

Ein verwickelter Schwanz

Dirk Laurie

Die kürzeste und beste Verbindung zweier Wahrheiten im Reellen verläuft oft durch das Komplexe.

Jacques Hadamard (1945)

„Mein Schwanz ist lang und traurig“, sagte die Maus und wandte sich seufzend an Alice.

„Das ist fürwahr ein langer Schwanz“, bestätigte Alice und sah verwundert zum Schwanz der Maus hinunter, „aber warum nennst du ihn traurig?“

Lewis Carroll (1865)

Problem 1

Welchen Wert hat $\lim_{\epsilon \rightarrow 0} \int_{\epsilon}^1 x^{-1} \cos(x^{-1} \log x) dx$?

1.1 Auf den ersten Blick

Dieses Problem scheint wirklich alle möglichen Gemeinheiten für uns bereit zu halten. Der Graph des Integranden sieht beängstigend aus, man werfe nur einen Blick auf Abb. 1.1 (das Stück nahe 0 haben wir unterschlagen, damit der Graph noch auf die Seite passt). Es ist kaum vorstellbar, dass der erforderliche Grenzwert existiert. Dennoch verlassen wir uns zunächst darauf, dass Professor Trefethen nichts Unmögliches verlangte. Im Laufe des Kapitels werden wir die Existenz dann tatsächlich auch beweisen können (siehe S. 34).

Fassen wir den Grenzwert als das Integral

$$S = \int_0^1 x^{-1} \cos(x^{-1} \log x) dx \tag{1.1}$$

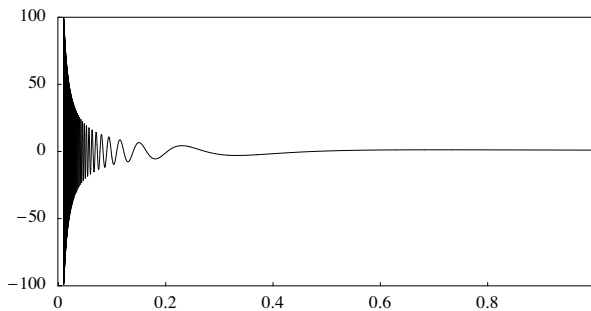


Abb. 1.1. Ein Integrand mit einem verwickelten Schwanz.

auf, so liegen zwei Eigenheiten vor, die jede für sich genommen die numerische Integration bereits schwierig macht:

- Der Integrand ist in der Nähe der linken Grenze unbeschränkt.
- Der Integrand oszilliert unendlich oft im Integrationsintervall.

Für den Rest dieses Kapitels fassen wir Integrale wie (1.1) als uneigentliche Riemann-Integrale auf, da das entsprechende Lebesgue-Integral in Ermangelung absoluter Integrierbarkeit nicht existiert.

1.2 Allgemeine oszillatorische Integrale

Wir beginnen mit der Betrachtung allgemeiner Integrale $\int_a^b g(x) dx$, deren Integrand g unendliche viele Extremstellen im Intervall (a, b) hat. Solche Integrale verweigern sich den üblichen Quadraturformeln (Gauß-Quadratur usw.), die ein polynomartiges Verhalten von g voraussetzen, und stürzen selbst jene automatischen Integratoren in Schwierigkeiten, die das Integrationsintervall adaptiv zerlegen.

Zur Vereinfachung wollen wir annehmen, dass a der einzige Häufungspunkt der Extremstellen von g ist. Die Grundstrategie für solche unendlich oszillatorischen Integrale (die auf Longman [Lon56] zurückgeht; siehe auch [Eva93, Kap. 4]) wird zwar üblicherweise für unendliche Integrationsbereiche formuliert (siehe [PdDKÜK83, S. 80]), aber endliche Intervalle können genauso behandelt werden:

1. Wähle eine Folge $a_k \rightarrow a$ mit $b = a_0 > a_1 > a_2 > a_3 > \dots$. Diese Folge zerlegt $(a, b]$ in die Teilintervalle $(a_k, a_{k-1}]$, $k = 1, 2, 3, \dots$
2. Integriere g über jedes der Teilintervalle. Dem liegt die Idee zugrunde, dass für eine hinreichend feine Unterteilung der Integrand g sich auf den einzelnen Teilintervallen nicht länger unerfreulich verhält und dort dann jedes vernünftige numerische Verfahren funktionieren wird. Setze

$$s_k = \int_{a_k}^{a_{k-1}} g(x) dx.$$

3. Summiere die unendliche Reihe $\sum_{k=1}^{\infty} s_k$. Das Resultat ist der gesuchte Wert des Integrals.

Man beachte, dass eine Singularität in a (wie sie in unserem Fall vorliegt) das Problem nicht schwieriger macht: Die Unterteilung häuft sich in a , was stets ein guter Weg zur Beherrschung einer Singularität ist. (Eine Singularität in b hätte dem Problem hingegen eine weitere Schwierigkeit hinzugefügt.)

Damit diese Strategie funktioniert, müssen die Reihenglieder s_k entweder sehr schnell gegen Null konvergieren (was selten genug passiert) oder sich hinreichend harmlos verhalten, so dass ein Algorithmus zur Extrapolation der Folge der Teilsummen anwendbar ist. Sinnvolle Möglichkeiten zur Wahl der Unterteilungspunkte a_k sind:

- Die Nullstellen von g . Das war die ursprüngliche Wahl von Longman.
- Die Extremstellen von g .
- Ist $g(x) = f(x)w(x)$ mit einem monotonen w , so ist es gewöhnlich einfacher, und auch effektiver, die Extremstellen von f statt derjenigen von g zu verwenden.

Im Fall einer Funktion über dem Intervall $[a, \infty)$, für welche der Abstand aufeinanderfolgender Nullstellen sich einem konstanten Wert h nähert, hat Lyness [Lyn85] dargelegt, dass es auf das Finden der Nullstellen nicht ankommt: Wählt man die Unterteilungspunkte äquidistant im Abstand h , so funktioniert die Methode immer noch. Lyness erwähnt zwar nicht wirklich die Verwendung der Extremstellen anstatt der Nullstellen, kommt dem aber nahe, wenn er die Mitte zwischen aufeinanderfolgenden Nullstellen des Integranden als Unterteilungspunkte vorschlägt.

