

# Kapitel 6

## Numerische Integration

In diesem Kapitel geht es um die *numerische Auswertung von bestimmten Integralen*

$$I(f) = \int_c^d dx f(x) \quad .$$

Numerische Methoden kommen dann zur Anwendung, wenn

- der Integrand nur punktweise (z.B. in Tabellenform) gegeben ist. In einem solchen Fall empfiehlt es sich, durch die gegebenen Punkte eine *Interpolationskurve* zu legen und diese dann stückweise zu integrieren (s. Abschnitt 6.1).
- der Integrand zwar als Funktion gegeben ist, das Integral jedoch nicht analytisch integrierbar ist, wie z.B.

$$\int_c^d dx \frac{\sin x}{x} \quad .$$

- das Integral zwar im Prinzip analytisch integrierbar ist, die analytische Auswertung jedoch unverhältnismäßig viel Rechenaufwand erfordern würde <sup>1</sup>.

In den beiden letzten Fällen werden fast ausschließlich sogenannte *Quadraturformeln* eingesetzt, von denen ab dem Abschnitt 6.2 ausführlich die Rede sein wird.

### 6.1 Numerische Integration punktweise gegebener Integranden.

Angenommen, eine zu integrierende Funktion  $f(x)$  sei nur bzgl. der  $n$  Stützpunkte

$$[x_i \mid f(x_i)] \quad i = 1, \dots, n$$

---

<sup>1</sup>Hier soll jedoch ausdrücklich auf die immer populärer werdende Computer-Software hingewiesen werden, die auch analytische Rechenoperationen erlaubt, wie z.B. die Systeme REDUCE, MAPLE, Mathematica usw.

bekannt, wobei der Einfachheit halber angenommen werden soll, daß der erste und letzte Stützpunkt,  $x_1$  und  $x_n$ , gleichzeitig die Integrationsgrenzen sind. In einem solchen Fall kann  $f(x)$  durch die entsprechende Interpolationskurve (z.B. eine kubische Spline-Kurve)  $I(x)$  ersetzt werden:

$$\begin{aligned} \int_{x_1}^{x_n} dx f(x) &\approx \int_{x_1}^{x_n} dx I(x) = \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} dx P^i(x) \\ &= \sum_{i=1}^{n-1} \int_{x_i}^{x_{i+1}} dx \left[ a_i + b_i(x - x_i) + c_i(x - x_i)^2 + d_i(x - x_i)^3 \right] \end{aligned}$$

Diese Integrale können leicht analytisch ausgewertet werden, und man erhält:

$$\int_{x_1}^{x_n} dx f(x) \approx \sum_{i=1}^{n-1} \left[ a_i(x_{i+1} - x_i) + \frac{b_i}{2}(x_{i+1} - x_i)^2 + \frac{c_i}{3}(x_{i+1} - x_i)^3 + \frac{d_i}{4}(x_{i+1} - x_i)^4 \right] \quad (6.1)$$

## 6.2 Verwendung von Quadraturformeln.

Für alle weiteren Ausführungen soll gelten: die (reellwertige) Integrandenfunktion  $f(x)$  ist analytisch gegeben und im gesamten Integrationsintervall  $[c; d]$  stetig.

### 6.2.1 Das grundsätzliche Problem

besteht in der näherungsweise Berechnung von bestimmten Integralen mit Hilfe von sogenannten Quadraturformeln:

$$I(f) = \int_c^d dx f(x) \quad \rightarrow \quad Q_m(f) = \frac{(d-c)}{2} \sum_{j=1}^m g_j f(x_j) \quad (6.2)$$

Die rechte Seite von (6.2), die Quadraturformel  $Q_m(f)$ , ist in der Lage, den Integralwert  $I(f)$  mit jeder gewünschten Genauigkeit<sup>2</sup> zu approximieren, wenn man

$m$ : den Grad der Quadraturformel

$x_1, \dots, x_m$ : die Stützpunkte im Intervall  $[c, d]$  sowie

$g_1, \dots, g_m$ : die Gewichts-Koeffizienten

geeignet wählt.

Die Differenz

$$E_V(f) = I(f) - Q_m(f)$$

nennt man den Verfahrensfehler der Quadratur.

Während der Grad der Quadraturformel, d.h. die Anzahl  $m$  der Summenterme, im Prinzip frei wählbar ist, müssen die Stützpunkte und die Gewichtungsfaktoren möglichst optimal gewählt werden. Im weiteren soll nun eine Optimierungsvorschrift für diese Größen angegeben werden.

<sup>2</sup>unter Vernachlässigung von Rundungsfehlern

### 6.2.2 Eine Optimierungsvorschrift für die $x_i$ und $g_i$ .

Für die Quadraturformel (6.2)  $m$ -ten Grades gibt es offenbar  $2m$  zu optimierende Parameter. Bei zahlreichen Quadraturformeln wird jedoch diese Anzahl noch durch zusätzliche *einschränkende Bedingungen* vermindert. Im allgemeinen sind daher nur

$$m' \leq 2m$$

Parameter zu optimieren.

Man geht nun von einem allgemeinen Polynom vom Grade  $m' - 1$  aus:

$$P_{m'-1}(x) = a_1 + a_2x + a_3x^2 + \dots + a_{m'}x^{m'-1} \quad (6.3)$$

Es sollen nun die Parameter der Quadraturformel so gewählt werden, daß für alle Polynome vom Grade  $m' - 1$  oder darunter *kein Verfahrensfehler auftritt*, d.h. daß gilt:

$$\int_c^d dx P_{m'-1}(x) \stackrel{!}{=} Q_m(P_{m'-1}) \quad (6.4)$$

Die Gleichung (6.4) stellt somit eine *Optimierungsvorschrift* für die zu optimierenden Parameter dar.

### 6.2.3 Die Trapezformel

ist eine Quadraturformel zweiten Grades, d.h.

$$Q^T(f) = \frac{(d-c)}{2} [g_1 f(x_1) + g_2 f(x_2)] \quad ,$$

wobei als Nebenbedingung  $x_1 = c$  und  $x_2 = d$  gilt. Es bleiben somit noch  $m' = 4 - 2 = 2$  Parameter zu optimieren, nämlich die Gewichtsfaktoren  $g_1$  und  $g_2$ . Durch Anwendung der Vorschrift (6.4) erhält man

$$\int_c^d dx (a_1 + a_2x) = \frac{(d-c)}{2} [g_1(a_1 + a_2c) + g_2(a_1 + a_2d)] \quad .$$

Das exakte Integral ist leicht auszuwerten, und durch Koeffizientenvergleich bzgl.  $a_1$  und  $a_2$  erhält man:

$$g_1 = 1 \quad \text{und} \quad g_2 = 1 \quad .$$

Die Trapezformel lautet somit

$$Q^T(f) = \frac{(d-c)}{2} [f(c) + f(d)] \quad . \quad (6.5)$$

Der Name 'Trapezformel' hat geometrische Gründe (s. Abb. 6.1).

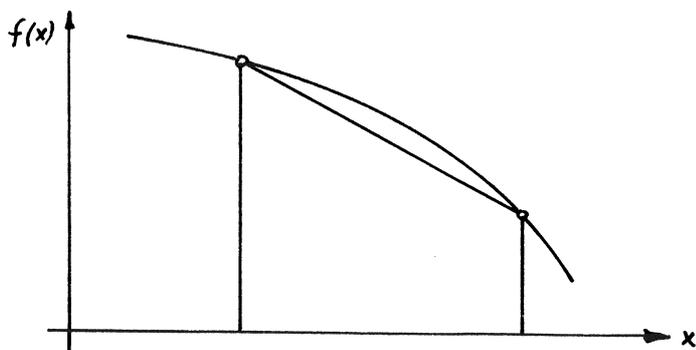


Abbildung 6.1: Anwendung der Trapezformel.

## 6.2.4 Die Simpson-Formel

ist eine Quadraturformel dritten Grades, d.h.

$$Q^S(f) = \frac{(d-c)}{2} [g_1 f(x_1) + g_2 f(x_2) + g_3 f(x_3)] \quad .$$

Als Nebenbedingung wird eine *äquidistante* Stützpunktverteilung gefordert, also

$$x_1 = c \quad x_2 = \frac{c+d}{2} \quad x_3 = d \quad .$$

Die Optimierung der 3 Gewichtungsfaktoren ergibt  $g_1 = g_3 = 1/3$ ;  $g_2 = 4/3$ , was zur Simpson-Formel

$$Q^S(f) = \frac{(d-c)}{6} \left[ f(c) + 4f\left(\frac{c+d}{2}\right) + f(d) \right] \quad (6.6)$$

führt.

## 6.2.5 Verfahrensfehler von Trapez- und Simpsonformel.

Die Verfahrensfehler der Quadraturformeln (6.5) und (6.6) können mittels Anwendung des Mittelwertsatzes der Integralrechnung bestimmt werden. Sie lauten

$$E_V^T = -\frac{(d-c)^3}{12} f''(\xi) \quad \xi \in [c, d] \quad (6.7)$$

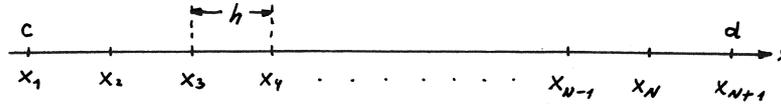
und

$$E_V^S = -\frac{(d-c)^5}{2880} f^{(4)}(\xi) \quad \xi \in [c, d] \quad . \quad (6.8)$$

Die Verfahrensfehler steigen also mit der 3. bzw. der 5. Potenz der Breite des Integrationsintervalls an. Das bedeutet natürlich, daß eine Anwendung von (6.5) bzw. (6.6) für größere Intervallbreiten zu vollkommen obsoleten Ergebnissen führt!

## 6.3 Die summierte Trapez- und Simpsonformel.

Die einzige Möglichkeit, mittels Trapez- und Simpsonformel zu brauchbaren Ergebnissen zu kommen, ist eine Aufteilung des gegebenen Intervalls  $[c, d]$  in eine Reihe von  $N$  Subintervallen mit der Intervallbreite  $h$  :



$$h = \frac{(d - c)}{N} \quad x_i = c + (i - 1)h \quad (i = 1, \dots, N + 1)$$

Nun wendet man die Trapezformel (6.5) auf jedes einzelne Subintervall an und summiert über die  $N$  Teilergebnisse: Dies ergibt<sup>3</sup>

$$Q^{ST}(f) = \sum_{i=1}^N \frac{h}{2} [f(x_i) + f(x_{i+1})]$$

bzw.

$$Q^{ST}(f) = h \left[ \frac{1}{2}f(c) + f(x_2) + f(x_3) + \dots + f(x_N) + \frac{1}{2}f(d) \right] \quad (6.9)$$

Ebenso kann man die Simpsonformel (6.6) auf *jeweils 2 nebeneinander liegende* Subintervalle anwenden, was voraussetzt, daß man für  $N$  eine gerade Zahl wählt.<sup>4</sup> Eine Summierung über die  $N/2$  Teilergebnisse ergibt

$$Q^{SS}(f) = \frac{h}{3} \left[ f(c) + 2 \sum_{i=1}^{N/2-1} f(x_{2i+1}) + 4 \sum_{i=1}^{N/2} f(x_{2i}) + f(d) \right]$$

bzw. ausgeschrieben:

$$Q^{SS}(f) = \frac{2h}{3} \left[ \frac{1}{2}f(c) + 2f(x_2) + f(x_3) + 2f(x_4) + \dots + f(x_{N-1}) + 2f(x_N) + \frac{1}{2}f(d) \right] \quad (6.10)$$

### 6.3.1 Verfahrensfehler der summierten Quadraturformeln.

Auch für die summierten Formeln (6.9) und (6.10) werden die Verfahrensfehler ohne Ableitung angegeben:

$$E_V^{ST}(f) = -\frac{(d - c)}{12} \cdot h^2 \cdot f''(\xi) \quad \xi \in [c, d] \quad (6.11)$$

bzw.

$$E_V^{SS}(f) = -\frac{(d - c)}{180} \cdot h^4 \cdot f^{(4)}(\xi) \quad \xi \in [c, d] \quad (6.12)$$

<sup>3</sup>'ST': Summierte Trapezformel, 'SS' : Summierte Simpsonformel.

