

HEWLETT-PACKARD

HP-34C

Programmierbarer wissenschaftlicher
Taschenrechner

Bedienungs- und
Programmierhandbuch



HEWLETT
PACKARD

Hewlett-Packard GmbH:
Berner Straße 117, Postfach 560140, 6000 Frankfurt 56, Tel. (0611) 5004-1

Hewlett-Packard S. A., Europa-Zentrale:
7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz, Tel.(022) 827000

HP-34C

Programmierbarer wissenschaftlicher
Taschenrechner

Bedienungs- und Programmierhandbuch

August 1979



00034-90004

Printed in Singapore

"Der technische und wirtschaftliche Erfolg unseres Unternehmens kann nur gesichert werden, wenn wir unseren Kunden technisch überlegene Produkte anbieten, die einen echten Bedarf decken und einen dauerhaften Wert darstellen. Und wenn wir durch eine Vielzahl von Service-Leistungen sowie durch technische Beratung vor und nach dem Verkauf den Kunden in der Anwendung dieser Produkte unterstützen.

Erklärung über die Unternehmensziele
von Hewlett-Packard

Als die Ingenieure Hewlett und Packard im Jahre 1939 das Unternehmen gründeten, begannen sie mit einem technisch überlegenen Produkt – einem Tongenerator.

Heute liefern wir mehr als 4000 verschiedene Qualitätsprodukte, die für einige der anspruchsvollsten Kunden auf dem Weltmarkt konstruiert und gefertigt werden.

Seit 1972, als wir unseren ersten Taschenrechner vorstellten, haben wir weit über eine Million Einheiten verkauft. Zu den Anwendern gehören Nobelpreisträger, Astronauten, berühmte Bergsteiger, Geschäftsleute, Ärzte, Wissenschaftler und Studenten.

Jeder unserer Taschenrechner wird mit höchster Präzision hergestellt. Er hilft dem Anwender die Aufgaben seines Berufslebens zu meistern.

Sie decken somit einen wirklichen Bedarf und haben für den Kunden einen bleibenden Wert.

INHALTSVERZEICHNIS



DER PROGRAMMIERBARE WISSENSCHAFTLICHE TASCHENRECHNER HP-34C	7b
Tastenfeld und Speicher	7b
Verzeichnis der Tastenfunktionen	8
Verzeichnis der Programmfunctionen	12
ABSCHNITT 1: EINLEITUNG	15
Manuelle Lösung eines Problems	16
Programmierte Lösung eines Problems	17
Die Vorteile des Permanent-Speichers	19
ABSCHNITT 2: BESONDERE EIGENSCHAFTEN DES HP-34C	21
Tastenfeld	21
Speicherregister und Programmspeicher	22
Die Speicherregister des HP-34C	22
Abändern von Zahlen	23
Absolutwert	23
Ganzzahliger Anteil einer Zahl	24
Dezimaler Anteil einer Zahl	24
Mathematische Funktionen	25
Fakultät	25
Gammafunktion	26
Berechnung prozentualer Unterschiede	27
Statistik-Funktionen	28
Summationen	28
Löschen und Korrigieren von Eingabedaten	31
Mittelwert	32
Standardabweichung	34
Lineare Regression	36
Linearer Schätzwert	39
Korrelationskoeffizient	40
Vektor Addition und Subtraktion	40

ABSCHNITT 3: EINFÄCHE PROGRAMMIERUNG

Was ist ein Programm?	43
Warum ein Programm schreiben?	43
Drei Rechner-Modi	44
Programmspeicher	45
Tasten-Codes	47
Übungsaufgaben	48
Löschen des Programms	48
Erstellen eines eigenen Programms	49
Beginn eines Programms	49
Beenden eines Programms	50
Laden eines Programms	50
Ausführung eines Programms	52
Aufsuchen einer Marke	52
Ausführung der gespeicherten Programmzeilen	53
Automatische Speicherplatzzuteilung	55
Umwandlung von Speicherregistern in Programmspeicher	55
Umwandlung von Programmspeicher in Speicherregister	58
Verwendung von 	59
Ein drittes Programm	60
Unterbrechung der Programmausführung	62
Anhalten eines Programms	62
Kurzfristige Unterbrechung der Programmausführung	64
Programm-Stops	65
Marken	66
Flußdiagramm	67
Übungsaufgaben	71
Programmiertechniken	75
Horner-Schema	79
Weitere Anwendungen	82
Übungsaufgaben	82
ABSCHNITT 4: PROGRAMMKORREKTUR	83
Nicht speicherbare Operationen	83
Pythagoras-Programm	85
Schrittweise Ausführung eines Programms	86
Abändern eines Programms	88
Schrittweise Anzeige ohne Ausführung des Programms	89
Rücksprung zu Speicherzeile 000	90



Vorrücken zu einer bestimmten Speicherstelle	90
Einfügen von Programmschritten in längeren Programmen	92
Schrittweises Zurückrücken im Speicher	92
Ausführen des abgeänderten Programms	94
Löschen von Programmzeilen	97
Übungsaufgaben	97
ABSCHNITT 5: PROGRAMMVERZWEIGUNGEN, ENTSCHEIDUNGEN UND FLAGS	101
Unbedingte Sprünge und Programmschleifen	101
Übungsaufgaben	103
Vergleichsoperationen und bedingte Programmverzweigung	106
Übungsaufgaben	114
Flags	118
Verwendung von Flags	119
Übungsaufgabe	119
ABSCHNITT 6: UNTERPROGRAMME	125
Anwendungsbeispiele für Unterprogramme	132
Grenzen bei der Verwendung von Unterprogrammen	135
Verwendung von  am Ende des belegten Programmspeichers	137
ABSCHNITT 7: FORTGESCHRITTENE PROGRAMMIERUNG	139
Verwendung des I-Registers	139
Speichern einer Zahl im I-Register	139
Austausch der Inhalte von X und I	140
Inkrementieren und Dekrementieren des I-Registers	140
Gültigkeitsgrenzen von ISG und DSE	147
Übungsaufgabe	147
Verwendung des I-Registers zur Kontrolle der Anzeige, der Speicherregister und der Programmsteuerung	149
Steuerung des Anzeigeformates mit dem I-Register	150
Austausch der Inhalte von X und (i)	154
Indirektes Speichern und Zurückrufen von Daten	157
Steuerung von Programmverzweigungen und Unterprogrammen mit dem I-Register	161
Übungsaufgabe	167
Steuerung von Programmverzweigungen und Unterprogrammen durch Adressierung der Zeilenummer	169

ABSCHNITT 8: BESTIMMUNG DER LÖSUNGEN EINER GLEICHUNG	173
Verwendung von 	173
Wenn keine Lösung gefunden wird	179
Wahl der anfänglichen Schätzwerte	182
Die Arbeitsweise von 	186
Die Genauigkeit der Lösung	188
Auswertung der Ergebnisse	192
Verwendung von in Programmen	199
Zusätzliche Information über 	200

ABSCHNITT 9: NUMERISCHE INTEGRATION	201
Verwendung von 	201
Genauigkeit von 	207
Verwendung von in Programmen	211
Zusätzliche Informationen über 	212

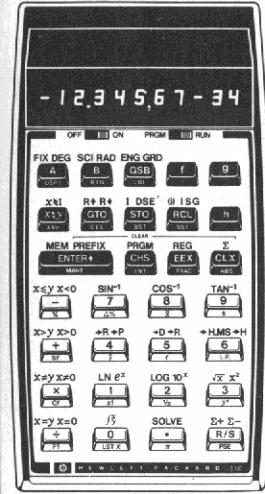
ANHANG A: FORTGESCHRITTENE VERWENDUNG VON 	213
Verwendung von mit Polynomen	213
Bestimmung mehrerer Nullstellen	213
Bestimmung von lokalen Extremwerten einer Funktion	223
Verwendung der Ableitung	223
Verwendung einer angenäherten Steigung	226
Iterative Berechnung	229
Begrenzung der Rechenzeit	231
Zählen der Iterationen	231
Festlegen einer Toleranzgrenze	232
Verwendung von und 	232

ANHANG B: DIE ARBEITSWEISE VON 	235
Genauigkeit, Fehlerabschätzung und Rechenzeit	236
Die Genauigkeit der zu integrierenden Funktion	240
Funktionen, die eine physikalischen Zusammenhang wiedergeben	241
Rundungsfehler bei internen Rechnungen	242

Fehlerschranke und Anzeigeformat	243
Berechnung von Integralen mit maximaler Genauigkeit	246
Der augenblickliche Näherungswert des Integrals	249
Umstände, die zu falschen Ergebnissen führen können	251
Faktoren, die die Rechenzeit verlängern	260
Unterteilung des Integrationsintervalls	262
Transformation der Variablen	266
ANHANG C: PFLEGE UND WARTUNG	269
ANHANG D: UNERLAUBTE OPERATIONEN	277
ANHANG E: STACK-LIFT UND LAST X	281

DER PROGRAMMIERBARE WISSENSCHAFTLICHE TASCHENRECHNER HP-34C

DER PROGRAMMIERBARE WISSENSCHAFTLICHE TASCHENRECHNER HP-34C



Der programmierbare wissenschaftliche Taschenrechner HP-34C

Automatischer Rechenregister-Stapel

T
Z
Y
X (Anzeige)

Last X

Programmspeicher

Programmspeicher fest	variabel
000-	071-
001-	072-
002-	073-
067-	207-
068-	208-
069-	209-
070-	210-

Die Grundstellung der Speicherzuteilung sieht 70 Zeilen Programmspeicher und 20 Speicherregister vor. Der Rechner wandelt bei Bedarf automatisch ein Register zur Zeit in sieben Programmzeilen um, wobei er mit Register R₉ anfängt und mit R₀ aufhört.

Speicherregister fest	variabel
I	R ₀
R ₁	R _{.0}
R ₂	R _{.1}
R ₃	R _{.2}
R ₄	R _{.3}
R ₅	R _{.4}
R ₆	R _{.5}
R ₇	R _{.6}
R ₈	R _{.7}
R ₉	R _{.8}

VERZEICHNIS DER TASTENFUNKTIONEN

Sobald eine der nachstehenden Funktionstaster des Tastenfeldes gedrückt wird, führt der Rechner die zugeordnete Operation aus. Eingetastete Zahlen und Rechenergebnisse werden angezeigt. Bis auf einige besonders angegebene Ausnahmen können alle hier aufgeführten Tastenoperationen sowohl von Hand über das Tastenfeld als auch im Rahmen eines gespeicherten Programms automatisch ausgeführt werden.

PRÄFIX-TASTEN

f Vortaste zur Auswahl der Funktion, deren goldfarbenes Symbol oberhalb der Taste steht.

g Vortaste zur Auswahl der Funktion, deren blaues Symbol oberhalb der Taste steht.

h Vortaste zur Auswahl der Funktion, deren schwarzes Symbol auf der abgeschrägten Vorderseite der Taste steht.

CLEAR PREFIX (nicht programmierbar) hebt die Wirkung einer teilweise eingetasteten Anweisung auf, wie z. B. **f**, **f SCI**, **g**, **h SF**, **STO** +, usw.

ZAHLENEINGABE

ENTER Trennt aufeinanderfolgende Zahlen bei der Eingabe. Dupliziert den Inhalt des **X**-Registers nach **Y**.

CHS Vorzeichenumkehr der angezeigten Zahl bzw. Exponenten.

EEX Exponenteneingabe: Zahl nach dieser Taste erscheint als Zehnerexponent

• Dezimalpunkt

ABÄNDERN VON ZAHLEN

INT Ganzzahligter Wert: Diese Taste beläßt nur den ganzzahligen Teil der angezeigten Zahl in der Anzeige.

FRAC Gebrochener Anteil: Darstellung des gebrochenen Anteils der angezeigten Zahl durch Abtrennen des ganzzahligen Anteils.

ABS Gibt den Absolutwert der angezeigten Zahl an.

UMORDNEN DER STACK-REGISTER

R↓ Zyklisches Vertauschen der Stack-Register-Inhalte nach oben).

R↑ Zyklisches Vertauschen der Stackregister nach unten).

XY Vertauscht die Inhalte von **X**- und **Y**-Register.

CLX Löscht das angezeigte **X**-Register.

STO Speichertaste: In Verbindung mit einer Zifferntaste wird die angezeigte Zahl im entsprechenden



SPEICHERN VON DATEN

Register (R₀ – R₉, R_{.0} – R_{.9}, I) gespeichert. Ermöglicht auch Registerarithmetik in Verbindung mit Arithmetiktasten.



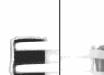
RCL Speicherabruf: Abruf eines gespeicherten Wertes aus einem der Register (R₀ – R₉, R_{.0} – R_{.9}, I) in das angezeigte **X**-Register.



LST X Abruf des letzten angezeigten Wertes vor der letzten Rechenoperation.



CLEAR REG Löscht alle Speicherregister (R₀ – R₉, R_{.0} bis R_{.9}).



WAHL DES ANZEIGEFORMATES



FIX Festkommaanzeige: Danach wird die entsprechende Zifferntaste für die gewünschte Stellenzahl gedrückt.



SCI Gleitkommaanzeige: Danach wird die entsprechende Zifferntaste für die gewünschte Stellenzahl gedrückt.



ENG Anzeige im technischen Format: Danach folgt eine Zifferntaste für die gewünschte Anzahl der Nachkommastellen.



DSP I Anzahl der Nachkommastellen entsprechend einer im **I**-Register enthaltenen Zahl (0–9).



MANT Mantisse (nicht programmierbar). Dient zur kurzzeitigen Anzeige aller 10 signifikanten Stellen der Mantisse der im **X**-Register stehenden Zahl.



PROZENTRECHNUNGEN

% Berechnung des prozentualen Unterschiedes zwischen dem Inhalt des **X**- und **Y**-Register.

% Berechnung des Prozentsatzes einer Zahl (% x von y).

MATHEMATISCHE FUNKTIONEN

+ - × ÷ Tasten für arithmetische Grundrechnungsarten.

vx Berechnet die Quadratwurzel der angezeigten Zahl.

x2 Berechnet das Quadrat der angezeigten Zahl.

xl Berechnet die Fakultät xl oder die Gammafunktion (x+1) der angezeigten Zahl.

vx Berechnet den Kehrwert der angezeigten Zahl.

PI Abruf der Konstanten (3,14159..) in das Anzeigeregister.

IB Berechnung des bestimmten Integrals **f(x)dx**. Der Ausdruck f(x) steht im Programmspeicher.

SOLVE Berechnung einer reellen Wurzel einer Gleichung **f(x)=0**. Der Ausdruck f(x) steht im Programmspeicher.

STATISTISCHE FUNKTIONEN

CLEAR S löscht die Statistik-Register R₀ bis R₅.

$\Sigma+$ Berechnet verschiedene Summen der Eingabewerte x und y in den Statistik-Registern R₀ bis R₅.

$\Sigma-$ Entfernt die Werte x und y aus der Summenbildung in den Statistik-Registern R₀ bis R₅.

\bar{x} Berechnet die Mittelwerte der x- und y-Werte.

S Berechnet die Stichproben-Standardabweichung für summierte x- und y-Werte.

LR Lineare Regression. Berechnet Achsen Schnittpunkt und Steigung für die lineare Funktion, die die beste Näherung für die mit **$\Sigma+$** summierten Werte darstellt. Der Wert des Y-Achsenabschnitts steht im X-Register. Der Wert der Steigung steht im Y-Register.

\hat{y} Lineare Schätzung. Berechnet zu einem gegebenen Wert x den zugehörigen Schätzwert y mittels der Methode der kleinsten Quadrate.

r Berechnet die Güte der Anpassung für die mit **$\Sigma+$** summierten x- und y-Werte mit der angenäherten linearen Funktion.

KOORDINATEN-UMWANDLUNG

$\leftrightarrow P$ Wandelt rechtwinklige Koordinaten (x, y) in Polarkoordinaten mit Betrag r und Winkel 0 um.

$\leftrightarrow R$ Wandelt Polarkoordinaten mit Betrag r und Winkel 0 in rechtwinklige x und y Koordinaten um.

TRIGONOMETRIE

DEG Schaltet den Rechner auf Altgrad bei trigonometrischen Funktionen.

RAD Schaltet den Rechner auf Bogenmaß bei trigonometrischen Funktionen.

GRD Schaltet den Rechner auf Neugrad bei trigonometrischen Funktionen.

SIN, **COS**, **TAN** Berechnet den Sinus, Cosinus und Tangens des angezeigten Wertes.

SIN⁻¹, **COS⁻¹**, **TAN⁻¹** Berechnet den Arcussinus und Argustangens des angezeigten Wertes.

$\leftrightarrow D$ Wandelt Bogenmaß in Grad um.

$\leftrightarrow R$ Wandelt Grad in Bogenmaß um.

$\rightarrow H.M.S$ Wandelt dezimal angezeigte Stunden oder Winkel in Stunden, Minuten, Sekunden bzw. Grad, Minuten, Sekunden um.

$\rightarrow H$ Wandelt Winkel (oder Zeit) in der Form Grad (oder Stunden), Minuten, Sekunden in Dezimalform um.

STEUERUNG DES I-REGISTERS

$\leftrightarrow I$ Tauscht den Inhalt des angezeigten X-Registers mit dem des I-Registers aus.

$x \leftrightarrow i$ Tauscht den Inhalt des angezeigten X-Registers mit dem Register aus, das durch den Inhalt des I-Registers adressiert wird.



I REGISTER Speicherregister für Inkrement/Dekrement-Operationen und indirekte Operationen in Verbindung mit STO, GSB und DSP. Auch als einfaches Speicherregister verwendbar.

INDIREKT Wird in Verbindung mit **STO**- und **RCL**-Anweisungen zu indirekten Datenspeicherung und Speicherregister-Arithmetik verwendet.

DSPI zeigt eine durch den Inhalt des I-Registers bestimmte Anzahl von Stellen nach dem Komma an.

DSE Dekrement und Sprung wenn gleich oder weniger. Subtrahiert die Schrittweite von der Laufvariablen. Überspringt eine Programmzeile, wenn der neue Wert der Laufvariable gleich oder kleiner als der Endwert ist.

ISG Inkrement und Sprung wenn größer. Addiert die Schrittweite zu der Laufvariablen. Überspringt eine Programmzeile, wenn der neue Wert der Laufvariable größer als der Endwert ist.

LOGARITHMEN UND EXPONENTIAL-FUNKTIONEN

y^x Allgemeine Exponentialfunktion: Erhebt die Zahl y (im Y-Register) zur Potenz x (im X-Register).

10^x Umkehrfunktion des dekadischen Logarithmus: Erhebt 10 zur Potenz der Zahl im X-Register.

e^x Natürliche Exponentialfunktion (Basis e = 2,718..) der Zahl x.

LOG Dekadischer Logarithmus. Berechnet den dekadischen Logarithmus (Basis 10) der Zahl im X-Register.

LN Natürlicher Logarithmus: Berechnet den natürlichen Logarithmus (Basis e = 2,718..) der Zahl im X-Register.

VERZEICHNIS DER PROGRAMMFUNKTIONEN

Mehrere der folgenden Tasten sind nur im PRGM-Modus wirksam. Andere verhalten sich unterschiedlich im PRGM-Modus und im RUN-Modus. Alle Funktionen werden ausführlich auf den angegebenen Seiten erklärt.

[MEM] Anzeige des augenblicklichen Status der Programmspeicher-/Datenregisterzuteilung (Seite 59).

[A] [B] Frei definierbare Programm-tasten, die sowohl als Programm-marken als auch für die Programmausführung (Seite 66) verwendet werden.

0, 1, 2, 3, 4, 5, 6, 7, 8, 9 Bezeichnung der Marken

Mit vorausgehenden **[LBL]** wird der Anfang einer Routine bezeichnet (Seite 66).

[LBL] Marke. Mit nachfolgendem A, B oder 0 bis 9 wird der Anfang eines Programms oder Unterprogramms bezeichnet (Seite 49).

ÜBER DAS TASTENFELD AUSGEFÜHRT

[GTO] Go to. Bewirkt mit **[A] [B]** oder 0 bis 9 den Sprung zur entsprechenden Marke. Es werden keine Befehle ausgeführt.

[GSB] Sprung zu einem Unterprogramm. Startet, gefolgt von **[A], [B]**, oder 0 bis 9 die Ausführung gespeicherter Programmzeilen ab der entsprechenden Marke. (Seite 125).

[GTO] □ nnn Bewirkt einen Sprung zu der durch nnn bezeichneten Zeilennummer (Seite 90).

[BST] Einzelschritt zurück. Setzt den Rechner um eine Zeile im belegten Programmspeicher zurück. (Seite 92).

[SST] Einzelschritt vor. Rückt den Rechner um eine Zeile im belegten Programmspeicher vor. (Seite 86).

IM RAHMEN EINES GESPEICHERTEN PROGRAMMS AUSGEFÜHRT

[GTO] Go to. Gefolgt von **[A], [B]** oder 0 – 9. Bewirkt die Unterbrechung der Programmausführung, die Suche nach dem ersten Auftreten der Marke und die Wiederaufnahme der Programmausführung ab dieser Stelle (Seite 101).

[GSB] Sprung zu einem Unterprogramm. Gefolgt von **[A], [B]** oder 0 – 9 bewirkt Sprung zur bezeichneten Marke und Ausführung des entsprechenden Programmabschnittes als Unterprogramm. (Seite 125).

[DEL] Löscht eine Programmzeile. Entfernt die angezeigte Programmzeile aus dem Programmspeicher: alle nachfolgenden Anweisungen rücken um eine Zeile vor. (Seite 95).

CLEAR [PRGM] Löscht alle Programmspeicher und führt die Programmausführung zur Zeile 000 zurück. (Seite 48).



[PSE] (Pause) unterbricht die Programmausführung für etwa eine Sekunde und zeigt den Inhalt des X-Registers an. Im Anschluß an diese Unterbrechung führt der Rechner die Programmausführung selbstständig fort. (Seite 64).



[RS] Start/Stop. Startet die Ausführung gespeicherter Programmbefehle ab der augenblicklichen Position. Während der Ausführung von Programmbefehlern gedrückt, hält **[RS]** das Programm an. (Seite 62).



[RTN] Rücksprung zur Zeile 000 des Programmspeichers von jeder Zeile im belegten Programmspeicher oder Sprung zur entsprechenden Zeile im Programmspeicher von der letzten Zeile eines Unterprogramms. (Seite 50).



[SF] Flag setzen. Setzt das durch die Flag-Bezeichnung (0, 1, 2, oder 3) angegebene Flag (Seite 118).



[CF] Flag löschen. Löscht das durch die Flag-Bezeichnung (0, 1, 2 oder 3) angegebene Flag (Seite 118).



[F?] prüft, gefolgt von einer der Ziffern 0, 1, 2 oder 3, den Status des entsprechenden Flags. Falls das Flag gesetzt (wahr) ist, fährt der Rechner mit der Ausführung aufeinanderfolgender Programmzeilen fort.



Ist das Flag dagegen gelöscht (falsch), wird die Programmzeile übersprungen und dann mit der Programmausführung fortgefahren. Die Flags F2 und F3 sind im Anschluß an diesen Test gelöscht.



X=Y	X\neqY	X>Y	X\leqY
X\neq0	X=0	X>0	X\leq0

Bedingte Sprungbefehle. Führen logische Vergleiche zwischen den Inhalten des X-Registers und denen des Y-Registers oder 0 aus. Ist die Bedingung erfüllt, fährt der Rechner in der natürlichen Reihenfolge mit der Ausführung gespeicherter Programmzeilen fort. Ist die Bedingung dagegen nicht erfüllt, wird die darauffolgende Programmzeile übersprungen und dann mit der weiteren Programmausführung fortgefahren. (Seite 106).



ABSCHNITT 1: EINLEITUNG

Herzlichen Glückwunsch!

Mit Ihrem HP-34C besitzen Sie einen überaus vielseitigen programmierbaren Rechner mit Permanent-Speicher, mit dem Sie, dank des leistungsfähigen Hewlett-Packard-Logiksystems, die komplexesten Rechnungen durchführen können. Dabei können Sie zwischen verschiedenen Benutzungsweisen wählen:

Wissenschaftlicher Taschenrechner: Mit jeder der Mehrzwecktasten auf dem HP-34C Tastenfeld stehen Ihnen bis zu vier unterschiedliche Funktionen mit einem Höchstmaß an Rechnerleistung zur Verfügung.