Wir sollten noch etwas dazu sagen, wie heikel die Verwendung der Nullstellen von g als Unterteilungspunkte ist. Die Schwierigkeit entsteht, weil (anders als bei der Menge der Extremstellen) die Menge der Nullstellen nicht invariant gegen die Verschiebung des Graphen von g nach oben oder unten ist. Nur eine einzige Position führt zu Nullstellen in regelmäßigen Abständen; alle anderen führen auf ein humpelndes Muster von abwechselnd zu kurzen und zu langen Schritten. Also sollten wir eigentlich nicht die Nullstellen von g verwenden, sondern die Lösungen der Gleichung $g(a_k) = c$ für das korrekte c . In vielen Fällen ist die Wahl von c offensichtlich: Wenn der Wert von g an den Extremstellen grundsätzlich ± 1 ist, dann ist $c = 0$ offenbar sinnvoll; ist jedoch $\lim_{x \rightarrow a} g(x) = g_0$, dann ist die Wahl $c = g_0$ der einzige Weg, um überhaupt zu einer unendlichen Folge s_k zu gelangen.

Im allgemeinen ist die Verwendung der Extremstellen die bessere Wahl. Sie ist robuster und die Reihenglieder sind kleiner. Es ist zwar richtig, dass

wir anders als bei der Verwendung der Nullstellen nicht garantieren können, dass die Reihe alternierend ist, aber nichtsdestoweniger ist sie es in der Praxis gewöhnlich doch.

1.3 Dieses spezielle oszillatorische Integral

Kommen wir auf den vorliegenden Fall zurück, so können wir offensichtlich schlicht die Nullstellen von g verwenden. Mithin erfüllen die Unterteilungspunkte

$$a_k^{-1} \log a_k = - \left(k - \frac{1}{2} \right) \pi, \quad k = 1, 2, 3, \dots$$

Im Fall der Extremstellen ersetzen wir $k - 1/2$ durch k . Zur Lösung dieser nichtlinearen Gleichung werden wir in §1.5 mehr zu sagen haben.

Für $k = 1, 2, 3, \dots, 17$ berechnen wir die Nullstellen des Integranden und ermitteln

$$s_k = \int_{a_k}^{a_{k-1}} x^{-1} \cos(x^{-1} \log x) dx.$$

Danach wiederholen wir das Ganze mit den Extremstellen des Cosinusfaktors statt der Nullstellen.

Der Integrand ist über jedem einzelnen Teilintervall äußerst harmlos und so gut wie jedes numerische Verfahren wird daher funktionieren: Wir wählen die Romberg-Integration (siehe Anhang A, S. 299), die fester Bestandteil fast jedes einführenden Numeriklehrbuchs seit Henricis Klassiker [Hen64] von 1964 ist. Romberg-Integration liefert die ersten 17 Glieder der beiden Reihen mit insgesamt 2961 bzw. 3601 Funktionsauswertungen. Die Resultate (berechnet auf einer Maschine mit doppeltgenauer IEEE-Arithmetik, also mit etwas weniger als 16 signifikanten Ziffern) finden sich im linken Teil der Tabelle 1.1.

Beide Reihen konvergieren sehr langsam: Die Glieder der ersten verhalten sich wie $O(k^{-1})$, die der zweiten wie $O(k^{-2})$. Jedoch sind beide Reihen alternierend, was theoretisch bedeutsam ist, da man darauf einen Konvergenzbeweis aufbauen kann, siehe §1.8. Es ist aber auch praktisch bedeutsam, da so gut wie jeder brauchbare Extrapolationsalgorithmus funktionieren wird. So liefert eine schlichte Iteration der Δ^2 -Methode von Aitken (siehe Anhang A, S. 309) die im rechten Teil der Tabelle 1.1 aufgelisteten Glieder der beschleunigten Reihen.

In beiden Fällen scheinen die Reihenglieder rasant gegen Null zu streben und ihre Summen, 0.3233674316777786 bzw. 0.3233674316777784, werden vermutlich ein ganzes Stück genauer sein als die verlangten 10 Ziffern. Wir misstrauen der letzten Ziffer und erhalten

$$S \doteq 0.323367431677778,$$

ein in allen 15 gezeigten Ziffern korrektes Resultat.

Tabelle 1.1. Die ersten 17 Reihenglieder, die durch Integration zwischen den Nullstellen bzw. Extremstellen entstehen. Außerdem sind jene Glieder aufgeführt, die sich aus der Beschleunigung der Reihen mittels Iteration der Δ^2 -Methode von Aitken ergeben.

s_k zw. Nullstellen	s_k zw. Extremstellen	Beschl. (zw. Nullst.)	Beschl. (zw. Extremst.)
0.5494499236517820	0.3550838448097824	0.5494499236517820	0.3550838448097824
-0.3400724368128824	-0.0431531965722963	-0.2100596710635879	-0.0384770911855535
0.1905558793876738	0.0172798308777912	-0.0078881065973032	0.0076628418755175
-0.1343938122973787	-0.0093857627537754	0.0328626109424927	-0.0009933325213392
0.1043876588340859	0.0059212509664411	-0.0408139994356358	0.0001062999924531
-0.0855855139972820	-0.0040870847239680	-0.0002275288448133	-0.0000160316464409
0.0726523873218591	0.0029962119905360	0.0000467825881499	0.0000010666342681
-0.0631902437669350	-0.0022935621330502	-0.0000029629978848	-0.0000001739751838
0.0559562976074883	0.0018138408858293	0.0000004098092536	0.0000000090009066
-0.0502402983462061	-0.0014714332449361	-0.0000000298386656	-0.0000000013911841
0.0456061367869889	0.0012183087499507	0.0000000036996310	0.0000000000923884
-0.0417708676657172	-0.0010257939520507	-0.0000000002614832	-0.0000000000089596
0.0385427179214103	0.0008758961712063	0.0000000000282100	0.0000000000011693
-0.0357870041817758	-0.0007568532956294	-0.0000000000024653	-0.0000000000000529
0.0334063038418840	0.0006607085396203	0.0000000000001224	0.0000000000000079
-0.0313283690781963	-0.0005819206203306	-0.0000000000000221	-0.0000000000000007
0.0294984671675406	0.0005165328013114	-0.0000000000000018	-0.0000000000000000

1.4 Komplexe Integration

Die Kenntnis der analytischen Fortsetzung einer reellen Funktion, von der reellen Achse weg in die komplexe Ebene hinein, ist von unschätzbarem Wert. Sie liefert numerische Methoden von großer Kraft und Vielseitigkeit.