<sup>4</sup>In der Literatur gibt es ähnliche Formeln auch für *ungerade*  $N$ , wie die sog. '3/8-Formel' (s. z. B. [2], S. 191).

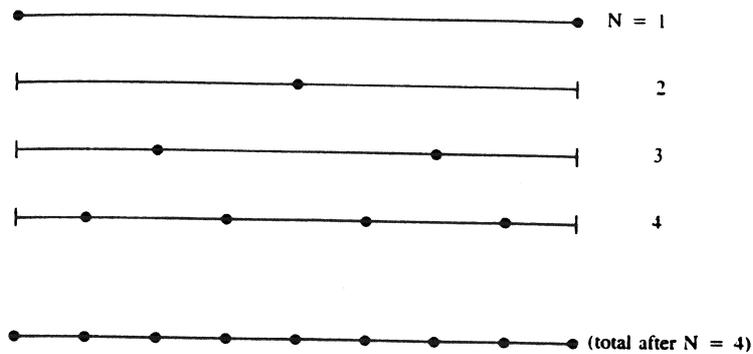


Abbildung 6.2: Erhöhung der Subintervalle bei der Anwendung der Trapezformel.

Diese Fehler sind also nur mehr linear von der Breite des Integrationsintervalls abhängig, dafür aber proportional zu  $h^2$  bzw.  $h^4$ : Das heißt, daß durch eine immer feinere Aufteilung von  $[c, d]$  die Verfahrensfehler beliebig klein gemacht werden können.

## 6.4 Die ökonomische Auswertung der Trapezformel.

Im allgemeinen wird man keine Information darüber haben, wieviele Subintervalle in die Rechnung einbezogen werden müssen, um ein numerisches Ergebnis der gewünschten Genauigkeit zu erhalten (eine a-priori Fehlerabschätzung auf Basis von (6.11) ist wegen der Unkenntnis von  $f''(\xi)$  nicht möglich). Man muß sich also an das gewünschte Ergebnis 'herantasten', wobei man die Stützpunktanzahl sukzessive erhöht. Am günstigsten ist es, die Stützpunktanzahl immer wieder zu verdoppeln (s. Abb. 6.2).

Man geht dabei von der simpelsten Näherung aus, nämlich von zwei Stützpunkten ( $M = 1$  Intervall), und man erhält

$$Q_{M=1}^{ST} = (d - c) \left[ \frac{1}{2}f(c) + \frac{1}{2}f(d) \right] \quad M = \text{Subintervall-Anzahl.}$$

Eine Verdoppelung der Intervalle ( $M = 2$ ) erfordert die Berechnung eines weiteren Funktionswertes mit dem Argument  $(c + d)/2$ , und es ergibt sich

$$Q_{M=2}^{ST} = \frac{(d - c)}{2} \left[ \frac{1}{2}f(c) + f\left(\frac{c + d}{2}\right) + \frac{1}{2}f(d) \right] = \frac{1}{2} \left[ Q_{M=1}^{ST} + (d - c)f\left(\frac{c + d}{2}\right) \right] .$$

Eine weitere Verdoppelung auf 4 Subintervalle erfordert lt. Abb. 6.2 die Berechnung von  $f(c + (d - c)/4)$  und  $f(c + 3(d - c)/4)$ :

$$Q_{M=4}^{ST} = \frac{1}{2} \left\{ Q_{M=2}^{ST} + \frac{(d - c)}{2} \left[ f \left( c + \frac{(d - c)}{4} \right) + f \left( c + \frac{3(d - c)}{4} \right) \right] \right\} .$$

Offensichtlich gilt bei jeder Verdoppelung der Subintervalle die Rekursionsformel

$$Q_{2M}^{ST} = \frac{1}{2} \left[ Q_M^{ST} + \frac{(d - c)}{M} \sum_{j=1(2)}^{2M-1} f \left( c + j \frac{(d - c)}{2M} \right) \right] . \quad (6.13)$$

(6.13) bildet die Basis für die Programme QTRAP und TRAPZD im Abschnitt 6.5.

### 6.4.1 Genauigkeitsabfragen

Durch die fortgesetzte Anwendung der Formel (6.13) erhält man also eine Folge von Näherungswerten für ein bestimmtes Integral nach Maßgabe der summierten Trapezformel:

$$Q_1^{ST} ; Q_2^{ST} ; Q_4^{ST} ; Q_8^{ST} \dots ,$$

wobei der untere Index die Zahl der Subintervalle in  $[c,d]$  darstellt.

Nun stellen sich die Fragen: 'Wie genau sind die erhaltenen numerischen Werte?' bzw. 'Wieviele Subintervall-Verdoppelungen sind nötig, um eine bestimmte vorgegebene (absolute oder relative) Genauigkeit zu erreichen?'

Das Problem einer solchen *internen Fehlerdiagnose* liegt darin, daß während des Programmablaufes eine möglichst zuverlässige Fehlerabschätzung, basierend auf dem vorliegenden Datenmaterial, erfolgen muß. Dabei kann man z. B. wie folgt vorgehen:

Wenn man die Rundungsfehler im Moment außer Acht läßt, kann man unter Verwendung von Glg. (6.11) und unter Berücksichtigung von  $h \propto 1/M$  schreiben:

$$I_{\text{exakt}} = Q_M^{ST} + C(M)/M^2 \quad (6.14)$$

$$I_{\text{exakt}} = Q_{2M}^{ST} + C(2M)/(2M)^2 ,$$

wobei  $C(M)$  und  $C(2M)$  die in (6.11) enthaltene Konstante für die Zahl der Subintervalle  $M$  und  $2M$  ist.

Nimmt man nun an, daß näherungsweise

$$C(M) \approx C(2M) = C \quad (6.15)$$

gilt (was durchaus nicht immer der Fall ist!), so stellen die Gleichungen (6.14) ein lineares Gleichungssystem für die Unbekannten  $I_{\text{exakt}}$  und  $C$  dar. Für  $C$  erhält man nach kurzer Rechnung

$$\frac{(2M)^2 (Q_{2M}^{ST} - Q_M^{ST})}{3}$$

bzw. für den Verfahrensfehler des numerischen Bestwertes  $Q_{2M}^{ST}$

$$E_V[Q_{2M}^{ST}] \approx \frac{Q_{2M}^{ST} - Q_M^{ST}}{3}. \quad (6.16)$$

Daraus kann man Abbruchbedingungen für ein auf (6.13) beruhendes Programm gewinnen:

*Die Rekursion (6.13) wird solange fortgesetzt, bis der vorgegebene absolute oder relative Fehler  $GEN$  unterschritten wird, also bis gilt:*

$$\frac{|Q_{2M}^{ST} - Q_M^{ST}|}{3} < GEN \quad \text{oder} \quad \frac{|Q_{2M}^{ST} - Q_M^{ST}|}{3|Q_{2M}^{ST}|} < GEN. \quad (6.17)$$

Beachten Sie bitte, daß die relative Fehlerabfrage versagt, wenn das numerische Ergebnis gegen Null geht. Es ist auch nochmals darauf hinzuweisen, daß die einfachen Abfragen (6.17) nur dann funktionieren, wenn (1) die Näherung (6.15) erfüllt ist und wenn (2) der Verfahrensfehler den Rundungsfehler dominiert.

## 6.5 Die Programme QTRAP und TRAPZD.

Quelle: [9], S. 111ff mit Änderungen.

INPUT-Parameter:

**C,D:** Grenzen des Integrationsintervalls.

**JMAX:** maximale Anzahl von Aufrufen von TRAPZD.

**IREL:** IREL=1 relative Fehlerabfrage, sonst absolute Fehlerabfrage.

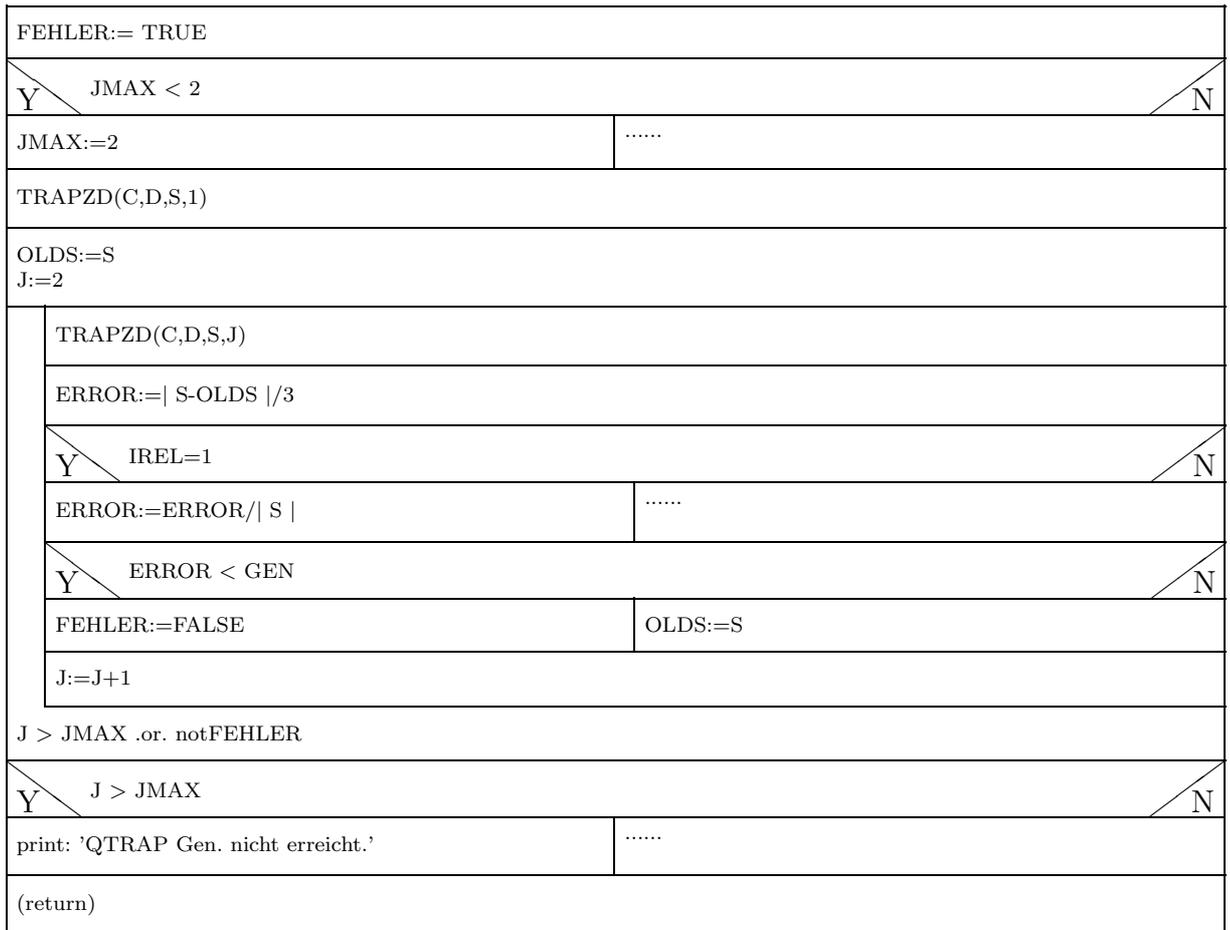
**GEN:** relative oder absolute Genauigkeit.