Verwendung fertiger Programme: Im „Allgemeinen Handbuch für technisch-wissenschaftliche Rechner“ sind zahlreiche fertige Programme aus Mathematik, Statistik, Vermessungswesen, kaufmännischen und anderen Bereichen enthalten, die Sie nur Schritt für Schritt einzutasten brauchen. Damit können Sie auch ohne besondere Vorkenntnisse viele Aufgabenstellungen sofort mit Ihrem Rechner lösen.

Verwendung selbsterstellter Programme: Die Programmierung Ihres HP-34C ist sehr einfach. Es sind dabei keinerlei Erfahrungen im Umgang mit programmierbaren Rechnern oder Kenntnisse über Programmiersprachen nötig. Dabei verfügt Ihr HP-34C über eine Vielzahl herausragender Eigenschaften, die auch erfahrene Computer-Fachleute zu schätzen wissen:

- Permanent-Speicher, der Programme und Daten im Rechner erhält, selbst wenn dieser ausgeschaltet wird.
- Automatische Speicherplatzzuweisung: Die Voreinstellung des Rechners von 70 Programmzeilen und 20 Datenregistern wird bei Bedarf – in Schritten von 7 Zeilen – automatisch in eine Neuzuordnung von bis zu 210 Zeilen und einem Datenregister geändert.
- Präfix- und nachfolgende Funktionstasten werden zu einem kombinierten Code zusammengefaßt. Damit wird der Programmspeicherplatz optimal genutzt.
- Eine Reihe leistungsfähiger Operationen für die Korrektur und Abänderung gespeicherter Programme.
- Bedingte und unbedingte Programmverzweigungen.
- Sechs Unterprogrammebenen, vier Flags und zwölf übersichtlich angeordnete, wiederverwendbare Programmmarken.
- Direkte und indirekte Datenspeicherung, Programmverzweigungen und Unterprogrammaufrufe.

- Die Tasten **SOLVE** und **INT**, die zur Berechnung von Nullstellen von Funktionen und Integralen verwendet werden.

Darüber hinaus kann der HP-34C durch seinen wiederaufladbaren Batteriesatz ortsunabhängig betrieben werden.

Wenn Sie noch keine Erfahrungen mit HP-Rechner gesammelt haben und mit dem UPN-Logik-System noch nicht vertraut sind, ist es empfehlenswert, zunächst das „Allgemeine Handbuch für technisch-wissenschaftliche Rechner“ durchzuarbeiten. Erst dann sollten Sie dieses Handbuch zu Rate ziehen. Auch wenn Sie schon einen anderen HP-Rechner besitzen, werden Sie in der erwähnten Broschüre einige neue Einzelheiten finden.

Die Beispiele auf den folgenden Seiten werden Ihnen zeigen, wie einfach Ihr Rechner zu bedienen ist, gleichgültig, ob Sie ein Rechenproblem nun manuell oder automatisch mit Hilfe eines Programmes lösen.

MANUELLE LÖSUNG EINES PROBLEMS

Bevor Sie weiterlesen, sollten Sie mit der manuellen Lösung von Rechenproblemen gut vertraut sein. Andernfalls sollten Sie den Einführungsschnitt des „Allgemeinen Handbuchs“ nochmals durcharbeiten.

Wollen wir anfangen? Schieben Sie den EIN-AUS-Schalter in Stellung ON und versichern Sie sich, daß sich der PRGM-RUN-Schalter in Stellung RUN befindet. Drücken Sie jetzt **FIX** 4, damit die Anzeige des HP-34C mit der auf den folgenden Seiten angegebenen Anzeige übereinstimmt.*

Um die Verwandtschaft zwischen der manuellen Lösung eines Problems vom Tastenfeld aus und der Verwendung eines entsprechenden Programms zu verdeutlichen, wollen wir die nachfolgende Aufgabe einmal nach beiden Verfahren lösen.

Es sei die Oberfläche einer Kugel zu berechnen. Die dafür verwendete Formel lautet: $A = \pi d^2$

wobei A = Oberfläche

π = Kreiskonstante Pi ($\pi = 3,141592654..$)

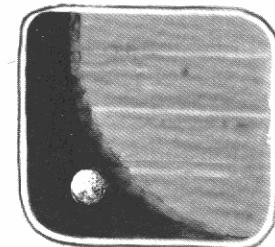
d = Durchmesser der Kugel

* Das in diesem Handbuch verwendete Anzeigeformat ist immer **FIX** 4, wenn es nicht ausdrücklich anders angegeben ist.



Beispiel: Ganymed, einer der 12 Jupitermonde, hat einen Durchmesser von 3200 Meilen. Zur Berechnung der Oberfläche dieses Himmelskörpers können Sie in der angegebenen Reihenfolge die nachstehenden Tasten drücken:

(Versichern Sie sich, daß der Symbol-Schalter in Stellung RUN steht.)



Tastenfolge	Anzeige	
3200	3.200,	Durchmesser von Ganymed
g X²	10.240.000,00	Quadrat des Durchmessers
h T	3,1416	Kreiskonstante π
x	32.169.908,78	Oberfläche von Ganymed in Quadratmeilen

PROGRAMMIERTE LÖSUNG EINES PROBLEMS

Nehmen wir einmal an, Sie wollten nicht nur die Oberfläche von Ganymed, sondern die Oberflächen aller 12 Jupitermonde berechnen. Dazu könnten Sie die obige Tastenfolge 12mal drücken, wobei Sie jedesmal einen anderen Durchmesser d einzusetzen hätten. Einfacher und schneller ist es jedoch, ein Programm zu erstellen, das die Oberfläche einer beliebigen Kugel aus ihrem Durchmesser berechnet, ohne jedesmal alle Tastenoperationen von Hand ausführen zu müssen.

Um die Oberfläche einer Kugel über ein Programm zu berechnen, müssen Sie zuerst das entsprechende Programm schreiben, es dann in den Rechner laden und schließlich ausführen, um das Ergebnis zu erhalten.

Erstellen des Programms. Ihr Programm existiert schon! Es ist kaum mehr als die Folge von Tasten, die Sie zur manuellen Lösung eines Problems drücken würden.

Laden des Programms. Um die Folge der Programmschritte in den Rechner einzugeben, gehen Sie folgendermaßen vor:

1. Schalten Sie den PRGM-RUN-Schalter in Stellung PRGM (programmieren).
2. Drücken Sie **CLEAR PRGM** zum Löschen des Programmspeichers.

3. Drücken Sie die folgenden Tasten in der angegebenen Reihenfolge. (Die dabei in der Anzeige auftretenden Zahlen sind im Augenblick noch nicht von Bedeutung. Sie stellen allerdings eine wichtige Information dar und werden später genau erklärt.)

Tastenfolge

LBL A

Der Programmstart wird definiert.

X²

Diese Tasten müßten Sie auch drücken, wenn Sie das Problem manuell vom Tastenfeld aus lösen würden.

RTN

Das Programmende wird definiert.

Die Tastenfolge ist jetzt im Rechner gespeichert.

Ausführung des Programms. Um das Programm zur Berechnung der Oberfläche einer beliebigen Kugel aus ihrem Durchmesser zu starten, gehen Sie vor wie folgt:

1. Schalten Sie den PRGM-RUN-Schalter in Stellung RUN.
2. Tasten Sie den Wert des Durchmessers ein.
3. Drücken Sie A zum Starten des Programms.

Wenn Sie A drücken, führt der Rechner die gespeicherten Programmschritte automatisch aus und kommt so zum gleichen Ergebnis, das Sie auch beim schrittweisen Rechnen von Hand erhalten hätten.

Als Beispiel wollen wir die Oberfläche von Ganymed, mit einem Durchmesser von 3200 Meilen, berechnen:

Tastenfolge Anzeige

3200

3.200,

A

32.169.908,78 Quadratmeilen

Mit Hilfe des einmal gespeicherten Programms können Sie jetzt die Oberfläche sämtlicher Jupitermonde und auch jeder anderen Kugel aus ihrem Durchmesser berechnen. Dazu lassen Sie den Rechner im RUN-Modus stehen und geben den entsprechenden Wert für den Durchmesser ein. Anschließend brauchen Sie nur noch A zu drücken!

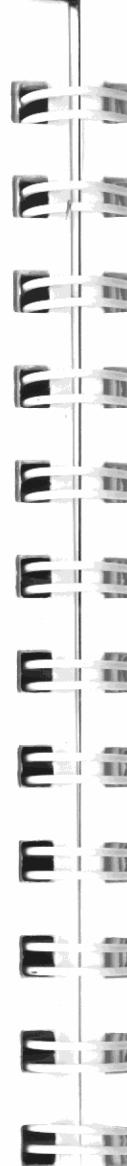
Nehmen wir an, Sie wollten die Oberfläche des Jupitermondes Io berechnen, der einen Durchmesser von 2310 Meilen hat:

Tastenfolge Anzeige

2310 A

16.763.852,56 Quadratmeilen

Nun berechnen Sie die Oberfläche für die Monde Europa, mit einem Durchmesser von 1950 Meilen, und Callisto, mit einem Durchmesser von 3220 Meilen.



Tastenfolge

1950 A

Anzeige

11.945.906,07

Oberfläche von Europa
in Quadratmeilen

3220 A

32.573.289,27

Oberfläche von Callisto
in Quadratmeilen

So einfach ist das Programmieren. Der Rechner merkt sich eine Folge von Tastenfunktionen und führt sie aus, sooft sie es wünschen. Tatsächlich kann Ihr HP-34C bis zu 210 einzelne Operationen speichern (und eine noch wesentlich höhere Anzahl von Tastenfunktionen, da viele Operationen zwei oder mehr Tastenfunktionen benötigen).

DIE VORTEILE DES PERMANENT-SPEICHERS

Ihr Rechner verfügt über einen Permanent-Speicher – eines der fortschrittlichsten Speichersysteme, mit denen ein wissenschaftlicher Rechner ausgerüstet werden kann. Alle Programme, der Inhalt der 21 Datenregister und das Anzeigeformat bleiben durch den Permanent-Speicher erhalten, wenn der Rechner ausgeschaltet wird. Sie können also das Programm oder die Programme, die Sie am häufigsten benutzen, tage- und wochenlang im Rechner speichern.

Der Permanent-Speicher ist besonders vorteilhaft, wenn Sie Daten speichern oder Batteriestrom sparen wollen, oder den Rechner auf Ihre Bedürfnisse zugeschnitten haben (d.h., wenn Sie mit 20 % Ihrer Programme 80 % Ihrer Probleme lösen). Sie sparen dadurch viel Zeit, weil Sie diese häufig verwendeten Programme nicht immer wieder eintasten müssen – sie bleiben im Rechner gespeichert. Außerdem wird die durch menschliches Versagen bedingte Fehlerquote reduziert – je weniger Tasten Sie zu drücken haben, desto weniger Fehler können Sie machen!

Der entscheidende Vorteil des Permanent-Speichers besteht vielleicht darin, daß Sie den Rechner auf Ihre persönlichen Bedürfnisse ausrichten können. Am einfachsten geschieht dies, indem Sie eine Liste der von Ihnen am häufigsten benötigten Programme aufstellen, und diese Programme dann der Wichtigkeit nach geordnet im Rechner speichern.

Wenn Sie auf ein sich wiederholendes Problem stoßen, schreiben Sie das Programm nur einmal und verwenden es jedesmal, wenn Sie es benötigen. Auf diese Weise können Sie auch ein oder zwei Lieblingsprogramme im Rechner speichern.

Der Permanent-Speicher erhält jedoch nicht nur Programme, sondern speichert auch Daten in bis zu 21 Speicherregistern abhängig von der augenblicklichen Programmspeicher-/Speicherregister-Zuteilung. Konstanten,

Summen und Zwischenergebnisse können jederzeit zurückgerufen werden. Und weil der Permanent-Speicher auch den Anzeige-Modus speichert, entspricht die Anzeige beim Einschalten jeweils dem zuletzt verwendeten **FIX**-, **SCI** - oder **ENG**-Anzeigeformat.

Der Permanent-Speicher erhöht in vielen Situationen auch die Lebensdauer der Batterie. Bei ausgeschaltetem Rechner kann der Permanent-Speicher Ihre Programme einen Monat und länger speichern. Wenn Sie Ihren Rechner benutzen, bleibt die Anzeige nicht so lange eingeschaltet, weil Sie weniger Programme eintasten müssen – und hier sparen Sie Batteriestrom.

ABSCHNITT 2: BESONDERE EIGENSCHAFTEN DES HP-34C

Viele der Bedienungselemente Ihres HP-34C sind im „Allgemeinen Handbuch für technisch-wissenschaftliche Rechner“ beschrieben. Darüber hinaus verfügt der HP-34C jedoch über einige Besonderheiten, die auch bei HP-Rechnern neu sind und auf den folgenden Seiten näher erläutert werden.

TASTENFELD

Den meisten Tasten auf dem Tastenfeld des HP-34C sind bis zu vier verschiedene Funktionen zugeordnet. Die Symbole dieser Funktionen stehen auf der Tastenoberseite, der abgeschrägten Tastenvorderseite und – in blauer bzw. goldfarbener Schrift – oberhalb der Taste auf dem Rechnergehäuse.

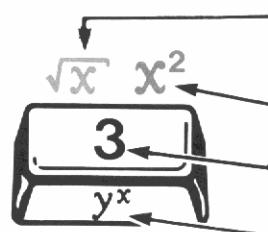
Für die Auswahl der gewünschten Funktion stehen Ihnen die drei Präfixtasten **f**, **g** und **h** zur Verfügung. Wenn Sie zuerst eine dieser Präfixtasten und dann die gewünschte Funktionstaste drücken, führt der Rechner die Operation aus, deren Symbol (in entsprechender Farbe) auf der abgeschrägten Tastenvorderseite oder oberhalb der Taste steht.

Zur Ausführung der Funktion, deren Symbol auf der Tastenoberseite steht, drücken Sie einfach diese Funktionstaste.

Zur Ausführung der Funktion, deren schwarzes Symbol auf der abgeschrägten Tastenvorderseite steht, drücken Sie zuerst die schwarze Präfixtaste **h** und anschließend die Funktionstaste.

Zur Ausführung der Funktion, deren goldfarbenes Symbol oberhalb der Taste steht, drücken Sie zuerst die goldfarbene Präfixtaste **f** und anschließend die Funktionstaste.

Zur Ausführung der Funktion, deren blaues Symbol oberhalb der Taste steht, drücken Sie zuerst die blaue Präfixtaste **g** und anschließend die Funktionstaste.



Drücken Sie zur Ausführung dieser Funktion zuerst **f** und dann **3**.

Drücken Sie zur Ausführung dieser Funktion zuerst **g** und dann **3**.

Drücken Sie zur Ausführung dieser Funktion **3**.

Drücken Sie zur Ausführung dieser Funktion zuerst **h** und dann **3**.

Beachten Sie, daß bei allen Tasten mit vier Funktionen bis auf **ENTER**, das goldfarbene Symbol oberhalb und links von der Taste und das blaue Symbol oberhalb und rechts von der Taste steht.

SPEICHERREGISTER UND PROGRAMMSPEICHER

Neben den vier Stack-Registern und dem LAST-X-Register verfügt Ihr HP-34C über einen automatisch gesteuerten Programm- und Datenregister-speicher.

In der Grundstellung besteht die Programmspeicher-/Datenregisterzuteilung aus 70 Programmzeilen und 20 Datenregistern und dem I-Register. Je nach Bedarf können ein oder mehrere der Datenregister in jeweils 7 Zeilen Programmspeicher umgewandelt werden. Und wenn Sie **g MEM** drücken, teilt der HP-34C Ihnen jederzeit die augenblickliche Speicherplatzzuteilung mit.

Mit diesem wichtigen Thema werden wir uns noch bei der Programmierung eingehend befassen.

DIE SPEICHERREGISTER DES HP-34C

SPEICHERN UND ZURÜCKRUFEN VON ZAHLEN

Die zwanzig Speicherregister Ihres HP-34C werden mit R₀ bis R₉ bezeichnet. (Das zusätzliche I-Register wird später behandelt.) Wie Sie im „Allgemeinen Handbuch für technisch-wissenschaftliche Rechner“ erfahren haben, wird die im X-Register angezeigte Zahl mittels **STO** (speichern), gefolgt durch eine Zifferntaste (0-9), in einem der Zahl entsprechenden Datenregister gespeichert. Der Inhalt eines Registers R₀ bis R₉ wird in das angezeigte X-Register mittels **RCL** (zurückrufen), gefolgt durch eine Zifferntaste (0-9), zurückgerufen.

Beachten Sie jedoch, daß Speichern und Zurückrufen von Daten in Verbindung mit den Registern R₀ bis R₉ eine zusätzliche Taste benötigt, die bei diesen Registerbezeichnungen dem Dezimalpunkt entspricht.

Wenn Sie beispielsweise die Zahl π in R₅ speichern wollen:

Tastenfolge	Anzeige
h T	3,1416
STO g 5	3,1416

Die Zahl π wird nach R₅ gespeichert

Um π aus R₅ zurückzurufen

Tastenfolge	Anzeige
CLX	0,0000
RCL g 5	3,1416

SPEICHERREGISTER-ARITHMETIK

Die Register R₀ bis R₉ werden, wie im „Allgemeinen Handbuch für technisch-wissenschaftliche Rechner“ beschrieben, für direkte Speicherregister-Arithmetik verwendet.

Alle Speicherregister (R₀ bis R₉, R₀ bis R₉ und I) können für indirekte Speicherregister-Arithmetik (die in Abschnitt 7, Fortgeschrittene Programmierung behandelt wird) benutzt werden.

ABÄNDERN VON ZAHLEN

Neben **CHS** gibt es noch drei weitere Tasten, die Zahlen verändern: **ABS**, **Frac** und **INT**. Diese Tasten werden häufig in Programmen zur Zahlenmanipulation verwendet.

ABSOLUTWERT

Manche Berechnungen benötigen den Absolutwert oder Betrag einer Zahl. Zur Berechnung des Absolutwertes einer Zahl im Anzeigenregister X, drücken Sie **h ABS**. Um beispielsweise den Absolutwert von -3 zu berechnen:

Tastenfolge	Anzeige
3 CHS h ABS	-3, 3,0000 = I-3I

GANZZAHLIGER ANTEIL EINER ZAHL

Um den ganzzahligen Anteil einer Zahl im X-Register zu bestimmen und anzuzeigen, drücken Sie **h INT**. Um beispielsweise bei der Zahl 111,222 den Dezimalteil abzuschneiden:

Tastenfolge	Anzeige
111,222	111,222
h INT	111,0000
	Es verbleibt nur der ganzzahlige Anteil
333,444 CHS	-333,444
h INT	-333,0000
	Wieder verbleibt nur der ganzzahlige Anteil.

Wenn Sie **h INT** drücken, geht der Dezimalteil der Zahl verloren. Das Vorzeichen bleibt unverändert. Sie können allerdings die vollständige Zahl wieder aus dem Last-X-Register zurückrufen.

Tastenfolge	Anzeige
h LSTX	-333,4440
	Die ursprüngliche Zahl.

DEZIMALER ANTEIL EINER ZAHL

Um den ganzzahligen Teil einer Zahl abzuschneiden und nur den Dezimalteil zu erhalten, drücken Sie **h FRAC**. Um beispielsweise nur den Dezimalteil von 555,666 zu erhalten:

Tastenfolge	Anzeige
555,666	555,666
h FRAC	0,6660
	Es verbleibt nur der Dezimalteil der Zahl,
777,888 CHS	-777,888
h FRAC	-0,8880
	Wieder verbleibt nur der Dezimalteil der Zahl

Wenn Sie **h FRAC** drücken, geht der ganzzahlige Anteil der Zahl verloren. Das Vorzeichen bleibt unverändert. Natürlich ist die ursprüngliche Zahl in LAST X verfügbar.

Tastenfolge	Anzeige
h LSTX	-777,8880

MATHEMATISCHE FUNKTIONEN

FAKULTÄT

Die Taste **x!** (Fakultät) erleichtert wesentlich die Berechnung von Kombinationen und Permutationen. Wenn Sie **h x!** drücken, wird die Fakultät der positiven ganzen Zahl im angezeigten X-Register berechnet.

Beispiel: Berechnen Sie die Anzahl der Möglichkeiten, wie sich acht Personen bezüglich ihrer Reihenfolge für eine fotografische Gruppenaufnahme aufstellen können.



Methode: $8! = 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1$

Tastenfolge	Anzeige
8 h x!	40.320,0000

Der Fotograf kann 40.320 unterschiedliche Aufnahmen der 8 Personen machen.

Beispiel: Die gewünschte Nahaufnahme lässt nur eine gleichzeitige Aufnahme von 5 Personen zu. Wieviel unterschiedliche Gruppenaufnahmen sind möglich, wenn jeweils 5 aus 8 Personen auf einem Bild zu sehen sind?

Methode: Die Anzahl der Aufnahmen wird durch

$$\frac{8!}{(8-5)! 5!} \quad \text{gegeben.}$$

Tastenfolge	Anzeige
8 h x!	40.320,0000
8 ENTER+	8,0000
5 –	3,0000
h x!	6,0000
5 h x!	120,0000
x	720,0000
÷	56,0000

Der Fotograf kann 56 unterschiedliche Aufnahmen von 5 Personen machen.

GAMMAFUNKTION

Die Γ -Taste* kann auch zur Berechnung der durch $\Gamma(x)$ bezeichneten Gammafunktion verwendet werden, die in der höheren Mathematik und Statistik auftritt. Wenn Sie Γ drücken, wird $\Gamma(x+1)$ berechnet. Um also die Gammafunktion einer Zahl zu berechnen, subtrahieren Sie erst 1 von der Zahl, und mit dem Ergebnis im X-Register drücken Sie dann Γ .

Beispiel: Berechnen Sie $\Gamma(2,7)$:

Tastenfolge	Anzeige	
2,7	2,7	Geben Sie die Zahl ein.
[ENTER+] 1 \square	1,7000	Subtrahieren Sie 1.
Γ \square	1,5447	$\Gamma(2,7)$

Beispiel: Berechnen Sie $\Gamma(-2,7)$

Tastenfolge	Anzeige	
2,7 [CHS]	-2,7	Geben Sie die Zahl ein.
[ENTER+] 1 \square	-3,7000	Subtrahieren Sie 1.
Γ \square	-0,9311	$\Gamma(-2,7)$

Da $\Gamma(x)$ nicht definiert ist, wenn die Funktionstaste \square mit einer negativen ganzen Zahl oder Null verwendet wird, ist $\Gamma(x+1)$ nicht definiert, wenn x eine negative Ganzzahl ist. Wenn x gegen einen dieser Werte geht, geht der Absolutwert von $\Gamma(x+1)$ gegen ∞ .

* Die Γ -Taste kann sowohl für die Berechnung der Fakultät wie der Gammafunktion verwendet werden, da $\Gamma(x+1) = \Gamma(n+1) = n!$ gilt, wenn x eine nicht-negative Ganzzahl n ist. Die Gammafunktion kann als eine Verallgemeinerung der Fakultät-Funktion betrachtet werden, da in diesem Fall die Zahl im X-Register nicht auf positive ganze Zahlen beschränkt ist. Umgekehrt kann die Fakultät-Funktion als ein spezieller Fall der Gammafunktion betrachtet werden.

Da $9.999999999 \times 10^{99}$ die größte Zahl ist, die im HP-34C darstellbar ist, ergibt sich im Rechner die Überlauf-Anzeige -9.999999 99, wenn Sie mit einer negativen Ganzzahl im X-Register Γ drücken. Obwohl abhängig von x der Wert $\Gamma(x+1)$ positiv oder negativ sein kann, wenn x eine negative Ganzzahl ist, zeigt der Rechner immer ein Minuszeichen in der Überlaufanzeige, wenn x eine negative Ganzzahl ist. Dies unterscheidet den Wert von der Überlauf-Anzeige 9.999999 99, die bei großen positiven Werten von x entsteht, wenn $\Gamma(x)$ gegen ∞ geht, aber immer positiv bleibt.

BERECHNUNG PROZENTUALER UNTERSCHIEDE

Die Taste $\Delta\%$ (prozentuale Änderung oder Unterschied) dient zur Berechnung prozentualer Differenzen und ist eine Funktion von zwei Variablen. Zur Berechnung des prozentuellen Unterschiedes zwischen zwei Zahlen verfahren Sie wie folgt:

1. Geben Sie die Basiszahl ein (gewöhnlich der zuerst auftretende Wert).
2. Drücken Sie [ENTER].
3. Geben Sie die zweite Zahl ein.
4. Drücken Sie $\Delta\%$

$$\text{Die verwendete Formel lautet: } \Delta\% = \frac{100(x - y)}{y}$$

Mit der oben angegebenen Eingabefolge bedeutet ein positives Ergebnis einen Anstieg und ein negatives Ergebnis einen Rückgang.