Unter Verwendung der Euler'schen Formel $e^{ix} = \cos x + i \sin x$ erhalten wir

$$\begin{aligned}
 S &= \int_0^1 \operatorname{Re} \left(\frac{e^{(i \log x)/x}}{x} \right) dx = \int_0^1 \operatorname{Re} \left(e^{((i/x)-1) \log x} \right) dx \\
 &= \int_0^1 \operatorname{Re} \left(x^{i/x-1} \right) dx.
 \end{aligned}$$

Nach dem Cauchy'schen Integralsatz hängt das Kurvenintegral zwischen zwei Punkten der komplexen Ebene nicht vom Integrationsweg ab, solange jedenfalls der Weg und seine betrachtete Alternative beide in einem einfach zusammenhängenden Gebiet liegen, in welchem die Funktion analytisch ist. Somit dürfen wir oben die reelle Variable x durch eine komplexe Variable z ersetzen, wenn wir sorgfältig einen geeigneten Integrationsweg wählen.

Wie man der perspektivischen Ansicht in Abb. 1.2 entnimmt, ist in unserem Fall der Integrand äußerst harmlos, sobald wir uns von der reellen

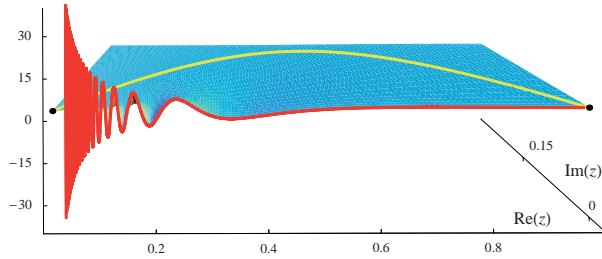


Abb. 1.2. Der Integrand des Problems 1 in der komplexen Ebene. Die rote Linie zeigt die ursprüngliche Formulierung, in welcher der Integrationsweg entlang der reellen Achse verläuft. Die gelbe Linie zeigt den Weg, welchen wir für die Integrationsmethode im Komplexen verwenden.

Achse weg in den positiven Quadranten begeben. Ein brauchbarer Integrationsweg wendet der reellen Achse sofort den Rücken zu, um dann später im Punkt (1,0) zu ihr zurückzukehren.

Wenn wir den Weg durch $z = z(t)$, $t \in [a, b]$, mit $z(a) = 0$ und $z(b) = 1$ parametrisieren, erhalten wir

$$S = \operatorname{Re} \int_a^b z(t)^{i/z(t)-1} z'(t) dt. \tag{1.2}$$

Auch dieses Integral ist ein uneigentliches Riemann-Integral entlang von $z(t) = t$, was dem Integrationsweg der ursprünglichen Problemformulierung entspricht, aber es wird, sofern wir den Integranden in $t = 0$ stetig fortsetzen, zu einem eigentlichen Integral entlang eines Wegs in der Art von Abb. 1.2.

Wir betrachten beispielsweise die einfache Parabel

$$z(t) = t + it(1 - t), \quad z'(t) = 1 + i - 2it, \quad t \in [0, 1]. \tag{1.3}$$

Abbildung 1.3 zeigt eine Darstellung des Realteils des Integranden in (1.2). Die Oszillationen kann man zwar noch unterm Mikroskop sehen, aber die Amplitude verschwindet zusehends mit $t \rightarrow 0$. Diese Funktion eignet sich unmittelbar zur Integration durch vorgefertigte Software.

Eine Sitzung mit Octave

```
>> function y=func(t);
>> z=t+i*t.*(1-t); y=real(z.^(i./z-1).*(1+i*(1-2*t)));
>> endfunction
>> quad('func',0,1)
```

ans = 0.323367431677773

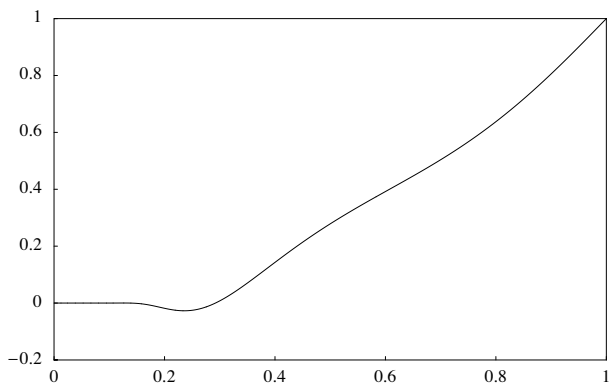


Abb. 1.3. Integrand entlang des Wegs $z(t) = t + it(1 - t)$.

Es ist ermutigend, dass die ersten 14 Ziffern mit dem früher berechneten Ergebnis übereinstimmen. Wir wollen es jetzt mit mehr Ziffern versuchen. Dazu verwenden wir die Langzahlarithmetik von PARI/GP. Hier müssen wir den Fall $t = 0$ extra behandeln, da PARI/GP anders als Octave eine geschlossene Quadraturformel benutzt (d.h. eine, die den Integranden in den Endpunkten des Integrationsintervalls auswertet).