OUTPUT-Parameter:

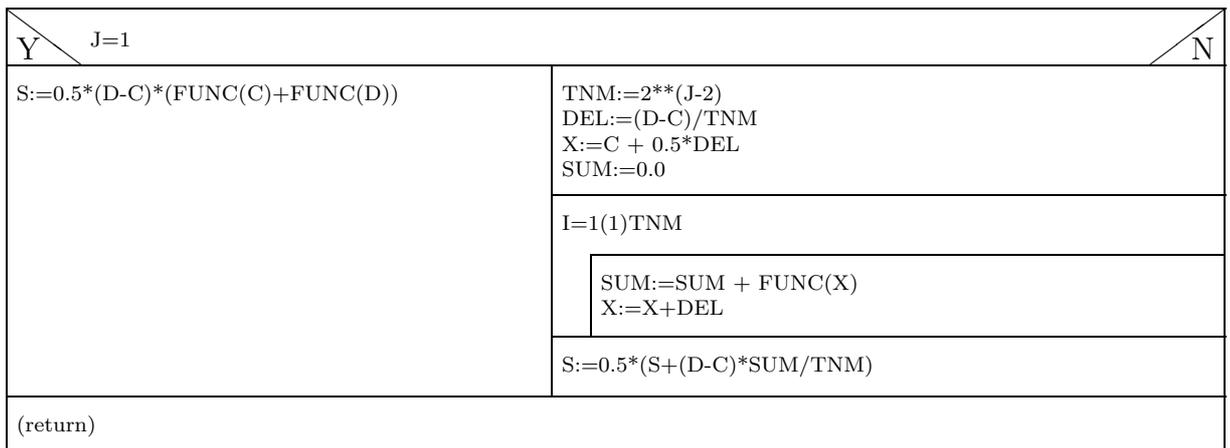
**S:** Näherungswert für das bestimmte Integral.

**FEHLER:** Fehlerdiagnostik: FEHLER = FALSE Berechnung ok  
FEHLER = TRUE Genauigkeit  
wurde nicht erreicht.

### Struktogramm 16 — QTRAP(C,D,JMAX,IREL,GEN,S,FEHLER)



### Struktogramm 17 — TRAPZD(C,D,S,J)



Anmerkung: Die Routine TRAPZD muß hintereinander für J=1,2,3,... aufgerufen werden.

Der folgende einfache Test soll Ihnen das korrekte Funktionieren der Programme QTRAP und TRAPZD dokumentieren:

$$I_{\text{exakt}} = \int_0^1 dx \frac{\ln(1+x)}{x(1+x)} = \frac{\pi^2}{12} - \frac{\ln(2)^2}{2} = 0.582240\dots$$

Fordert man vom Programm QTRAP (F90-DOUBLE PRECISION) eine absolute Genauigkeit GEN=0.000001, so erhält man

```
TEST FUER DIE ROUTINEN  QTRAP  UND  TRAPZD
NUM. METHODEN IN DER PHYSIK  WS 2000/2001          28-8-2000
H. Sormann  F90  DOUBLE PRECISION
```

J	Funkt. aufrufe	num. Erg.	wahrer abs. Fehler	geschaetzter abs. Fehler	GEN
2	3	0.60695347	0.247129E-01	0.221111E-01	0.100000E-05
3	5	0.58858506	0.634453E-02	0.612280E-02	0.100000E-05
4	9	0.58383827	0.159774E-02	0.158226E-02	0.100000E-05
5	17	0.58264071	0.400184E-03	0.399185E-03	0.100000E-05
6	33	0.58234062	0.100093E-03	0.100030E-03	0.100000E-05
7	65	0.58226555	0.250263E-04	0.250223E-04	0.100000E-05
8	129	0.58224678	0.625675E-05	0.625651E-05	0.100000E-05
9	257	0.58224209	0.156420E-05	0.156418E-05	0.100000E-05
10	513	0.58224092	0.391051E-06	0.391050E-06	0.100000E-05

```
EXAKT =      0.5822405265
Ergebnis =  0.5822409175
geforderte Genauigkeit (abs) = 0.000001000
Dazu waren      513 Funktionsaufrufe noetig.
```

## 6.6 Das Romberg-Verfahren.

Wie aus den Gleichungen (6.11) und (6.12) für die Verfahrensfehler der summierten Trapezformel (Ordnung  $m=2$ ) und der summierten Simpsonformel (Ordnung  $m=3$ ) hervorgeht, sind diese gegeben durch

$$E_V^{(m)} \text{ proportional } h^{2m-2} ,$$

wobei  $m$  die Ordnung der Quadraturformel bedeutet. Wegen  $h \propto 1/M$  ( $M$  = Anzahl der Subintervalle) gilt also:

$$E_V^{(m)}(M) = \frac{C(M)}{M^{2m-2}} . \quad (6.18)$$

Man geht nun genauso vor wie im Abschnitt 6.4.1, d.h. man stellt den (unbekannten) exakten Integralwert durch die Gleichungen

$$I_{\text{exakt}} = Q_M^{(m)} + \frac{C(M)}{M^{2m-2}}$$

bzw.

$$I_{\text{exakt}} = Q_{2M}^{(m)} + \frac{C(2M)}{(2M)^{2m-2}}$$

dar, wobei die Konstante  $C$  als näherungsweise unabhängig von  $M$  angenommen wird, also

$$C(M) \approx C(2M) = C \quad .$$

Unter diesen Umständen stellen die obigen Gleichungen ein lineares System für  $C$  und  $I_{\text{exakt}}$  dar, wobei die Lösung

$$I_{\text{exakt}} = \frac{1}{4^{m-1} - 1} \left( 4^{m-1} Q_{2M}^{(m)} - Q_M^{(m)} \right)$$

lautet. Dabei ist ' $I_{\text{exakt}}$ ' wegen der Approximation von  $C$  natürlich nicht wirklich die exakte Integrallösung, sondern nur eine Verbesserung.

Man kann zeigen:

' $I_{\text{exakt}}$ ' entspricht dem Ergebnis einer Quadraturformel mit  $2M$  Subintervallen und *einer um Eins höheren Ordnung*:

$$Q_{2M}^{(m+1)} = \frac{1}{4^{m-1} - 1} \left( 4^{m-1} Q_{2M}^{(m)} - Q_M^{(m)} \right) \quad . \quad (6.19)$$

Verifikation der Glg. (6.19) durch ein Beispiel:

Geht man von der Trapezformel ( $m = 2$ ) sowie von zwei Subintervallen ( $M = 2$ ) aus, so erhält man

$$Q_2^{(2)} = \frac{(d-c)}{2} \left[ \frac{1}{2} f(c) + f\left(\frac{c+d}{2}\right) + \frac{1}{2} f(d) \right] \quad .$$

Eine Verdoppelung auf vier Subintervalle ergibt

$$Q_4^{(2)} = \frac{(d-c)}{4} \left[ \frac{1}{2} f(c) + f\left(c + \frac{(d-c)}{4}\right) + f\left(\frac{c+d}{2}\right) + f\left(c + 3\frac{(d-c)}{4}\right) + \frac{1}{2} f(d) \right] \quad .$$

Setzt man die beiden letzten Gleichungen in die Formel (6.19) ein, so ergibt sich

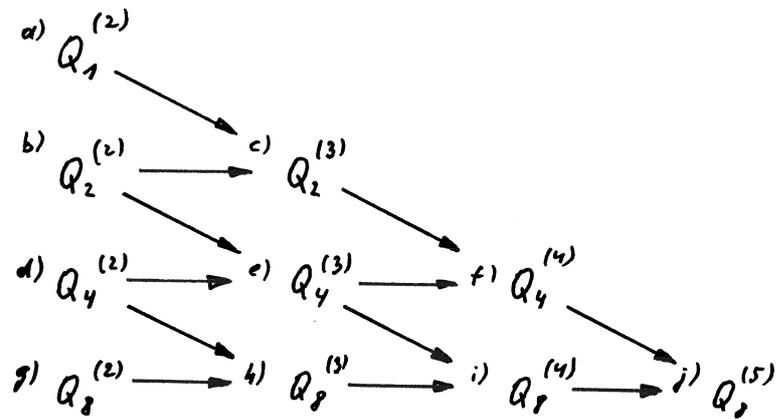
$$Q_4^{(3)} = \frac{(d-c)}{6} \left[ \frac{1}{2} f(c) + 2f\left(c + \frac{(d-c)}{4}\right) + f\left(\frac{c+d}{2}\right) + 2f\left(c + 3\frac{(d-c)}{4}\right) + \frac{1}{2} f(d) \right] \quad ,$$

was exakt dem Ergebnis der summierten Simpsonformel (Quadraturformel der Ordnung 3) für 4 Subintervalle entspricht:

$$Q_4^{(3)} \equiv Q_4^{SS}$$

Man erhält also durch Anwendung der Formel (6.19) auf 2 Trapezformeln genau die Simpsonformel.

Die Strategie von Romberg, die auf der Gleichung (6.19) beruht, ist mit Hilfe des folgenden Schemas leicht zu verstehen:



Die Kleinbuchstaben links neben den  $Q$ -Symbolen bezeichnen die Reihenfolge der Berechnung, und die zusammenlaufenden Pfeile bedeuten die Anwendung von (6.19). D.h., von Zeile zu Zeile verdoppelt sich die Anzahl der Subintervalle, von Spalte zu Spalte erhöht sich die Ordnung der Quadraturformel. Es ist nun plausibel anzunehmen, daß die Diagonalelemente des obigen Schemas schneller zum gewünschten Integralwert konvergieren als die Zeilen- und Spaltenelemente, weil ja entlang der Diagonale sowohl die Subintervall-Zahl als auch die Ordnung der Quadratur erhöht werden. Man kann in der Tat zeigen, daß für viele praktische Anwendungen der Romberg-Strategie

$$|Q_{2M}^{(m)} - Q_M^{(m-1)}| \approx |Q_M^{(m-1)} - I_{\text{exakt}}| \quad (6.20)$$

gilt, was bedeutet, daß

*die Differenz zwischen zwei aufeinanderfolgenden Diagonalelementen des Romberg-Schemas eine gute Näherung für den absoluten Fehler des Elementes  $Q_M^{(m-1)}$  darstellt.*

Diese Fehlerabschätzung hat aber eine unschöne Seite: man muß offensichtlich, um die Genauigkeit des Diagonalelementes der Zeile  $M$  bestimmen zu können, die gesamte folgende Zeile  $2M$  berechnen!

### 6.6.1 Das Programm ROMB.

Quellen: [9], S. 114f, [2], S. 205ff und 407f, mit Änderungen.

Das Programm ROMB berechnet einen Näherungswert für ein bestimmtes Integral.

