werden. Mit Hilfe von Zuweisungen an Hilfsvariablen vom Typ `operand` können mehrfach auftretende Teilausdrücke ersetzt und damit Rechenzeit gespart werden, da die ersetzten Teilausdrücke insgesamt nur einmal ausgewertet werden müssen. Die Operatoren der Klasse `operand` dienen nur zum Aufbau der Baumstruktur. Der eigentliche Datentyp zur Repräsentation mathematischer Funktionen ist die Klasse `integrand`, die einen Zeiger auf ein Objekt vom Typ `operand`, den Kopf des Formelbaums, enthält. Der Gebrauch der Klasse läßt sich am einfachsten an einem kurzen Programmfragment erklären, das Einschließungen von partiellen Ableitungen der Funktion $f(x, y) = \sinh(xy)$ an der Stelle $(1, 2)$ berechnet.

```
integrand f(2,2);
f = sinh(x*y);
cout << f.taylor(2,1,interval(1, 1), interval(2,2)) << endl;
```

Die Objekte `x` und `y` vom Typ `operand` repräsentieren zwei unabhängige Variablen. Der Konstruktoraufruf der ersten Zeile stellt den Speicherplatz für die Taylorkoeffizienten bereit. Sollen Taylokoeffizienten höherer Ordnung als ursprünglich geplant berechnet werden, kann nachträglich mit einer Hilfsfunktion der zu Verfügung gestellte Speicherplatz vergrößert werden. Die Zuweisung der nächsten Zeile sorgt dafür, daß dem Objekt `f` ein Zeiger auf eine vollständige Kopie der rechten Seite zugewiesen wird. Zum Schluß wird eine Einschließung der partiellen Ableitung $\frac{\partial^3 f}{\partial x^2 \partial y}$ an der Stelle $(1, 2)$ ausgegeben. Neben den hier beschriebenen stehen noch eine Reihe weiterer Operatoren und Funktionen zur Verfügung, deren Bedeutung in der Headerdatei `f_ari.h` erklärt wird.

5.4 Klassen für die eindimensionale Integration

5.4.1 Benutzerschnittstelle

CLAVIS bietet die in Tabelle 5.1 aufgeführten Klassen zur verifizierten Bestimmung nicht singulärer, schwach singulärer oder stark singulärer Integrale an. Diese können in einfacher Weise – wie im nächsten Beispiel gezeigt – eingesetzt werden. Das folgende Programm berechnet eine Einschließung des Fourierkoeffizienten α_{20} der Funktion aus Abschnitt 2.6.4 mit einem Restgliedurchmesser, der kleiner als 10^{-10} ist[1]

[1]Genau genommen wird die zu 10^{-10} nächstgelegene Maschinenzahl als oberen Schranke für den Durchmesser des Restglieds verwendet, da 10^{-10} im endlichen Binärformat nicht exakt darstellbar ist.

Quadraturklassen	
Name	Argumente Konstruktor
integral	const integrand&
sqrt_integral	const integrand&
cauchy_integral	const integrand&, real lambda
hadamard_integral	const integrand&, real lambda

Abbildung 5.1: Quadraturklassen

```
#include"quadrature.h"

int main() {
  real r = 0.875, eps = 1e-10;
  int  k = 20;
  interval Pi = 4 * atan(interval(1,1));
  integrand f;

  f = 1/ Pi  * (1 - r * cos(x))/( 1 - 2*r* cos(x) + sqr(r) )
      * cos(k*x);
  integral example(f);
  example.integrate(0,Pi,eps);
  cout <<   SetPrecision(15,9) << example << endl;
  return 0;
}
```

Die Ausgabe des Programms sieht wie folgt aus.

```
number of intervals : 12
#f                  : 164
approximationsum    : [ 3.460437938E-002, 3.460437939E-002]
d(approximationsum) :  8.897049764E-014
remainder           : [-2.364451155E-011, 2.364451155E-011]
d(remainder)        :  4.728902309E-011
enclosure           : [ 3.460437936E-002, 3.460437942E-002]
d(enclosure)        :  4.737800430E-011
```

5.4.2 Implementierung und Erweiterbarkeit

Ein wichtiger Aspekt beim Design der Quadraturklassen ist die Erweiterungsmöglichkeit. Wollen wir z. B. ein Integrationsverfahren für eine neue Gewichtsfunktion implementieren, so muß gewährleistet sein, daß die schon vorhandenen Methoden möglichst einfach weiterverwendet werden können, ohne nachträglich verändert werden zu müssen. Dies wird durch Einführen einer abstrakten Basisklasse `base_interval` und den von ihr abgeleiteten Klassen erreicht. Integrale über Teilbereiche werden durch Objekte

der abgeleiteten Klassen der Basisklasse `base_interval` dargestellt. Diese enthalten den Integrationsbereich, Einschließungen des Restglieds und der Quadratursumme, die Angabe der verwendeten Quadraturformel und der verwendeten Restglieddarstellung, eine Referenz auf den Integranden sowie die Funktionen `add_quadraturesum` zur Berechnung der Quadratursumme und `subdivide` zur Zerlegung des Integrationsbereiches und Bestimmung des Restglieds.