Eine Sitzung mit PARI/GP

```
? func(t) = if(t==0,return(0),\
  z = t+I*t*(1-t); real(z^(I/z-1)*(1+I*(1-2*t))))
? intnum(t=0,1,func(t))
```

```
0.3233674316777787613993700868
```

Tatsächlich können wir durch eine Rechnung in noch längerer Arithmetik bestätigen, dass $S \doteq 0.32336743167777876139937008$.

1.5 Lambert'sche W-Funktion

Um die Zerlegung von $(0,1]$ für die Unterteilungsmethode von §1.2 zu finden, müssen wir für die Punkte a_k mehrere Gleichungen der Form

$$a_k^{-1} \log a_k = -b_k$$

lösen. Das ist zwar keine besonders schwierige Aufgabe, da die linke Seite monoton ist und man immer Bisektion verwenden kann, aber es stellt sich heraus, dass sich die Lösung mit Hilfe einer speziellen Funktion, die es

verdiente besser bekannt zu sein, in geschlossener Form schreiben lässt. Es sei $z = b_k$ und $w = -\log a_k$, dann gilt

$$we^w = z. \quad (1.4)$$

Für $z \in [0, \infty)$ hat diese Gleichung eine eindeutige reelle Lösung $w = W(z)$.

Die Funktion $-W(-z)$ war bereits Euler bekannt, der sie als konfluenten Fall einer Funktion zweier Variablen erhielt, die zuvor von Lambert untersucht worden war. Obwohl sich die Geschichte der Funktion W bis zum Jahr 1758 zurückverfolgen lässt, taucht sie im berühmten *Handbook of Mathematical Functions* [AS84] nicht auf. Der Name *Lambert'sche W-Funktion*, unter dem sie neuerdings bekannt ist, wurde ihr in den frühen 1990er Jahren von den Entwicklern von Maple gegeben.¹ Einen reizvollen Überblick über Geschichte, Eigenschaften und Anwendungen der Lambert'schen W -Funktion liefert der Aufsatz von Corless et al. [CGH⁺96], welcher auch unsere Quelle für die im folgenden aufgeführten Fakten ist.²

Euler kannte die Maclaurin'sche Entwicklung von W :

$$W(z) = z - \frac{2^1}{2!}z^2 + \frac{3^2}{3!}z^3 - \frac{4^3}{4!}z^4 + \dots \quad (1.5)$$

Für $z_0 \neq 0$ lässt sich die Taylorentwicklung von $W(z)$ in Potenzen von $z - z_0$ leicht finden, da die Ableitungen von W durch

$$\frac{d^n}{dz^n} W(z) = \frac{p_n(W(z))}{e^{nW(z)}(1+W(z))^{2n-1}}, \quad n = 1, 2, 3, \dots, \quad (1.6)$$

gegeben sind, wobei die Polynome p_n die Beziehungen $p_1(w) = 1$ und

$$p_{n+1}(w) = (1 - 3n - nw)p_n(w) + (1 + w)p_n'(w), \quad n = 1, 2, 3, \dots$$

erfüllen. Es sei angemerkt, dass p_n ein Polynom vom Grad $n - 1$ ist. Man sollte der Versuchung widerstehen, den Ausdruck (1.6) durch Substitution von $e^{W(z)} = z/W(z)$ zu „vereinfachen“, da die Formel dann völlig unnötig eine hebbare Singularität bekäme.

Einige Programmierumgebungen stellen Implementierungen der Lambert'schen W -Funktion zur Verfügung. Andererseits ist es aber einfach, ein eigenes Unterprogramm zu schreiben. In [CGH⁺96] findet sich der Vorschlag, die nichtlineare Gleichung (1.4) iterativ zu lösen, aber eine Kombination von Iteration mit der Taylorentwicklung

$$W(z) = W(z_k) + \frac{1}{1!}(z - z_k)W'(z_k) + \frac{1}{2!}(z - z_k)^2W''(z_k) + \dots$$

¹ Die Entwickler von Mathematica entschieden sich für den Namen `ProductLog`.

² Exzellenter Ausgangspunkt für Informationen zur Lambert'schen W -Funktion ist ebenso <http://www.apmaths.uwo.ca/~rcorless/frames/PAPERS/LambertW/>.

ist sogar noch einfacher. Für eine gegebene Approximation w_k berechnen wir $z_k = w_k e^{w_k}$, so dass also $w_k = W(z_k)$. Danach addieren wir solange weitere Glieder der Taylorentwicklung, bis keine Verbesserung mehr erzielt wird oder ein zusätzliches Glied nicht länger der Mühe wert wäre. An welcher Stelle genau der Abbruch erfolgen sollte, wird von den Details der Implementierung abhängen, insbesondere davon, um wieviel teurer die Berechnung von e^z im Vergleich zu Multiplikation und Division ist.

Man beachte, dass der Faktor $e^{W(z_k)}$ in (1.6) bereits während der Berechnung von z_k ermittelt wurde. Wenn wir den Nenner $q_n(W(z)) = e^{W(z)n}(1+W(z))^{2n-1}$ abspeichern, so können wir jede weitere Ableitung in $n+1$ Multiplikationen und einer Division berechnen. Natürlich sollten wir die Koeffizienten der Polynome p_n vorab bis zum höchsten benötigten Grad aufgestellt haben.

Wie in [CGH⁺96] bemerkt wird, braucht nur die letzte Iteration in der vollen verlangten Genauigkeit ausgeführt zu werden. In unserer PARI/GP-Implementierung der Auswertung von W auf d Ziffern verwenden wir acht Glieder der Taylorentwicklung mit einem Anfangswert, der auf $d/7.5$ Ziffern genau ist und aus einem rekursiven Aufruf des Programms stammt. Auf der untersten Ebene verwenden wir den Startwert $w_0 = \log(1+z)$. Dies ist völlig sachgerecht für die Auswertung von W als reellwertiger Funktion von $z \geq 0$. Bei sehr hohen Genauigkeiten benötigt unsere Routine weniger Zeit als zwei Auswertungen von e^z . Für die Feinheiten der Auswertung von W als mehrblättriger komplexer Funktion verweisen wir auf [CGH⁺96].