INPUT-Parameter:

**C,D:** Grenzen des Integrationsintervalls.

**KMIN:** Ordnung des Romberg-Prozesses, *ab welchem die Fehlerdiagnose beginnt*. Diskussion siehe S. 178f.

**KMAX:** maximale Ordnung des Romberg-Prozesses ( $\geq 3$ ).

**IREL:** IREL=1 relative Fehlerabfrage, ansonsten absolute Fehlerabfrage.

**GEN:** relativer oder absoluter zu unterschreitender Fehler.

OUTPUT-Parameter:

**S:** Näherungswert für das bestimmte Integral.

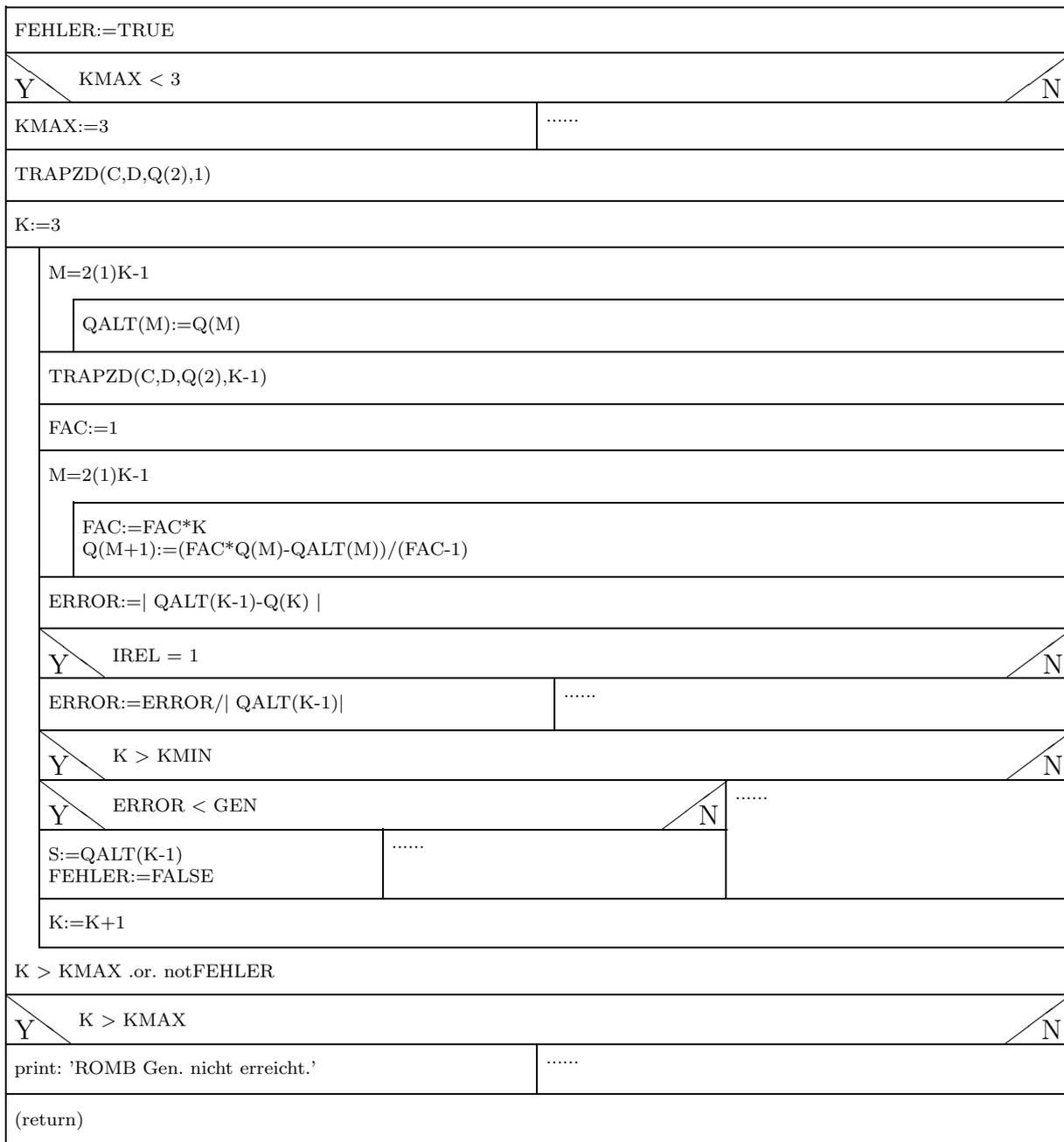
**ERROR:** Abschätzung für den absoluten bzw. relativen Fehler dieser Näherung.

**FEHLER:** Fehlerdiagnostik: FEHLER = FALSE Berechnung ok  
FEHLER = TRUE Genauigkeit  
wurde nicht erreicht.

Das Programm ROMB benötigt die bereits besprochene Routine TRAPZD.

Anmerkung: Was die Leistungsfähigkeit des Romberg-Verfahrens betrifft, so ist (frei zitiert nach [9], S. 115) ... die Romberg-Routine sehr leistungsfähig für genügend glatte Integranden ... Unter solchen Umständen braucht das Programm ROMB *viel, viel weniger* Funktionsaufrufe als jedes andere Programm, das auf der Trapez- oder der Simpsonformel beruht.

**Struktogramm 18:** — ROMB(C,D,KMIN,KMAX,IREL,GEN,S,ERROR,FEHLER)



Das folgende Output-Protokoll zeigt Ihnen an Hand des Beispiels von S. 169 die Überlegenheit des Romberg-Algorithmus gegenüber der Trapezmethode:

```
NUM. METH. IN DER PHYSIK  WS 2000/2001  ROMBERG-FORMEL  28-8-2000
INTEGRAL  Skriptum S. 169
H. Sormann  F90  DOUBLE PRECISION
```

m	Funkt. aufrufe	num. Erg.	wahrer abs. Fehler	geschaetzter abs. Fehler	absGEN
2	3	0.67328680	0.910463E-01	0.884444E-01	0.100000E-05
3	5	0.58484236	0.260183E-02	0.253878E-02	0.100000E-05
4	9	0.58230358	0.630557E-04	0.623029E-04	0.100000E-05
5	17	0.58224128	0.752829E-06	0.749125E-06	0.100000E-05

```
EXAKT =      0.5822405265
Ergebnis =  0.5822412793
geforderte Gen. (abs) = 0.100000E-05
Dazu waren      17 Funktionsaufrufe noetig.
```

## 6.6.2 Romberg-Verfahren: Tests und Anmerkungen

Das eben erläuterte Verfahren wird im folgenden auf Integrale vom Typus

$$I_n = \int_1^9 dx \cos \left[ \frac{n\pi}{2} x \right] e^{-0.2x} \quad n = 1, 2, \dots \quad (6.21)$$

angewendet<sup>5</sup>. Es ist unmittelbar klar, daß die numerische Auswertung solcher Integrale mit größer werdendem  $n$  schwieriger wird, da die Oszillation der Integrandenfunktion immer intensiver wird. Zur Demonstration zeigt die Abb. 6.3 die Integrandenfunktion für  $n = 20$ . Es ist zu erwarten, daß in solchen Fällen für die numerische Integration eine erhebliche Zahl von Stützpunkten erforderlich ist.

In den folgenden Tests geht es hauptsächlich um die Zuverlässigkeit der auf Glg. (6.20) basierenden Fehlerdiagnostik im Romberg-Programm. Das Verhalten des *absoluten* Fehlers ist in den *oberen* Diagrammen der Abbildungen 6.4 (für  $n = 2$ ) und 6.5 (für  $n = 100$ ) dargestellt.

Sowohl für den schwach ( $n = 2$ ) als auch für den stark oszillierenden Integranden ( $n = 100$ ) ist im Bereich  $5 \leq m \leq 16$  eine sehr gute Übereinstimmung zwischen dem 'wahren' Fehler und dem vom Programm geschätzten Fehler zu beobachten. Die Kurven liegen praktisch aufeinander, und die Fehler zwischen numerischer Näherung und exakter Lösung nehmen mit fortschreitendem Romberg-Prozess (=mehr Stützpunkte, höhere Ordnung der Quadraturformel) monoton ab. Auf diese Weise kann der absolute Verfahrensfehler bis auf  $\approx 10^{-16}$  reduziert werden. Natürlich wird dieser Wert für den schwach oszillierenden Integranden bedeutend früher erreicht ( $m = 11$ )

---

<sup>5</sup>Alle Ergebnisse dieses Abschnittes wurden mit einem F90-Programm, DOUBLE-PRECISION, erhalten.

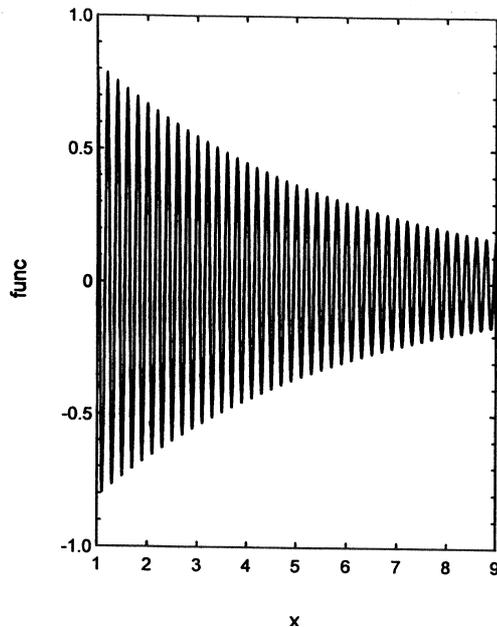


Abbildung 6.3: Die Integrandenfunktion von (6.21) für  $n = 20$ .

als für den stark oszillierenden Integranden ( $m = 16$ ).

Ab  $m = 11$  (für  $n = 2$ ) bzw.  $m = 16$  (für  $n = 100$ ) kann der Fehler des numerischen Ergebnisses nicht mehr weiter reduziert werden, er nimmt im Gegenteil sogar leicht zu! Die Ursache dafür sind natürlich *Rundungsfehler*, die mit Zunahme der Stützpunktzahl größer werden und bei einem gewissen  $m = 10$  den Verfahrensfehler übersteigen.

Man muß also beim Numerischen Integrieren (und nicht nur dort) zur Kenntnis nehmen:

*Auf Grund der Rundungsfehler kann die Genauigkeit einer numerischen Integration nur bis zu einer gewissen Grenze gesteigert werden. Ab dieser Grenze bringt eine weitere Erhöhung des Rechenaufwandes keine Verbesserung der Ergebnisse, sondern oft sogar eine Verschlechterung.*

Man sieht auch, daß der numerisch ermittelte Fehler ab  $m = 16$  plötzlich auf Null zurückgeht. Überlegen Sie sich die Ursache für dieses Verhalten.

Interessant ist das Verhalten des absoluten Fehlers für kleine Werte von  $m$ : in diesem Bereich weicht die Fehlerabschätzung (6.20) stark vom 'wahren' Fehler ab. Der Grund dafür liegt darin, daß in diesem Bereich die Stützpunktdichte viel zu klein ist, um die (oszillierende) Integrandenfunktion numerisch korrekt beschreiben zu können.

Zusammenfassend kann jedoch gesagt werden, daß die Abschätzung des *absoluten* Fehlers im Romberg-Programm sehr gut funktioniert.