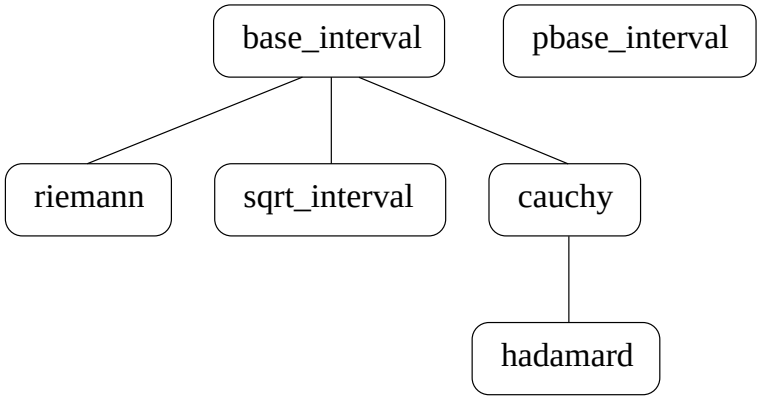
```
class base_interval{
  // constructors
  .....
 protected:
  real dremainder;
  real inf, sup;
  real r_inf, r_sup;
  interval quadsum;
  bool quad;
  int  n, m;
  integral& ri;

  virtual void add_quadraturesum() = 0;
  virtual void subdivide() = 0;
};
```

Die Klasse `base_interval` ist eine abstrakte Klasse. Die Erzeugung einer Instanz ist nicht erforderlich und auch nicht möglich, da mit ihr nur ein Konzept festgelegt wird, dessen Ausprägungen in den abgeleiteten Klassen spezifiziert wird. Objekte vom Typ `base_interval` treten nicht als eigenständige Objekte, sondern nur als Teilobjekte in Objekten abgeleiteter Klassen auf. Die rein virtuellen Funktionen `subdivide` und `add_quadraturesum` werden nicht für die Basisklasse definiert, sondern müssen für jede abgeleitete Klasse gesondert definiert werden. Dies ist deswegen sinnvoll, weil z. B. die Zerlegung des Integrationsbereiches eines Cauchy-Hauptwert-Integrals nicht mit der eines Riemann-Integrals übereinstimmt. Der entscheidende Vorteil dieses virtuellen Mechanismus ist, daß die bei einer Zerlegung auftretenden Teilintervalle als Objekte der abgeleiteten Klasse über Zeiger auf die Basisklasse `base_interval` referenziert und die virtuellen Funktionen `subdivide` und `add_quadraturesum` bequem über diese Zeiger aufgerufen werden können. Zum Übersetzungszeitpunkt ist es unerheblich, welchen Typ das Objekt besitzt, auf das der Zeiger verweist. Wichtig ist nur, daß für alle abgeleiteten Klassen die virtuellen Funktionen definiert sind. Erst zur Laufzeit wird der Aufruf einer virtuellen Funktion abhängig vom Typ des Objektes, das gerade referenziert wird, interpretiert. Diese Möglichkeit, mit einem Funktionsaufruf eine Fülle von verschiedenen Funktionen gleichen Typs aus einer Vererbungsstruktur auszuwählen, wird auch Polymorphismus genannt (siehe [76, 83]). In unserem Fall heißt dies, daß wir eine Elementfunktion `gaa()` (siehe weiter unten) realisieren können, die ohne Kenntnisse der Implementierungsdetails der abgeleiteten Klassen (u. a. der Funktion `subdivide`)

die adaptive Zerlegung des Ausgangsintervalls vornimmt. Der Zeiger `old_range` kann beliebige Objekte einer abgeleiteten Klasse von `base_interval` referenzieren. Beim Aufruf von `old_range->subdivide` wird der dynamische Typ, d. h. der Typ des Objektes, auf das `old_range` verweist, bestimmt und die entsprechende Implementierung von `subdivide` aufgerufen. Der mit polymorphen Klassen verbundene zusätzliche Aufwand spielt hier keine Rolle, da bei den meisten Compilern virtuelle Funktionen über indirekte Funktionsaufrufe realisiert werden und der zeitliche Mehraufwand für einen indirekten Funktionsaufruf verglichen mit dem Aufwand der Bestimmung der Zerlegung eines Teilbereichs verschwindend gering ist. Die Berechnung des Approximationsterms kann auf die gleiche Weise mit Hilfe der virtuellen Funktion `add_quadraturesum` realisiert werden. Diese Polymorphie der Klassen für die Integrationsbereiche erlaubt uns, die Implementierung von Funktionen auf einem abstrakteren Niveau, die eine einfache Ergänzung bestehender Integrationstypen garantieren, da zusätzliche Klassen für neue Integrationstypen abgeleitet werden können, ohne daß zentrale Funktionen wie z. B. `gaa` verändert werden müssen.

```
void  integral::gaa(){    // global adaptive algorithm
  base_interval* old_range;
  while( (Rsup - Rinf) > eps_abs && heapsize < max_heapsize ){
    old_range = pop();
    old_range->subdivide();
  }
  .....
}
```

Zur Zeit sind folgende Klassen für Teilbereiche des Integrationsintervalls implementiert.