1.6 Transformation auf eine Halbachse

Als Faustregel kann man sich merken, dass die Integration beschränkter Funktionen über unendlichen Intervallen einfacher ist als die Integration unbeschränkter Funktionen über endlichen Intervallen. Die einfachste Transformation in dieser Hinsicht ist $u = 1/x$ und führt auf ein Integral, das mit der Kurvenintegralmethode gelöst werden kann. Eine weitere naheliegende Transformation ist $t = -\log x$, welche das folgende Integral liefert:

$$S = \int_0^\infty \cos(te^t) dt. \quad (1.7)$$

Auf den ersten Blick scheint die Transformation nichts gebracht zu haben, da eine Integration zwischen den Nullstellen oder den Extremstellen genau die gleichen unendlichen Reihen wie zuvor liefert. Auch sieht der Integrand nur unwesentlich weniger hässlich aus (siehe Abb. 1.4).

Der transformierte Integrand besitzt jedoch eine einfachere Form und eignet sich daher besser für analytische Manipulationen. Insbesondere erhalten wir mittels partieller Integration äquivalente Ausdrücke, in denen

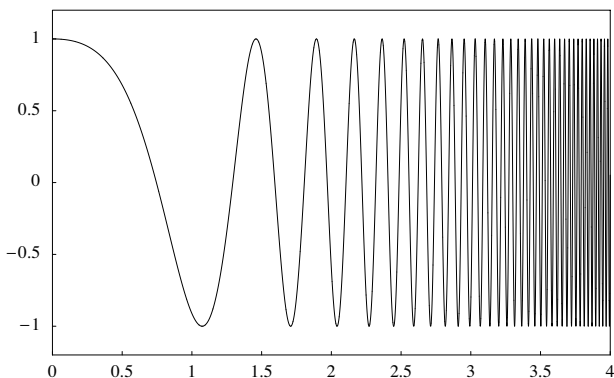


Abb. 1.4. Integrand nach der Transformation $t = -\log x$.

der Integrand mit wachsendem t rasch abfällt. Für die ersten paar Terme kann noch direkt mit (1.7) gearbeitet werden; möchte man aber zu einer allgemeinen Formel gelangen, so ist es einfacher, das Integral $\int_0^\infty \cos(xe^x) dx$ als $\operatorname{Re} \int_0^\infty e^{iW^{-1}(x)} dx$ aufzufassen. Dies erlaubt uns, aus einem raffinierten Trick zur wiederholten partiellen Integration Vorteil zu schlagen.

Es sei I_0 ein Integral der Form $I_0 = \int e^{cg(x)} dx$ mit $f = g^{-1}$, d.h. $f(g(x)) = x$ für alle x im Integrationsbereich. Aufgrund der Identität $g'(x)f'(g(x)) = 1$ erhalten wir

$$I_0 = \int f'(g(x))g'(x)e^{cg(x)} dx = c^{-1}f'(g(x))e^{cg(x)} - c^{-1}I_1,$$

$$\begin{aligned} I_1 &= \int \frac{d}{dx}(f'(g(x)))e^{cg(x)} dx = \int f''(g(x))g'(x)e^{cg(x)} dx \\ &= c^{-1}f''(g(x))e^{cg(x)} - c^{-1}I_2, \end{aligned}$$

$$I_2 = \int \frac{d}{dx}(f''(g(x)))e^{cg(x)} dx = \int f'''(g(x))g'(x)e^{cg(x)} dx$$

sowie allgemein

$$\begin{aligned} \int e^{cg(x)} dx &= -e^{cg(x)} \left(-c^{-1}f'(g(x)) + (-c)^{-2}f''(g(x)) \right. \\ &\quad \left. + \dots + (-c)^{-k}f^{(k)}(g(x)) \right) + (-c)^{-k} \int f^{(k+1)}(g(x))g'(x)e^{cg(x)} dx. \end{aligned}$$

Im vorliegenden Fall haben wir $c = i$, $f(y) = W(y)$, $g(x) = xe^x$, und alle Ableitungen von f verschwinden, wenn das Argument gegen unendlich

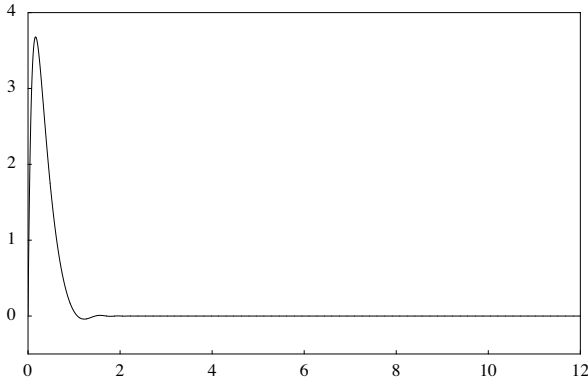


Abb. 1.5. Integrand des Restterms nach dreifacher Anwendung partieller Integration.

geht. Setzen wir die Ableitungen $W^{(k)}(0)$ ein, die wir der Maclaurin'schen Entwicklung (1.5) entnehmen, so bleibt uns

$$\int_0^\infty e^{ixe^x} dx = i1^0 - i^22^1 + i^33^2 - \dots + i^k(-k)^{k-1} + i^k \int_0^\infty W^{(k+1)}(xe^x)e^x(1+x)e^{ixe^x} dx.$$

Brechen wir diese Entwicklung bei $k = 3$ ab und nehmen den Realteil, so finden wir mit Hilfe von (1.6), dass

$$S = 2 - \int_0^\infty \left(\frac{6}{(1+t)^3} + \frac{18}{(1+t)^4} + \frac{25}{(1+t)^5} + \frac{15}{(1+t)^6} \right) e^{-3t} \sin(te^t) dt.$$

Der transformierte Integrand hat zwar eine unangenehm große Steigung im Ursprung und formt eine hässliche Spitze, fällt aber recht schnell ab (siehe Abb. 1.5). Für 16 Ziffern reicht es, den Bereich $0 \leq t \leq 12$ zu betrachten. Romberg-Integration ist noch angemessen; der Integrand ist jedoch nicht mehr so einfach und wir benötigen 131 073 Funktionsauswertungen, um die Resultate der bisherigen Methoden zu bestätigen.