Beispiel: Ihre Münzsammlung wird in 1974 auf \$ 475 geschätzt. Eine Schätzung 1979 ergab einen Wert von \$ 735. Welchem Prozentsatz entspricht diese Wertsteigerung von 1974 bis 1979.



Tastenfolge**Anzeige**475 **ENTER**735 **%**475,0000
54,7368Schalten Sie das Anzeigeformat auf **FIX** 4.

Prozentuale Wertsteigerung.

STATISTIK-FUNKTIONEN**SUMMATIONEN**

Wenn Sie die Taste **$\Sigma+$** (Summationstaste) drücken, werden gleichzeitig mehrere Summen und Produkte der Inhalte des X- und Y-Registers berechnet. Um diese Summen für die verschiedenen statistischen Funktionen verfügbar zu halten, werden Sie automatisch in die Speicherregister R_0 bis R_5 geschrieben. Die einzige Situation, in der Daten in den Speicherregistern automatisch aufaddiert werden, ist im Zusammenhang mit dieser Taste.

Bevor Sie mit der Summation beginnen, die mit der Taste **$\Sigma+$** durchgeführt werden, sollten Sie mit Hilfe von **CLEAR** **Σ** die Speicherregister löschen.

Verfahren Sie dann folgendermaßen mit den Wertepaaren x und y:

1. Tasten Sie den y-Wert in das X-Register.
2. Drücken Sie **ENTER**, um den y-Wert in das Y-Register zu kopieren.
3. Tasten Sie den x-Wert in das X-Register ein.
4. Drücken Sie **$\Sigma+$** .

Der Vorgang ist ähnlich, wenn Ihr Statistikproblem statt zwei Variablen (x und y) nur eine Variable (x) einbezieht. Löschen Sie erst die Statistik-Register R_0 bis R_5 . Wenn der Inhalt des Y-Registers ungleich Null ist, sollte auch das Y-Register gelöscht werden. (Eine Zahl ungleich Null im Y-Register kann bei der Berechnung von s, r, L.R. oder \hat{y} für eine Variable zu der Fehleranzeige ERROR 3 führen.)

CLEAR REG löscht die Register R_0 bis R_5 , aber auch die Register R_6 bis R_9 , R_{10} bis R_{19} und das I-Register. Wenn Sie den Inhalt dieser Register retten wollen, sollten Sie statt **CLEAR REG** die Tasten **CLEAR Σ** drücken.

Nachdem Sie die Register gelöscht haben, verfahren Sie mit jedem x-Wert in folgender Weise:

1. Tasten Sie den x-Wert in das X-Register
2. Drücken Sie **$\Sigma+$**



Jedes Mal, wenn Sie **$\Sigma+$** drücken, geschieht im einzelnen folgendes:

1. Die Zahl im X-Register wird zum Inhalt von Register R_0 addiert.
2. Das Quadrat der Zahl im X-Register wird zum Inhalt von Register R_1 addiert.
3. Die Zahl im Y-Register des Stacks wird zum Inhalt des Registers R_2 addiert.
4. Das Quadrat der Zahl im Y-Register wird zum Inhalt von Register R_3 addiert.
5. Die Zahl im X-Register wird mit der Zahl im Y-Register multipliziert und das Produkt zum Inhalt von Register R_4 addiert.
6. Die Zahl 1 wird zum Inhalt von Register R_5 addiert. Wenn alle folgenden Schritte ausgeführt worden sind, wird die sich in Register R_5 ergebende Zahl im X-Register angezeigt.

Mit jedem Drücken der **$\Sigma+$** -Tasten wird die zuletzt im X-Register enthaltene Zahl in das LAST X-Register geschrieben. Der Wert im Y-Register bleibt unverändert.

In der untenstehenden Tabelle fassen wir noch einmal zusammen, wo die statistischen Summationen im Rechner gespeichert werden:

Register	Inhalt	
Anzeige (X)	n	Anzahl der eingegebenen Daten
R_0	n	Anzahl der eingegebenen Daten
R_1	Σx	Summe der x-Werte
R_2	Σx^2	Summe der Quadrate der x-Werte
R_3	Σy	Summe der y-Werte
R_4	Σy^2	Summe der Quadrate der y-Werte
R_5	Σxy	Summe der Produkte xy

Manchmal bestehen Datensätze aus x- und y-Werten, die sich nur geringfügig von einer gewissen Zahl unterscheiden. Sie können die Genauigkeit der statistischen Berechnungen dadurch erhöhen, indem Sie nur die Differenz zwischen den Variablen und einen dem Mittelwert der Zahlen entsprechenden Wert eingeben. Wenn Sie so vorgehen, muß das Ergebnis einer Berechnung von \bar{x} , \hat{y} oder des Achsenabschnittes nachträglich um diesen Wert erhöht werden.

Wenn beispielsweise die x-Werte aus 665999, 666000 und 666001 bestehen, sollten Sie als Eingabedaten -1, 0, 1 verwenden. Nach der Berechnung von x sollten Sie dann 666000 zum Ergebnis addieren. In einigen Fällen kann der Rechner s, r, L.R. oder \hat{y} nicht berechnen, wenn die Werte zu nah beieinander liegen. Der Rechner meldet sich in solchen Fällen mit der Fehleranzeige ERROR 3. Das geschieht jedoch nicht, wenn Sie die Daten vorher wie obenstehend normalisieren.

Anmerkung: Im Gegensatz zur Speicherregister-Arithmetik kann die $\Sigma+$ -Funktion zu einem Speicherüberlauf (d.h. Zahlen größer als 9,999999999 \times 10⁹⁹) in den Registern R₀ bis R₅ führen, ohne daß in der Anzeige die Fehlermeldung **Error 1** erfolgt, (d.h. wenn eine $\Sigma+$ oder $\Sigma-$ -Anweisung zu einem Speicherüberlauf führt, wird das Ergebnis in das entsprechende Statistik-Register geschrieben, ohne daß der Programmablauf unterbrochen wird).

Wenn Sie eine der Summen anzeigen wollen, brauchen Sie nur den entsprechenden Wert aus dem Speicherregister mittels **RCL** und der Registerangabe in die Anzeige zurückzurufen. Wenn Sie das unmittelbar nach $\text{f } \Sigma+$ oder $\text{g } \Sigma$ tun, wird der Wert n in X überschrieben, ohne daß der Stack angehoben wird.

Wenn Sie sowohl Σx wie Σy verwenden wollen, drücken Sie **RCL** $\text{f } \Sigma+$. Damit wird gleichzeitig Σx aus R₁ in das angezeigte X-Register und Σy aus R₃ in das Y-Register kopiert. Wenn Sie das unmittelbar nach $\text{f } \Sigma+$, $\text{g } \Sigma$, **CLX** oder **ENTER+** drücken, wird die Zahl im Y-Register erst in das Z-Register angehoben. Andernfalls werden die Zahlen im X- und Y-Register erst in das Z-Register bzw. in das T-Register angehoben.

Beispiel: Berechnen Sie Σx , Σx^2 , Σy , Σy^2 und Σxy für die nachfolgenden Wertepaare (x, y).

y	7	5	9
x	5	3	8

Tastenfolge	Anzeige	
f CLEAR Σ	0,0000	Löscht die Summationsregister (Vorhergehende Berechnungen haben Null im X-Register gelassen)
7 ENTER+	7,0000	1. Wertepaar, n = 1
5 $\text{f } \Sigma+$	1,0000	
5 ENTER+	5,0000	2. Wertepaar, n = 2
3 $\text{f } \Sigma+$	2,0000	
9 ENTER+	9,0000	3. Wertepaar, n = 3
8 $\text{f } \Sigma+$	3,0000	Anzahl der Eingaben aus R ₀ (n = 3)
RCL 1	16,0000	Summe der x-Werte in R ₁
RCL 2	98,0000	Summe der x ² -Werte in R ₂
RCL 3	21,0000	Summe der y-Werte in R ₃
RCL 4	155,0000	Summe der y ² -Werte in R ₄
RCL 5	122,0000	Summe der Produkte xy in R ₅
RCL 0	3,0000	Anzahl der Eingaben aus R ₀ (n = 3)

LÖSCHEN UND KORRIGIEREN VON EINGABEDATEN

Wenn Sie eine falsche Zahl eingetastet und $\text{f } \Sigma+$ noch nicht gedrückt haben, drücken Sie **CLX** und geben Sie den richtigen Wert ein.

Wenn einer der Werte geändert werden soll, oder Sie nach Drücken von $\Sigma+$ feststellen, daß fehlerhafte Daten eingegeben wurden, können Sie diesen Fehler unter Verwendung von $\Sigma-$ (Sigma minus) wieder rückgängig machen:

1. Geben Sie das fehlerhafte Datenpaar in X- und Y-Register ein (Sie können **LST X** verwenden, um einen einzelnen falschen Datenwert ins X-Register zurückzuholen).
2. Drücken Sie $\text{g } \Sigma-$, womit diese Daten wieder aus den Summen entfernt werden.
3. Geben Sie die korrekten Werte für x und y ein. (Auch wenn nur einer der Werte x oder y zu korrigieren war, sind beide Werte zu entfernen und erneut einzugeben.)
4. Drücken Sie $\text{f } \Sigma+$.

Nehmen wir beispielsweise an, daß das letzte Wertepaar im vorhergehenden Beispiel (8,6) statt (8,9) hätte heißen sollen.

Wir können die Summationen folgendermaßen korrigieren:

Tastenfolge	Anzeige	
9 ENTER+	9,0000	Zu ändernder y-Wert
8	8,	Zu ändernder x-Wert
$\text{g } \Sigma-$	2,0000	Die falschen Werte sind entfernt worden, und die Anzahl der Wertepaare ist n = 2
6 ENTER+	6,0000	Der richtige y-Wert
8	8,	Der richtige x-Wert
$\text{f } \Sigma+$	3,0000	Die richtigen Werte sind aufsummiert worden. Es sind wieder 3 Wertepaare berücksichtigt

Anmerkung: Obwohl $\text{g } \Sigma-$ zum Entfernen falsch eingegebener Wertepaare verwendet werden kann, werden eventuelle Rechenfehler, die bei der Summation in den Registern R₁ bis R₅ entstehen, nicht gelöscht. Es ist daher möglich, daß weitere Ergebnisse sich unterscheiden, wenn ein falsches Datenpaar (x, y) erst mit $\text{f } \Sigma+$ eingegeben und dann mit $\text{g } \Sigma-$ wieder gelöscht wurde.

Der Unterschied dürfte aber kaum eine Rolle spielen, wenn das falsche Datenpaar (x, y) nicht unverhältnismäßig größer war als das richtige Datenpaar. In solchen Fällen lohnt es sich, noch einmal von vorne zu beginnen und die Daten abermals (und diesmal vorsichtiger) einzugeben.

MITTELWERT

Die Funktionstaste **Σ** dient dazu, den Mittelwert (das arithmetische Mittel) der mittels **$\Sigma+$** eingegebenen Daten zu berechnen.

Wenn Sie **h** **f** drücken, geschieht folgendes:

- Der Inhalt der Stack-Register wird in gleicher Weise angehoben, wie es bei **RCL**   der Fall ist und auf Seite 30 beschrieben ist.
 - Der Mittelwert von x wird aus den Daten der Register R_0 und R_1 (sie enthalten n und Σx) berechnet. Der sich ergebende Wert wird nach der Gleichung

$$\bar{x} = \frac{\Sigma x}{n}$$

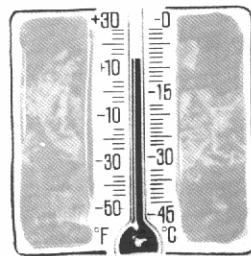
berechnet. Das Ergebnis steht danach im angezeigten X-Register.

3. Der Mittelwert von y wird aus den Daten der Register R_0 und R_3 (sie enthalten n und Σy) nach der Gleichung

$$\bar{y} = \frac{\Sigma y}{n}$$

berechnet. Der sich ergebende Wert wird ins Y-Register gebracht. Sie brauchen nur die Taste **X_y** zu drücken, um diesen Wert ins X-Register zu bringen.

Beispiel: In der folgenden Tabelle sind die Tageshöchst- und Tagesniedrigsttemperaturen einer Winterwoche in Fairbanks, Alaska, aufgeführt. Wie lauten die Mittelwerte der Höchst- und Niedrigstwerte für diese Woche?



	So	Mo	Di	Mi	Do	Fr	Sa
Höchsttemperatur	6	11	14	12	5	-2	-9
Niedrigsttemperatur	-22	-17	-15	-9	-24	-29	-35

Tastenfolge **Anzeige**
 **CLEAR**  **0.0000**

Löschen der Speicherregister
(es wird angenommen, daß keine Ergebnisse vorangegangener Rechnungen im Stack stehen)

Erstes Datenpaar ($n = 1$)

6 **ENTER** 22 22,
CHS **f** **$\Sigma+$** 1,0000

Zweites Datenpaar ($n = 2$)

11 ENTER 17

CHS 2,0000
14 **ENTER** 15 15,

CHS f Σ +
12 ENTER 9

CHS f Σ + 4,0000
5 ENTER \downarrow 24 24

CHS f Σ^+

29 CHS f Σ+ 6,0000

35 CHS

-21,5714

x:y

7. Wertepaar, $n = 7$
Mittelwert der Niedrigsttemperatur
(Mittelwert von x) wird im
X-Register angezeigt
Mittelwert der Höchsttemperatur
(Mittelwert von y) wird im
X-Register angezeigt

STANDARDABWEICHUNG

Mit den aufsummierten Daten in den Speicherregistern können Sie mit Hilfe der Taste **S** die Standardabweichung (als Maß für die Verteilung um den Mittelwert) berechnen. Dabei verwendet der HP-34C folgende Formeln:

$$s_x = \sqrt{\frac{n \Sigma x^2 - (\Sigma x)^2}{n(n-1)}} \quad s_y = \sqrt{\frac{n \Sigma y^2 - (\Sigma y)^2}{n(n-1)}}$$

Mit diesen Formeln wird die beste Schätzung der Standardabweichung der Grundgesamtheit aus der gewonnenen Stichprobe berechnet. Die durch diese Formeln angegebenen Standardabweichungen werden daher als Stichproben-Standardabweichungen bezeichnet.

Wenn Sie **H S** drücken, geschieht folgendes:

- Der Inhalt der Stack-Register wird in gleicher Weise angehoben, wie es bei **RCL f Σ+** der Fall ist und auf Seite 30 beschrieben ist.
- Aus den Inhalten n , Σx und Σx^2 der Statistikregister R_0 , R_1 und R_2 wird die Standardabweichung der Proben x berechnet und das Ergebnis ins X-Register gebracht und angezeigt.
- Aus den Inhalten n , Σy und Σy^2 der Statistikregister R_0 , R_3 und R_4 wird die Standardabweichung der Proben y berechnet und das Ergebnis ins Y-Register gebracht. Um diesen Wert ins X-Register und zur Anzeige zu bringen, brauchen Sie nur die Taste **X-Y** zu drücken.

Beispiel: Zeus Zahlenfresser, ein aufstrebender junger Mathematikprofessor der Mammoth Universität hat ein neues Verfahren zur Ermittlung der mathematischen Fähigkeiten der Studenten im ersten Jahr entwickelt. Um die Wirksamkeit des Tests zu überprüfen, lässt er 746 Studenten, die seinen Vorlesungen in der Differentialrechnung beiwohnen, den Test machen. Da ihn die Auswertung des Tests zu sehr in Anspruch nimmt, beschließt er 8 der 746 zufällig auszuwählen und die Standardabweichung aus dieser Stichprobe von 8 Tests zu berechnen.

Die Ergebnisse der acht gewählten Arbeiten waren 79, 94, 68, 86, 82, 78, 83 und 89. Welche Stichproben-Standardabweichung hat Professor Zahlenfresser berechnet?



	Tastenfolge	Anzeige	
	CLX ENTER	0,0000	Zu Beginn werden die X- und Y-Register gelöscht.
	f CLEAR Σ+	0,0000	Löscht die Statistik-Register.
	79 f Σ+	1,0000	Der erste Wert wird eingegeben. Beachten Sie, daß nur eine Variable vorhanden ist, und daher kein y-Wert in das Y-Register eingegeben werden muß.
	94 f Σ+	2,0000	Anzahl der bis jetzt eingegebenen Datenwerte.
	68 f Σ+	3,0000	Der letzte Datenwert.
	86 f Σ+	4,0000	Die Standardabweichung für
	82 f Σ+	5,0000	746 Studenten bei einem Stichprobenumfang von 8.
	78 f Σ+	6,0000	
	83 f Σ+	7,0000	
	89 f Σ+	8,0000	
	h S	7,8365	
			Wenn Ihre Daten nicht eine Stichprobe, sondern die Grundgesamtheit darstellen, ist die Standardabweichung Ihrer Daten die wahre Standardabweichung der Grundgesamtheit.
			Der Zusammenhang zwischen der Stichproben-Standardabweichung und der Standardabweichung einer Grundgesamtheit (σ) ist durch die folgende Gleichung gegeben:
	$\sigma = S \sqrt{\frac{n-1}{n}}$		
			Der Unterschied zwischen den Werten ist klein und kann in den meisten Fällen vernachlässigt werden. Wenn Sie jedoch die Standardabweichung für eine Grundgesamtheit genau berechnen wollen, so ist dies mit Ihrem HP-34C leicht durchzuführen.
			Addieren Sie einfach mittels der f Σ+ Taste den Mittelwert (x) der Daten zu den schon summierten Daten und drücken Sie dann H S . Das Ergebnis entspricht jetzt der wahren Standardabweichung der Grundgesamtheit der ursprünglichen Daten.
			Beispiel: Wir wollen annehmen, daß die Daten des vorherigen Beispiels den endgültigen Prüfungsergebnissen von Zahlenfressers Seminar über transzendente Funktionen entspricht. Da Zahlenfresser dieses Seminar zum erstenmal abgehalten hat, will er die Standardabweichung der Ergebnisse berechnen, um eine Aussage über die von ihm gestellte Prüfung machen zu können. Zahlenfresser bedient sich dazu seines HP-Rechners, in den er alle Daten eingibt und dann folgende Tasten drückt:

Tastenfolge	Anzeige	
H \bar{x} f $\Sigma +$	82,3750 9,0000	Mittelwert der Ergebnisse. Mittelwert wird zu den Daten summiert. Es werden neun Eingabedaten angezeigt.
H S	7,3304	Die Standardabweichung aller Ergebnisse der Schlußprüfung.

LINEARE REGRESSION

Lineare Regression ist eine Methode der Statistik zum Auffinden einer Geraden, die für eine gegebene Menge von Datenpunkten die bestmögliche Näherung darstellt und so eine Beziehung zwischen zwei statistischen Variablen herstellt. Besteht zwischen den einzelnen Datenpunkten ein fester räumlicher oder zeitlicher Abstand, so wird diese Gerade eine Trendlinie genannt. Wegen der ihr zugrunde liegenden Berechnungsart wird die lineare Regression oft auch als Methode der kleinsten quadratischen Abweichung bezeichnet.

Natürgemäß müssen wenigstens zwei Datenpunkte in den Rechner eingegeben werden, bevor eine passende Gerade bestimmt werden kann. Wenn Sie alle Punkte mit Hilfe der **$\Sigma +$** -Taste eingegeben haben, können Sie die Koeffizienten der linearen Gleichung

$$y = Ax + B$$

bestimmen, indem Sie **H LR** drücken. B ist der Schnittpunkt auf der y-Achse und erscheint in der Anzeige. A stellt den Anstieg der Geraden dar und wird im Y-Register gespeichert.

Wenn Sie **H LR** drücken geschieht folgendes:

- Der Inhalt der Stack-Register wird in gleicher Weise angehoben, wie es bei **RCL f $\Sigma +$** der Fall ist und auf Seite 30 beschrieben ist.
- Die Steigung (A) der Regressionsgeraden wird mit der Formel

$$A = \frac{n \Sigma xy - \Sigma x \Sigma y}{n \Sigma x^2 - (\Sigma x)^2}$$

berechnet. Der Wert der Steigung steht nach Ausführung der Rechnung im Y-Register.

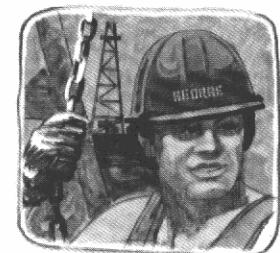
- Der Y-Achsenschnittpunkt (B) der Regressionsgeraden wird mit der Formel

$$B = \frac{\Sigma y \Sigma x^2 - \Sigma x \Sigma xy}{n \Sigma x^2 - (\Sigma x)^2}$$

berechnet. Der Wert des Y-Achsenschnittpunktes steht nach Ausführung der Rechnung im angezeigten X-Register.

Zur Betrachtung oder Weiterverwendung des Wertes A werden die Inhalte der X- und Y-Register einfach mit der Taste **X \leftrightarrow Y** getauscht.

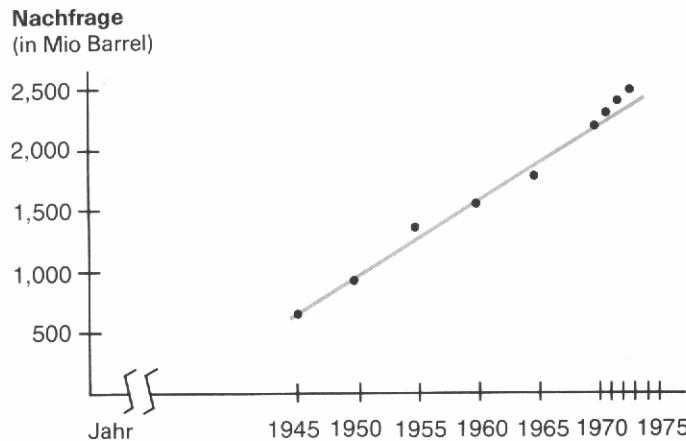
Beispiel: Herr Bettelmann von der Nimmer-satt-Olgesellschaft möchte die Steigung und den y-Achsenschnittpunkt der Regressionsgeraden für den Benzinverbrauch in den USA in Abhängigkeit von der Zeit seit 1945 bestimmen. Es sind ihm die in der nachfolgenden Tabelle aufgeführten Werte bekannt:



Benzinverbrauch

(in Mio Barrel)	696	994	1330	1512	1750	2162	2243	2382	2484
Jahr	1945	1950	1955	1960	1965	1970	1971	1972	1973

Lösung: Bettelmann könnte den Benzinverbrauch über die Zeit auftragen und so das untenstehende Diagramm erstellen:



Mit dem HP-34C braucht er jedoch nur die gegebenen Daten mit der **S+**-Taste einzugeben und dann **h** **L.R.** zu drücken.

Tastenfolge	Anzeige	
f CLEAR	0,0000	Löscht Statistik-Register (Annahme, daß Stack gelöscht ist)
696 ENTER	696,0000	
1945 f Σ+	1,0000	
994 ENTER	994,0000	
1950 f Σ+	2,0000	
1330 ENTER	1.330,0000	
1955 f Σ+	3,0000	
1512 ENTER	1.512,0000	
1960 f Σ+	4,0000	
1750 ENTER	1.750,0000	
1965 f Σ+	5,0000	
2162 ENTER	2.162,0000	



1970		6,0000
2243	 	2.243,0000
1971		7,0000
2382	 	2.382,0000
1972		8,0000
2484	 	2.484,0000
1973		9,0000
		-118.290,6
		61,1612

Die Eingabe aller Wertepaare ist
erfolgt.

-118.290,6295 Der y-Achsenabschnitt der Geraden.
61,1612 Die Steigung der Geraden.



LINEARER SCHÄTZWERT

Aus den in den Registern R_0 bis R_5 aufsummierten Daten kann zu einem neuen x-Wert ein Schätzwert \hat{y} berechnet werden, indem dieser x-Wert eingegeben und **[h] [y]** gedrückt wird.



Wenn zum Beispiel Herr Bettelmann im vorangegangenen Beispiel den Benzinverbrauch für die Jahre 1980 und 2000 voraussagen wollte, bräuchte er nur die neuen x-Werte einzutasten und  zu drücken.



	Tastenfolge	Anzeige	
	1980 [h] [y]	2.808,6264	Schätzwert in Mio Barrel für das Jahr 1980
	2000 [h] [y]	4.031,8512	Schätzwert in Mio Barrel für das Jahr 2000

KORRELATIONSKoeffizient

Um zu erkennen, wie gut die Daten mit der linearen Regression übereinstimmen, können Sie den Korrelationskoeffizienten r berechnen, indem Sie die Funktion Σ benutzen. Der Korrelationskoeffizient kann einen Wert zwischen -1 und $+1$ annehmen. Ein Wert von $r = 0$ entspricht überhaupt keiner Übereinstimmung, während $r = +1$ und $r = -1$ vollkommene Übereinstimmung mit der Regressionsgeraden bedeuten.