Die Verwaltung der Integrationsintervalle findet mit Hilfe eines Heaps statt. Ein Heap ist eine Sequenz, deren Elemente in einem Zustand gehalten werden, der es erleichtert, sie zu sortieren. Das erste Element der Sequenz ist das Element mit dem höchsten

Wert, in unserem Fall das Intervall mit dem größten Restglieddurchmesser. Die Operationen `push` und `pop` zum Einfügen und Löschen von Elementen benötigen lediglich logarithmischen Aufwand. Der Heap ist damit eine ideale Datenstrukur zur Verwaltung der Integrationsbereiche, wie Untersuchungen in [49] bestätigen. In C++ können Heaps mit den Heap-Algorithmen der Standardbibliothek, die die Heap-Operationen push_heap und pop_heap enthalten, implementiert werden. Zu diesem Zweck ist die Zeigerklasse `pbase_interval` implementiert, die einen Zeiger auf Objekte der Basisklasse `base_interval` enthält.

5.5 Klassen für die zweidimensionale Integration

Die Klassen für die Bestimmung mehrfach iterierter Integrale, im einzelnen sind dies die Klasse `integral` zur Bestimmung doppelt iterierter Riemann-Integrale, die Klasse `cauchyxriemann_integral` zur Bestimmung von Integralen des Typs Cauchy x Riemann und `cauchy_integral` zur Bestimmung doppelt iterierter Cauchy-Hauptwerte bieten die gleiche Funktionalität wie die zuvor besprochenen Quadraturklassen. Sie können ähnlich wie diese verwendet werden, wie das folgende kurze Programm zur Berechnung des doppelt iterierten Cauchy-Hauptwertes aus Abschnitt 4.4.3 zeigt.

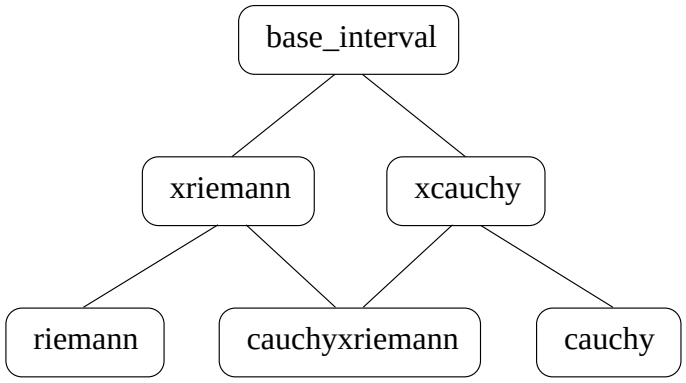
```
#include<iostream>
#include"cubature.h"

int main(){

  operand r, q;
  integrand f;

  q = exp(sqr(x));
  r = exp(sqr(y));

  f = sin(q) * q * sin(r) * r;

  cauchy_integral a(f, 1.25, 1.5);
  a.integrate(1, 2, 1, 2, 1e-10);
  cout << a << endl;

  return 0;
}
```

Die Hilfsvariablen `r` und `q` bewirken, daß die zugewiesenen Teilausdrücke bei der Auswertung der Funktion und Berechnung der partiellen Ableitungen nur einmal ausgewertet werden. Der Konstruktor initialisiert neben dem Integranden die Werte der singulären Stellen, ansonsten ist der Programmtext selbsterklärend. Die Ausgabe des

Programms enthält zuätzlich noch die Anzahl der Teilintervalle, die zur Bestimmung der Zerlegung benötigt werden, jedoch nicht in die Berechnung der Quadratursumme oder des Restglieds mit einfließen.

```
number of intervals: 258 (118)
#f                 : 43893

approximationsum   : [-7.6237054671106450E+001,-7.6237054670757345E+001]
d(approximationsum): 3.4910385693365E-010

remainder:         : [-4.9377684935693121E-011,4.9224237854969806E-011]
d(remainder)       : 9.8601922790663E-011

enclosure:         : [-7.6237054671155832E+001,-7.6237054670708118E+001]
d(enclosure)       : 4.4771297780244E-010
```

Die Integrationsbereiche sind nach dem gleichen Prinzip wie im eindimensionalen Fall implementiert. Folgende Klassen stehen zur Verfügung.



Die virutellen Klassen `xriemann` und `xcauchy` dienen nicht nur der besseren Strukturierung, sondern werden auch bei der Berechnung von Integralen über Randbereiche (siehe auch Abschnitt 4.4.2) benötigt.

Zum Schluß zeigen wir noch, wie mit Hilfe der Klasse `integral` eine Klasse zur Bestimmung von Integralen über Kreisringen gebildet werden kann.
Das Integral $\int_D f(s,t)\,d(s,t)$ mit

$$D = D(r_1, r_2, \phi_1, \phi_2) = \{(r\cos\phi, r\sin\phi) | r_1 \le r \le r_2, \phi_1 \le \phi \le \phi_2\}$$

kann bekannterweise durch die Substitutionsregel in ein Riemann-Integral über einen Rechteckbereich überführt werden

$$\int_D f(s,t)\,d(s,t) = \int_{r_1}^{r_2}\int_{\phi_1}^{\phi_2} r\, f(r\,\cos(\phi), r\,\sin(\phi))\,d\phi\,dr.$$