1.7 Darstellung als divergente Reihe

Setzen wir den Prozess der partiellen Integration unbegrenzt fort und ignorieren den Restterm, so erhalten wir die unendliche Reihe

$$2^1 - 4^3 + 6^5 - 8^7 + \dots \tag{1.8}$$

Diese Reihe ist zwar von niederschmetternder Einfachheit, aber auch gewaltig divergent. Derartige divergente Reihen mit einfachen analytischen

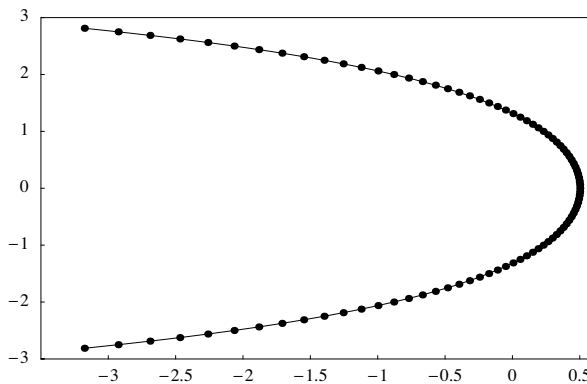


Abb. 1.6. Integrationsweg für (1.10). Die Stützstellen sind bezüglich der Parametrisierung äquidistant gewählt.

Termen als Glieder können formal in andere Ausdrücke umgewandelt werden, die gewöhnlich äquivalent zur ursprünglich gesuchten Größe sind. Solch ein Vorgehen liefert natürlich nur eine heuristische Motivation für eine Identität, die dann noch einer Rechtfertigung durch andere Mittel bedarf.

Universelle Methoden zur Konvergenzbeschleunigung (wie die in Anhang A vorgestellten) beeindruckt diese Reihe nur wenig. Wie im Fall der Kurvenintegralmethode von §1.4 führt uns jedoch der Umstand, einen analytischen Ausdruck der Reihenglieder zu besitzen, zu einer besseren Methode. Eine konvergente alternierende Reihe erfüllt die Identität

$$\sum_{k=1}^{\infty} (-1)^{k-1} a_k = \frac{i}{2} \int_{\mathcal{C}} f(z) \csc(\pi z) dz \quad (1.9)$$

unter folgenden Voraussetzungen (siehe Theorem 3.7):

1. Der Integrationsweg \mathcal{C} zerlegt die komplexe Ebene in das offene Gebiet Ω_R , das die Punkte $1, 2, 3, \dots$ enthält, und das offene Gebiet Ω_L , das die übrigen ganzen Zahlen enthält, wobei \mathcal{C} gegen den Uhrzeigersinn um Ω_R verläuft.
2. Es gilt $a_k = f(k)$, wobei f in Ω_R analytisch ist und geeignet für $z \rightarrow \infty$ abfällt.

Die Gestalt der divergenten Reihe (1.8) legt daher nahe, das Kurvenintegral

$$\frac{i}{2} \int_{\mathcal{C}} (2z)^{2z-1} \csc(\pi z) dz \quad (1.10)$$

zu betrachten.

Tabelle 1.2. Approximation des Kurvenintegrals (1.10) mittels Trapezsumme der Schrittweite h . Zudem wird die Anzahl der Auswertungen des Integranden aufgeführt.

h^{-1}	Anz. Auswert.	Approximation von S
2	11	0.334486489324265
4	23	0.323398015778690
8	45	0.323367431981211
16	91	0.323367431677779
32	181	0.323367431677779

Für die rasche Konvergenz des Kurvenintegrals eignet sich der parametrisierte Weg (siehe Abb. 1.6)

$$(x, y) = \left(1 - \frac{1}{2} \cosh t, -t\right), \quad -\infty < t < \infty.$$

Für 16 Dezimalziffern reicht die Berücksichtigung des Bereichs $|t| \leq 2.8$. Werten wir das Kurvenintegral mittels der Trapezsumme³ zur Schrittweite h in der Variablen t aus, so erhalten wir die Resultate in Tabelle 1.2. Es ist ermutigend festzustellen, dass die Resultate mit den bereits erhaltenen übereinstimmen.

Obwohl die Übereinstimmung auf 14 Ziffern hochgradig suggestiv ist, haben wir doch keinen Beweis der Gültigkeit dieser Methode. Im allgemeinen gibt es unendlich viele analytische Funktionen, die vorgegebene Werte in den positiven ganzen Zahlen interpolieren, und wir können nur im Fall konvergenter Reihen garantieren, dass (1.9) von der Wahl des Interpolanten unabhängig ist. Als Kuriosität und Herausforderung formulieren wir daher die *Vermutung*: Das uneigentliche Riemann-Integral (1.1) und das Kurvenintegral (1.10) besitzen den gleichen Wert.

1.8 Transformation auf ein Fourierintegral

Die am besten verstandenen unendlich-oszillatorischen Integrale sind Fourierintegrale. In §9.5 werden wir diese Integrale in größerer Ausführlichkeit diskutieren; hier beschränken wir uns darauf zu zeigen, dass das vorliegende Problem auch auf diesem Weg gelöst werden kann.

Das wesentliche Merkmal eines Fourierintegrals ist das Auftreten eines Faktors der Form $\sin(at + b)$. Im vorliegenden Fall haben wir den Faktor

³ In §3.6.1 erklären wir, warum Trapezsummen so außerordentlich gute Resultate für Kurvenintegrale analytischer Funktionen liefern: Die Konvergenz erfolgt *exponentiell schnell*.