Viel problematischer ist die Fehlerdiagnostik für den *relativen* Fehler, der an und für sich (s. Kapitel 1) viel mehr interessiert als der absolute Fehler. Das Dilemma wird in den *unteren* Diagrammen der Abbildungen 6.4 und 6.5 demonstriert:

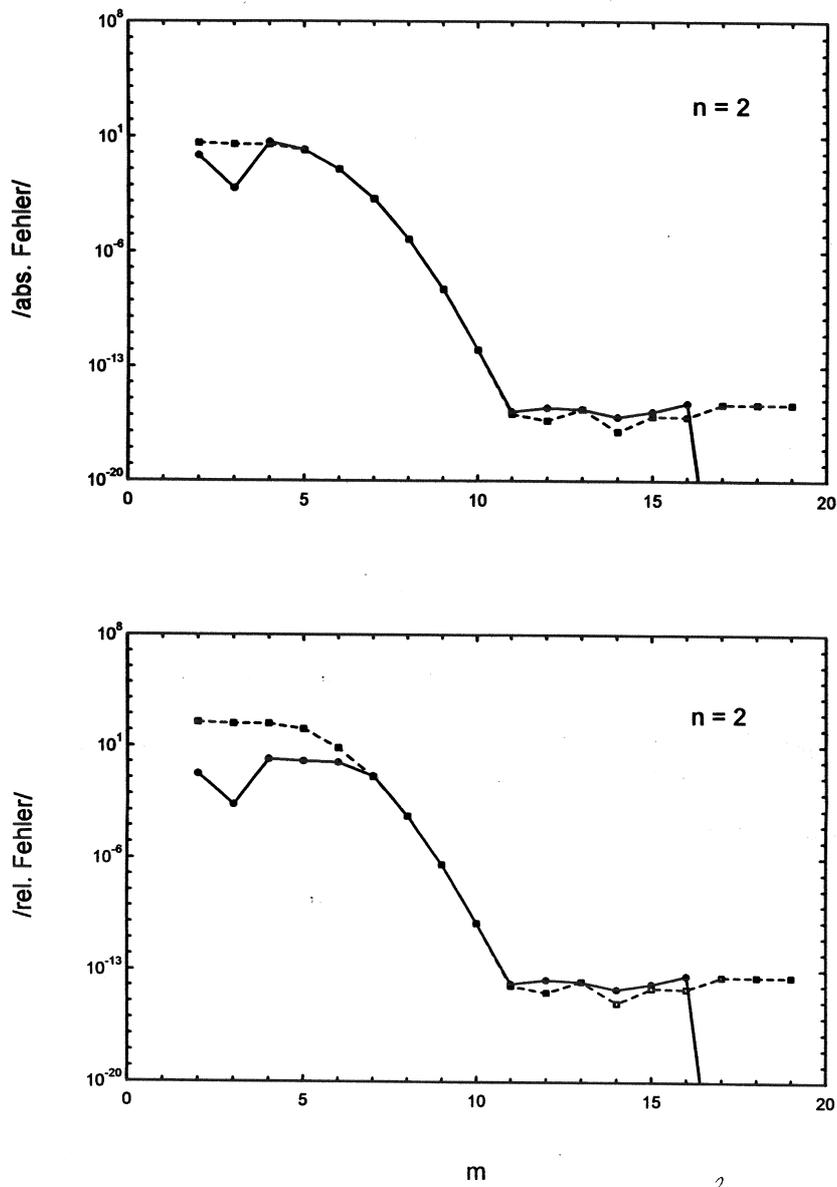


Abbildung 6.4: Romberg-Tests für das Integral (6.21),  $n = 2$ . Dargestellt ist der *absolute* (oben) und der *relative* Fehler (unten) der 'Romberg-Näherung' in bezug auf das exakte Ergebnis in Abhängigkeit von der Ordnung des Romberg-Prozesses  $m$ .

Volle Kurve: Fehlerabschätzung laut Glg. (6.20),  
gestrichelte Kurve: 'wahrer' Fehler.

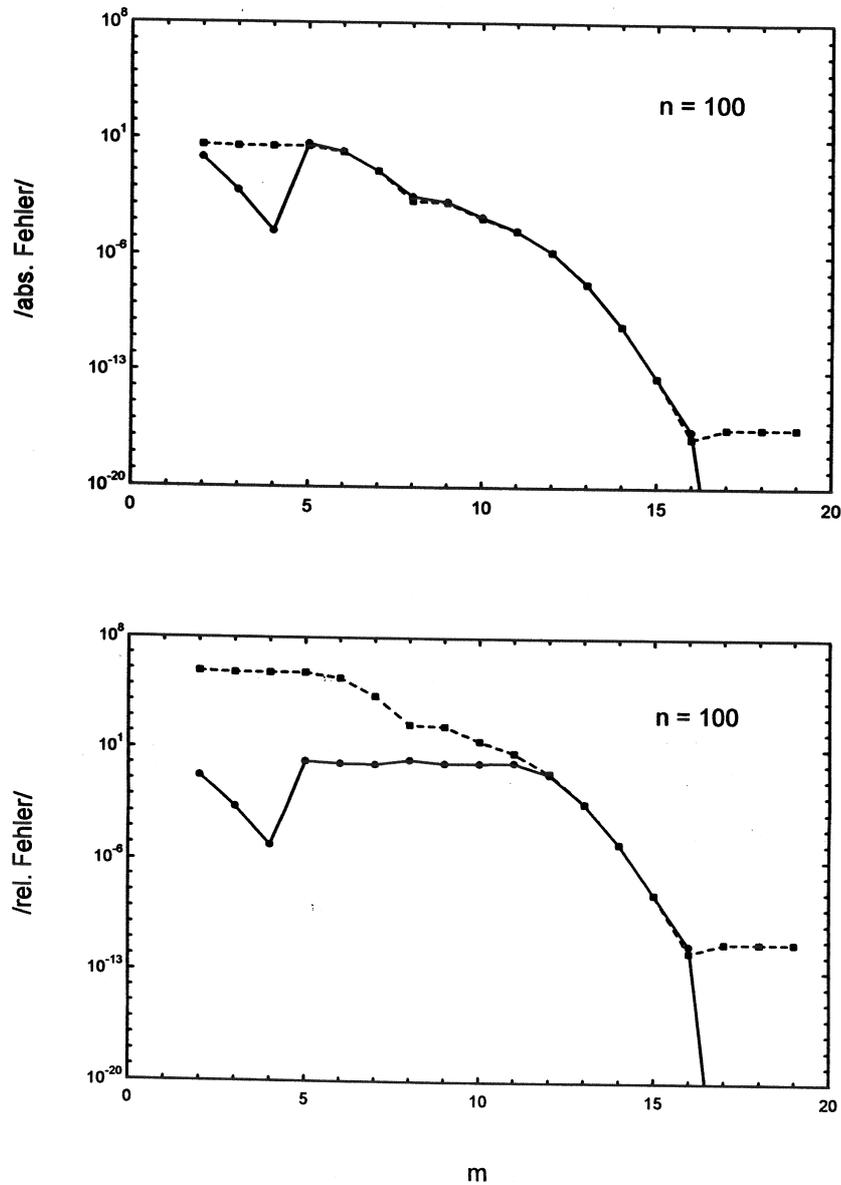


Abbildung 6.5: Romberg-Tests für das Integral (6.21),  $n = 100$ . Dargestellt ist der *absolute* (oben) und der *relative* Fehler (unten) der 'Romberg-Näherung' in bezug auf das exakte Ergebnis in Abhängigkeit von der Ordnung des Romberg-Prozesses  $m$ .

Volle Kurve: Fehlerabschätzung laut Glg. (6.20),  
gestrichelte Kurve: 'wahrer' Fehler.

Da für kleine und mittlere Werte  $m$  des Romberg-Prozesses die Integrationswerte noch sehr stark vom 'wahren' Integralwert abweichen, kommt es für den *relativen* Fehler in diesem Bereich von  $m$  zu sehr großen Unterschieden zwischen dem 'wahren' und dem geschätzten relativen Fehler (s. speziell Abb. 6.5 (unten), wo Unterschiede von bis zu 11(!) Zehnerpotenzen vorkommen). Das daraus resultierende Fehlverhalten des Romberg-Programms wird im folgenden gezeigt:

Angenommen, das Integral (6.21) soll auf eine *relative* Genauigkeit von  $\text{GEN}=10^{-5}$  berechnet werden. In diesem Fall liefert das Romberg-Programm das folgende Ergebnis:

```
NUM. METH. IN DER PHYSIK WS 2000/2001 ROMBERG-FORMEL 28-8-2000
INTEGRAL Skriptum (6.21) fuer n = 100
H. Sormann F90 DOUBLE PRECISION
```

m	Funkt. aufrufe	num. Erg.	wahrer rel. Fehler	geschaetzter rel. Fehler	relGEN
2	3	3.93611857	0.743152E+06	0.168200E+00	0.100000E-04
3	5	3.27406321	0.618154E+06	0.210069E-02	0.100000E-04
4	9	3.26718542	0.616855E+06	0.797950E-05	0.100000E-04

```
EXAKT =      0.0000052965
Ergebnis =  3.2671854208
geforderte Gen. (rel) = 0.100000E-04
Wahre Genauigkeit =      0.616855E+06
Dazu waren      9 Funktionsaufrufe noetig.
```

Auf Grund des Versagens der Fehlerdiagnose stoppt das Programm bereits bei  $m = 4$  ab und gibt ein Ergebnis, dessen relativer Fehler um mehr als 10 Zehnerpotenzen größer ist als GEN.

Zumindest eine gewisse Möglichkeit, einem solchen Verhalten gegenzu- steuern, bietet der im Abschnitt 6.6.1 angegebene Parameter KMIN. Dieser Wert gibt an, *ab welcher Stufe des Romberg-Prozesses die Fehlerdiagnostik beginnen soll*. Wenn man nun weiß, daß die Integrandenfunktion 'schwierig' ist (starke Oszillationen, Spitzen etc.), kann man durch die Eingabe eines geeignet großen Wertes von KMIN solche Probleme entschärfen.

Zum Beispiel gibt das Romberg-Programm für  $\text{KMIN}=6$  das richtige Ergebnis:

(\*) am Ende der Zeile bedeutet: keine Fehlerdiagnose.

m	Funkt. aufrufe	num. Erg.	wahrer rel. Fehler	geschaetzter rel. Fehler	relGEN	
2	3	3.93611857	0.743152E+06	0.168200E+00	0.100000E-04	*
3	5	3.27406321	0.618154E+06	0.210069E-02	0.100000E-04	*
4	9	3.26718542	0.616855E+06	0.797950E-05	0.100000E-04	*
5	17	3.26715935	0.616850E+06	0.144805E+01	0.100000E-04	*
6	33	-1.46384560	0.276380E+06	0.107212E+01	0.100000E-04	
7	65	0.10557102	0.199312E+05	0.984072E+00	0.100000E-04	
8	129	0.00168158	0.316489E+03	0.180823E+01	0.100000E-04	
9	257	-0.00135911	0.257604E+03	0.111862E+01	0.100000E-04	
10	513	0.00016122	0.294385E+02	0.114146E+01	0.100000E-04	
11	1025	-0.00002281	0.530572E+01	0.130154E+01	0.100000E-04	
12	2049	0.00000688	0.298361E+00	0.232677E+00	0.100000E-04	
13	4097	0.00000528	0.373724E-02	0.376226E-02	0.100000E-04	
14	8193	0.00000530	0.109633E-04	0.109709E-04	0.100000E-04	
15	16385	0.00000530	0.768462E-08	0.768636E-08	0.100000E-04	

EXAKT = 0.0000052965  
 Ergebnis = 0.0000052965  
 geforderte Gen. (rel) = 0.100000E-04  
 berechnete Gen. (rel) = 0.768636E-08  
 Wahre Genauigkeit = 0.768462E-08  
 Dazu waren 16385 Funktionsaufrufe noetig.