In der Headerdatei `polar.h`

```
#include"cubature.h"

extern operand s,t;

class polar_integral: public integral{
 public:
  polar_integral(integrand& f);
};
```

und der dazugehörigen Datei `polar.C`

```
#include"polar.h"

operand s(x * cos(y)),t(x * sin(y));

polar_integral::polar_integral(integrand& f):
   integral(f * x){}
```

wird eine von `integral` abgeleitete Klasse `polar_integral` definiert, mit der im folgenden Hauptprogramm das Integral der Funktion $f(s,t) = st$ über den Teil des Einheitskreises, der im ersten Quadranten liegt, berechnet wird.

```
#include"polar.h"

int main(){
 integrand f;
 interval Pi = 4.0 * atan(interval(1,1));
 f = s * t;
 polar_integral a(f);
 a.integrate(0, 1, 0, Pi/2, 1e-10);
 // r = 0..1, phi = 0..Pi/2, eps_abs = 1e-10
 cout << a << endl;
 return 0;
}
```

Die Ausgabe des Programms sieht wie folgt aus.

```
number of intervals: 2 (0)
#f                 : 72

approximationsum   : [1.2499999999999E-001,1.2500000000001E-001]
d(approximationsum): 1.1379786002408E-015

remainder:         : [0.0000000000000E+000,2.4004476060292E-015]
```

```
d(remainder)        : 2.4004476060292E-015

enclosure:          : [1.2499999999999E-001,1.2500000000001E-001]
d(enclosure)        : 3.5527136788005E-015
```

Sollen Klassen dieses Abschnittes zusammen mit Objekten der Quadraturklassen benutzt werden, so können auftretende Namenskonflikte durch die Einführung von Namensbereichen (siehe [83]) behoben werden.

Zum Schluß sei noch darauf hingewiesen, daß dieses Kapitel keinen Anspruch auf Vollständigkeit erhebt, sondern lediglich einen Überblick über die vorhandenen Möglichkeiten und den Gebrauch der C++-Bibliothek CLAVIS vermitteln soll. Zusätzlich zu den hier beschriebenen Funktionen existieren noch weitere, die in den entsprechenden Headerdateien dokumentiert sind.

Kapitel 6

Zusammenfassung und Ausblick

Die in dieser Arbeit vorgestellten Verfahren ermöglichen die effiziente Bestimmung schwach und stark singulärer Integrale mit automatischer Fehlerkontrolle. Hierfür werden Gauß-Quadraturformeln in Verbindung mit einer global adaptiven Strategie eingesetzt.
Bei der Entwicklung von verifizierenden Integrationsverfahren stellt sich grundsätzlich die Frage, welche Quadraturformeln am besten für diesen Zweck geeignet sind. Entscheidend hierbei ist neben der numerischen Auswertbarkeit der Approximationssumme vor allem eine möglichst enge und einfach berechenbare Einschließung des Verfahrensfehlers. In [42] wird zur Einschließung von nicht singulären (eindimensionalen) Integralen das Romberg-Verfahren verwendet, das die in dieser Arbeit vorgestellten Testprobleme deutlich schneller berechnet als die Methode mittels Taylorreihenansatz. Beim Vergleich eines adaptiven Romberg-Verfahrens mit einem adaptiven Gauß-Verfahren in [45] erweist sich das Gauß-Verfahren im Hinblick auf die benötigte Rechenzeit deutlich überlegen. Der Vorteil der Romberg-Verfahren liegt in der einfachen Berechnung der Stützstellen und Gewichte und in der Wiederverwendbarkeit der Stützstellen. Gauß-Quadraturformeln bieten dagegen maximalen Exaktheitsgrad und sehr niedrige Integrationskonstanten. Dies sind die Gründe, warum in neueren Veröffentlichungen ([80],[85] oder [9]) Gauß-Quadraturformeln den Vorzug erhalten.[1]
Mit diesen ist auch die numerische Berechnung von Integralen mit Gewichtsfunktionen möglich. Dazu müssen die Stützstellen, Gewichte und Integrationskonstanten der Quadraturformeln im voraus berechnet werden, da deren Berechnung aufwendiger als das eigentliche Integrationsproblem sein kann. Ein Verfahren für die verifizierte Bestimmung dieser Daten wird in Kapitel 2 angegeben.
Der Aufwand der in dieser Arbeit vorgestellten Integrationsverfahren wird durch eine global adaptive Strategie, bei der außer der Breite des Integrationsintervalls die Ordnung des Restglieds und die Anzahl der Stützstellen variiert werden, erheblich reduziert.

[1] In [9] werden neben den Gauß-Quadraturformeln noch die Clenshaw-Curtis-Formeln in die engere Wahl miteinbezogen. Da bei gleicher Stützstellenanzahl und Ordnung die relevanten Peano-Konstanten beim Clenshaw-Curtis-Verfahren größer als bei der Gauss-Quadratur sind, haben wir uns in dieser Arbeit ausschließlich auf Gauß-Quadraturformeln konzentriert.