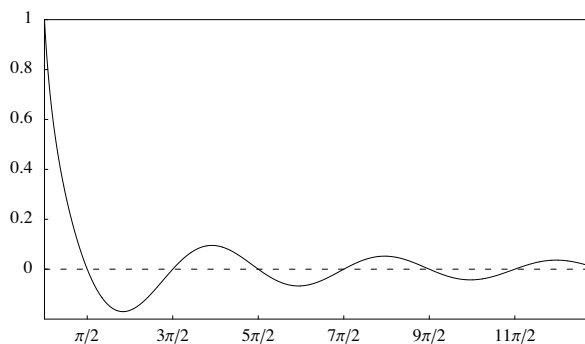


Abb. 1.7. *Integrand nach Transformation mit der Lambert'schen W-Funktion.*

$\cos(te^t)$, so dass wir die Transformation $u = te^t$, das heißt also $t = W(u)$, benötigen. Das ergibt den Ausdruck⁴

$$S = \int_0^{\infty} W'(u) \cos u \, du. \quad (1.11)$$

W' kann leicht aus (1.6) berechnet werden. Was oszillatorische Integranden über unendlichen Intervallen anbetrifft, verhält sich dieser Integrand völlig harmlos, wie der Graph in Abb. 1.7 zeigt.

In diesem Stadium unserer Diskussion können wir endlich beweisen, dass das ursprüngliche Integral konvergiert. Da die Nullstellen jetzt äquidistant sind, reicht es zu zeigen, dass der Vorfaktor von $\cos u$ für hinreichend großes u monoton gegen Null fällt. In diesem Fall liefert die Integration zwischen den Nullstellen eine alternierende Reihe mit monoton gegen Null strebenden Gliedern. Tatsächlich ist die Funktion $W'(u) = e^{-W(u)} / (1 + W(u))$ ein Produkt zweier monoton fallender Funktionen und als solche daher selbst monoton fallend.

Die Methode von Oura und Mori [OM99] (siehe §9.5 für eine ausführliche Beschreibung) arbeitet mit der Strategie des iterativen Verdoppelns des Parameters M wie in Tabelle 1.3 gezeigt. Wie wir sehen, sind sowohl die Anzahl der Funktionsauswertungen als auch die Anzahl korrekter Ziffern in etwa zu M proportional. Gestützt auf diese Beobachtung verlassen wir

⁴ Die Routine `NIntegrate` zur numerischen Integration in Mathematica stellt für solche Fourierintegrale eine Option zur Verfügung, die im wesentlichen die Methode von Longman aus §1.2 implementiert. Problem 1 wird dann auf 13 korrekte Ziffern mittels des folgenden kurzen Programmtexts gelöst:

```
NIntegrate[ProductLog'[u] Cos[u], {u, 0, ∞}, Method → Oscillatory]
```

```
0.3233674316777859
```

Tabelle 1.3. *Approximation von S durch Anwendung der Methode von Ooura und Mori auf das Integral (1.11). Zusätzlich wird die Anzahl der Auswertungen des Integranden aufgeführt.*

M	Anz. Auswert.	Approximation von S
2	6	0.333683545675313
4	13	0.323277481884531
8	26	0.323367418739372
16	53	0.323367431677750
32	111	0.323367431677779

uns auf alle Ziffern bis auf die letzte (die von Rundungsfehlern verschmutzt sein könnte) und stellen fest, dass

$$S \doteq 0.32336743167777.$$

1.9 Auf dem Weg zu 10 000 Ziffern

Hochgenaues Rechnen ist eine Kunst, die man am besten Experten überlässt. Glücklicherweise haben solche Experten proprietäre Software wie Maple und Mathematica, oder freie Software wie GNU MP, CLN, GiNaC, PARI/GP und zweifellos viele weitere mehr geschrieben. Soweit es die grundlegende Arithmetik und die elementaren Funktionen betrifft, können wir uns darauf verlassen, dass die Experten ihr Bestes gegeben haben. Auch wenn ein Experte es vielleicht besser als ein anderer Experte gemacht haben sollte, dürfte ein gewöhnlicher Nutzer solcher Pakete nicht darauf hoffen, dagegen antreten zu können.

Hier ist ein Beispiel von Zeitmessungen für einige handelsübliche Aufgaben auf meinem Computer. Für $x = \sqrt{3}$ führt Tabelle 1.4 die Laufzeiten in Uhrtakten (ein Uhrtakt ist die kleinste auf dem Computer messbare Zeit, etwa 0.002 Sekunden) auf, wie sie `ginsh` (eine einfache, interaktive und taschenrechnerartige Schnittstelle zu GiNaC) für drei Genauigkeitsstufen meldet. Zwei naheliegende Beobachtungen aus dieser Tabelle gelten im allgemeinen für alle Implementierungen mehrfach-genauer Arithmetik:

- Die Kosten einer Operation in d -ziffriger Arithmetik wachsen schneller als d wächst.
- Die Auswertung transzendenter Funktionen ist um ein vielfaches teurer als die arithmetischen Operationen.

Der zweite Band des Klassikers „The Art of Computer Programming“ von Donald Knuth [Knu81, §4.3.3] enthält eine umfassende Diskussion mehrfach-genauer Arithmetik.

Tabelle 1.4. Laufzeit (in Uhrtakten) verschiedener hochgenauer Berechnungen.

Ziffernanzahl:	10 000	20 000	40 000
$x \cdot x$	1	3	12
\sqrt{x}	3	11	49
$1/x$	6	24	70
$\cos x$	249	1038	3749
e^x	204	694	2168

Der übliche Weg zur Beurteilung der Komplexität einer d -ziffrigen Rechnung besteht im Auffinden eines Exponenten p , so dass die Zeit $t(d)$ für wachsendes d wie $O(d^p)$ skaliert. Im Fall der Multiplikation ist die bekannteste schnelle Methode die von Karatsuba mit $p = \log_2 3 \doteq 1.6$ – dies ist die von GiNaC verwendete Methode.

Knuth beschreibt mehrere Algorithmen, die zwei Zahlen in der Zeit $O(d^p)$ mit $p = 1 + \epsilon$ für beliebig kleines ϵ multiplizieren können, was asymptotisch optimal ist. Für relativ kleine Werte von d beschreibt jedoch ein Wert von p , der deutlich größer als 1 ist, das beobachtete Verhalten der optimalen Methoden besser. Der Übergangspunkt, von welchem an eine optimale Methode in der Praxis besser als die Methode von Karatsuba ist, liegt üblicherweise nahe bei $d = 10\,000$.