Beispiel: Berechnen Sie den Korrelationskoeffizienten für die vorangehend berechnete lineare Regression.

Tastenfolge	Anzeige	
h r	0,9931	Eine sehr gute Übereinstimmung

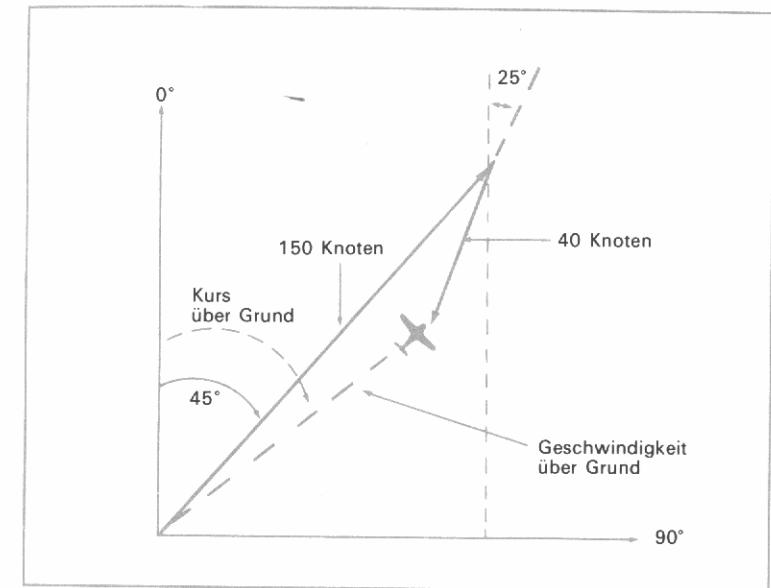
VEKTOR-ADDITION UND -SUBTRAKTION

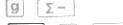
Die Taste $\Sigma+$ kann zum Summieren beliebiger Werte im X- und Y-Register verwendet werden. Besonders nützlich ist diese Funktion, wenn Vektoren addiert oder mit $\Sigma-$ subtrahiert werden sollen. Dazu sind die in Polarkoordinaten gegebenen Koordinaten zuvor mit Hilfe der Taste $F \rightarrow R$ in rechtwinkligen Koordinaten umzuformen. Das Ergebnis kann mit der Taste $\Sigma \rightarrow P$ wieder in die polare Form zurückverwandelt werden.

Wenn Sie nur die Summen Σx und Σy verwenden wollen, die Sie in den Speicherregistern aufaddiert haben, können Sie die Taste **RCL** und anschließend $\Sigma+$ drücken. Damit bringen Sie Σx ins angezeigte X-Register und Σy ins Y-Register, wobei die Inhalte dieser beiden Stack-Register überschrieben werden. Der Stack wird nicht angehoben. Diese Eigenschaft ist vor allem bei Vektorberechnungen von Vorteil.

Beispiel: Ein Flugzeug fliegt mit einer Eigengeschwindigkeit (gegenüber der es umgebenden Luft) von 150 Knoten (= nautische Meilen pro Stunde). Es steuert einen Kurs von 45° . Bedingt durch einen Gegenwind aus 25° mit 40 Knoten wird es auf seinem Flugweg versetzt. Wie groß ist die Geschwindigkeit über Grund und der Kurs über Grund, den es tatsächlich zurücklegt?

Lösungsweg: Der gesuchte Vektor (Geschwindigkeit über Grund, Kurs über Grund) ist gleich der Differenz zwischen den Vektoren (Eigengeschwindigkeit, Steuerkurs) = $(150, 45^\circ)$ und (Windgeschwindigkeit, Windrichtung) = $(40, 25^\circ)$.



Tastenfolge	Anzeige	
 [CLEAR] [Σ]	0,0000	Löscht die Statistik-Register
 [DEG]	0,0000	Wählt Winkel-Modus <Grad>
45 [ENTER]	45,0000	Winkel für 1. Vektor wird ins Y-Register eingegeben
150  [F] [R]	150, 106,0660	Betrag für 1. Vektor wird eingegeben Umformung in rechtwinkelige Koordinaten
25 [ENTER]	1,0000	1. Vektor wird in den Speicherregistern R ₁ und R ₃ addiert (zu Null)
40  [F] [R]	25,0000	Winkel für 2. Vektor wird ins Y-Register eingegeben
40  [Σ+]	40, 36,2523	Betrag für 2. Vektor wird eingegeben Umformung in rechtwinkelige Koordinaten
 [RCL]  [F]  [Σ+]	0,0000 69,8137 113,2417	2. Vektor wird subtrahiert Rückruf der Register R ₁ und R ₃
 [g]  [F]	113,2417	Tatsächliche Geschwindigkeit über Grund
 [x:y]	51,9389	Tatsächlicher Kurs über Grund

ABSCHNITT 3: EINFACHE PROGRAMMIERUNG

WAS IST EIN PROGRAMM?

Ein Programm ist nichts weiter als eine Folge von Tastenbefehlen, die Sie auch im Falle der manuellen Lösung eines Problems vom Tastenfeld aus drücken müßten. Der HP-34C speichert diese Tastenfolge und führt sie anschließend auf den Druck einer einzigen Taste hin automatisch aus. Für jede Wiederholung des Programmablaufs ist lediglich diese eine Taste zu drücken. Zur Programmierung des HP-34C ist keinerlei Programmiererfahrung erforderlich.

WARUM EIN PROGRAMM SCHREIBEN?

Programme sind besonders dann sinnvoll, wenn die Lösung einer Aufgabe die wiederholte Durchführung bestimmter Rechenschritte erfordert. Nachdem Sie einmal die zur Lösung eines Problems erforderliche Tastenfolge bestimmt und in den Programmspeicher eingetastet haben, brauchen Sie sich um die einzelnen Programmschritte keine Gedanken mehr zu machen. Jetzt nimmt Ihnen der Rechner die <Kleinarbeit> ab, und Sie brauchen sich nicht mehr zu sorgen, ob Sie nun eine falsche Taste gedrückt haben oder nicht, und können sich der kreativen Arbeit widmen.

Bevor wir fortfahren, wollen wir uns noch einmal die hervorragenden Programmereigenschaften des HP-34C vor Augen halten:

- Eine leicht verständliche Programmiersprache.
- Zwölf im Programm verwendbare (und mehrfach verwendbare) Marken zur Kennzeichnung von Programmen und Programmabschnitten.
- Zusammengefaßte Tasten-Codes. Eine aus einer Tastenfolge bestehende Anweisung wie  oder  + 1 benötigt nur eine Programmzeile im Programmspeicher.
- Automatische Speicherplatzzuteilung. Die Speicherplatzzuteilung variiert zwischen der Voreinstellung von 21 Speicherregistern/70 Programmzeilen und 1 Speicherregister/210 Programmzeilen. Die Speicherumwandlung geschieht automatisch, wobei jeweils ein Speicherregister sieben Programmzeilen entspricht.
- Entscheidungsmöglichkeiten für kompliziertere Programme.
- Einfach verwendbare Korrektur-Funktionen zum Abändern und Ergänzen von Programmen.

- Sechs Unterprogrammebenen und vier Flags zur Vereinfachung von sonst komplizierten Programmen.
- Indirektes Speichern und Zurückrufen von Daten. Indirekte Programmverzweigungen und Unterprogrammaufrufe zur automatischen Steuerung von Daten, Entscheidungen und Programmabläufen.
- Schleifensteuerung und Inkrement/Dekrement-Zähler. In der Summe bieten Ihnen diese Eigenschaften die Werkzeuge, um mit Zuversicht an die Lösung schwierigster Probleme heranzugehen.

DREI RECHNER-MODI

Ihr HP-34C verfügt über drei verschiedene Operations-Modi:

1. Manueller RUN-Modus
2. Programmier-Modus PRGM
3. Automatischer RUN-Modus

MANUELLER RUN-MODUS

Die Funktionen und Operationen, die Sie im ersten Teil dieses Handbuchs und in der Broschüre *«Rechnen mit dem HP-Rechner»* kennengelernt haben, werden einzeln manuell ausgeführt. Im Zusammenhang mit dem automatischen Rechenregister-Stapel (Stack) können Sie mit diesen Funktionen leicht alle Berechnungen durchführen.

PROGRAMMIER-MODUS PRGM

Im Programmier-Modus werden die Ihnen bekannten Funktionen und Operationen nicht ausgeführt, sondern in den Programmspeicher genannten Teil des Rechners gespeichert und erst später ausgeführt. Um den Rechner in den Programmier-Modus zu schalten, schieben Sie einfach den PRGM-RUN-Schalter in Stellung PRGM. Bis auf die folgenden Operationen können alle Funktionen der Tastatur für spätere Ausführungen gespeichert werden, wenn sich der Rechner im PRGM-Modus befindet.

Die folgenden Operationen können nicht gespeichert werden:

f CLEAR	PRGM	f CLEAR	PREFIX	g	MEM
h	SST	GTO	<input checked="" type="checkbox"/> nnn	h	MANT
h	BST	h	DEL	STO	ENTER ↑

Die obenstehenden Operationen bis auf **h MANT** und **STO ENTER**↑ sind bei der Eingabe und Korrektur eines Programms nützlich*.

- * **h MANT** bleibt im PRGM-Modus ohne Wirkung. Mit **STO ENTER**↑ wird im PRGM- oder RUN-Modus eine Selbstprüfroutine ausgelöst, die die Stack-Register, das LAST X-Register und die Flags löscht, auf den Winkel-Modus Grad schaltet und den Rechner an den Programmablaufstart (Zeile 000) zurücksetzt.

AUTOMATISCHER RUN-MODUS

Sie haben schon gesehen, daß der HP-34C automatisch eine Folge von Operationen, die im Rechner gespeichert sind, ausführt, wenn er sich im RUN-Modus befindet.

Jetzt wird nicht mehr jede Taste einzeln manuell gedrückt, sondern der Rechner führt die gespeicherten Operationen sequentiell im automatischen RUN-Modus aus. Kennzeichnend für diesen Ablauf ist, daß gewöhnlich nur eine Taste gedrückt wird, damit das Programm am Anfang der Operationsliste gestartet wird.

Die gesamte gespeicherte Operationsfolge wird dann um ein Vielfaches schneller ausgeführt, als es manuell möglich wäre.

PROGRAMMSPEICHER

Wie Sie bereits zu Beginn dieses Handbuchs erfahren haben, besteht ein Programm im wesentlichen aus der Tastenfolge, über die Sie das Rechenproblem auch «von Hand» über das Tastenfeld gelöst hätten. Der Rechner speichert die einzelnen Tastenbefehle im sogenannten Programmspeicher. Wenn Sie den **PRGM**  -Schalter in Stellung PRGM schieben, können Sie sich den Inhalt des Programmspeichers Zeile für Zeile ansehen.

Drücken Sie als erstes **GTO** 000, um den HP-34C an den Beginn des Programmspeichers zurückzuführen. Schalten Sie anschließend in den PRGM-Modus um. In der Anzeige sehen Sie 000-.

Der Programmspeicher besteht aus 70 bis 210 Programmzeilen mit einer zusätzlichen Marke als Programmspeicheranfang, welche durch das in der Anzeige stehende 000- gekennzeichnet wird. Der Programmspeicher ist als getrennt von Stack, LAST X, I und den verfügbaren Speicherregistern zu betrachten.*



* «Verfügbare» Speicherregister beziehen sich auf Datenregister, die nicht in Programmspeicher umgewandelt sind.

Programmspeicher

000-	← Programmspeicher-anfang
001-	
002-	
003-	
068-	
069-	← Minimale Programmspeicher-zuteilung
070-	
208-	
209-	← Maximale Programmspeicher-zuteilung
210-	

Mit dem **PRGM** -Schalter in Stellung PRGM erkennen Sie im linken Teil der Anzeige die Zeilennummer, welche die augenblickliche Position des Rechners im Programmspeicher kennzeichnet. Drücken Sie jetzt **CLEAR**, **PRGM**, **H**, **LBL**, **A**, die erste Tastenoperation des Programms zur Mondflächenberechnung (siehe Seite 17). Die Anzeige wechselt folgendermaßen:

Zeilennummer **001 – 25, 13, 11**

Wie Sie an der im linken Teil der Anzeige sichtbaren Schrittnummer erkennen, steht der Rechner jetzt an der ersten Speicherzeile. Die übrigen Zahlen, die in der Anzeige erscheinen, sind zweistellige Tasten-Codes als Bezeichnung der Anweisungen, die in dieser Programmspeicherzeile abgespeichert sind.

Drücken Sie **9**, **X²**. Die Anzeige ändert sich wie folgt:

002 – 15 3

Die links stehende Zahl 002 besagt, daß Sie den Inhalt der Zeile zwei in der Anzeige sehen.

Dabei wird jede vollständige Anweisung in eine einzelne Speicherzeile geschrieben, unabhängig davon, wieviele Einzeltasten sie umfaßt.



So kann eine Speicherzeile beispielsweise eine einzelne Taste (wie **C HS**) oder eine 3-Tasten-Operation (wie **STO** + 6) enthalten. (Addiert den Inhalt des angezeigten X-Registers zum Inhalt des Registers R₆.)



Welcher Zusammenhang besteht zwischen der Anzeige und der eigentlichen zum Programmbeispiel zusammengesetzten Tastenfolge? Diese Frage bringt uns zu Tasten-Codes und damit einen Schritt weiter in der Beherrschung des HP-34C.

**TASTEN-CODES**

Wir wollen noch einmal das eben eingegebene Programm betrachten. Drücken Sie **H BST**. In der Anzeige steht jetzt der Inhalt der ersten Programmzeile des Flächenberechnungsprogrammes der Jupitermonde:



Zeilennummer 001 – 25, 13, 11 Tasten-Code



Wie Sie wissen, kennzeichnet der Code 001 im linken Teil der Anzeige die Zeilennummer des Programmspeichers. Das nächste Ziffernpaar 25 stellt die Taste **H**, 13 die Taste **LBL** und 11 die Taste **A** dar. Jeder einzelne Tasten-Code setzt sich aus zwei Ziffern zusammen, von denen die erste die Zeilennummer und die zweite die Spaltennummer der Position dieser Taste auf dem Tastenfeld des Rechners angibt. So bezeichnet zum Beispiel der Tasten-Code 25 die zweite Zeile und fünfte Spalte (d. h. fünfte Taste in der zweiten Zeile). Wenn Sie von der Oberkante des Tastenfeldes ausgehend zwei Zeilen nach unten zählen und dann die fünfte Taste in dieser Zeile aufsuchen, sehen Sie, daß es sich um die Taste **H** handelt. Auf diese Weise wird jede Taste auf dem Tastenfeld (bis auf die Tasten 0 bis 9) dargestellt.



Der Einfachheit halber sind die Zifferntasten mit ihren alternativen Funktionen als 0 bis 9 codiert. Wir wollen es anhand eines Beispieles probieren. Drücken Sie **H SST** einmal. Die Anzeige enthält jetzt die zweite Zeile des Flächenberechnungsprogrammes, **9 X²**:



Zeilennummer 002 – 15 3 Tasten-Code



Wir wissen jetzt schon, daß 002 die Zeilennummer angibt, und daß 15 die Taste in der ersten Zeile und der fünften Spalte oder die **9**-Taste darstellt.



Da die Präfix-Taste **9** Teil dieser Anweisung ist, bezeichnet der Tasten-Code 3 jetzt die Funktion **X²**. In der Rechnersprache ist **9 X²** eine "umgeschaltete Funktion" oder "Alternativfunktion" der **3**-Taste, gerade wie das "**B**" eine umgeschaltete Funktion auf einer Schreibmaschine ist.



Die Tastenfolge für die Berechnung der Kugeloberfläche und die entsprechenden Anzeigen sind nachstehend noch einmal angegeben. Geben Sie die einzelnen Schritte ein und prüfen Sie jedesmal den angezeigten Code.

Tastenfolge	Anzeige
h T1	003– 25 73
x	004– 61
h RTN	005– 25 12

In diesem Fall hat ein Programm, das insgesamt 10 Tasten umfaßt, nur 5 Zeilen im Programmspeicher belegt.

Übungsaufgaben

1. Wie lauten die Tastencode für die folgenden Operationen:

h **Vx**, **g** **GRD**, **f** **-H.MS**, **STO** **+ 1**?

(Antwort: 25 2; 15 13; 14 6; 23, 51, 1.)

2. Wie viele Programmzeilen würden benötigt, um die folgenden Programmabschnitte zu speichern?

a) 2 **ENTER** 3 **+**

b) 10 **STO** 6 **RCL** 6 **x**

c) 100 **STO** 1 50 **STO** **x** 1 **RCL** 2 **h** **T1** **x**

(Antwort: a) 4, b) 5, c) 10.)

LÖSCHEN EINES PROGRAMMS

Durch den Permanent-Speicher Ihres Rechners bleiben alle Programme im Rechner erhalten, selbst wenn Sie das Gerät ausschalten.

Wenn Sie den Programmspeicher löschen wollen, schalten Sie den Rechner ein, schieben Sie den **PRGM** **RUN** -Schalter in Stellung PRGM und drücken Sie **f** **CLEAR** **PRGM**. Alle Programmzeilen, die zuvor Programmbefehle enthielten, stehen nach **f** **CLEAR** **PRGM** wieder zum Speichern von neuen Programmbefehlern zur Verfügung. Wenn das gelöschte Programm eine Länge von mehr als 70 Zeilen hatte, werden die überschüssigen Zeilen automatisch wieder den Datenregistern zugeteilt. Beachten Sie, daß wenn Sie **f** **CLEAR** **PRGM** im RUN-Modus drücken, der Rechner nach Zeile 000 zurückgesetzt wird, aber der Programmspeicher nicht gelöscht wird.

Wenn die Stromversorgung des Rechners unterbrochen wird (d.h. die Batterie versagt) kann es eintreten, daß die Informationen im Programmspeicher und in den Datenregistern verloren gehen. Wenn die Ursache für den Stromausfall behoben ist und der Rechner eingeschaltet wird, erscheint Pr Error in der Anzeige, um auf einen eventuellen Verlust hinzuweisen.

ERSTELLEN EINES EIGENEN PROGRAMMS

Sie haben bereits in der Einleitung zu diesem Handbuch ein einfaches Programm zur Berechnung der Kugeloberfläche bei gegebenem Durchmesser erstellt.

Wir wollen jetzt ein weiteres Programm schreiben, im Rechner speichern und ausführen, um Ihnen die Verwendung anderer Besonderheiten Ihres HP-34C zu zeigen.

Angenommen, Sie wollen über das Tastenfeld Ihres HP-34C die Fläche eines Kreises nach der Formel $A = \pi r^2$ berechnen. Sie würden dabei als erstes den Radius r eintasten und diesen Wert anschließend mit **g** **X²** quadrieren. Als nächstes würden Sie mit **h** **T1** den Wert der Kreiskonstanten Pi in die Anzeige rufen. Abschließend würden Sie **x** drücken, um damit den quadrierten Radius mit der Zahl Pi zu multiplizieren.

Erinnern Sie sich daran, daß ein Programm nichts weiter als die Tastenfolge ist, mit der Sie das gleiche Rechenproblem auch vom Tastenfeld aus lösen würden. Daher sind zur Erstellung eines Programms für die Berechnung der Fläche eines beliebigen Kreises die gleichen Tasten zu verwenden, die Sie sonst zur Berechnung des Ergebnisses (von Hand) gedrückt hätten.

Die Fläche eines Kreises können Sie nach der Formel $A = \pi r^2$ mit der nachstehenden Tastenfolge berechnen:

g **X²**
h **T1**
x

Diese Tastenfunktionen werden Sie auch als Bestandteil des Programms in den Speicher eintasten.

Dazu kommen normalerweise zwei weitere Operationen: **h** **LBL** **A** und **h** **RTN**. **h** **LBL** **A** wird Adressmarke genannt und kennzeichnet den Programmangfang. **h** **RTN** wird als Programmende verwendet.

BEGINN EINES PROGRAMMS

Den Programmbeginn kennzeichnen Sie mit **LBL** (label = Marke) und einer der Buchstabentasten (**A** oder **B**) oder einer Zifferntaste (0 bis 9).

Die Verwendung dieser Marken ermöglicht es, mehrere Programme zur gleichen Zeit im Rechner zu speichern und unabhängig voneinander zu verwenden.

BEENDEN EINES PROGRAMMS

Das Ende Ihres Programms bezeichnen Sie mit einer **[h] RTN** (Rücksprung-) Anweisung. Wenn der Rechner bei der Ausführung eines Programms an einen RTN-Befehl kommt, verzweigt die Programmausführung nach Zeile 000 und hält an (wenn **RTN** nicht als Teil eines Unterprogramms ausgeführt wurde). Mehr über Unterprogramme später).

Anmerkung: Wenn das laufende Programm das Ende des benutzten Programmspeichers erreicht, verhält sich der Rechner als ob er auf eine **[h] RTN**-Anweisung getroffen wäre. Wenn es sich bei der Programmierung herausstellen sollte, daß als letzte Anweisung ein **[h] RTN** stehen müßte, wird dieser Programmbebefhl überflüssig. Eine Programmzeile kann somit gespart werden.

Wenn Sie ein Programm an einer bestimmten Zeile im Programmspeicher anhalten wollen, ohne nach Zeile 000 zurückzukehren, laden Sie die Anweisung **[RS]** in diese Zeile. Wenn der Rechner im Verlauf der Programmausführung zu dem **[RS]**-Befehl kommt, hält das Programm einfach an. Schalten Sie jetzt vom RUN-Modus in den PRGM-Modus. In der Anzeige können Sie den Inhalt der auf die **[RS]**-Anweisung folgenden Zeile betrachten.

(Beachten Sie, daß der Rechner nach Zeile 000 zurückkehrt und anhält, nachdem er die letzte Anweisung im Programmspeicher ausgeführt hat, sofern diese keine **[RS]**-, **GSB** - oder **GTO** -Anweisung oder ein **[RS]**-Befehl als Teil eines Unterprogramms ist. Es ist also im Normalfall nicht erforderlich, eine **[RS]**-Anweisung als letzten Befehl in den Programmspeicher zu laden.)

Das vollständige Programm zur Berechnung der Kreisfläche bei gegebenem Radius sieht jetzt wie folgt aus:

[h] LBL A

Markiert den Anfang des Programms und gibt ihm einen <Namen>.

[g] X²

Quadriert den Radius.

[h] π

Ruft den Zahlenwert π in die Anzeige.

[x]

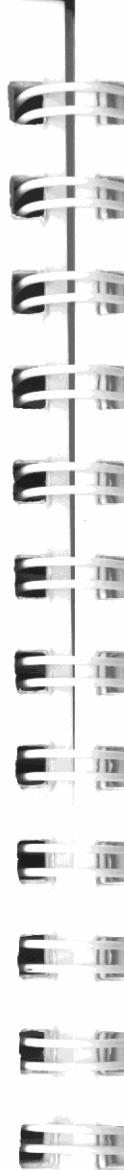
Multipliziert r^2 mit π ; das Ergebnis wird angezeigt.

[h] RTN

Markiert das Programmende und hält das laufende Programm an.

LADEN EINES PROGRAMMS

Befindet sich der Rechner im PRGM-Modus, werden die Operationen und Funktionen nicht ausgeführt, sondern als Befehlsfolge zur späteren Ausführung in den Programmspeicher geladen. Bis auf die neun Operationen, die auf Seite 44 aufgeführt sind, können alle Tastenfunktionen zur späteren Ausführung in den Programmspeicher geladen werden.



Um den Rechner zum Laden des Programms vorzubereiten:

1. Schieben Sie den PRGM-RUN-Schalter in Stellung PRGM.
2. Drücken Sie **[f] CLEAR [PRGM]**, um alle im Speicher befindlichen Programme zu löschen.

An den Ziffern 000 links in der Anzeige erkennen Sie, daß sich der Rechner am Programmspeicheranfang befindet. Die links in der Anzeige stehenden Ziffern geben immer die augenblickliche Zeilennummer im Programmspeicher an.

Um das Programm zur Berechnung der Kreisfläche zu laden, drücken Sie folgende Tasten:

[h] LBL A
[g] X²
[h] π
[x]
[h] RTN

Drücken Sie die erste Taste des Programms, **[h]**.