Voraussetzung für eine variable Ordnung des Restglieds bei fester Stützstellenanzahl ist die Einschließung der Integrationskonstanten. So erlaubt die verifizierte Bestimmung der Integrationskonstanten der Hunter-Formeln (siehe Kapitel 3) die Angabe eines adaptiven Verfahrens, das mit deutlich weniger Funktionsauswertungen und Teilbereichen auskommt. Die numerischen Beispiele am Ende des Kapitel 3 verdeutlichen, daß sich hierdurch die Geschwindigkeit der Berechnung teilweise mehr als verdreifacht. Ein Blick auf die Ergebnisse der Kapitel 2 und 3 zeigt aber auch, daß die bloße Angabe der Anzahl der benötigten Funktionsauswertungen oder Restgliedberechnungen nur im sehr begrenzten Maße Rückschlüsse auf die tatsächlich benötigte Rechenzeit zuläßt.
Die in Kapitel 3 konstruierten Quadraturformeln stimmen, falls die Kollokationsstelle eine Nullstelle der Funktion zweiter Art ist, mit den in [38] angegebenen überein. Mit Hilfe der hergeleiteten Restglieddarstellungen der Quadraturformeln ist die Einschliessung stark singulärer Integrale, deren Integranden eine doppelte Polstelle (doppelte Nullstelle im Nenner) im Inneren des Integrationsintervalls besitzen, auf ähnliche Weise wie bei den Cauchy-Hauptwerten möglich.
Die Übertragung der adaptiven Strategie auf den mehrdimensionalen Fall ermöglicht die Einschließung mehrfach geschachtelter singulärer Integrale. Für die Berechnung der bei der Zerlegung auftretenden Teilintegrale werden Produktformeln verwendet und die verschiedenen Möglichkeiten der Restgliedeinschließung untersucht. Die Tests am Ende des Kapitels 4 belegen, daß mit dem angegebenen Verfahren auch komplizierte Integrale berechnet werden können.

Zur numerischen Integration existieren umfangreiche Softwarepakete.[2] Das bekannteste unter ihnen ist QUADPACK [70], das zahlreiche in Fortran geschriebene Routinen zur numerischen Bestimmung eindimensionaler Integrale enthält. Viele Programmbibliotheken (z. B. NAG [66] oder IMSL [37]) enthalten nur leicht veränderte oder direkt aus QUADPACK übernommene Routinen. Zweidimensionale Integrale können mit Hilfe von CUBPACK++ [13], einer in C++ geschriebenen Bibliothek, berechnet werden. CUBPACK++ stellt eine Vielzahl verschiedener Bereiche (u. a. Rechtecke, Parallelogramme, Dreiecke oder Kreisscheiben), über die integriert werden kann, zur Verfügung, ist aber nicht für die numerische Integration von singulären Funktionen geeignet.[3] Vergleichbare Programmpakete mit Einschließungsmethoden existieren zur Zeit nicht. Vorhandene Integrationsroutinen mit automatischer Ergebnisverifikation sind oft auf spezielle Integrationsprobleme zugeschnitten, besitzen keine einheitlichen Schnittstellen oder sind in Programmiersprachen geschrieben, die den Anforderungen eines modernen Softwaredesigns nicht mehr gerecht werden.

Die in dieser Arbeit besprochenen Verfahren sind in C++ unter Verwendung von C–XSC in Form einer Klassenbibliothek implementiert. Dies garantiert direkten Zugriff auf eine Vielzahl existierender C bzw. C++ Bibliotheken. Aber auch das Einbinden

[2] Ein Überblick über bestehende Softwarepakete und Programmbibliotheken wird in [49] gegeben.

[3] Keines der numerischen Beispiele des Kapitels 4, auch das iterierte Riemann Integral aus Abschnitt 4.4.1 nicht, konnten von CUBPACK++ bestimmt werden.

von Routinen aus Softwarepaketen, die nicht in C++ erstellt sind, ist möglich. So kann man z. B. mit den FORTRAN-Routinen aus ORTHPOL [27] zunächst Näherungen der Stützstellen einer Quadraturformel bestimmen und aus diesen anschließend, wie in Abschnitt 2.3.4 erwähnt, Einschließungen ermitteln. Bei der Realisierung der Quadratur- und Kubaturklassen wurde besonderer Wert auf eine einfache Erweiterbarkeit der Klassenbibliothek gelegt, da für jede Gewichtsfunktion eigene Klassen angelegt werden müssen. Für den praktischen Gebrauch wäre eine Ergänzung der bestehenden Integralklassen von CLAVIS, um ein möglichst großes Spektrum von Gewichtsfunktionen (vergleichbar mit dem von QUADPACK) abzudecken, wünschenswert. Diese Aufgabe wird durch die Vererbungsmechanismen von C++ programmtechnisch, wie in Kapitel 5 ausgeführt, erheblich vereinfacht. Ein Parser für eine interaktive Eingabe des Integranden und einer automatischen Erkennung der Gewichtsfunktionen könnte den Benutzerkomfort zusätzlich steigern. Vom theoretischen und praktischen Standpunkt wäre ein Vergleich des Aufwands der hier verwendeten Methode mit dem bei der Verwendung einer komplexen Restgliedabschätzung, wie sie in [77] gegeben wird, interessant. Die Behandlung und Implementierung zusätzlicher mehrdimensionaler Integrationsbereiche, die sich besser als Rechteckbereiche für eine adaptive Aufteilung des Integrationsbereichs eignen, können zu einer höheren Flexibität und Effizienz der mehrdimensionalen Integrationsklassen beitragen.
Beim Vergleich des Laufzeitverhaltens von verifizierenden Integrationsverfahren mit den in QUADPACK oder CUBPACK++ verwendeten Verfahren (siehe auch [12]) schneiden erstere in der Regel schlechter ab, da die Berechnung des Verfahrensfehlers unter Berücksichtigung eventuell auftretender Rundungsfehler zusätzlichen Aufwand verursacht. Dafür bieten die Ergebnisse von Einschließungsverfahren eine ganz andere Qualität, da sie garantierte obere und untere Schranken des gesuchten Integralwertes liefern. Zwar enthält z. B. QUADPACK auch Routinen, die zusätzlich zu den Näherungen Fehlerabschätzungen mit ausgeben. Aus diesen lassen sich jedoch keine garantierte Schranken für das tatsächliche Ergebnis ableiten. Hinzu kommt noch, daß diese Schätzungen ebenso wie der Näherungswert mit Rundungsfehlern behaftet sind. Auf der anderen Seite stellen verifizierende Verfahren zusätzliche Voraussetzungen an die zu integrierende Funktion, die ihren Einsatz in bestimmten Situationen unmöglich machen. Aus diesen Gründen stellen die bisher entwickelten verifizierende Integrationsverfahren keinen Ersatz, sondern eine sinnvolle Ergänzung herkömmlicher Integrationsverfahren dar.