## 6.7 Die Gauss'sche Quadraturformel.

Ausgangspunkt ist wieder die Gleichung (6.2), wobei vorläufig nur das Integrationsintervall  $[-1, +1]$  betrachtet wird:

$$\int_{-1}^{+1} dx f(x) \approx \sum_{j=1}^m g_j f(x_j) \quad . \quad (6.22)$$

Zum Unterschied von der Trapez- und der Simpsonformel werden nun *keinerlei einschränkende Bedingungen* bzgl. der Optimierung der Parameter  $g_1 \dots g_m$  und  $x_1 \dots x_m$  gemacht. Es sind also für eine Gauss-Quadraturformel  $m$ -ten Grades insgesamt  $2m$  Parameter zu optimieren. Die Optimierungsvorschrift (6.4) lautet also in diesem Fall

$$\int_{-1}^{+1} dx P_{2m-1}(x) = g_1 P_{2m-1}(x_1) + \dots + g_m P_{2m-1}(x_m) \quad .$$

Man kann nun zeigen, daß die obige Gleichung erfüllt ist, wenn man für  $x_1 \dots x_m$  die Nullstellen des Legendre-Polynoms  $m$ -ter Ordnung ( $L_m$ ) verwen-

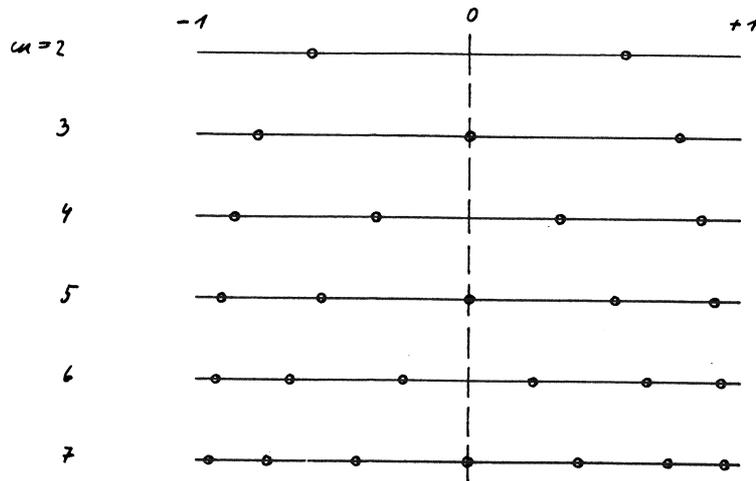


Abbildung 6.6: Stützpunktverteilung bei der Gauss-Quadratur.

det. Aus diesem Grund wird diese Quadraturformel auch als *Gauss-Legendre-Formel* bezeichnet.

(o. B:) Die zugehörigen Gewichtungsfaktoren ergeben sich aus

$$g_j = \frac{2}{(1 - x_j^2) \left[ \frac{d}{dx} L_m(x) \right]_{x=x_j}^2} .$$

Was die Verteilung der Stützpunkte im Intervall  $[-1, +1]$  betrifft, so ist diese natürlich nicht mehr äquidistant (s. Abb. 6.6), sondern hat die folgenden Eigenschaften:

- Die Stützstellen sind bzgl. des Intervallmittelpunktes symmetrisch verteilt. Jedem Stützpunkt  $x_j$  im rechten Halbintervall entspricht ein Punkt  $-x_j$  im linken Halbintervall, wobei beide Punkte denselben Gewichtungsfaktor haben.
- Die Stützpunkt-Dichte nimmt zu den Intervallgrenzen hin zu.
- Die Intervallgrenzen selbst sind keine Stützpunkte.<sup>6</sup>

Bisher wurden alle Überlegungen zur Gauss-Quadratur für das spezielle Intervall  $[-1, +1]$  durchgeführt. Die entsprechende Quadraturformel

$$\int_{-1}^{+1} dx f(x) \approx \frac{(+1) - (-1)}{2} \sum_{j=1}^m g_j f(x_j)$$

kann nun mittels der *linearen Transformation*

$$z_j = \frac{d+c}{2} + \frac{d-c}{2} x_j$$

<sup>6</sup>Die Gauss-Formel ist eine *offene* Quadraturformel, während die Trapez- und die Simpson-Formel *geschlossene* Quadraturformeln sind.

leicht auf die allgemeine Form

$$\int_c^d dx f(x) \approx \frac{d-c}{2} \sum_{j=1}^m g_j f(z_j) \quad , \quad (6.23)$$

gebracht werden, wobei der Vorfaktor  $(d-c)/2$  meistens in die Gewichtungsfaktoren mit einbezogen wird:

$$w_j \equiv \frac{(d-c)}{2} g_j \quad .$$

### 6.7.1 Der Verfahrensfehler bei der Gauss-Quadratur.

Wegen der Kompromisslosigkeit bei der Optimierung der Parameter ist zu erwarten, daß die Gauss-Quadraturformel allen anderen Quadraturformeln überlegen ist. Dies ist in vielen Fällen auch tatsächlich so: Betrachtet man nämlich den Verfahrensfehler der Gauss-Quadratur (s. z.B. [16], S.199)

$$E_V^G = \frac{(d-c)^{2m+1} (m!)^4}{(2m+1)[(2m)!]^3} f^{(2m)}(\xi) \quad \xi \in [c, d] \quad , \quad (6.24)$$

so erkennt man, daß der dominante Nenner-Term  $[(2m)!]^3$  bei steigendem  $m$  für eine rasante Verkleinerung des Verfahrensfehlers sorgt!

### 6.7.2 Die Berechnung der Gauss-Parameter.

Ein wichtiges Kriterium für die Anwendung der Gauss-Quadratur ist die bequeme Verfügbarkeit der Gauss-Parameter  $z_j$  und  $w_j$  in (6.23). Es gibt wohl keine ernstzunehmende Programm-Library, die nicht über ein einschlägiges Generierungsprogramm verfügt. 'Numerical Recipes' enthält die Routine GAULEG in FORTRAN ([9], S. 125f), PASCAL ([9], S. 700f) und C ([10], S. 152).

Die Routinen GAULEG.F90 bzw. GAULEG.C berechnen die Parameter der Gauss-Quadratur ('Gauss-Legendre-Parameter') für ein beliebiges Intervall und eine beliebige Ordnung:

INPUT-Parameter:

**X1,X2:** Intervallgrenzen für die Integration.

**N:** Grad der Gauss-Quadratur.

OUTPUT-Parameter:

**X( ):** Feld mit den Abszissenwerten  $x_j$  der Stützpunkte.

**W( ):** Feld mit den Gewichtungsfaktoren  $w_j$ .

### 6.7.3 Das Programm GAUINT.

Das Programm GAUINT berechnet näherungsweise ein bestimmtes Integral mittels der Gauss'schen Quadraturformel. Die erforderlichen Parameter werden von der Routine GAULEG (s. o.) berechnet.

INPUT-Parameter:

**C,D:** Integrationsgrenzen.

**IREL:** IREL=1 relative Fehlerabfrage, ansonsten absolute Fehlerabfrage.

**GEN:** relative oder absolute Genauigkeit.

**MANF:** Anfangs-Ordnung der Quadratur.

**MINKR:** Erhöhung der Ordnung bei Nicht-Erreichen der Genauigkeit.

**MMAX:** maximale Ordnung der Quadratur.

OUTPUT-Parameter:

**S:** Näherung für das bestimmte Integral.

**ERROR:** Abschätzung für den absoluten bzw. relativen Fehler dieser Näherung.

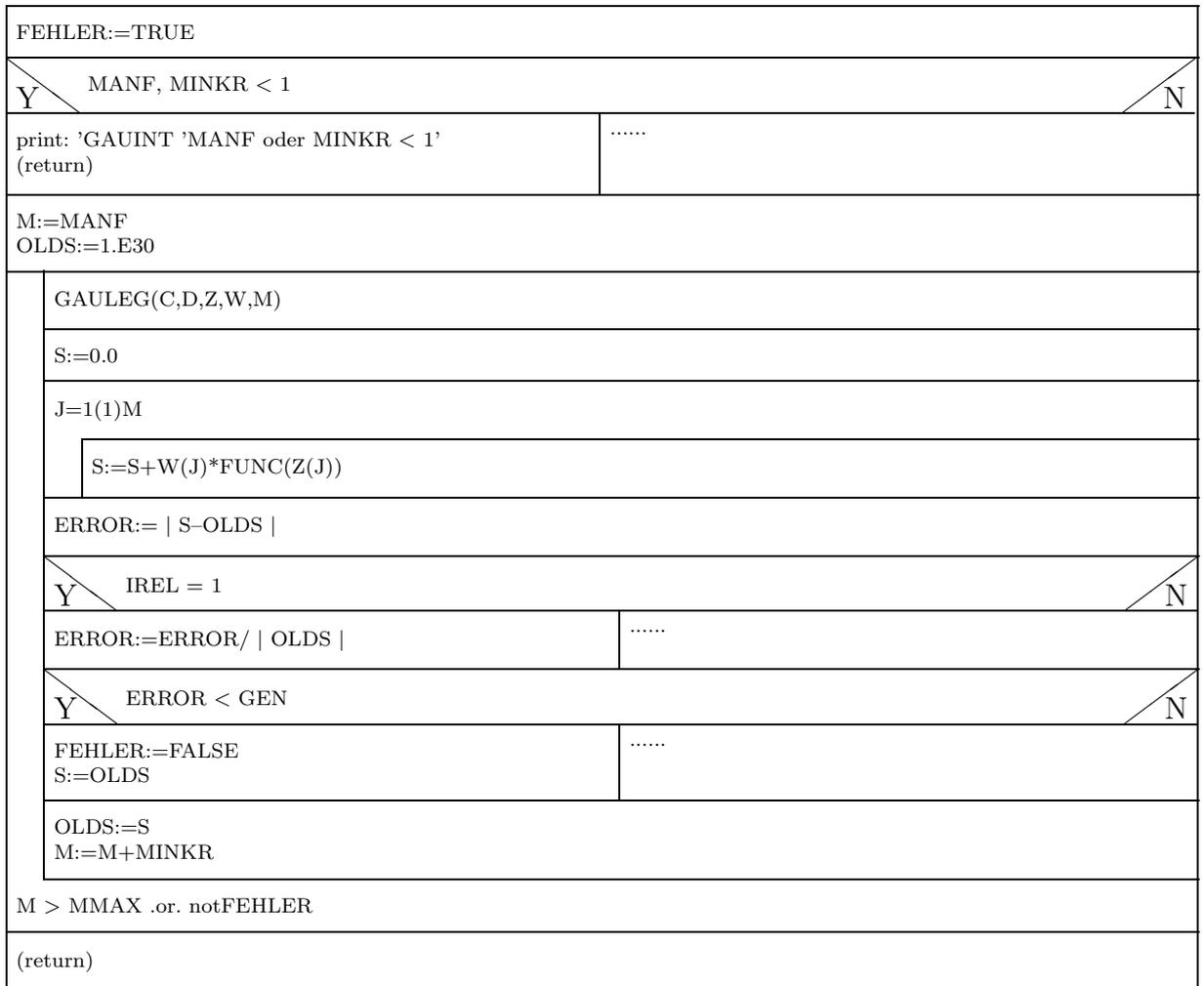
**FEHLER:** logische Fehlervariable: FALSE, wenn alles o.k., sonst TRUE.

Interne Parameter:

**Z( ),W( ):** eindimensionale Felder für die Gauss-Parameter.

Struktogramm 19: Das Programm GAUINT.

**Struktogramm 19** — GAUINT(C,D,IREL,GEN,MANF,MINKR,MMAX,S,ERROR,FEHLER)



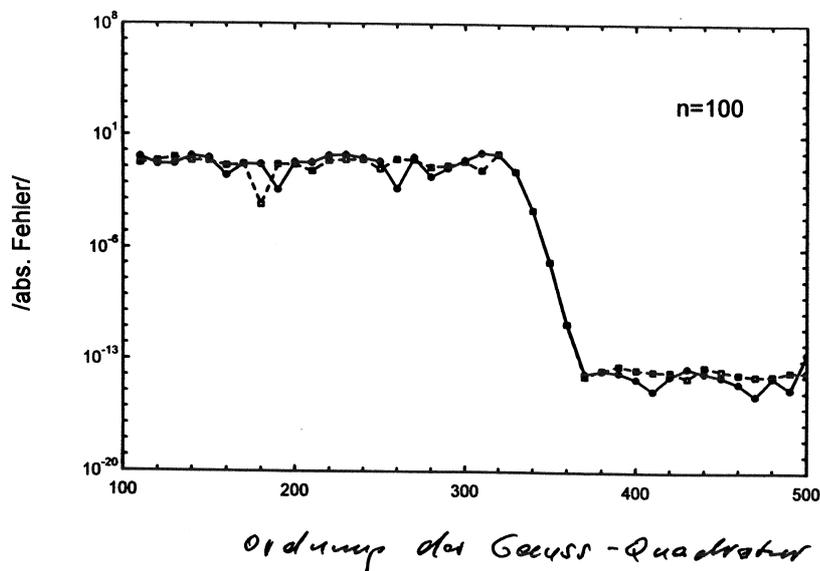


Abbildung 6.7: Anwendung des Programms GAUINT auf das Integral (6.21) für  $n = 100$ . Dargestellt sind der 'wahre' absolute Fehler (strichlierte Kurve) und der numerisch abgeschätzte Fehler (volle Kurve) als Funktion der Ordnung der Gauss-Legendre-Formel.