Tastenfolge **Anzeige**
[h] **000-**

Beachten Sie, daß sich die Anzeige nicht verändert hat. Drücken Sie jetzt die zweite und dritte Taste des Programms.

Tastenfolge **Anzeige**
LBL **000-**
A **001-25,13,11**

Jetzt erscheint die Zeilennummer (001) des Programmspeichers in der Anzeige; dies ist ein Anzeichen dafür, daß die vollständige Programmanweisung in dieser Zeile des Programmspeichers abgelegt wurde. Die im rechten Teil der Anzeige erscheinenden Tasten-Codes geben die entsprechenden Operationen an. Der Code 25 steht für **[h]**, 13 für **LBL** und die Zahl 11 für **[A]**. Die einzelnen Anweisungen werden stets erst in dem Moment in den Programmspeicher geladen, wenn Sie alle zugehörigen Tasten (entweder 1, 2 oder 3 aufeinanderfolgende Tasten) gedrückt haben.

Geben Sie jetzt auch die übrigen Programmschritte in den Rechner ein. Beachten Sie dabei die in der Anzeige erscheinende Nummer der Programmspeicherzeile und vergleichen Sie den Tasten-Code mit den auf dem Tastenfeld gedrückten Funktionstasten.

Tastenfolge	Anzeige
9	002- 15 3
h	003- 25 73
x	004- 61
h	005- 25 12

Das Programm für die Berechnung einer Kreisfläche zu vorgegebenem Radius steht jetzt im Programmspeicher des HP-34C zur Verfügung. Beachten Sie, daß in die Programmspeicherzeile 000 keine Informationen geschrieben werden konnten.

AUSFÜHRUNG EINES PROGRAMMS

Programme werden im automatischen RUN-Modus ausgeführt. Dazu muß der **PRGM** -Schalter wieder in Stellung RUN geschoben, eventuell erforderliche Daten eingetastet und anschließend eine der Programmtasten **A** oder **B** (entsprechend der Marke zu Beginn Ihres Programms) gedrückt werden.

Verwenden Sie das gespeicherte Programm, das im Rechner zur Verfügung steht, zur Berechnung der Fläche verschiedener Kreise mit den Durchmessern 3 Zoll, 6 Meter und 9 Meilen:

Schieben Sie als erstes den **PRGM** -Schalter in Stellung RUN.

Tastenfolge	Anzeige
3 A	28,2743
6 A	113,0973
9 A	254,4690

Quadratzoll Quadratmeter Quadratmeilen

AUFSUCHEN EINER MARKE

Als Sie nach Eintasten des Programms den **PRGM** -Schalter in Stellung RUN zurückgesetzt haben, befand sich der Rechner an der Zeile 005 im Programmspeicher. In diese Zeile hatten Sie die letzte Anweisung des gespeicherten Programms geschrieben. Mit dem Drücken der Taste **A** begann der Rechner nun den Programmspeicher ab der Zeile 005 nach einer **LBL** **A**-Anweisung abzusuchen. Während dieses Suchvorgangs wurden keine sonstigen Anweisungen ausgeführt.

Da Zeile 005 keine Anweisung **h LBL A** enthielt und außerdem keine weiteren Zeilen des Programmspeichers belegt sind, ist der Rechner nach Zeile 000 zurückgekehrt und hat den Suchvorgang von dort aus fortgesetzt.



AUSFÜHRUNG DER GESPEICHERTEN PROGRAMMZEILEN

Nachdem der Rechner in Zeile 001 die Anweisung **h LBL A** gefunden hatte, wurde der Suchprozeß abgebrochen und mit der Ausführung des Programms begonnen. Bei diesem »Abarbeiten« der gespeicherten Programmschritte geht der Rechner in der Reihenfolge vor, in der Sie die Anweisungen eingetastet haben. Als erstes wird die **g x^2**-Funktion in Zeile 002 ausgeführt, anschließend **h T** in Zeile 3 usw., bis eine **h RTN**- oder **R/S**-Anweisung erreicht wird. Da der Rechner in Zeile 005 eine **h RTN**-Anweisung ausführt, kehrt er nach Zeile 000 zurück, bricht die Ausführung des Programms ab und zeigt den Inhalt des X-Registers an.

Es ist im allgemeinen besser mit den Marken **A** und **B** den Anfang eines Programms zu kennzeichnen und **LBL 0** bis 9 als Unterprogrammarken zu verwenden (Unterprogramme werden noch gesondert behandelt). Warum? Bei den Marken **A** und **B** genügt ein Tastendruck zum Starten des Programms, wie dies bei unserer Kreisflächenberechnung der Fall war.

Wenn Sie aber mehrere kurze Programme in Ihrem HP-34C speichern wollen, können Sie die Marken 0 bis 9 zur Adressierung dieser einzelnen Programme verwenden. Bei numerischen Marken ist eine weitere Taste **GSB** zur Ausführung des Programms erforderlich. Wir wollen das veranschaulichen, indem wir die Befehlsfolge für die Kreisflächenberechnung unter Verwendung der Marke 0 in den Rechner laden: Schieben Sie den PRGM/RUN-Schalter in Stellung PRGM.

Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL 0	001- 25, 13, 0
g	002- 15 3
h	003- 25 73
x BST GSB	004- 61
h RTN	005- 25 12

Schalten Sie den PRGM/RUN-Schalter wieder in Stellung RUN;

Führen Sie jetzt das Programm aus, wobei Sie die Eingabedaten von Seite 52 verwenden. Da wir die Marke geändert haben, drücken Sie jetzt **GSB 0** anstatt **A**.

Tastenfolge	Anzeige
3 GSB 0	28,2743
6 GSB 0	113,0973
9 GSB 0	254,4690

Bei dem Versuch, ein Programm über eine Marke auszuführen, die nicht im Programmspeicher enthalten ist, meldet sich der Rechner mit der Fehlermeldung Error 4 in der Anzeige. Wenn beispielsweise Ihr HP-34C im Augenblick nur das eben eingetastete Programm zur Berechnung der Kreisfläche enthält, können Sie dies auf einfache Weise ausprobieren, indem Sie eine der anderen Programmtasten drücken.

Vergewissern Sie sich zuvor, daß der PRGM RUN -Schalter in Stellung RUN steht.

Tastenfolge	Anzeige
B	Error

Zum Löschen dieser Fehleranzeige können Sie **CLX** oder eine beliebige andere Taste auf dem Tastenfeld drücken. Sie können die Anzeige **Error** auch dadurch löschen, daß Sie den PRGM RUN -Schalter in Stellung PRGM schieben. Die augenblickliche Position im Programmspeicher wird dadurch nicht verändert.

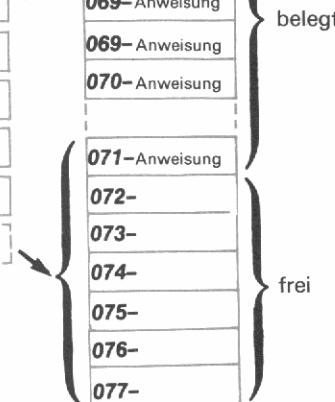
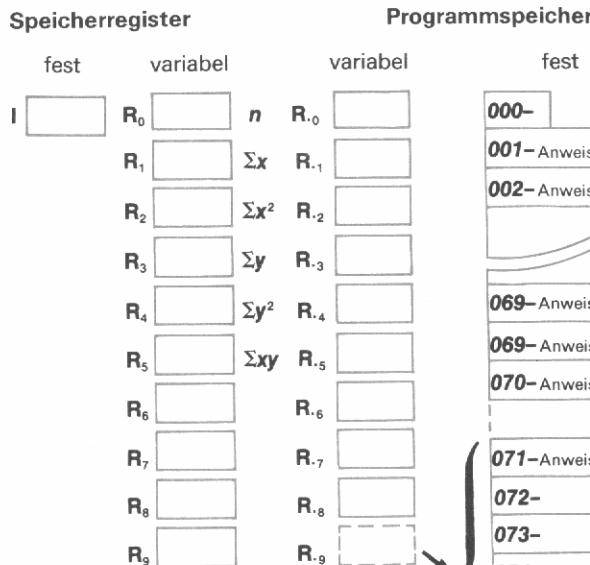
AUTOMATISCHE SPEICHERPLATZZUTEILUNG

UMWANDLUNG VON SPEICHERREGISTERN IN PROGRAMMSPEICHER

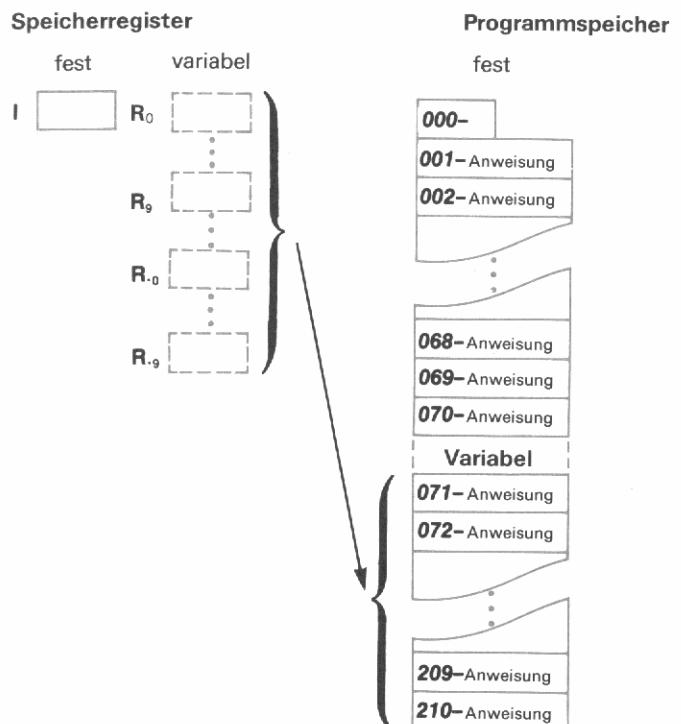
Die Vielseitigkeit Ihres HP-34C zeigt sich erneut in der automatischen Speicherplatzzuteilung, die je nach Bedarf Speicherregister in Programmspeicher umwandelt. In der Voreinstellung besteht der Rechner aus 70 Programmzeilen und 20 Speicherregistern (und dem I-Register, das in Abschnitt 7 beschrieben ist). Wenn der Programmspeicher aus 70 oder weniger Zeilen besteht, sieht die Speicherplatzzuteilung folgendermaßen aus:

Speicherregister			Programmspeicher	
fest	variabel	variabel	fest	
I	<input type="text"/>	n	R. ₀	000-
R ₁		Σx	R. ₁	001-
R ₂		Σx^2	R. ₂	002-
R ₃		Σy	R. ₃	⋮
R ₄		Σy^2	R. ₄	⋮
R ₅		Σxy	R. ₅	068-
R ₆			R. ₆	069-
R ₇			R. ₇	070-
R ₈			R. ₈	Variabel -keiner-
R ₉			R. ₉	

Wenn Sie die 71. Programmzeile eintasten, wird das Register R₉ in 7 zusätzliche Programmzeilen umgewandelt. Die Speicherplatzzuteilung sieht jetzt folgendermaßen aus:



Wenn Sie alle 210 Zeilen des Programmspeichers belegen wollen, sieht die Speicherplatzzuteilung wie folgt aus:



Der Programmspeicher, die vier Stack-Register und das LAST X-Register sind voneinander getrennt. Beachten Sie, daß von den ursprünglichen 21 Speicherregistern (R₀ bis R₉, R₀ bis R₉) nur das nicht umwandelbare I-Register übrig geblieben ist. Was geschah mit den Speicherregistern R₀ bis R₉ und R₀ bis R₉? Sie wurden in Blöcken von sieben Zeilen pro Register in Programmspeicher umgewandelt.

Die rechts aufgeführte Tabelle zeigt, welche Programmzeilen nach einer Neuzuordnung den Speicherregistern entsprechen.

Wenn alle 210 Programmzeilen belegt sind, erscheint bei dem Versuch, eine weitere Anweisung einzugeben, die Fehlermeldung ERROR 4 in der Anzeige.

Die zusätzliche Anweisung wird ignoriert und die ursprünglichen 210 Zeilen bleiben unverändert.

Wie Sie sehen, wird jedesmal, wenn der augenblicklich verfügbare Programmspeicher ganz belegt ist, das niedrigste verbleibende Speicherregister automatisch in sieben zusätzliche Programmzeilen umgewandelt, wenn ein weiterer Programmbebefl eingegeben wird. Wenn beispielsweise die ersten 77 Zeilen belegt sind und dann ein weiterer Befehl in Zeile 78 eingegeben wird, wandelt der Rechner Register R₈ in 7 zusätzliche Programmzeilen (78–84) um.

Anmerkung: Der HP-34C wandelt Register in der Reihenfolge R₉ bis R₀ und dann R₉ bis R₀ in Programmspeicher um.

Es ist daher sinnvoll, für **STO** - und **RCL** -Anweisungen die Speicherregister in der umgekehrten Reihenfolge, also mit R₀ beginnend, zu verwenden. So vermeiden Sie auch mit **STO** - und **RCL** -Anweisungen Register zu adressieren, die in Programmspeicher umgewandelt wurden. Beachten Sie außerdem, daß die ursprünglich in Speicherregistern enthaltenen Daten nach der Umwandlung in Programmspeicher verloren gehen.

UMWANDLUNG VON PROGRAMMSPEICHER IN SPEICHERREGISTER

Wenn **f CLEAR PRGM** im PRGM-Modus gedrückt wird, wandelt der Rechner den gesamten variablen Programmspeicher (Zeilen 071–210) in die Speicherregister R₀ bis R₉ um. Wenn Sie dagegen nur einzelne Programmzeilen löschen, können Sie auch Teile des variablen Speichers in Speicherregister umwandeln, ohne den ganzen Programmspeicher zu löschen. (Das Löschen von Programmzeilen wird in Abschnitt 4 behandelt.)

R ₉ → 071–077	R ₉ → 141–147
R ₈ → 078–084	R ₈ → 148–154
R ₇ → 085–091	R ₇ → 155–161
R ₆ → 092–098	R ₆ → 162–168
R ₅ → 099–105	R ₅ → 169–175
R ₄ → 106–112	R ₄ → 176–182
R ₃ → 113–119	R ₃ → 183–189
R ₂ → 120–126	R ₂ → 190–196
R ₁ → 127–133	R ₁ → 197–203
R ₀ → 134–140	R ₀ → 204–210



VERWENDUNG VON **MEM**

Die im PRGM- oder RUN-Modus ausgeführte **MEM**-Funktion gibt den augenblicklichen Status der Speicherplatzzuteilung wieder. Wenn Sie **g MEM** drücken, wird (1) die vor der Umwandlung eines weiteren Speicherregisters verbleibende Anzahl freier Programmzeilen und (2) die Bezeichnung des nächsten umzuwandelnden Registers (R₉ bis R₀, R₉ bis R₀) angezeigt.

Wenn beispielsweise 44 Zeilen des Programmspeichers belegt sind, wenn Sie **g MEM** drücken, werden Sie folgende Anzeige sehen:

p-26

Anzahl noch zu belegender Programmzeilen, bevor der Rechner automatisch ein Speicherregister in 7 weitere Programmzeilen umwandelt.

Wenn **g MEM** gedrückt wird, wenn 173 Zeilen des Programmspeichers belegt sind, werden Sie folgende Anzeige sehen:

p-02

Anzahl noch zu belegender Programmzeilen, bevor der Rechner automatisch ein Speicherregister in 7 weitere Programmzeilen umwandelt.

Wenn **g MEM** gedrückt wird, wenn 205 Zeilen des Programmspeichers belegt sind, werden Sie folgende Anzeige sehen:

p-05

Anzahl zu belegender Programmzeilen, bevor der ganze Programmspeicher belegt ist.

Solange Sie **MEM** gedrückt halten, wird die Speicherplatzzuteilung angezeigt. Wenn Sie die **MEM**-Taste loslassen, kehrt der Rechner zur ursprünglichen Anzeige zurück. Auf diese Weise können Sie jederzeit feststellen, wieviel Programmzeilen und wieviel Register Ihnen zur Datenspeicherung zur Verfügung stehen. Wegen seiner besonderen Funktionen ist das I-Register ein permanentes Speicherregister, welches bei der **MEM**-Funktion unberücksichtigt bleibt.

r-9

Das nächste Speicherregister, das umgewandelt wird.

r-4

Das nächste Speicherregister, das umgewandelt wird (d. h. das ranghöchste verbleibende Register).

r-

Weitere Speicherregister können nicht mehr in Programmspeicher umgewandelt werden.

Anmerkung: Bedenken Sie, daß die Statistik-Funktionen die Register R₀ bis R₅ verwenden. Wenn ein oder mehrere dieser sechs Register in Programmsspeicher umgewandelt sind, erscheint bei dem Versuch, die Statistik-Funktionen zu verwenden, die Fehlermeldung ERROR 2 in der Anzeige.

EIN DRITTES PROGRAMM

Wir wollen jetzt ein anderes Programm erstellen, um daran weitere Einzelheiten zur Programmierung des HP-34C aufzuzeigen. Nehmen wir einmal an, Sie wollen ein Programm schreiben, das die Volumenänderung eines kugelförmigen Ballons bei Veränderung des Durchmessers berechnet. Dazu verwenden wir die Formel:

$$\text{Volumenänderung} = \frac{1}{6}\pi (d_1^3 - d_0^3)$$

Dabei ist mit d₀ der ursprüngliche Durchmesser des Ballons und mit d₁ der neue Durchmesser gemeint. Wenn Sie d₀ in das Y-Register und d₁ in das X-Register eingeben, können Sie die Rechnung manuell ausführen, indem Sie die Tasten drücken, die in der folgenden Tabelle in der linken Spalte angegeben sind.

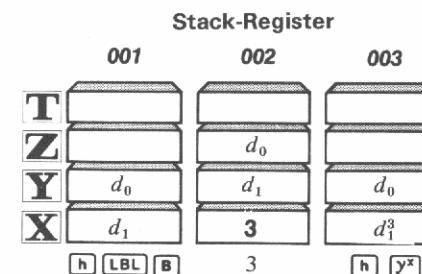
Die Programmtasten für dieses Problem sind dabei identisch mit der manuellen Tastenfolge. Schieben Sie den PRGM-RUN-Schalter in Stellung PRGM und drücken Sie die folgenden Tasten:

Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL B	001- 25, 13, 12
3	002- 3
h yx	003- 25 3 d ₁ ³
x ² y	004- 21
3	005- 3
h yx	006- 25 3 d ₀ ³
-	007- 41 d ₁ ³
h π	008- 25 73
x	009- 61 Multiplikation mit π
6	010- 6
÷	011- 71 Division durch 6
h RTN	012- 25 12



Beachten Sie, daß keine ENTER-Anweisung verwendet wurde, um die Zahl 3 in Zeile 002 von den Ziffern zu trennen, die Sie später eingegeben werden. Die Verwendung von **ENTER+** nach der **LBL**-Anweisung würde in diesem Beispiel zu keinem Fehler führen, doch ist sie unnötig.

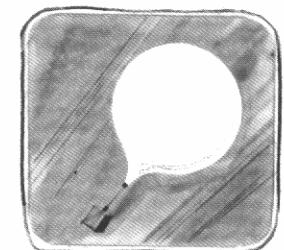
Warum? Wenn ein Programm die **LBL**-Anweisung ausführt, wird der Stack-Lift wirksam gemacht. Wenn dann durch das Programm (automatisch) eine neue Zahl in das X-Register eingegeben wird, wird der Stack automatisch angehoben. Mit d₀ im Y-Register und im d₁ im X-Register findet bei der Ausführung der Programmbefehle folgender Ablauf statt:



Im Anhang E <Stack-Lift und LAST X> können Sie nachlesen, wie alle Operationen den Stack beeinflussen.

Beispiel:

Wie ändert sich das Volumen eines Ballons, der von einem ursprünglichen Durchmesser von 30 Meter auf 35 Meter aufgeblasen wird?



Tastenfolge	Anzeige	
30 ENTER+	30,0000	Eingabe des ursprünglichen Wertes für den Durchmesser (d ₀)
35 B	8.312,1306	Eingabe des neuen Durchmessers nach X und Starten des Programms. Das Resultat (in m ³) wird angezeigt.

UNTERBRECHUNG DER PROGRAMMAUSFÜHRUNG

Gelegentlich soll ein Programm anhalten, damit Sie Daten eingeben oder Zwischenergebnisse anzeigen können. Diese Programmunterbrechung kann auf verschiedene Weise erfolgen, indem Sie entweder **R/S** oder **PSE** als Bestandteil eines Programms mit eingeben.

ANHALTEN EINES PROGRAMMS

R/S wirkt unterschiedlich, je nachdem, ob es als Bestandteil eines gespeicherten Programms ausgeführt oder vom Tastenfeld aus gedrückt wird.

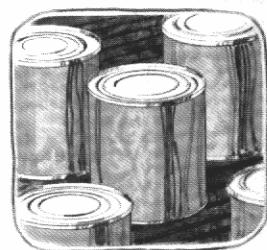
R/S hat vom Tastenfeld aus gedrückt folgende Wirkung:

1. **R/S** hält ein augenblicklich laufendes Programm an.
2. Wenn die Ausführung eines Programms zuvor angehalten oder noch nicht begonnen wurde und der Rechner im RUN-Modus arbeitet, startet **R/S** die Programmausführung ab der augenblicklichen Speicherposition.

(Solange Sie die Taste **R/S** gedrückt halten, zeigt der Rechner die Schritt-nummer und den Code der gespeicherten Anweisung an; nach Loslassen der Taste wird das Programm gestartet.)

Sie können diese Eigenschaften der **R/S**-Anweisung zum Anhalten des Programms für die Eingabe von Daten bzw. das Ablesen von Zwischen-ergebnissen verwenden. Nachdem Sie die Daten eingegeben haben, können Sie das Programm mit **R/S** vom Tastenfeld aus erneut starten.

Programmbeispiel: Eine Konservenfabrik möchte die Volumen verschiedener zylinder-förmiger Blechdosen berechnen. Außerdem möchte sie die Grundfläche jeder Dose notieren, bevor das Volumen berechnet wird.



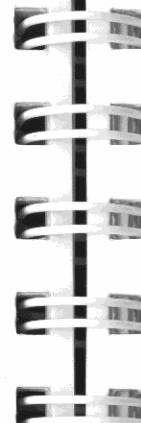
Das nachstehende Programm berechnet zu jeder Dose die Grundfläche und hält dann an. Wenn Sie den Wert notiert haben, können Sie das Programm erneut starten, damit es das Volumen berechnet. Die verwendete Formel lautet:

$$\text{Volumen} = \text{Grundfläche} \times \text{Höhe} = \pi r^2 \times h.$$



Bevor das Programm gestartet wird, werden der Radius (*r*) und die Höhe der Dose (*h*) in die Register **X** und **Y** eingegeben. Um dieses Programm in den Programmspeicher zu schreiben, schalten Sie den **PRGM** **RUN**-Schalter in Stellung **PRGM**.

Geben Sie jetzt die nachfolgende Tastenfolge ein:



Tastenfolge

f **CLEAR** **PRGM**

h **LBL** **A**
g **x²**
h **π**
x
R/S
x
h **RTN**

Anzeige

000-

001- 25, 13, 11
002- 15 3
003- 25 73
004- 61
005- 74
006- 61
007- 25 12

Anzeige

000-

Löschen des Programmspeichers.
Zeile 000 wird angezeigt.

Berechnet *r*²
Speichert π nach *x*
Berechnet die Grundfläche
Hält zur Anzeige der Grundfläche an
Volumen der Dose

Schalten Sie zur Ausführung dieses Programms den Moduswahlschalter in Stellung **RUN** (**PRGM** **RUN**).

Vervollständigen Sie jetzt mit Hilfe dieses Programms die nachstehende Tabelle:

Höhe cm	Radius cm	Grundfläche cm ²	Volumen cm ³
25	10,0	?	?
8	4,5	?	?



Tastenfolge

25 **ENTER**
10 **A**

25,0000

314,1593

Anzeige

25,0000

314,1593

Höhe *h* nach **Y** speichern
Programm hält zur Anzeige der Grundfläche an



R/S

7,853,9816

8,0000

63,6173

Anzeige

7,853,9816

8,0000

63,6173

Volumen der 1. Dose

Höhe *h* nach **Y** speichern

Programm hält zur Anzeige der Grundfläche an



R/S

508,9380

508,9380

Volumen der 2. Dose

Nach Eingabe der Höhe in das **Y**-Register und des Radius in das **X**-Register wird durch Drücken von **A** im automatischen RUN-Modus die Grundfläche berechnet. Das Programm hält an, sobald es auf die erste **R/S**-Anweisung trifft. Wird jetzt **R/S** gedrückt, wird das Volumen berechnet und das Programm kehrt nach Zeile 000 zurück und hält an.