Literaturverzeichnis

[1] E. Adams, U. Kulisch (eds.), *Scientific Computing with Automatic Result Verification* , Academic Press, San Diego 1993 (ISBN 0-12-044210-8)

[2] G. Alefeld, J. Herzberger, *Einführung in die Intervallrechnung*, Reihe Informatik/12 B.I.-Wissenschaftsverlag

[3] V.I. Lebedev & O.V. Baburin, *Calculation of the principal value, weights and nodes of the Gauss quadrature formulae of integrals*, U.S.S.R. Comp. Math. and Phys., v.5, No. 3 (1965), pp. 81-92

[4] W. Barrett, *Convergence properties of Gaussian quadratures formulae*, Computer J., v. 3, (1960), pp. 272-277

[5] H. Brass, *Quadraturverfahren*, Vandenhoeck & Ruprecht in Göttingen, (1977)

[6] Jarle Berntsen et al., *On the subdivision strategie in adaptive quadrature algorithm*, J. of Comp. and Appl. Math.v. 35, (1991), pp. 119-132

[7] James L. Blue, *A Legendre Polynomial Integral*, Math. Comp., v. 33, (1979), pp. 739-741

[8] G. Booch, *Objektorientierte Analyse und Design*, Addison-Wesley 1994

[9] Chin-Yun Chen, *Adaptive numerische Quadratur und Kubatur mit automatischer Ergebnisverifikation*, Dissertation Universität Karlsruhe 1998

[10] C.W. Clenshaw, *A note on the summation of Chebyshev series*, Math. Comp., v. 9, (1955), pp. 118-120. MR 17, 194

[11] G.F. Corliss, *Computing Narrow Inclusions for Definite Integrals* , in [41], (1987), pp. 150-169

[12] G.F. Corliss, *Performance of self-validating adaptive quadrature* in 'Numerical integration' (P. Keast, G. Fairweather) NATO ASI Series, 1987

[13] R. Cools, D.P. Laurie, l. Pluym, *Algorithm 764: Cubpack++ - a C++-package for automatic two-dimensional cubature*, ACM Trans. Math Software 23 (1997), pp. 1-15

[14] G. Criscuolo,G. Mastroianni, *On the Uniform Convergence of Gaussian Quadrature Rules for Cauchy Principal Value Integrals*, Numer. Math., v. 54, (1989), pp. 445-461

[15] Ph.J. Davis, Ph. Rabinowitz, *Methods of Numerical Integration*, Academic Press, San Diego 1984

[16] K. Diethelm, *Numerische Approximation von Cauchy-Hauptwert-Integralen unter theoretischen und rechnerorientierten Aspekten*, Dissertation 1994 Universität Hildesheim

[17] M.C. Eiermann, *Automatic, guaranteed integration of analytic function*, BIT 29 (1989), pp. 270-282

[18] D. Elliott, D.F. Paget, *Gauss Type Quadrature Rules for Cauchy Principal Value Integrals*, Math. of Comp., v.33, (1979), pp.301-309

[19] D. Elliott, D.F. Paget, *On the Convergence of a Quadrature Rules for Evaluating Certain Cauchy Principal Value Integrals: An Addendum*, Numer. Math., v.25, (1976), pp.287-289

[20] G. Evans, *Practical Numerical Integration*, John Wiley & Sons 1993

[21] H. Fischer, *Schnelle automatische Differentiation, Einschließungsmethoden und Anwendungen*, Dissertation 1990 Universität Karlsruhe

[22] L. Gatteschi, *On Some Orthogonal Polynomial Integrals*, Math. Comp.,v. 35, (1980), pp.1291-1298

[23] W. Gautschi, *Construction of Gauss-Christoffel quadrature formulas*, Math. Comp., v. 22, (1968), pp.251-270, MR 37 # 3755

[24] W. Gautschi, *On the Construction of Gaussian Quadrature Rules from Modified Moments*, Math. Comp., v. 24, (1970), pp. 245-260

[25] W. Gautschi, *On the Preceding Paper "A Legendre Polynomial Integral" by James Blue*, Math. Comp., v. 33, (1979), pp. 742-743

[26] W. Gautschi, *Orthogonal Polynomials: applications and computation*, Acta Numerica (1996), pp. 45-119