Parameter: MANF=100    MINKR=10.

#### 6.7.4 Genauigkeitsabfragen

Beim Romberg-Verfahren gibt es mit (6.20) eine Formel, die auf einfache Weise eine in vielen Fällen zuverlässige Fehlerabschätzung ermöglicht. Diese Methode ist auch im Programm GAUINT realisiert, d.h., man geht davon aus, daß die Differenz zweier aufeinanderfolgender Näherungen der Gauss-Quadratur (mit den Ordnungen  $M$  und  $M+MINKR$ ) eine vernünftige Abschätzung des absoluten Fehlers des Ergebnisses mit der Ordnung  $M$  darstellt. Meine Tests haben gezeigt, daß diese Aussage (für nicht zu große  $MINKR$ )  $\approx$  unabhängig vom  $MINKR$  ist.

#### 6.7.5 Ein Qualitätsvergleich von GAUINT mit QTRAP und ROMB

Im folgenden soll das Gauss-Legendre-Verfahren auf das 'schwierige' Integral (6.21) für  $n = 100$  angewendet werden. Die Abb. 6.7 zeigt das Verhalten des absoluten Fehlers in Abhängigkeit von der Ordnung der Quadraturformel, d.h. von der Zahl der verwendeten Funktionswerte, wobei wieder der 'wahre' Fehler dem numerisch geschätzten Fehler gegenübergestellt ist.

Die Ergebnisse sind im Prinzip ähnlich wie beim Romberg-Verfahren: bis zu einer 'kritischen Ordnung' von etwa  $m=320$  ist der Verfahrensfehler sehr groß; danach fällt er innerhalb eines relativ kleinen Bereichs ( $m$  zwischen 320 und 370) auf den erreichbaren Mindestfehler von ca.  $10^{-14}$  ab. Zumindest in diesem Bereich ist die Fehlerabschätzung gemäß Abschnitt 6.7.4 recht zuverlässig.

Abb. 6.7 zeigt beeindruckend die große Leistungsfähigkeit der Gauss-

Quadratur gegenüber den anderen in diesem Kapitel behandelten Methoden: selbst bei diesem sehr schwierig auszuwertenden Integral genügen etwa 370 'Gauss-Stützpunkte', um eine Genauigkeit des Integrals von  $< 10^{-14}$  zu erreichen! Beim Romberg-Verfahren ist dafür etwa eine Ordnung 15 nötig (s. Abb. 6.5 oben), was weit mehr 10000 Stützpunkte bedeutet!

Diese Überlegenheit der Gauss-Quadratur wird zwar auch im folgenden sehr deutlich, muß aber dennoch - wie sogleich demonstriert wird - kritisch betrachtet werden:

Für einen Vergleich der in diesem Kapitel präsentierten Verfahren soll als *Qualitätskriterium* die Rechenzeit herangezogen werden, die zur numerischen Berechnung eines bestimmten Integrals mit einer bestimmten Genauigkeit erforderlich ist. Da i.a. die zeitintensivste Komponente aller präsentierten Programme die Auswertung der Integrandenfunktion ist, kann man das Qualitätskriterium wie folgt definieren:

*Jenes numerische Verfahren zur Berechnung bestimmter Integrale ist das beste, das zur Erlangung einer bestimmten Genauigkeit die kleinste Zahl von Stützpunkten benötigt.*

Es soll nun als Test für die Trapezformel (Programm QTRAP), das Romberg-Verfahren (Programm ROMB) und die Gaussformel (Programm GAUINT) das bereits früher (s. S. 169) verwendete bestimmte Integral

$$\int_0^1 dx \frac{\ln(1+x)}{x(1+x)} = \frac{\pi^2}{12} - \frac{1}{2}(\ln 2)^2 = 0.5822405 \dots$$

numerisch ausgewertet werden. Die Ergebnisse dieses Tests sind in der Abb. 6.8 zusammengefaßt:

- Der Abbildung ist sofort zu entnehmen, daß die Trapezformel sowohl dem Romberg-Verfahren wie auch der Gauss-Formel stark unterlegen ist. Sie benötigt weit mehr als 1 Million (!) Stützpunkte, um den durch den Rundungsfehler gegebenen *minimalen Fehler* von ca.  $10^{-15}$  zu erreichen.
- Das Romberg-Verfahren schafft das schon bei  $\approx 100$  Stützpunkten.
- Auf den ersten Blick konkurrenzlos: die Gauss-Legendre-Quadratur, die lediglich 10 (!) Stützpunkte benötigt, um das Test-Integral bis auf einen absoluten Fehler von  $10^{-15}$  auszuwerten!
- Dazu ist jedoch zu bemerken, daß beim Gauss-Verfahren – zum Unterschied vom Romberg-Verfahren – eine Reihe von Stützpunkten *im Laufe des 'Herantastens'* an die gewünschte Genauigkeit berechnet werden müssen, die in die eigentliche Berechnung des Näherungswertes nicht eingehen. Diese unökonomische Eigenschaft des Programms GAUINT reduziert etwas die Überlegenheit dieser Methode gegenüber dem Verfahren von Romberg.

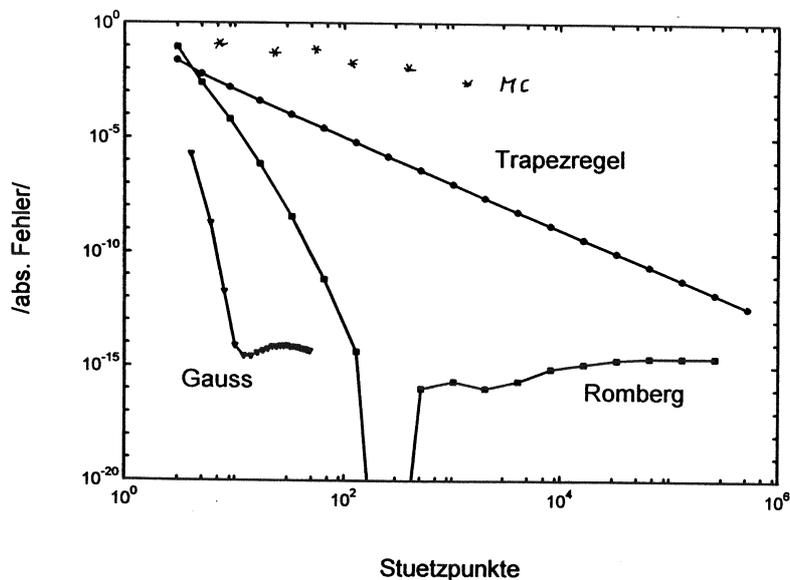


Abbildung 6.8: Ein Genauigkeitstest für QTRAP, ROMB und GAUINT.  
 (\*) 'Monte-Carlo'-Ergebnisse.

- Nochmals soll hier auf das *Rundungsfehlerverhalten* der Quadraturformeln hingewiesen werden:  
 Es ist wegen dieses Fehlers nicht möglich, die Genauigkeit des numerisch ermittelten Integrationsergebnisses durch Erhöhung der Stützpunktanzahl beliebig zu steigern. Wie das Diagramm 6.8 deutlich zeigt, kippen die Fehlerkurven für alle getesteten Methoden bei ungefähr derselben Genauigkeit (Fehler ca.  $10^{-14}$ – $10^{-15}$ )!
- Es muß aber an dieser Stelle auch betont werden, daß ein einzelner Test natürlich nicht ausreicht, klare Aussagen über das allgemeine Verhalten der verschiedenen Methoden zu machen. Trotzdem bietet er einen Einblick in das typische Verhalten dieser Programme in vielen praktischen Fällen!

## 6.8 Numerische Auswertung uneigentlicher Integrale.

Definition: Ein uneigentliches Integral liegt vor, wenn

- die obere und/oder die untere Integrationsgrenze unbeschränkt sind/ist (Typ 1),
- die zu integrierende Funktion an einer der beiden oder an beiden Intervallgrenzen oder im Inneren des Intervalls unbeschränkt ist (Typ 2),

wobei trotz dieser Unbeschränktheiten die Integralfläche *endlich* ist.

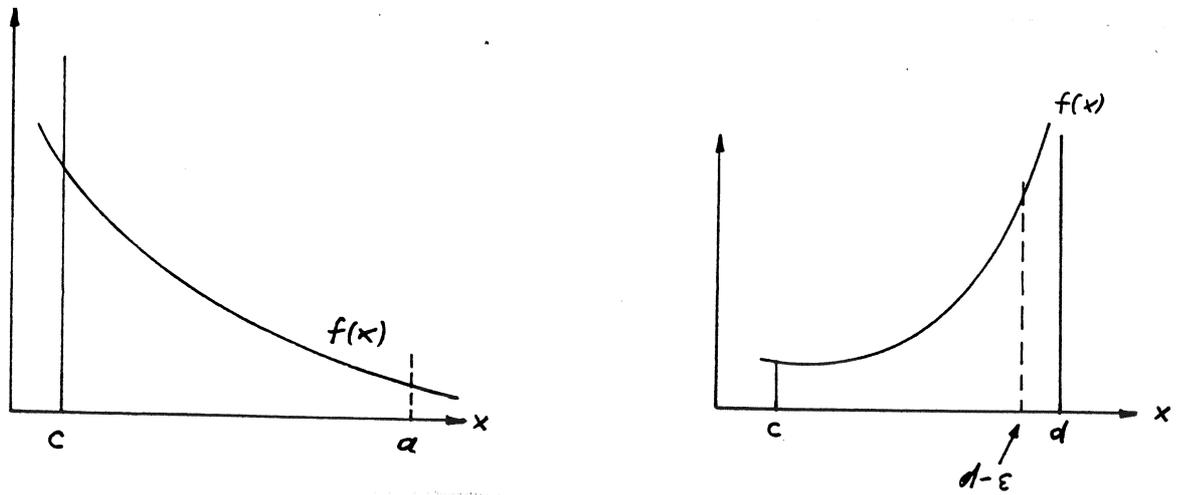


Abbildung 6.9: Uneigentliche Integrale vom Typ 1 und vom Typ 2.

- Hier sollen - mathematisch nicht ganz korrekt - auch jene Integrale als uneigentliche Integrale angesehen werden, bei welchen der Integrand an einer der beiden (oder an beiden) Grenzen des Integrationsintervalls eine *unbestimmte Form* annimmt (Typ 3).

Selbstverständlich können die Probleme (1)-(3) auch in Kombination vorkommen!

Mathematische Formulierung:

Typ 1 (s. Abb. 6.9, links):

$$I(a) = \int_c^a dx f(x) \quad \lim_{a \rightarrow \infty} I(a) \rightarrow I$$

Typ 2 (s. Abb. 6.9, rechts):

$$I(\epsilon) = \int_c^{d-\epsilon} dx f(x) \quad \lim_{\epsilon \rightarrow 0} I(\epsilon) \rightarrow I$$

### 6.8.1 Auswertung von Integralen vom Typ 1.

Eine durchaus praktikable, wenn auch nicht sehr elegante Methode, mit Integralen der Art

$$\int_c^\infty dx f(x) \quad \int_{-\infty}^d dx f(x) \quad \int_{-\infty}^\infty dx f(x)$$

fertigzuwerden, ist die folgende: man setzt anstatt 'unendlich' eine hinlänglich große Zahl  $a_1$  und berechnet das Integral

$$I_1 = \int_c^{a_1} dx f(x) \quad .$$

Dann erhöht man  $a_1$  auf  $a_2$  und erhält  $I_2$ .