KURZFRISTIGE UNTERBRECHUNG DER PROGRAMM-AUSFÜHRUNG

Eine **[PSE]**-Anweisung innerhalb eines Programms unterbricht die Ausführung eines Programms für einen Moment, so daß Werte angezeigt werden können, die nicht notiert werden müssen. Die Länge der Programmunterbrechung beträgt ca. 1 Sekunde. Falls Sie längere Pausenzeiten wünschen, können Sie mehrere **[PSE]**-Anweisungen hintereinander in das Programm einfügen.

Um zu sehen, wie **[PSE]** wirkt, wollen wir das Programm zur Berechnung des Zylindervolumens etwas ändern. In dem neuen Programm wird jetzt die Grundfläche nur noch für einen Augenblick angezeigt, bevor das Programm selbstständig mit der Berechnung des Volumens fortfährt.

Außerdem zeigt dieses Beispiel, wie man sich einem Problem auf verschiedene Arten nähern kann.

Schalten Sie zum Eintasten des Programms in Stellung PRGM und drücken Sie **[CLEAR PRGM]** zum Löschen des Programmspeichers. Geben Sie jetzt die nachfolgenden Programmschritte ein.

Tastenfolge	Anzeige
[CLEAR PRGM]	000-
[LBL A]	001- 25, 13, 11
[X²]	002- 15 3 Berechnet r^2
[T]	003- 25 73 Speichert Pi nach X
[x]	004- 61 Berechnet die Grundfläche
[PSE]	005- 25 74 Zeigt kurzfristig die Fläche an
[x]	006- 61 Berechnet das Volumen
[RTN]	007- 25 12

Dieses Programm geht ebenso davon aus, daß die Höhe zuvor in das Y-Register und der Radius in das X-Register eingegeben wird. Wenn Sie die Tastenfolge eingegeben haben, schalten Sie den **PRGM RUN**-Schalter in die Stellung RUN und berechnen Sie jetzt die noch fehlenden Tabellenwerte.

Höhe cm	Radius cm	Grundfläche cm^2	Volumen cm^3
20	15	?	?
10	5	?	?

	Tastenfolge	Anzeige	
	20 [ENTER]	20,000	Eingabe der Höhe nach Y
	15 [A]	706,8583	Grundfläche wird eine Sekunde lang angezeigt
	10 [ENTER]	14.137,1669	Volumen der ersten Dose
	5 [A]	10,0000	Eingabe der Höhe nach Y
		78,5398	Grundfläche wird eine Sekunde lang angezeigt
		785,3982	Volumen der zweiten Dose

PROGRAMM-STOPS

Gelegentlich wird es vorkommen, daß das Programm wegen eines Fehlers irgendeiner Art unterbrochen wird. Die folgende Aufstellung der möglichen Ursachen für das Anhalten eines laufenden Programms kann als Hilfe bei der Fehlersuche betrachtet werden.

- Ausführung von **[RTN]**.
Wenn **[RTN]** nicht als Bestandteil eines Unterprogrammes ausgeführt wird, kehrt der Rechner nach Zeile 000 zurück und hält an.
- Erreichen des Endes des Programmspeichers.
Wenn die letzte Anweisung im Programmspeicher nicht **GTO**, **RTN**, **GSB** oder **R/S** ist oder Bestandteil eines Unterprogrammes, kehrt der Rechner sofort nach Zeile 000 zurück und hält an.
- Drücken einer beliebigen Taste.
Wird während eines Programmlaufs eine beliebige Taste gedrückt, wird das Programm angehalten. Achten Sie daher darauf, daß Sie nicht während eines Programmlaufs versehentlich eine Taste drücken.

In dem Entwurf des Rechners ist vorgesehen, daß ein Programm nicht innerhalb einer Zahleneingabe anhalten kann. Wenn Sie eine Taste drücken, während ein laufendes Programm Zahlen in das X-Register eingibt, wird die Zahl vollständig geschrieben und der nächste Programmbebefl ausgeführt, bevor der Rechner anhält.

Ist das Programm angehalten, kann es durch Drücken der Taste **R/S** im RUN-Modus wieder fortgesetzt werden. Wenn Sie **R/S** drücken, wird das Programm mit dem nächsten Schritt fortgesetzt, als ob keine Unterbrechung stattgefunden hätte.

FEHLER-STOPS

Operationen, die eine Fehlermeldung zur Folge haben (siehe Fehleranzeigen im Anhang D) unterbrechen den Programmlauf sofort. Der Rechner zeigt die Meldung **<Error>** und eine Zahl an. Wenn Sie den Rechner kurzzeitig in den PRGM-Modus umschalten, zeigt der Rechner den Programmschritt und den Tastencode der unerlaubten bzw. fehlerverursachenden Operation an.

ÜBERLAUF

Ihr HP-34C ist so konzipiert worden, daß es jederzeit möglich ist, den Grund für ein Anhalten bei der Programmausführung in der Anzeige zu erkennen. Wenn das Ergebnis einer Rechnung im **X**-Register größer als $9.99999999 \times 10^{99}$ ist, zeigt der Rechner in jeder Position der Anzeige eine 9 mit entsprechendem Vorzeichen an, und ein eventuell laufendes Programm wird angehalten. Wenn Sie jetzt in den PRGM-Modus umschalten, erkennen Sie leicht, welche Operation für den Überlauf verantwortlich war.

Wenn ein Überlauf in einem der Speicherregister durch Speicherregister-Arithmetik auftreten würde, hält der Rechner mit der Fehlermeldung **ERROR 1** in der Anzeige an. Der Inhalt des entsprechenden Speicherregisters bleibt dabei unverändert. Wenn Sie die Fehlermeldung löschen, erscheint der vorherige Inhalt des **X**-Registers in der Anzeige.

Wenn das Ergebnis einer Zahl kleiner ist als 10^{-99} , ersetzt der Rechner diese Zahl durch Null. Ein laufendes Programm wird dadurch nicht angehalten.

Dies wird als Unterlauf bezeichnet.

MARKEN

Die Marken (**A**, **B** und 0 bis 9) haben im Programm die Funktion einer Adresse – sie sagen dem Rechner, wo er die Ausführung gespeicherter Programme zu beginnen oder wieder aufzunehmen hat. Wenn der Rechner während der Ausführung eines Programms zu einer Marke kommt, die Bestandteil dieses Programms ist, überspringt der Rechner diese Marke und setzt die Ausführung der nachfolgenden Programmschritte fort. Wenn Sie beispielsweise im nachstehend abgebildeten Programmteil die Taste **A** drücken, wird die Ausführung der Programmzeilen mit **LBL A** beginnen; Die Marke **LBL 3** wird überlesen und der Rechner wird nach **H RTN** in Zeile 000 angehalten.



Hier beginnt die Ausführung der Programmzeilen wenn Sie **A** drücken.

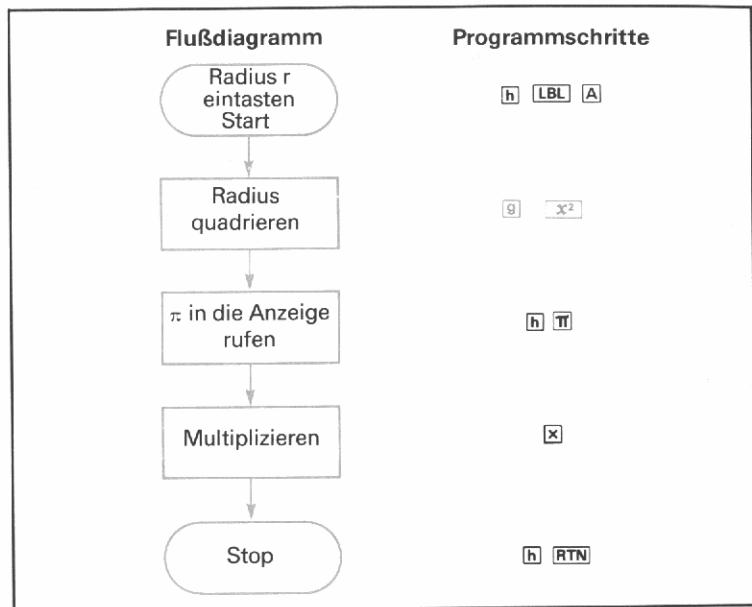
Hier steht kein **RTN**.
Daher wird **LBL 3** überlesen.

Die Programmausführung verzweigt an dieser Stelle nach Zeile 000 und hält an.

FLUSSDIAGRAMM

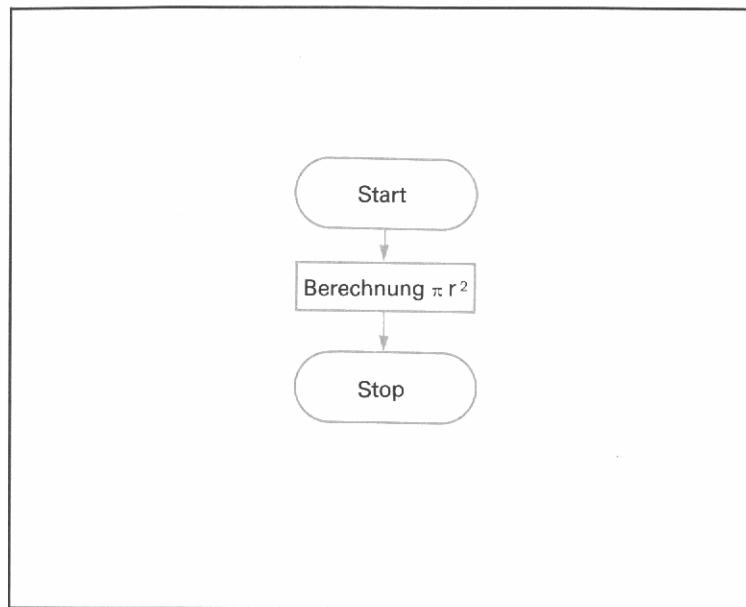
Wir wollen an dieser Stelle die Erklärung über den Rechner selbst für einen Moment unterbrechen und uns mit einem sehr wertvollen Hilfsmittel bei der Programmierung – dem Flußdiagramm – befassen. Die Entwicklung eines «Flußdiagramms» ist eine wertvolle Hilfe bei der Entscheidung, wie ein bestimmtes Problem gelöst werden soll. Als Zwischenschritt bei der Entwicklung eines Programms hilft Ihnen dieses Fluß- oder Ablaufdiagramm bei der Ermittlung des günstigsten Lösungsweges. In diesem Stadium ist es aufgrund der Übersichtlichkeit noch recht einfach, Änderungen am Lösungsgang vorzunehmen oder logische Fehler zu erkennen.

Ein Flußdiagramm kann so einfach oder so ausführlich sein, wie Sie das möchten. Nachstehend ist ein Ablaufdiagramm für die Berechnung der Kreisfläche nach der Formel $A = \pi r^2$ angegeben. Vergleichen Sie einmal die Anweisungen des Flußdiagramms mit den einzelnen Schritten des entsprechenden Programms:



Die Ähnlichkeiten sind deutlich zu erkennen.

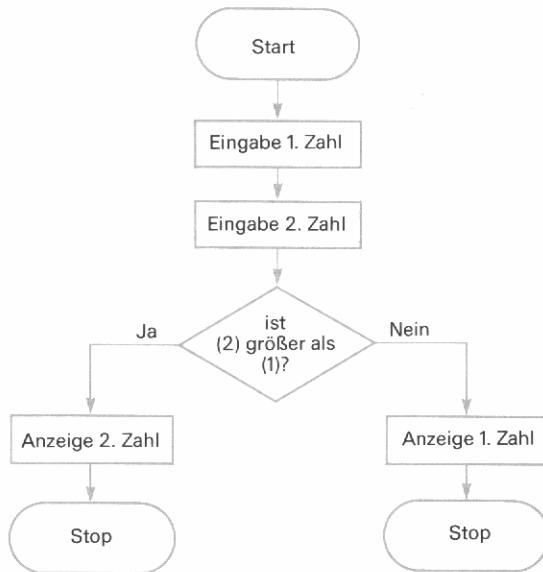
Manchmal gibt ein Flußdiagramm die Programmschritte exakt wieder, wie das oben der Fall war. Manchmal ist es jedoch sinnvoller, einen ganzen Programmteil durch ein einziges Element im Flußdiagramm darzustellen. Für die Flächenberechnung eines Kreises sieht das folgendermaßen aus:



Hier werden mehrere Programmschritte durch ein Element im Flußdiagramm dargestellt. Diese Technik wird allgemein angewandt und ermöglicht dem Betrachter, sich mittels des Flußdiagramms ein Bild von dem gesamten Programm zu machen.

Sie sehen, daß ein Flußdiagramm linear von oben nach unten gezeichnet wird. Das gibt die Richtung des Programms von Anfang bis Ende an. Die im Flußdiagramm verwendeten Symbole haben manchmal unterschiedliche Bedeutung. In diesem Handbuch jedoch verwenden wir abgerundete Kästchen, um Anfang und Ende von Programmen oder Routinen anzudecken, und Rechtecke, um Funktionen, die eine Eingangsgröße erhalten, diese verarbeiten und einen Ausgangswert liefern, darzustellen. Rauten werden dort eingesetzt, wo eine Eingangsgröße eine von zwei Ausgängen zur Folge haben könnte.

Wenn Sie beispielsweise ein Programm schreiben wollen, welches von zwei Zahlen nur die größere anzeigen soll, dann könnte das Programm in Form eines Flußdiagramm folgendermaßen dargestellt werden:



Nachdem Sie das Flußdiagramm gezeichnet haben, würden Sie jedes Element des Flußdiagramms durch eine Befehlsfolge ersetzen. Wenn das Programm in den Rechner geladen und getestet wird, so würde, wenn (2) größer als (1) ist, die Antwort auf die Frage, ist (2) größer als (1)?, Ja lauten. Das Programm würde mit der linken Verzweigung fortfahren, die Zahl (2) anzeigen und anhalten. Wenn die Antwort auf die Abfrage Nein ist, würde das Programm mit der rechten Verzweigung fortfahren, die Zahl (1) anzeigen und anhalten. Sie werden später die vielfältigen Entscheidungsmöglichkeiten mit Ihrem HP-34C kennenlernen.

Während Sie dieses Handbuch durcharbeiten, werden Sie mit Flußdiagrammen mehr vertraut werden. Benutzen Sie die Flußdiagramme, die hier die Beispiele und Probleme erläutern zum schnelleren Verständnis der Eigenschaften des Rechners. Verwenden Sie eigene Flußdiagramme, um Programme zu erstellen und editieren, um Fehler zu beseitigen und um Programme zu dokumentieren.

Übungsaufgaben

Die folgenden vier Übungsaufgaben sind anhand des bis jetzt behandelten Stoffes zu lösen. Mögliche Lösungen werden auch angeboten. Doch ziehen Sie den größten Nutzen aus den Übungen, wenn Sie Ihre eigenen Lösungen bilden, bevor Sie die gegebenen Lösungen betrachten. Auch gibt es für die meisten Probleme mehr als eine Lösung. Vielleicht finden Sie bessere Lösungswege als wir.

1. Sie haben gesehen, wie ein Programm für die Berechnung der Kreisfläche zu gegebenem Radius geschrieben, im Rechner gespeichert und anschließend ausgeführt wird. Schreiben Sie jetzt ein Programm, das umgekehrt zu gegebener Kreisfläche A den zugehörigen Radius nach der Formel $r = \sqrt{A/\pi}$ berechnet. Achten Sie darauf, daß als erstes der PRGM [PRGM] RUN - Schalter in Stellung PRGM zu schieben und der Programmspeicher mit [CLEAR PRGM] zu löschen ist. Kennzeichnen Sie dieses Programm mit [LBL] A und [RTN]. Nachdem Sie das Programm eingetastet haben, berechnen Sie die Radien, die folgenden Kreisflächen entsprechen: 28,2743 Quadratzoll, 113,0973 Quadratmeter und 254,4690 Quadratmeilen.

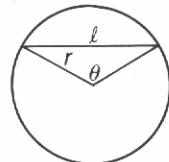
(Ergebnisse: 3,0000 Zoll, 6,0000 Meter, 9,0000 Meilen)

2. Schreiben Sie ein Programm zur Berechnung der Länge l einer Bogensehne gegenüber einem Winkel θ auf einem Kreis mit Radius r .

Verwenden Sie die Formel $l = 2r \sin \frac{\theta}{2}$.

Definieren Sie dieses neue Programm mit **H** **LBL** **B** und vervollständigen Sie folgende Tabelle:

r (Meter)	θ	1
25	30	?
50	45	?
100	90	?



Halten Sie bei der Dateneingabe an die Reihenfolge r, θ .

(Ergebnisse: 12.9410 Meter, 38.2683 Meter, 141.4214 Meter)

Wenn Sie bei diesem Problem auf einige Schwierigkeiten stoßen sollten, schauen Sie sich den Absatz **«Ein drittes Programm»** auf Seite 60 an.

3. Erstellen Sie ein Programm, das in Grad Celsius gegebene Temperaturen nach der Beziehung $F = 1,8^\circ\text{C} + 32$ in Grad Fahrenheit umrechnet. Kennzeichnen Sie dieses Programm mit **H** **LBL** **0** und **H** **RTN** und verwenden Sie es zur Umwandlung folgender Temperaturen: -40°C , 0°C und $+72^\circ\text{C}$.

(Ergebnisse: $-40,0000^\circ\text{F}$, $32,0000^\circ\text{F}$, $161,6000^\circ\text{F}$)

4. Schreiben Sie im Anschluß an Aufgabe 3 ein Programm, das die in Grad Fahrenheit ausgedrückten Temperaturen wieder in Grad Celsius zurückrechnet. Verwenden Sie dabei die Beziehung $C = \frac{5}{9} \times (F - 32)$ und markieren Sie das Programm mit **H** **LBL** **1**. Tasten Sie die Programmschritte jetzt im Anschluß an das in Aufgabe 3 eingegebene Programm in den Speicher ein. Jetzt können Sie die in der dritten Aufgabe in Grad Fahrenheit umgerechneten Temperaturen wieder in Grad Celsius umrechnen.

Wenn Sie die Programme der dritten und vierten Aufgabe in der angegebenen Weise erstellt und in den Rechner eingegeben haben, können Sie beliebige Temperaturwerte durch Drücken der Tasten **GSB** 0 von $^\circ\text{C}$ in $^\circ\text{F}$ und durch Drücken der Tasten **GSB** 1 von $^\circ\text{F}$ in $^\circ\text{C}$ umrechnen.

Haben Sie noch Fragen? Vielleicht lesen Sie sich nochmal den Stoff auf Seite 53 durch, der sich mit **GSB** und den Marken 0 bis 9 zur Adressierung von einzelnen Programmen befaßt.

1. $r = \sqrt{A/\pi}$

Fläche eintasten
Start

in das X-Reg.
laden

Dividieren-

$\sqrt{\text{Quotient}}$

Stop

2. $l = 2r \sin \frac{\theta}{2}$

r, θ eintasten
Start

$\sin \frac{\theta}{2}$

$2r$

Multiplizieren

Stop

3. $F = 1,8^\circ\text{C} + 32$

$^\circ\text{C}$ eintasten
Start

$1,8^\circ\text{C}$

Konstante
addieren

Stop

4. $^\circ\text{C} = (F-32) \cdot \frac{5}{9}$

$^\circ\text{F}$ eintasten
Start

Konstante
subtrahieren

Zwischenresultat $\times \frac{5}{9}$

Stop

LÖSUNGEN ZU DEN ÜBUNGSAUFGABEN

	Tastenfolge	Anzeige
Radius eines Kreises	[h] [LBL] A [h] [Π] [÷] [f] [√X] [h] [RTN]	001-25,13,11 002- 25 73 003- 71 004- 14 3 005- 25 12
Länge einer Bogensehne	[h] [LBL] B 2 [÷] [f] [SIN] [x] 2 [x] [h] [RTN]	001-25,13,12 002- 2 003- 71 004- 14 7 005- 61 006- 2 007- 61 008- 25 12
Umwandlung von ° Celsius in ° Fahrenheit	[h] [LBL] 0 1 [x] 8 [x] 3 2 [+] [h] [RTN]	001-25,13, 0 002- 1 003- 73 004- 8 005- 61 006- 3 007- 2 008- 51 009- 25 12
Umwandlung von ° Fahrenheit in ° Celsius	[h] [LBL] 1 3 2 [−] 5 [x] 9 [÷] [h] [RTN]	010-25,13, 1 011- 3 012- 2 013- 41 014- 5 015- 61 016- 9 017- 71 018- 25 12



TASTENFOLGE ZUR LÖSUNG DER PROBLEME

1. nnFläche [A] = Radius
2. nnr [ENTER+] nnθ [B] = Länge der Bogensehne
3. nn°C [GSB] 0 = °F
4. nn°F [GSB] 1 = °C



Nochmals herzlichen Glückwunsch. Sie sind im Begriff, die vielen Eigenschaften Ihres HP-34C Rechners zu meistern.



PROGRAMMIERTECHNIKEN

Bei der Lösung von gewissen Arten von Problemen wird dieselbe Variable wiederholt während der Berechnung benötigt. Sie haben schon erfahren, daß es mehrere Lösungswege für Probleme dieser Art für Ihren HP-34C gibt. Es ist jedoch häufig wünschenswert, ein Programm zu schreiben, das sowohl bezüglich der Ausführungszeit als auch des benötigten Programmspeichers gleichermaßen günstig ist.



Wir wollen jetzt zwei unterschiedliche Lösungswege für ein Problem untersuchen, in dem eine Variable mehrfach verwendet wird.



Das Polynom $f(x) = x^4 - 3x^3 - x^2 - 4x - 1$ verwendet beispielsweise die Variable x viermal, d. h. x^4, x^3, x^2, x . Das bedeutet, daß vier Potenzen von x zur Berechnung von $f(x)$ benötigt werden. Ihre Aufgabe ist es, ein Programm zu schreiben, mit dem Sie $f(x)$ mathematisch ausdrücken und das den Wert der Variablen x dann zur Verfügung stellt, wenn er bei der Ausführung des Programms benötigt wird. Das kann recht wirksam auf zwei verschiedene Arten bewerkstelligt werden. Entweder speichern Sie den Wert der Variablen, so daß Sie ihn später bei Bedarf einfach zurückrufen können, oder sie schreiben das Programm auf solche Weise, daß die Variable vor Ausführung des Programms lediglich in den Stack geladen werden muß. Die Vorteile der Stack-Lade-Methode gegenüber der Speichermethode sind folgende:



1. Es ist leicht, das Programm so zu gestalten, daß die Variable entweder unmittelbar oder nach einer **x,y**-Anweisung zur Verfügung steht. (Bedenken Sie, daß nach allen Operationen, die den Stack nach unten verschieben, der vorherige Inhalt des T-Registers in das Z-Register kopiert wird.) Auf diese Weise werden Sie weniger Programmzeilen benötigen, da **STO** - und **RCL** -Anweisungen unnötig werden.