[27] W. Gautschi, *Algoritm 726:ORTHPOL – A Package of Routines for Generating Orthogonal Polynomials and Gauss–Type Quadrature Rules*, ACM Transactions on Mathematical Software,Vol. 20, (1994), pp. 21-62

[28] A. Gienger, *Zur Lösungsverifikation bei Fredholmschen Integralgleichungen zweiter Art*, Dissertation Universität Karlsruhe 1997

[29] G. Golub & J.H. Welsch, *Calculation of Gauss quadrature rules*, Math. Comp., v. 23, (1969), pp. 221-230

[30] W. Hackbusch, *Integralgleichungen*, B. G. Teubner Stuttgart (1997)

[31] R. Hammer, M. Hocks, U. Kulisch, D. Ratz, *C++ Toolbox for Verified Computing*, Springer-Verlag Berlin Heidelberg 1995

[32] G. Hämmerlin, KH. Hoffmann, *Numerische Mathematik*, Grundwissen Mathematik 7, Springer Verlag 1989

[33] P. Hermle, Diplomarbeit Universität Karlsruhe 1993

[34] W. Hess, Diplomarbeit Universität Karlsruhe 1996

[35] H. Heuser, *Funktionalanalysis*, B.G. Teubner Stuttgart 1986

[36] D.B. Hunter, *Some Gauss-Type Formulas for the Evaluation of Cauchy Principal Values of Integrals*, Numer. Math., v. 19, (1972), pp. 419-424

[37] Visual Numerics Inc., *IMSL MATH/LIBRARY* Houston, TX, 1994

[38] N.I. Iokamidis, *On the uniform convergence of Gaussian quadrature rules for Cauchy principal values and their derivatives*, Math Comp. 44(1985), pp. 191-198

[39] E. Isaacson, H.B. Keller, *Analyse numerischer Verfahren*, Verlag Harri Deutsch, Frankfurt a.M. und Zürich 1973

[40] ISO/IEC 14882-1998, *Information Technology - Programming Languages - C++*

[41] E. Kaucher, U. Kulisch, Ch. Ullrich (Eds.): *Computerarithmetic: Scientific Computation and Programming Languages* , B.G. Teubner Verlag, Stuttgart, 1987

[42] R. Kelch, *Ein adaptives Verfahren zur Numerischen Quadratur mit automatischer Ergebnisverifikation* , Dissertation Universität Karlsruhe (TH) 1989

[43] R. Klatte, U. Kulisch, M. Neaga, D. Ratz, CH. Ullrich, *PASCAL-XSC – Language Reference with Examples* , Springer-Verlag, 1992

[44] R. Klatte, U. Kulisch, A. Wiethoff, C. Lawo, M. Rauch *C–XSC A C++ Class Library for Extended Scientific Computing*, Springer-Verlag 1993

[45] W. Krämer, S. Wedner *Two adaptive Gauss-Legendre type algorithms for the verified computation of definite integrals*, Reliable Computing 2(3), pp. 241-253, 1996

[46] W. Krämer, S. Wedner, *Computing Narrow Inclusions for Cauchy Principal Value Integrals*, in 'Scientific Computing and Validated Numerics' (G. Alefeld, A. Frommer, B. Lang, eds.), Mathematical Research, Volume 90, Akademie Verlag 1996

[47] W. Krämer, U. Kulisch, R. Lohner, *Numerical Toolbox for Verified Computing II*, Springer-Verlag, Berlin (Vorabdruck)

[48] G.S. Krenz, *Using weight functions in self-validating quadrature* in 'Numerical integration' (P. Keast, G. Fairweather) NATO ASI Series, 1987

[49] A.R. Krommer, C.W. Ueberhuber, *Computational Integration*, SIAM, Philadelphia, 1998

[50] U.W. Kulisch, *Grundlagen des numerischens Rechnens – Mathematische Begründung der Rechnerarithmetik* , Reihe Informatik, Band 19, B.I.-Wissenschaftsverlag, Mannheim, 1976

[51] U.W. Kulisch, W.L. Miranker (Eds.), *Computer Arithmetic in Theory in Practice* Academic Press New York 1981

[52] U.W. Kulisch (Hrsg), *Wissenschaftliches Rechnen mit Ergebnisverifikation*, Vieweg Braunschweig 1989

[53] E. Luik, *Fehlerabschäzungen bei Quadratur und Kubatur auf der Grundlage von Approximationsgraden*, Dissertation Universität Tübingen 1984

[54] R. Lohner, Habilitationsschrift Universität Karlsruhe 1994

[55] R. Lohner, *Interval Arithmetic in Staggered Correction Format*, in [1], pp. 301-321, 1993

[56] R. Lohner, Vortrag interne Tagung Oberjoch 1999

[57] R. Lohner, *Einschließungen bei Anfangs- und Randwertaufgaben* in [52]

[58] R. Lohner, *Einschließungen der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*, Dissertation Universität Karlsruhe (TH) 1988

[59] B. Mauch, *Verifizierte numerische Integration durch Einschließung der Restglieder von Quadraturformeln*, wissenschaftliche Arbeit zur Zulassung für das Lehramt an staatlichen Gymnasien, Universität Karlsruhe 1999

[60] G. Mayer, *Grundbegriffe der Intervallrechnung* in [52]