Es ist auch bekannt [Bre76], dass die meisten gebräuchlichen transzendenten Funktionen in der Zeit $O(M(d) \log d)$ ausgewertet werden können, wobei $M(d)$ die Zeit einer d -ziffrigen Multiplikation bezeichnet.

Das Übergewicht der Formel $O(d^p)$ bedeutet nicht, dass die tatsächliche Formel für $t(d)$ von so einfacher Form ist – das große O kehrt Faktoren wie $\log d$, $\log \log d$, usw., sowie einige möglicherweise recht große Konstanten unter den Teppich. Allerdings liefert uns die vereinfachte Formel eine Faustregel: Wenn d eine geometrische Progression bildet, dann sollte $t(d)$ es ungefähr auch tun. Man kann diese Argumentation sogar ohne Berechnung von p nutzen. Zum Beispiel haben wir im Fall von e^x die Zeiten $t(10\,000) = 204$ und $t(20\,000) = 694$, und prognostizieren daher, dass $t(40\,000)$ nahe bei $694^2/204 \doteq 2361$ liegen sollte, was auch tatsächlich der Fall ist. Da Terme der Form $\log d$ letztlich durch eine Potenz von d modelliert werden, neigt dieses Vorgehen zur Überschätzung der benötigten Zeit, was grundsätzlich erst einmal nicht schlecht ist.

Wir wollen nun das Verhalten der vielversprechenderen Algorithmen für Problem 1 im Hinblick darauf beurteilen, wie sehr die Laufzeit für größeres d wächst.

Integration über Teilintervalle und Extrapolation (§1.3)

Nehmen wir an, wir fänden zum Nulltarif eine Quadraturformel, die das Integral über einem Teilintervall auf d Ziffern genau mit $O(d)$ Stützstellen auswerten könnte (das ist zwar optimistisch, aber ich werde darlegen, dass diese Methode ohnehin nicht konkurrenzfähig wäre), und wir kämen mit $O(d)$ Teilintervallen aus. Das führt auf $O(d^2)$ Auswertungen des Integranden.

Komplexe Integration (§1.4)

Wenn wir mit 10 000 Ziffern rechnen, können wir uns den Luxus einer adaptiven Quadraturroutine nicht leisten; wir benötigen etwas für die vorliegende Aufgabe Maßgeschneidertes. Die Grundtechnik ist folgende:

1. Wähle den Integrationsweg sorgfältig.
2. Parametrisiere den Weg so, dass die Trapezsumme auf der Parameterachse eine asymptotische optimale Quadraturformel ist, d.h. eine Formel, für welche die Anzahl der korrekten Ziffern etwa proportional zur Anzahl der Funktionsauswertungen ist (siehe §3.6.1).

Was im einzelnen zu tun ist, wollen wir hier nicht weiter ausführen; stattdessen verweisen wir auf die ausführliche Diskussion vergleichbarer Techniken, angewendet auf weitere Probleme, in §§3.6 und 9.4. Nach so manchem Experiment empfehlen wir den folgenden parametrisierten Weg zur Auswertung von (1.2) auf eine Genauigkeit von d Ziffern:

$$z = \frac{\pi e^t}{\pi e^t + 2 - 2t} + \frac{2i}{\cosh t}, \quad \log(cd) < t < 1 + \log d,$$

mit $c \approx 1.53$; es taugt dann $2/(d+1)$ als Schrittweite der Trapezsumme.

Die divergente Reihe (§1.7)

Wir können diese Methode zwar nicht als unser Hauptinstrument einsetzen, aber wenn ihr Resultat mit dem unserer besser fundierten Methoden übereinstimmt, nutzt es und stärkt unsere Gewissheit.

Kurvenintegrale über unendlichen Integrationswegen benötigen typischerweise eine Schrittweite, die umgekehrt proportional zur Anzahl der verlangten Ziffern ist. Der doppelt-exponentielle Abfall des Integranden in (1.10) als Funktion von t (ein erster einfach-exponentieller Abfall stammt von der Parametrisierung, ein weiterer von der Exponentiation im Integranden) garantiert, dass die benötigte Anzahl von Stützstellen nur geringfügig schneller als der Kehrwert der Schrittweite wächst. Also können wir die Anzahl N der tatsächlich benutzten Stützstellen durch $N = O(d^{1+\epsilon})$ modellieren, wobei ϵ klein ist. (Ein genaueres Modell wäre $d \approx cN / \log N$, das

wir bei Kenntnis von c mittels eines Zweigs der Lambert'schen W -Funktion, der nicht durch (1.5) gegeben ist, nach N auflösen könnten – aber das ist eine andere Geschichte.)

An jeder dieser Stützstellen müssen wir eine Hyperbelfunktion, einen komplexen Kosekans und eine komplexe Potenz auswerten. Wir können davon ausgehen, dass dies alles zwar teurer als eine Auswertung von $\cos(xe^x)$ für reelle Argumente x ist, jedoch nicht soviel teurer, dass es die Vorteile von $O(d^{1+\epsilon})$ gegenüber $O(d^2)$ Funktionsauswertungen aufwäge.

Methode von Oura und Mori (§1.8)

Wie die beiden vorangehenden Methoden ist auch dies ein doppelt-exponentielles Integrationsverfahren, so dass weitgehend die gleiche Argumentation Anwendung findet wie im Fall der komplexen Integration. Der Vorteil besteht jetzt darin, dass die transzendenten Funktionen in reellen Argumenten ausgewertet werden und wir daher einen kleinen Laufzeitgewinn erwarten dürfen.

Wie wir gesehen haben, sollte uns die Auswertung der Lambert'schen W -Funktion nicht mehr als zwei Auswertungen von e^x kosten. Experimentell beobachten wir, dass $M = 1.5d$ für eine Genauigkeit von d Ziffern ausreicht.