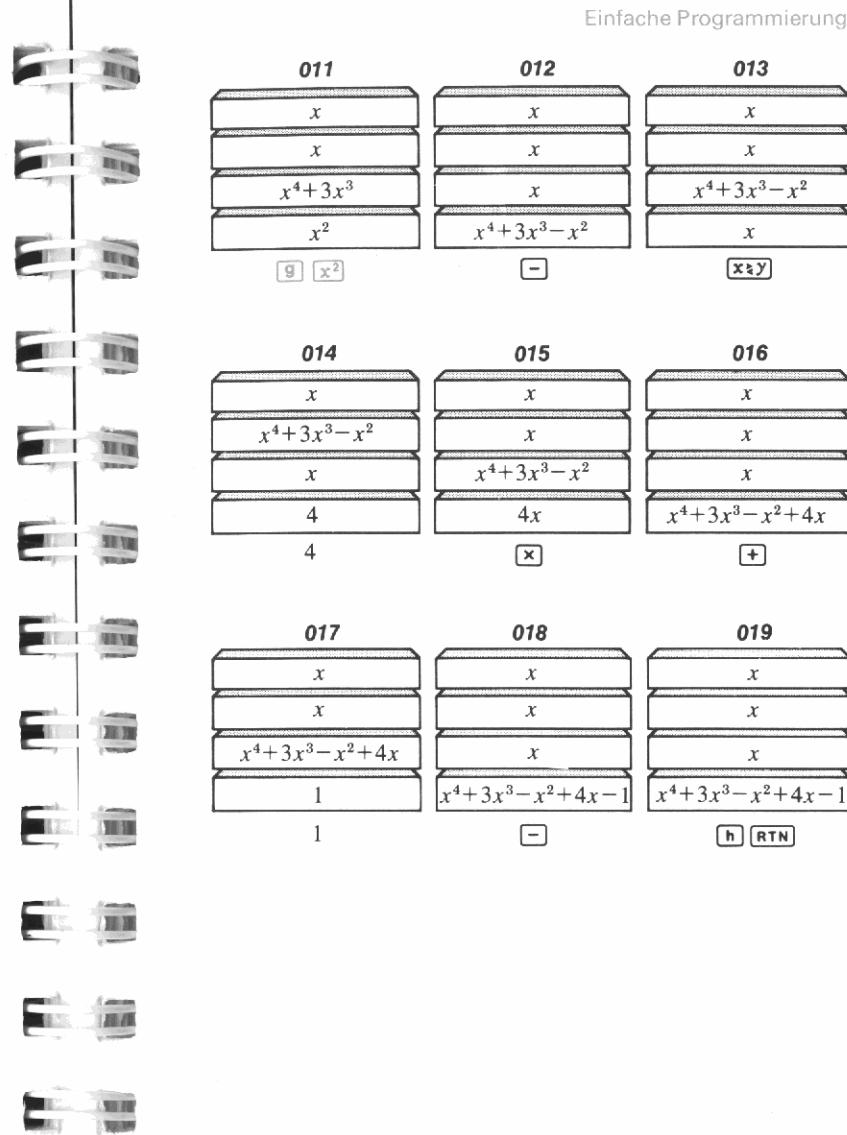
2. Ein Speicherregister bleibt für andere Zwecke frei.

3. Die Stack-Lade-Methode ist allgemein für die Berechnung von Polynomen und für Anwendungen in Verbindung mit **SOLVE** und **DISP** günstig. (**SOLVE** und **DISP** wird später noch behandelt.)

Jetzt wollen wir ein Programm untersuchen, das mittels der Stack-Lade-Methode den Ausdruck $x^4 + 3x^3 - x^2 + 4x - 1$ berechnet. Wir betrachten dabei nur die Programmanweisungen und den Stack-Inhalt. Schauen Sie sich jede Programmanweisung genau an, damit Sie verstehen, wie und warum jede Anweisung den Stack beeinflußt. Gehen Sie davon aus, daß der Wert x sich vor jeder Programmausführung schon im Stack befindet.

Stack-Register				
001	002	003	004	005
x	x	x	x	x^4
x	x	x	x^4	x
x	x	x^4	x	3
h LBL A	4	h y^x	x,y	

006	007	008	009	010
x	x	x	x	x
x	x^4	x	x	x
x^4	x^3	x^4	x	$x^4 + 3x^3$
x^3	3	$3x^3$	$x^4 + 3x^3$	x
h y^x	3	x	+	x,y



Beachten Sie, daß die Variable x nach Beendigung des Programms weiterhin im Stack vorhanden ist. Wenn Sie aus irgendeinem Grund nach der Berechnung von $f(x)$ die Variable in die Anzeige zurückrufen wollen, drücken Sie einfach xy .

Um die Stack-Lade-Methode auszuprobieren, tasten Sie das Programm zur Berechnung des obenstehenden Ausdruckes in den Rechner und lassen Sie das Programm einige Beispiele durchrechnen.

Schieben Sie den PRGM RUN -Schalter in Stellung PRGM.

Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL A	001-25,13,11
4	002- 4
h y^x	003- 25 3
xy	004- 21
3	005- 3
h y^x	006- 25 3
3	007- 3
x	008- 61
+	009- 51
xy	010- 21
g x^2	011- 15 3
-	012- 41
xy	013- 21
4	014- 4
x	015- 61
+	016- 51
1	017- 1
-	018- 41
h RTN	019- 25 12

Schieben Sie den PRGM RUN -Schalter in Stellung RUN und berechnen Sie den Wert des Ausdrucks mit folgenden Werten von x :
1, 2, 7.1935, ln17,5.

Tastenfolge	Anzeige	
1 ENTER+ ENTER+	1,0000	Die Variable wird in den Stack geladen f(x)
ENTER+	6,0000	
A 2 ENTER+ ENTER+	2,0000	
ENTER+	43,0000	Die Variable wird in den Stack geladen f(x)
A 7,1935 ENTER+ ENTER+ ENTER+	7,1935	
ENTER+ ENTER+	3.770,4359	
A 17,5 f LN ENTER+ ENTER+ ENTER+	2,8622	Die Variable wird in den Stack geladen (fx)
ENTER+ ENTER+	139,7118	
		HORNER-SCHEMA
		Wie Sie sehen, war das obenstehende Programm logisch aufgebaut und leicht zu schreiben; und es ergab die Resultate, die wir wollten. Wenn wir uns aber einer mathematischen Methode, die als Horner-Schema bekannt ist, bedienen, ist das Programm nicht nur logisch im Aufbau sondern auch einfacher und kürzer.
		Im Falle von ganzen, rationalen Funktionen der Form $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0$ werden bei der Verwendung des Horner-Schemas alle Potenzen x^n, x^{n-1}, \dots, x^1 der Variable auf x^1 reduziert. Der Ausdruck wird daher als eine Serie von arithmetischen Operationen der Variablen x und der Koeffizienten $a_n, a_{n-1}, \dots, a_1, a_0$ angegeben.
		Wir wollen beispielsweise das Horner-Schema auf das schon berechnete Polynom anwenden:
	$x^4 + 3x^3 - x^2 + 4x - 1$	
	$(x^3 + 3x^2 - x + 4) x - 1$	
	$((x^2 + 3x - 1) x + 4) x - 1$	
	$((((x + 3) x - 1) x + 4) x - 1)$	
		Jetzt können wir ein weiteres Programm wieder unter Verwendung der Stack-Lade-Methode schreiben. Diesmal benötigen wir wegen der neuen durch das Horner-Schema gegebenen Form von $f(x)$ nur sieben arithmetische Operationen an Stelle der sechs arithmetischen Operationen und drei Exponentialfunktionen im vorangegangenen Beispiel.

Stack-Register				
001	002	003	004	005
x	x	x	x	x
x	x	x	x	x
x	x	x	x	(x+3)x
x	3	(x+3)	(x+3)x	1
h LBL B	3	+	×	1
006	007	008		
x	x	x		
x	x	x		
x	x	((x+3)x-1)x		
((x+3)x-1)	((x+3)x-1)x	4		
-	×	4		
009	010	011		
x	x	x		
x	x	x		
x	x	(((x+3)x-1)x+4)x		
((x+3)x-1)x+4	(((x+3)x-1)x+4)x	1		
+	×	1		
012	013			
x	x			
x	x			
x	x			
(((x+3)x-1)x+4)x-1	(((x+3)x-1)x+4)x-1			
-	h RTN			

Die obenstehende Befehlsfolge benötigt nur 13 Programmzeilen. Damit werden gegenüber der ersten Methode 6 Programmzeilen bei der Berechnung des identischen Ausdrucks eingespart. Tasten Sie das Programm ein und führen Sie das Programm für die schon vorher berechneten Beispiele aus.

Schieben Sie den PRGM RUN -Schalter in Stellung PRGM. Wenn Sie seit der letzten Berechnung von $f(x)$ keine weiteren Anweisungen ausgeführt haben, werden Sie 000– in der Anzeige sehen. Ist dies nicht der Fall, drücken Sie GTO **□ 000**. (GTO wird später ausführlich behandelt.)

Tastenfolge	Anzeige
h LBL B	001-25,13,12
3	002- 3
+	003- 51
×	004- 61
1	005- 1
-	006- 41
×	007- 61
4	008- 4
+	009- 51
×	010- 61
1	011- 1
-	012- 41
h RTN	013- 25 12

Schieben Sie den PRGM RUN -Schalter in Stellung RUN und berechnen Sie den Ausdruck für die schon einmal verwendeten Werte von x .

Tastenfolge	Anzeige
1 ENTER↑ ENTER↑	1,0000
ENTER↑	
B	6,0000
2 ENTER↑ ENTER↑	2,0000
ENTER↑	
B	43,0000
7,1935 ENTER↑ ENTER↑	7,1935
ENTER↑	
B	3.770,4359
17,5 f LN ENTER↑ ENTER↑	2,8622
ENTER↑	
B	139,7118

Die Variable wird in den Stack geladen $f(x)$

Haben Sie irgendeinen Unterschied festgestellt? Die Rechenzeit für jede Variable war um etwa 2 Sekunden kürzer, als es vorher bei der Berechnung von $f(x)$ mit Programm A der Fall war.

WEITERE ANWENDUNGEN

Sie haben gesehen, wie einfach und praktisch die Stack-Lade-Methode ist, wenn ein Ausdruck zu berechnen ist, in dem eine Variable mehrfach vorkommt. Und wenn es möglich ist, das Horner-Schema für ein Problem anzuwenden, sparen wir noch mehr Platz und Zeit. Später werden wir die Lösung von Gleichungen und numerischer Integration eingehend behandeln. Dann werden Sie erfahren, wie die automatische Stack-Lade-Methode, die in den **SOLVE** - und **OPTN**-Operationen enthalten ist, die Leistungsfähigkeit und Anwendbarkeit Ihres HP-34C noch weiter steigert.

ÜBUNGSAUFGABEN

Verwenden Sie die Stack-Lade-Methode und das Horner-Schema in einem Programm, um die folgenden Polynome für Werte von x von 1,5, 3,73 und -4,25 zu berechnen.

$$1. \ 2x^5 - x^4 + 2x^2 + x + 1$$

$$2. \ 0,97 \sin^3 x + 0,04 \sin^2 x - 1,73 \sin x - 1$$

(Ergebnisse: 1. 17,1250, 1.283,0102, -3.066,5371.
2. -1,0452, -1,1121, -0,8720.)

ABSCHNITT 4: PROGRAMMKORREKTUR

Es ist oft wünschenswert, bereits im Rechner gespeicherte Programme abändern oder ergänzen zu können. Auf dem Tastenfeld Ihres HP-34C finden Sie eine Reihe von Funktionen, die das Überarbeiten Ihrer Programme einfach gestalten. Sie ermöglichen das Austauschen einzelner Programmweisen, ohne daß die übrigen Programme erneut in den Speicher geladen werden müssen.

Vielleicht erinnern Sie sich noch, daß zu Beginn der Ausführungen über die Programmierung 9 Tastenfunktionen genannt wurden, die nicht für eine spätere Ausführung im Rechner gespeichert werden können. Davon gehören 7 Tasten zu den Korrektur-Operationen, die das nachträgliche Abändern und Ergänzen der Programme ermöglichen.

NICHT SPEICHERBARE OPERATIONEN

CLEAR PRGM ist eine der Tastenfeldoperationen, die nicht in den Programmspeicher geschrieben werden können. Wenn Sie den **PRGM** **RUN** - Schalter in Stellung **PRGM** schieben und dann **f CLEAR PRGM** drücken, wird der ganze Programmspeicher gelöscht und der Rechner an den Beginn des Speichers (Zeile 000) zurückgesetzt.

Beachten Sie, daß mit **f CLEAR PRGM** die Winkel-Modi Bogenmaß oder Neugrad nicht in Grad zurückgeschaltet werden.

SST (Einzelschritt vor) ist eine weitere nichtspeicherbare Operation.

Wenn Sie **SST** im **RUN**-Modus drücken, zeigt der Rechner die nachfolgende Programmspeicherzeile so lange an, wie Sie die Taste gedrückt halten; nach Loslassen der Taste wird die in dieser Zeile gespeicherte Anweisung ausgeführt. Sie können also **SST** in Abhängigkeit von der Stellung des **PRGM/RUN**-Schalters sowohl zur schrittweisen Ausführung eines Programms, als auch zur reinen Anzeige der aufeinanderfolgenden Programmzeilen verwenden.

h BST (Einzelschritt zurück) ist eine nichtspeicherbare Tastenfunktion, die jeweils die vorangegangene Programmzeile anzeigt. Wenn Sie **h BST** im **PRGM**-Modus drücken, rückt der Rechner um eine Programmspeicherposition zurück und zeigt die in dieser Zeile gespeicherte Anweisung an. Wenn Sie dagegen den **PRGM** **RUN** -Schalter in Stellung **RUN** schieben und

dann die Taste **H BST** gedrückt halten, wird die Programmzeilen-Nummer und der Tasten-Code der vorhergehenden Programmzeile angezeigt. Nach Loslassen der Taste **BST** erscheint wieder der ursprüngliche Inhalt des X-Registers in der Anzeige. Die gespeicherte Programmanweisung wird dabei nicht ausgeführt.

CLEAR PREFIX ist eine nicht speicherbare Operation die im PRGM-Modus wie im RUN-Modus die Wirkung einer der folgenden Tasten aufhebt:

f f	FIX	H LBL	STO (– + , × , ÷)	GTO
g f	SCI	STO	STO (– . + , × , ÷)	GTO T f y
h f	ENG	STO •	RCL	GTO • f SOLVE
STO	f	RCL •	GTO • n h F?	
RCL	f	GTO • nn h SF		
		GSB h CF		
		GSB f		

Wenn Sie **f** gedrückt haben, können Sie diesen Tastendruck rückgängig machen, indem Sie einfach **CLEAR PREFIX** drücken. Haben Sie dagegen **g** oder **H** gedrückt, und wollen diese Präfixtaste rückgängig machen, drücken Sie einfach **f CLEAR PREFIX**. Alle anderen aufgeführten Tasten wurden mit **f CLEAR PREFIX** gelöscht.

GTO • nnn ist eine weitere Tastenfeldoperation, die nicht als Anweisung im Rechner gespeichert werden kann. (Dagegen kann **GTO**, gefolgt von einer beliebigen Zifferntaste (0 bis 9) jederzeit als Programmschritt im Rechner gespeichert werden. Die **GTO**-Anweisung wird an späterer Stelle noch ausführlicher besprochen werden.) Wenn Sie **GTO •** und anschließend eine dreistellige Schrittnummer eintasten, rückt der Rechner zu der angegebenen Speicherstelle vor oder zurück, so daß der nächste Programmschritt in dieser Zeile gespeichert oder das Programm von dort aus gestartet werden kann. Dabei werden keine Programmschritte ausgeführt. Es ist in diesem Fall außerdem nicht von Bedeutung, ob der PRGM-RUN-Schalter in Stellung PRGM oder RUN steht. Haben Sie die angegebene Tastenfolge im RUN-Modus gedrückt, können Sie sich von der richtigen Position im Programmspeicher dadurch überzeugen, daß Sie den PRGM **RUN**-Schalter kurzzeitig in Stellung PRGM schieben. Die Anweisung **GTO • nnn** ist besonders im PRGM-Modus von Nutzen, da Sie damit jede beliebige Speicherstelle erreichen und so beliebige Programmschritte entfernen oder abändern können.

Achtung: Der Versuch, eine **GTO •** Anweisung auszuführen, die auf eine Programmspeicherzeile zielt, die nicht belegt ist oder vom Rechner noch nicht in Programmspeicher umgewandelt wurde, gilt als illegale Operation und ergibt die Fehlermeldung ERROR 4.

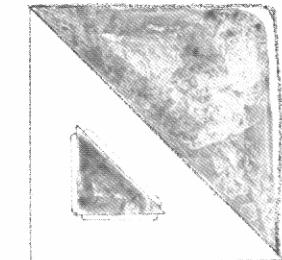
Mit **H DEL** (Programmschritt löschen) können Sie einen beliebigen Programmschritt aus dem Programmspeicher des Rechners löschen. Auch diese Operation kann nicht selbst als Bestandteil eines Programms im Rechner gespeichert werden. Wenn Sie den PRGM **RUN**-Schalter in Stellung PRGM schieben und **H DEL** drücken, wird die in der augenblicklichen Position gespeicherte Anweisung entfernt, und alle nachfolgenden Programmschritte rücken um eine Speicherzelle nach oben. Wird **H DEL** im RUN-Modus gedrückt, hat es keine Wirkung.

Mit der **MEM** (Speicher)-Funktion wird die aktuelle Speicherzuteilung angezeigt. Der Rechner-Modus spielt dabei keine Rolle. In Abschnitt 3 "Einfache Programmierung" können Sie auf Seite 59 die Beschreibung der **MEM** Funktion nochmal durchlesen.

Wir wollen jetzt ein Programm über das Tastenfeld in den Rechner eingeben und diese Korrekturfunktionen zur Überprüfung und Abänderung verwenden.

PYTHAGORAS-PROGRAMM

Das folgende Programm berechnet die Länge der Hypotenuse eines rechtwinkligen Dreiecks, wenn die beiden anderen Seitenlängen gegeben sind. Dabei wird die Formel $c = \sqrt{a^2 + b^2}$ verwendet (Satz des Pythagoras).



Nachfolgend sind die einzelnen Programmschritte zur Lösung dieser Aufgabe angegeben. (Im wesentlichen die gleichen Tasten, die Sie zur Berechnung von c auch über das Tastenfeld gedrückt hätten.) Dabei wird angenommen, daß die Zahlenwerte für die Seiten a und b in den entsprechenden Registern X und Y des Stacks stehen.

Zum Eintasten des Programms schieben Sie als erstes den PRGM **RUN**-Schalter in Stellung PRGM, anschließend drücken Sie **f CLEAR PRGM**, damit der Programmspeicher gelöscht wird und der Rechner zur Speicherzeile 000 vorrückt.

Geben Sie jetzt folgende Tastenfolge ein:

h LBL A	001-25,13,11
g X²	002- 15 3
x:y	003- 21 a
g X²	004- 15 3 a + b
f √X	006- 14 3 Programmende; der Rechner kehrt zur
h RTN	007- 25 12 Zeile 000 zurück und hält dort an.

Schalten Sie den Rechner zurück in den RUN-Modus und führen Sie dann das Programm aus. Berechnen Sie z. B. die Länge der Hypotenuse eines rechtwinkligen Dreiecks, dessen eine Seite, a, 22 Meter und dessen andere Seite, b, 9 Meter lang ist. (Beachten Sie, daß in diesem Fall die Daten in beliebiger Reihenfolge eingegeben werden können).

Tastenfolge	Anzeige
22 ENTER+	22,00
9	9,
A 23,77	Länge der Seite c in Metern

Zur Berechnung der Hypotenuse eines rechtwinkligen Dreiecks mit den Seiten a = 73 Meilen und b = 99 Meilen:

Tastenfolge	Anzeige
73 ENTER+	73,00
99	99,
A 123,000	Länge der Seite c in Meilen

Wir wollen uns einmal ansehen, wie die nichtspeicherbaren Korrektur-Operationen des HP-34C zur Überprüfung und Abänderung dieses Programms verwendet werden können.

SCHRITTWEISE AUSFÜHRUNG EINES PROGRAMMS

Sie können gespeicherte Programme im RUN-Modus durch wiederholtes Drücken der Taste **SST** (Einzelschritt vor) Zeile für Zeile ausführen.

Verwenden Sie das Pythagoras-Programm zur Berechnung der Hypotenuse eines Dreiecks mit den Seiten a = 73 Meilen und b = 99 Meilen; führen Sie das Programm jetzt einmal Schritt für Schritt aus:

Schieben Sie als erstes den **PRGM** - **RUN** - Schalter in Stellung RUN.



Tastenfolge	Anzeige	
73 ENTER+	73,00	
99	99,	Jetzt sind die Daten eingegeben und das Programm kann gestartet werden.
		Wenn Sie jetzt SST gedrückt halten, wird der Tasten-Code der nächsten Anweisung angezeigt. Nach Loslassen der Taste wird dieser Programmbefehl ausgeführt. (Erinnern Sie sich, daß nach der letzten Ausführung des Programms der Rechner nach der h RTN -Anweisung in Zeile 007 wieder an den Speicheranfang zurückgesetzt wird.)
Tastenfolge	Anzeige	
h SST	001-25,13,11	Solange Sie SST gedrückt halten, wird der Tasten-Code für h LBL A angezeigt.
	99,00	Nach Loslassen von SST wird h LBL A ausgeführt.
		Beachten Sie, daß Sie in diesem Falle nicht A gedrückt haben. Wenn Sie das Programm schrittweise ausführen, wird mit h SST das Programm mit der augenblicklichen Zeile des Programmspeichers fortgesetzt. Hier war es Zeile 001.
		Fahren Sie mit der schrittweisen Programmausführung fort. Wenn Sie SST erneut drücken und festhalten, wird der Tasten-Code der nächsten Anweisung angezeigt. Nach Loslassen von SST wird auch dieser Programmbefehl ausgeführt.
Tastenfolge	Anzeige	
h SST	002- 15 3	Tasten-Code für g X² wurde ausgeführt.
	9801,0000	
		Wenn Sie im RUN-Modus SST ein drittes Mal drücken, erscheint der Inhalt der Speicherzeile 003 in der Anzeige. Nach Loslassen von SST wird die in dieser Zeile gespeicherte Anweisung, x:y , ausgeführt und der Rechner hält an.
Tastenfolge	Anzeige	
h SST	003- 21	Tasten-Code für x:y wurde ausgeführt.
	73,0000	

Setzen Sie die zeilenweise Ausführung des Programms mit Hilfe von **SST** fort. Wenn Sie auf diese Weise auch die **h RTN**-Anweisung in Zeile 007 fortgeführt haben, kehrt der Rechner nach Zeile 000 zurück und zeigt das Ergebnis in gleicher Weise an, wie er das auch bei der automatischen Ausführung der gespeicherten Programmschritte getan hätte.
(Wir werden uns später mit **RTN** ausführlicher befassen).

Tastenfolge	Anzeige
h SST	004- 15 3
	5.329,0000
h SST	005- 51
	15.130,0000
h SST	006- 14 3
	123,0041
h SST	007- 25 12
	123,0041
	Endergebnis

Achtung: Der Rechner rückt mit **h SST** nicht in freien Programmspeicher vor. Wenn Sie bei der schrittweisen Ausführung die letzte belegte Programmzeile erreicht haben, springt der HP-34C lediglich zur Speicherzeile 000 zurück. Dies ist modusunabhängig, wobei im RUN-Modus der ursprüngliche Inhalt der Anzeige unverändert bleibt und im PRGM-Modus 000- und damit der Anfang des Programmspeichers angezeigt wird.

Sie haben gesehen, wie Sie sich mit Hilfe von **SST** schrittweise durch ein gespeichertes Programm tasten können. Diese Möglichkeit ist besonders beim Erstellen und Korrigieren von Programmen von großem Nutzen. Wir wollen uns jetzt ansehen, wie **SST BST GTO** 000 und **GTO** nnn im RUN-Modus zum Abändern eines gespeicherten Programms eingesetzt werden können.

ABÄNDERN EINES PROGRAMMS

Wir wollen das Pythagoras-Programm jetzt derart abändern, daß der Rechner an einigen Stellen des Programms automatisch die Stackregister-Inhalte anzeigen.

Dazu fügen wir die Anweisung **h PSE** ein, um das Programm anzuhalten, den Inhalt des X-Registers für etwa eine Sekunde anzuzeigen und dann das Programm fortzusetzen. (**PSE** wird später noch behandelt.)

Das ursprüngliche Programm sieht folgendermaßen aus:

Tastenfolge	Anzeige
h LBL A	001-25,13,11
g X²	002- 15 3
x:y	003- 21
g X²	004- 15 3
+	005- 51
f X	006- 14 3
h RTN	007- 25 12

An diesen Stellen soll jeweils eine **h PSE**-Anweisung eingefügt werden.

SCHRITTWEISE ANZEIGE OHNE AUSFÜHRUNG DES PROGRAMMS

Im PRGM-Modus können Sie sich mit Hilfe von **SST** zu der gewünschten Stelle im Programmspeicher vortasten, ohne dabei die gespeicherten Programmbefehle auszuführen. Wenn Sie in den PRGM-Modus umschalten, sehen Sie, daß der Rechner auf Grund der **h RTN**-Anweisung an den Speicheranfang (Zeile 000) zurückgesetzt wurde. Wenn Sie jetzt einmal **SST** drücken, rückt der Rechner nach Zeile 001 vor und zeigt den Inhalt dieser Speicherzeile an. Dabei werden keine gespeicherten Anweisungen ausgeführt.

Schieben Sie den **PRGM**  **RUN**-Schalter in Stellung PRGM.

Tastenfolge	Anzeige
h SST	001 25,13,11

Programmspeicherzeile 001 wird angezeigt.