Ist nun  $|I_2 - I_1|$  absolut oder relativ genügend klein, kann  $I_2$  als Näherungswert für das uneigentliche Integral angesehen werden.

Diese einfache Methode ist jedoch nicht immer brauchbar. So müßte z.B., wenn man das uneigentliche Integral

$$I = \int_0^{\infty} \frac{dx}{(1+x^2)^{4/3}}$$

auf 5 Dezimalen genau berechnen will, für '∞' eine Zahl > 1000 gesetzt werden (s. [8], S.219)!

In solchen Fällen ist eine *nicht-lineare Transformation der Integrationsvariablen* zu empfehlen. So kann mit

$$t = \frac{1}{1+x}$$

das obige Integral in die Form

$$I = \int_0^1 dt \frac{t^{2/3}}{[t^2 + (1-t)^2]^{4/3}}$$

übergeführt werden. Dabei wird also das nach oben offene Intervall  $[0, \infty]$  auf das geschlossene Intervall  $[0, 1]$  abgebildet. In dieser Form kann das Integral numerisch sehr genau ausgewertet werden (im folgenden eine Auswertung mittels GAUINT):

2 Stuetzpunkte	I = .103746E+01
6 Stuetzpunkte	I = .111758E+01
12 Stuetzpunkte	I = .112029E+01
20 Stuetzpunkte	I = .112031E+01
30 Stuetzpunkte	I = .112028E+01
42 Stuetzpunkte	I = .112027E+01
56 Stuetzpunkte	I = .112026E+01
72 Stuetzpunkte	I = .112026E+01
90 Stuetzpunkte	I = .112026E+01
110 Stuetzpunkte	I = .112025E+01
132 Stuetzpunkte	I = .112025E+01

## 6.8.2 Auswertung von Integralen vom Typ 2.

Als Beispiel diene das uneigentliche Integral

$$I = \int_0^1 dx \frac{e^x}{\sqrt{x}} = 2.9253034918... \quad (6.25)$$

mit einer Unendlichkeitsstelle am Nullpunkt. Für die Auswertung gibt es nun die folgenden Möglichkeiten:

1. Man ignoriert einfach die Singularität. Dies ist jedoch nur möglich, wenn man die Gauss-Quadratur oder eine andere *offene* Quadraturformel verwendet, weil ja in diesem Fall der Integrand nicht an der Stelle  $x = 0$  ausgewertet werden kann!

Wie man allerdings der Abb. 6.10 entnehmen kann, muß man diese Ignoranz der Polstelle mit einer sehr schlechten Konvergenz des Verfahrens bezahlen!

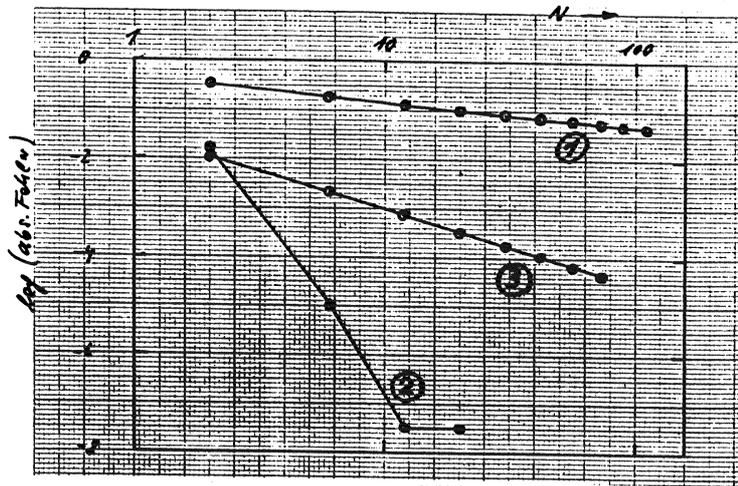


Abbildung 6.10: Numerische Auswertung des Integrals (6.25). Bedeutung der Nummern siehe Text.

2. Man eliminiert die Singularität des Integranden durch *geeignete analytische Umformungen*, wie z.B. mittels der Variablen-Transformation

$$x = t^2 \quad ,$$

die zum Integral

$$I = 2 \int_0^1 dt e^{t^2}$$

führt. Dieses Integral hat ausgezeichnete Konvergenz-Eigenschaften.

3. Auch eine *partielle Integration* des Integrals führt zum Ziel:

$$I = 2e - 2 \int_0^1 dx \sqrt{x} e^x \quad .$$

Im Diagramm 6.10 ist dargestellt, wie das Gauss-Verfahren für die drei besprochenen Varianten konvergiert.

### 6.8.3 Auswertung von Integralen vom Typ 3.

Integranden mit Unbestimmtheitsstellen an den Intervallgrenzen bereiten offenen Quadraturformeln überhaupt keine Probleme. Verwendet man geschlossene Quadraturformeln, muß man allerdings die Unbestimmtheitsstellen explizite berücksichtigen. So würde z.B. das Funktionsunterprogramm für den Integranden des Integrals

$$I = \int_0^1 dx \frac{\sin x}{x}$$

wie folgt aussehen:

```
FUNCTION FUNC(x: real): real;
BEGIN
  IF(x = 0.0) THEN func:=1.0 ELSE func:=sin(x)/x
END;
```

## 6.9 Variationen der Gauss-Quadratur.

Die soeben besprochene Gauss-Quadraturformel (6.22) ist nur ein Spezialfall einer Gruppe von Quadraturformeln für die bestimmten Integrale

$$I = \int_c^d dx W(x) f(x) \approx \sum_{i=1}^m w_i f(x_i) \quad , \quad (6.26)$$

wobei der als *Gewichtsfunktion*  $W(x)$  bezeichnete Teil des Integranden fix gegeben ist. Die Formel (6.22) ergibt sich also aus (6.26) durch  $W(x) \equiv 1$ . Da die Stützpunkte in diesem Fall (für  $c = -1$  und  $d = +1$ ) die Nullstellen der Legendre-Polynome sind, wird (6.22) auch die *Gauss-Legendre-Formel* genannt.

Andere, für die Praxis wichtige Spezialfälle von  $W(x)$  sind

$$W(x) = e^{-x} \quad \text{für das Intervall } [0, \infty)$$

sowie

$$W(x) = e^{-x^2} \quad \text{für das Intervall } (-\infty, \infty).$$

Da die entsprechenden Stützpunkte die Nullstellen der *Laguerre'schen* bzw. der *Hermite'schen* Polynome darstellen, heißen die davon hergeleiteten Quadraturformeln

die Gauss-Laguerre-Formel

$$\int_0^{+\infty} dx e^{-x} f(x) \approx \sum_{i=1}^m w_i^{Lag} f(x_i^{Lag}). \quad (6.27)$$

beziehungsweise die Gauss-Hermite-Formel:

$$\int_{-\infty}^{+\infty} dx e^{-x^2} f(x) \approx \sum_{i=1}^m w_i^H f(x_i^H). \quad (6.28)$$

Formeln dieser Art sind speziell für die numerische Behandlung von uneigentlichen Integralen vom Typ (1) (s. Abschnitt 6.8) interessant.

Viele Libraries bieten Programme an, die die Stützpunkte und Gewichtungsfaktoren einer ganzen Reihe solcher Sonderfälle berechnen (die *Numerical Recipes C* enthalten z. B. die Routinen GAULAG (Gauss-Laguerre), GAUHER (Gauss-Hermite) und andere); siehe dazu auch Abschnitt 6.11).

## 6.10 Mehrfach-Integrale

sind im Prinzip kein Problem. Wenn man sich auf Mehrfach-Integrale mit konstanten Grenzen beschränkt, so gilt einfach

$$\begin{aligned} & \int_{c_1}^{d_1} \int_{c_2}^{d_2} \cdots \int_{c_k}^{d_k} dx_1 dx_2 \cdots dx_k \quad f(x_1, x_2, \dots, x_k) \quad (6.29) \\ & \approx \frac{(d_1 - c_1)(d_2 - c_2) \cdots (d_k - c_k)}{2^k} \cdot \sum_{j_1=1}^{m_1} g_{j_1} \sum_{j_2=1}^{m_2} g_{j_2} \cdots \sum_{j_k=1}^{m_k} g_{j_k} \cdot f(x_{j_1}^{(1)}, x_{j_2}^{(2)}, \dots, x_{j_k}^{(k)}) \quad . \end{aligned}$$

Die Anwendbarkeit solcher *mehrdimensionalen Quadraturformeln* ist allerdings in der Praxis wegen des immens hohen Rechenaufwandes auf Integrale nicht zu hoher Ordnung ( $k = 1, 2, 3(?), \dots$ ) beschränkt.

Für Integrale höherer Ordnung bzw. für Mehrfach-Integrale mit komplizierten Integranden sind *statistische Methoden (Monte-Carlo-Methoden)* von großer Bedeutung, insbesondere dann, *wenn keine allzu hohe Genauigkeit gefordert wird!*

## 6.11 Software-Angebot

**NAG:** In dieser Library finden Sie z. B. mit der Routine *d01bcf* ein gutes Beispiel für ein Programm, das die numerische Auswertung sechs verschiedener Gauss-Integralformeln ermöglicht. Das Programm liefert nicht das Ergebnis der Quadraturformel, sondern lediglich die Stützpunkte (Feld *ABSCIS*) und die Gewichtungsfaktoren (Feld *WEIGHT*) für die Quadraturformel.

Die Headline des Programmes (in FORTRAN) lautet wie folgt:

```
SUBROUTINE D01BCF (ITYPE, A, B, C, D, N, WEIGHT, ABSCIS, IFAIL)
INTEGER ITYPE, N, IFAIL
REAL*8 A, B, C, D, WEIGHT(N), ABSCIS(N)
```

·  
·

ITYPE=0      Gauss-Legendre       $I = \int_a^b dx f(x)$

ITYPE=1      Gauss-Jacobi       $I = \int_a^b dx (b-x)^c (x-a)^d f(x)$

ITYPE=2      Gauss-Exponential       $I = \int_a^b dx |x - \frac{a+b}{2}|^c f(x)$

ITYPE=3      Gauss-Laguerre       $I = \int_a^\infty dx |x-a|^c e^{-bx} f(x)$   
when  $b > 0$

ITYPE=4      Gauss-Hermite       $I = \int_{-\infty}^{+\infty} dx |x-a|^c e^{-b(x-a)^2} f(x)$

ITYPE=5      Gauss-Rational       $I = \int_a^\infty dx \frac{|x-a|^c}{|x+b|^d} f(x)$   
when  $a+b > 0$ .

Die Ermittlung des Näherungswertes für das Integral lautet in allen Fällen

$$I \approx \sum_{i=1}^n w_i f(x_i).$$

Ebenfalls in **NAG:**      *d01daf* 2-D quadrature, finite region.

*d01fbf* Multi-dimensional Gauss.

*d01gbf* Multi-dimensional Monte Carlo.

**QUADPACK:** (p.d.) Paket, erhältlich über NETLIB für die Berechnung bestimmter Integrale und Integraltransformationen von Funktionen einer Veränderlichen.

**TOMS:** (p.d.) NETLIB/TOMS/655 enthält ebenfalls Programme für Variationen des Gauss-Quadraturformel.