[61] G. Monegato, *The Numerical Evaluation of One-Dimensional Cauchy Principal Value Integral*, Computing, v. 29 (1982), pp. 337-354

[62] G. Monegato, *Numerical Evaluation of hypersingular integrals*, Journal of Computational an Applied Mathematics 50(1994), pp. 9 -31

[63] G. Monegato, *Convergence of product formulas for the numerical evaluation of certain two-dimensional Cauchy principal value integrals*,
Numer. Math. 43 (1984), pp. 161 - 173

[64] A. Neumaier, *Interval Methods for systems of equations* ,
Cambridge University Press New York 1990

[65] N.I. Muskhelishvili *Singular integral equations*, Groningen, Holland (1953)

[66] NAG Ltd., NAG Fortran Library Manual–Mark 17, Oxford UK 1996

[67] K. Niederdrenk, *Funktionen einer reellen Veränderlichen*,
Rechnerorientierte Ingenieurmathematik, Vieweg

[68] D.F. Paget, D. Elliott, *An Algorithm for the Numerical Evaluation of Certain Cauchy Principal Value Integrals*, Numer. Math.,v. 19 (1972), pp. 373-385

[69] T.E. Price, *Orthogonal polynomials for nonclassical weight functions*,
SIAM J. Numer. Anal., v. 16, No. 6, (December 1979), pp. 999-1006

[70] R. Piessens, E. de Doncker-Kapenga, C.W. Überhuber, D.K. Kahaner, *QUADPACK A Subroutine Package for Automatic Integration*,
Springer Series in Computational Mathematics 1, Springer Verlag, (1983)

[71] L.B. Rall, *Differentiation and Generating of Taylor Coefficients in PASCAL-SC*, in [51]

[72] D. Ratz, *Inclusion Isotone Extended Interval Arithmetic*, Interner Bericht 5/1996 (`http://www.uni-karlsruhe.de/~iam/html/reports/reports.html`)

[73] J.R. Rice, *A Theory of Condition*, J. SIAM Numer. Anal.,v. 3 (1966), No. 2

[74] R.A. Sack and A.F. Donovan, *An algorithm for Gaussian quadrature given modified moments*, Numer.Math., v. 18, (1972), pp. 465-478

[75] A. Sard, *Integral Representations of Remainders*, Duke.Math., J. 18, (1948), pp. 333-345

[76] M. Schader, S. Kuhlins, *Programmieren in C++*, Springer Verlag (1995)

[77] T. Schiera, *Ableitungsfreie Fehlerabschätzungen bei numerischer Integration holomorpher Funktionen*, Dissertation Universität Karlsruhe 1994

[78] H. D. Shapiro, *Increasing robustness in global adaptive quadrature through interval heuristics*, ACM Trans. Math. Software 10 (1984), pp. 117-139

[79] Ch. E. Stewart, *On the numerical evaluation of singular integrals of Cauchy type*, J. Soc. Ind. Appl. Math. 8, 342-353 (1960)

[80] U. Storck, *Verified Calculation of the Nodes and Weights for Gaussian Quadrature Formulas*, Interval Computation, St. Petersburg 1993 (ISSN 0135-4868)

[81] U. Storck, *Verifizierte Berechnung mehrfach geschachtelter singulärer Integrale der Gaskinetik*, Dissertation Universität Karlsruhe 1995

[82] A. Stroud & D. Secrest, *Gaussian Quadrature Formulas*, Prentice-Hall, Englewood Cliffs, N. J. , 1966, MR 34 #2185

[83] B. Stroustrup, *Die C++ Programmiersprache*, 3. Auflage, Addison-Wesley (1998)

[84] G. Szegö, *Orthogonal Polynomials*, Amer. Math. Soc., New York 1959

[85] S. Wedner, *Numerische Quadratur mit automatischer Ergebnisverifikation*, Diplomarbeit Universität Karlsruhe Oktober 1994

[86] J. Weissinger, *Numerische Mathematik auf Personal-Computern Teil 1*, B.I.- Wissenschaftsverlag 1984

[87] J.C. Wheeler, *Modified moments and Gaussian quadratures*, Rocky Mountain J. of Comp., v. 4(2), (1974), pp. 287-296

[88] J. Zierep, *Grundzüge der Strömungslehre*, Springer-Verlag Berlin Heidelberg 1997

Lebenslauf

Stefan Peter Wedner

geboren	am 26. Mai 1966 in Karlsruhe
Eltern	Hans-Peter Wedner und Sigrid Wedner geb. Riesch
Schulbildung	1972 - 1976 Grundschule Oberreut
	1976 - 1985 Bismarck-Gymnasium Karlsruhe
Reifeprüfung	am 11. Juni 1985
Zivildienst	Individuelle Schwerstbehinderten Betreuung
	Karlsruhe August 1985 - April 1987
Studium	der Mathematik mit Nebenfach Informatik an der
	Universität Karlsuhe (TH) 1987 - 1995
Diplomprüfung	im Studiengang Mathematik am 17.11.1995
Berufstätigkeit	angestellt am Institut für Angwandte Mathematik der
	Universität Karlsruhe seit Januar 1996
Eheschließung	am 31.März 2000 mit Jutta Wedner geb. Loisch
Kinder	Marius Wedner geb. am 9. Juni 2000