Wie Sie sehen, steht der Rechner bei Zeile 001 des Programmspeichers. Wenn Sie jetzt die Tasten für eine beliebige speicherbare Operation drücken, wird diese Anweisung in die nächste Zeile des Programmspeichers, also Zeile 002, gespeichert. Dabei werden alle übrigen Programmschritte um eine Speicherzeile nach unten verschoben, so daß keine Informationen überschrieben werden! Um jetzt die **h PSE**-Anweisung einzufügen, so daß der Rechner anhält und den Inhalt des X-Registers anzeigt:

Tastenfolge	Anzeige
h PSE	002- 25 74

Wir wollen uns jetzt ansehen, was beim Drücken der Tasten im Programmspeicher geschehen ist. Wenn Sie in Speicherzeile 001 stehen und dann **h PSE** drücken, ändert sich der Inhalt des Programmspeichers wie folgt:

Vorher	Nachher
001 h LBL A	001 h LBL A
002 h PSE	002 h PSE
003 x²	003 g x²
004 x²y	004 x²y
005 +	005 g x²
006 f ✓x	006 +
007 h RTN	007 h RTN

Hier wird die Anweisung **h PSE** eingefügt

Alle nachfolgenden Programm schritte werden um eine Zeile nach unten verschoben

Rücksprung zur Speicherzeile 000

Beim Einschalten des HP-34C wird der Rechner automatisch an den Anfang des Programmspeichers gesetzt. Auch haben Sie schon erfahren, daß der Rechner auf Zeile 000 zurückgesetzt und zusätzlich der gesamte Programmspeicher gelöscht wird, wenn Sie im PRGM-Modus **f CLEAR PRGM** drücken. Sie können den Rechner aber auch an den Speicheranfang setzen und dabei die bestehenden Programme im Programmspeicher erhalten, wenn Sie **GTO** **000** im PRGM- oder RUN-Modus und **h RTN** oder **f CLEAR PRGM** im RUN-Modus drücken.

Um den Rechner mit dem Pythagoras-Programm im Speicher auf Zeile 000 zu setzen:

Tastenfolge	Anzeige
GTO 000	000-

VORRÜCKEN ZU EINER BESTIMMTEN SPEICHERSTELLE

Wenn Sie leicht einsehen werden, ist die wiederholte Verwendung von **SST** zum Vorrücken im Programmspeicher dann zeitraubend und mühsam, wenn die gewünschte Speicherstelle weit von der augenblicklichen Position im Speicher entfernt liegt.

Sie können, um dieses zu vermeiden, die Operation **GTO** **nnn** verwenden, die Sie bereits für das Rücksetzen des Rechners auf Zeile 000 benutzt haben.

Wenn Sie entweder im PRGM- oder RUN-Modus **GTO** **nnn** drücken, entspringt der Rechner augenblicklich zu der Speicherzeile, die durch die dreistellige Ziffernfolge nnn angegeben wird. Dabei werden keine Programmabfahrt ausgeführt. Im PRGM-Modus wird dabei automatisch die ange



wählte Zeilennummer und der Code der dort gespeicherten Anweisung angezeigt; befinden Sie sich dagegen im RUN-Modus, können Sie kurzzeitig in den PRGM-Modus umschalten, um diese Speicherzeile anzuzeigen. Das Anzeigen der Programmzeilen, weitere Suchvorgänge oder die Ausführung des Programms beginnen dann ab dieser Position im Programmspeicher. Eingetastete Programmzeilen werden mit der darauffolgenden Zeile beginnend in den Speicher geschrieben.

Wenn Sie beispielsweise erreichen wollen, daß der Wert der in Zeile 006 aufaddierten Quadrate angezeigt wird, müssen Sie im Anschluß an diese Programmzeile eine **h PSE**-Anweisung einfügen. Drücken Sie dazu als erstes **GTO** (gehe nach), gefolgt von der Taste für den Dezimalpunkt und den entsprechenden Zifferntasten für die dreistellige Angabe der gewünschten <Sprungadresse>. Dann können Sie **h PSE** drücken und damit den gewünschten Anzeigebefehl in die darauffolgende Speicherzeile schreiben. Denken Sie daran, daß dabei alle nachfolgenden Programmzeilen um eine Speicherzeile nach unten verschoben werden. Zum Einfügen von **h PSE** im Anschluß an die in Zeile 006 gespeicherte **+**-Anweisung muß sich der Rechner im PRGM-Modus befinden:

Tastenfolge	Anzeige
GTO 006	006- 51
h PSE	007- 25 74

Beim Einfügen der **h PSE**-Anweisung in Speicherzeile 007 wurde der Programmschritt, der zuvor in dieser Zeile gespeichert war, in die Speicherzeile 008 geschoben. In gleicher Weise wurden alle darauffolgenden Programmzeilen um eine Speicherzeile nach unten gerückt.

Beim Einfügen der **h PSE**-Anweisung im Anschluß an Zeile 007 wurden die Inhalte des Programmspeichers wie folgt verschoben:

Vorher	Nachher
001 h LBL A	001 h LBL A
002 h PSE	002 h PSE
003 g x²	003 g x²
004 x²y	004 x²y
005 g x²	005 g x²
006 +	006 +
007 f ✓x	007 h PSE
008 h RTN	008 f ✓x
	009 h RTN

Hier wurde der **h PSE**-Befehl eingefügt

Die darauffolgenden Programmzeilen wurden um eine Position nach unten verschoben

EINFÜGEN VON PROGRAMMSCHRITTEN IN LÄNGEREN PROGRAMMEN

Sobald die ursprünglichen 70 Zeilen des Programmspeichers mit Programmschritten belegt sind, wandelt der Rechner automatisch Speicherregister im verfügbaren Programmspeicher in Blöcken von jeweils 7 Zeilen um. Dies geschieht blockweise, sobald jede neue Programmspeicherzuteilung mit Programmschritten belegt ist. Wenn ein Programm genau 70 Zeilen lang ist und eine weitere Anweisung an einer beliebigen Stelle des Programms eingefügt wird, wandelt der Rechner automatisch ein Speicherregister (in diesem Fall R₉) in 7 zusätzliche Zeilen verfügbaren Programmspeichers um. Der Rechner hätte dann 77 Zeilen verfügbaren und 71 Zeilen belegten Programmspeicher (siehe „Variable Speicherplatzzuweisung“ auf Seite 55–58). Sind 77 Zeilen belegt, wird noch ein Speicherregister (R₈) in 7 Programmzeilen umgewandelt, wenn eine weitere Anweisung eingefügt wird. Dies kann fortgeführt werden, bis maximal 210 Programmzeilen belegt sind. Zu diesem Zeitpunkt nimmt der Rechner keine weiteren Programmanweisungen mehr an. Wenn Sie dennoch versuchen sollten, eine weitere Anweisung einzugeben, wenn alle 210 Zeilen belegt sind, erscheint die Fehlermeldung ERROR 4 in der Anzeige. Der Programmspeicherinhalt bleibt unverändert.

(Denken Sie daran, daß Sie mit jederzeit den augenblicklichen Programmzeilen/Speicherregister-Status feststellen können.)

SCHRITTWEISES ZURÜCKRÜCKEN IM SPEICHER

Die Taste **BST** (Einzelschritt zurück) wird im Zusammenhang mit der Korrektur von Programmen dazu verwendet, Schritt für Schritt im Programmspeicher zurückzurücken. Wenn Sie **H BST** im RUN-Modus drücken, rückt der Rechner um eine Zeile im Programmspeicher zurück und zeigt die dort gespeicherte Anweisung (und Zeilennummer) so lange an, wie Sie **BST** gedrückt halten – nach Loslassen der Taste erscheint wieder der letzte Inhalt des X-Registers in der Anzeige.

Im PRGM-Modus wird natürlich die Zeilennummer und der Tasten-Code der entsprechenden Anweisung angezeigt. Anweisungen werden weder im PRGM-Modus noch im RUN-Modus ausgeführt.

Anmerkung: Befindet sich der HP-34C am Programmspeicheranfang (Zeile 000) wenn Sie **H BST** drücken, springt der Rechner auf die letzte Zeile des belegten Programmspeichers. Diese Eigenschaft ist dann sehr nützlich, wenn Sie schnell die Länge eines bestehenden Programms bestimmen

wollen oder ein neues Programm oder Unterprogramm unmittelbar auf ein sich im Rechner befindliches Programm oder Unterprogramm folgen lassen wollen.

Sie wollten noch eine weitere **H PSE**-Anweisung in das Pythagoras-Programm einfügen. Dieser Schritt ist im Anschluß an die **X_{XY}**-Anweisung, die augenblicklich in Zeile 004 des Programmspeichers steht, einzutasten. Der Rechner steht im Moment noch vom Einfügen des letzten **H PSE** bei Zeile 007 des Programms. Sie können jetzt die Taste **BST** dazu verwenden, schrittweise zur Zeile 004 zurückzurücken. Anschließend tasten Sie die **H PSE**-Anweisung in Zeile 005 ein. Zu Beginn: Vergewissern Sie sich, daß der PRGM -Schalter in Stellung PRGM steht.

Tastenfolge	Anzeige
H BST	008- 25 74
	007- 14 3

Der Rechner steht noch bei Zeile 008 des Programms.
Wenn Sie einmal **BST** drücken, wird der Inhalt der vorhergehenden Speicherzeile angezeigt.

Wenn Sie **H BST** drücken, rückt der Rechner um eine Programmspeicherzeile zurück. Dabei werden keine gespeicherten Programmbefehle ausgeführt. Fahren Sie mit dem Drücken von **BST** fort, bis Sie die Speicherzeile 004 erreicht haben:

Tastenfolge	Anzeige
H BST	005- 15 3
H BST	004- 21

Da Sie den **H PSE**-Befehl hinter die in Zeile 004 stehende **X_{XY}**-Anweisung speichern wollen, müssen Sie zuerst zur Zeile 004 zurückrücken. Ein eingetaster Programmbefehl wird grundsätzlich in die Speicherzeile geschrieben, die auf die angezeigte Programmzeile folgt. Wenn Sie also jetzt **H PSE** drücken, wird diese Anweisung in Speicherzeile 005 geschrieben und der nachfolgende Teil des Programms um eine Speicherzeile nach unten verschoben.

Tastenfolge	Anzeige
H PSE	005- 25 74

KONTROLLE DER PROGRAMMÄNDERUNG

Sie haben Ihr Pythagoras-Programm jetzt so abgeändert, daß die Stack-inhalte bei der Ausführung des Programms an mehreren Stellen automatisch angezeigt werden. Das Programm sieht jetzt wie folgt aus:

Tastenfolge	Anzeige
001 H LBL A	001–25,13,11
002 H PSE	002– 25 74
003 G X²	003– 15 3
004 X^Y	004– 21
005 H PSE	005– 25 74
006 G X²	006– 15 3
007 +	007– 51
008 H PSE	008– 25 74
009 F ✓X	009– 14 3
010 H RTN	010– 25 12

Wenn Sie kontrollieren wollen, ob diese Programmzeilen auch im Speicher Ihres HP-34C stehen, können Sie zur Speicherzeile 000 vorrücken und das Programm mit **SST** im PRGM-Modus Zeile für Zeile in die Anzeige bringen.

AUSFÜHREN DES ABGEÄNDERTEN PROGRAMMS

Um das abgeänderte Pythagoras-Programm auszuführen, müssen Sie Zahlenwerte für die Seiten a und b eingeben und anschließend **A** drücken.

Der HP-34C zeigt daraufhin den Inhalt des **X**-Registers an, quadriert die Seite b, tauscht die Inhalte von **X** und **Y** aus und zeigt dann erneut den Inhalt des **X**-Registers an (jetzt Seite a).

Dann quadriert der Rechner die Seite a, addiert sie zu b^2 und zeigt den **X**-Register-Inhalt ($a^2 + b^2$) ein drittes Mal an. Jetzt wird die Hypotenuse errechnet, worauf der Rechner zu Zeile 000 springt und anhält.

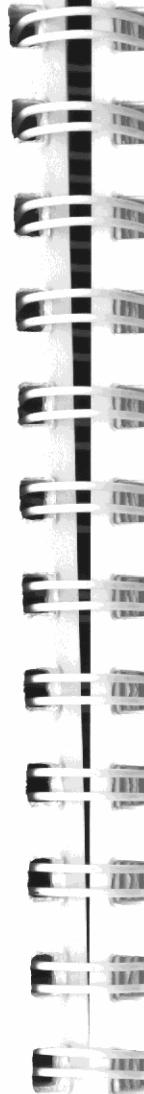
Verwenden Sie das Programm jetzt zur Berechnung der Hypotenuse eines rechtwinkligen Dreiecks mit den Seiten a = 22 Meter und b = 9 Meter:

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung RUN.

Tastenfolge	Anzeige
22 ENTER	22,0000
9 A	23,7697

Nach dreimaliger automatischer Anzeige des **X**-Registers zeigt der Rechner das Ergebnis in Meter an.

Wiederholen Sie die Rechnung jetzt für ein Dreieck mit den Seiten a = 73 Meilen und b = 99 Meilen (Ergebnis 123 Meilen).



Im Zusammenhang mit der Änderung und Korrektur gespeicherter Programme wird es oftmals nötig sein, einzelne Programmzeilen zu entfernen. Rücken Sie den Rechner dazu an die entsprechende Programmspeicherstelle vor und drücken Sie dann im PRGM-Modus die Tasten **H** **DEL** (Einzel-schritt löschen). Diese Anweisung gehört ebenfalls zu den Operationen, die nicht als Bestandteil eines Programms im Rechner gespeichert werden können. (Wenn Sie **H** **DEL** im RUN-Modus drücken, wird dadurch lediglich die Wirkung einer zuvor gedrückten Präfixtaste aufgehoben.) Nachdem eine Programmzeile mit **H** **DEL** aus einer Speicherzeile entfernt wurde, rücken alle nachfolgenden Programmzeilen um eine Zeile auf.

Gleichzeitig rückt der Rechner um eine Speicherzeile zurück und zeigt den entsprechenden Inhalt an.

Wenn Sie beispielsweise das augenblicklich im Rechner gespeicherte Pythagoras-Programm in der Weise erneut ändern wollen, daß das **X**-Register nur noch einmal, und zwar mit den Zwischenresultat $a^2 + b^2$ angezeigt wird, müssen Sie die beiden übrigen **H** **PSE**-Anweisungen in den Zeilen 002 und 005 entfernen. Zum Löschen dieser Programmzeile ist als erstes der Rechner an die entsprechende Stelle im Programmspeicher zu rücken. Dazu können Sie **SST**, **H** **BST** und **GTO** **nnn** verwenden.

Anschließend drücken Sie **H** **DEL** und löschen so die **H** **PSE**-Anweisung in Zeile 002.

Schieben Sie als erstes den **PRGM** **RUN**-Schalter in Stellung PRGM.

Tastenfolge	Anzeige
GTO 002	002– 25 74 Speicherzeile 002 wird angezeigt
DEL	001–25,13,11 Anweisung in Zeile 002 wird gelöscht und der Rechner rückt zur Zeile 001 zurück

Wenn Sie sich davon überzeugen wollen, daß der **H** **PSE**-Befehl gelöscht wurde, müssen Sie eine Zeile im Programmspeicher vorrücken.

Tastenfolge	Anzeige
H SST	002– 15 3 Die zuvor in Zeile 003 gespeicherte Anweisung steht jetzt in Zeile 002; alle nachfolgenden Programmzeilen rücken beim Drücken von H DEL ein Speicherzeile auf

Beim Drücken von **H** **DEL** hat sich der Inhalt des Programmspeichers wie folgt geändert:

Vorher

001	h	LBL	A
002	h	PSE	
003	g	X ²	
004	X ²	y	
005	h	PSE	
006	g	X ²	
007	+		
008	h	PSE	
009	f	✓X	
010	h	RTN	

Nachher

001	h	LBL	A
002	g	X ²	
003	X ²	y	
004	h	PSE	
005	g	X ²	
006	+		
007	h	PSE	
008	f	✓X	
009	h	RTN	

Um die **h PSE**-Anweisung aus der Programmspeicherzeile 004 zu entfernen, können Sie mit **SST** zu dieser Speicherposition vorrücken und dann die gespeicherte Anweisung mit **h DEL** löschen.

Tastenfolge

h SST
h SST
h DEL

Anzeige

003- 21
004- 25 74
003- 21

Nach Entfernen des **h PSE**-Befehls aus Zeile 004 zeigt der Rechner die Speicherzeile 003 an. Alle nachfolgenden Programmzeilen rücken um einen Speicherplatz vor

Wenn Sie das Programm wie gezeigt abgeändert haben, wird der Rechner den Inhalt des X-Registers jetzt nur noch einmal anzeigen, wenn $a^2 + b^2$ berechnet wurde. Anschließend wird der Wert der Hypotenuse berechnet und das Programm hält an.

Schieben Sie den **PRGM** **RUN** -Schalter in Stellung RUN und verwenden Sie das Programm zur Berechnung eines rechtwinkligen Dreiecks mit:

Seite a = 17 Meter, b = 34 Meter. Ergebnis für Seite c = 38,0132 Meter.

Summe der Quadrate = 1.445,0000.

Seite a = 550 Zoll, b = 740 Zoll. Ergebnis für Seite c = 922,0087 Zoll.

Summe der Quadrate = 850.100,0000.

Zum Auswechseln einer beliebigen Programmzeile genügt es, den Rechner entsprechend im Speicher zu positionieren und **h DEL** zu drücken. Im Anschluß daran können Sie die Tastenfolge für die abgeänderte Programm-anweisung eingeben.

LÖSCHEN VON PROGRAMMZEILEN IN LÄNGEREN PROGRAMMEN

Werden Programmanweisungen aus einem Programm gelöscht, das mehr als 70 Zeilen enthält, wird der Vorgang, der automatisch Datenregister in Programmspeicher umwandelt, umgekehrt. Wenn beispielsweise eine beliebige Anweisung aus einem Programm mit einer Länge von 78 Zeilen gelöscht wird, wandelt der Rechner die Zeilen 78–85 automatisch in das Speicherregister R₈ um. (Siehe »Automatische Speicherplatzzuteilung« Seite 55–57.)

Die verschiedenen Korrekturmöglichkeiten, über die Ihr HP-34C verfügt, machen es leicht, jede beliebige Stelle eines Programms zu erreichen, um sie dann abzuändern oder zu berichtigen. Wenn ein Programm bei der Ausführung wegen eines Speicherüberlaufs oder einer unerlaubten Operation anhält, können Sie in Stellung PRGM umschalten und sich die Anweisung (den Code des gespeicherten Programmschrittes) ansehen, der den Fehler verursacht hat. Sie können den Rechner auch mit **GTO** **nnn** zu einem offensichtlich fehlerhaften Programmteil vorrücken und dann die nachfolgenden Programmzeilen mit **SST** im RUN-Modus Zeile für Zeile für die Wirkung jeder einzelnen Operation verfolgen und den Fehler schnell einkreisen.

Übungsaufgaben

1. Vielleicht haben Sie bereits herausgefunden, daß der HP-34C über eine einzelne Tastenfeld-Operation (die **g** **→P**-Taste) verfügt, mit der Sie ohne weitere Rechenschritte die Hypotenuse (Seite c) eines rechtwinkligen Dreiecks berechnen können. Dazu sind zuvor die Werte für die Seiten a und b in das X- und Y-Register einzugeben. Ersetzen Sie jetzt im Pythagoras-Programm **X²** **X²** **X²** **+** **PSE** und **✓X** durch die einzige Anweisung **g →P**:

a) Verwenden Sie **GTO** **nnn** und **SST** zur Kontrolle, ob die Programmzeilen in Ihrem HP-34C der nachfolgenden Speicherliste entsprechen:

Tastenfolge	Anzeige
h LBL A	001-25,13,11
g X²	002- 15 3
x:y	003- 21
g X²	004- 15 3
+	005- 51
f √X	006- 25 74
h PSE	007- 14 3
h RTN	008- 25 12

Diese Programmzeilen sind durch **g P** zu ersetzen

- b) Rücken Sie unter Verwendung von **GTO** **nnn** zur Speicherzeile 007 vor. Dort steht die letzte Anweisung, die entfernt werden soll.
- c) Verwenden Sie **h DEL** im PRGM-Modus zum Löschen der Programmzeilen 007, 006, 005, 004, 003 und 002.
- d) Tasten Sie die **g P**-Anweisung in Speicherzeile 002.
- e) Überzeugen Sie sich davon, daß das abgeänderte Programm wie folgt aussieht:

Tastenfolge	Anzeige
h LBL A	001-25,13,11
g P	002- 15 4
h RTN	003- 25 12

- f) Schalten Sie zurück in den RUN-Modus und verwenden Sie das Programm zur Berechnung eines rechtwinkligen Dreiecks mit den Seiten $a = 73$ Fuß und $b = 112$ Fuß (Ergebnis: $c = 133,6899$).
- 2. Der Abteilungsleiter eines Geldinstitutes verwendet das nachfolgende Programm zur Berechnung des verzinsten Guthabens bei Sparkonten; dabei ist die Formel $FV = PV (1 + i)^n$ zu lösen, wobei FV den zukünftigen oder Endwert des Kapitals, PV den gegenwärtigen oder Anfangswert des Kapitals, i den dezimalen Wert des Periodenzinssatzes und n die Anzahl der Zinsperioden bezeichnet. Wenn PV zuvor in das Y-Register eingegeben wird, n entsprechend in das X-Register und der Jahreszinssatz 7,5 % beträgt, sieht das Programm wie folgt aus:

f CLEAR PRGM
001 h LBL B
002 f FIX 2
003 1
004 *
005 0
006 7
007 5
008 x:y
009 h yx
010 x
011 h RTN

- a) Tasten Sie die Schrittfolge in den Programmspeicher des Rechners ein.
- b) Verwenden Sie das Programm zur Berechnung des Endwertes, auf den 1000 DM innerhalb von 5 Jahren anwachsen.
(Ergebnis: 1435,63 DM)
Über welches Guthaben verfügen Sie nach 4 Jahren, wenn die anfängliche Einlage 2300 DM beträgt?
(Ergebnis: 3071,58 DM)
- c) Ändern Sie das Programm zur Berücksichtigung eines Jahreszinssatzes von 8 % (anstatt 7,5 %) ab.
- d) Verwenden Sie das Programm zur Berechnung der Endbeträge, auf die bei 8 % p.a. 500 DM in 4 Jahren und 2000 DM in 10 Jahren anwachsen sind (Ergebnisse: 680,24 DM; 4317,85 DM).
- 3. Das folgende Programm berechnet die Zeit, die ein aus der Höhe h abgeworfener Gegenstand braucht, bis er die Erdoberfläche erreicht. (Der Einfluß des Luftwiderstandes wird dabei außer Betracht gelassen.) Wenn Sie als Vorbereitungsschritt die Höhe h (in Meter) in das angezeigte X-Register eintasten und dann drücken, wird die Fallzeit

$$t = \sqrt{\frac{2h}{9,8 \text{ Meter/Sek.}^2}}$$
berechnet und angezeigt.

- a) Löschen Sie sämtliche Programme des Rechners, schalten Sie das Anzeigeformat auf **FIX** 4 und geben Sie die nachfolgenden Programmanweisungen ein.

Tastenfolge Anzeige

CLEAR PRGM	000-
h LBL B	001-25,13,12
2	002- 2
x	003- 61
9	004- 9
.	005- 73
8	006- 8
+	007- 71
f X	008- 14 3
h RTN	009- 25 12

- b) Berechnen Sie mit Hilfe dieses Programms die Fallzeit für einen Stein, der vom 300,51 Meter hohen Eiffelturm geworfen wird. Führen Sie die gleiche Rechnung für einen Gegenstand durch, der aus einem in 1000 Meter Höhe fliegenden Raumschiff abgeworfen wird.

(Ergebnisse: 7,8313 Sek. und 14,2857 Sek.)

- c) Ändern Sie das vorstehende Programm jetzt so ab, daß die Höhe in Fuß eingegeben werden kann. Es gilt:

$$t = \sqrt{\frac{2h}{32,1740 \text{ Fuß/Sek.}^2}}$$

- d) Ein Wetterballon platzt bereits wenige Sekunden nach seinem Aufstieg in eine Höhe von 550 Fuß. Berechnen Sie mit Hilfe des abgeänderten Programms die Fallzeit, nach der die angehängte Nutzlast den Erdboden erreicht. Wie lange fällt ein Stein, der vom 1350 Fuß hohen Gebäude des Welthandelszentrums in New York City geworfen wird?

(Ergebnisse: 5,8471 Sek. und 9,1607 Sek.)

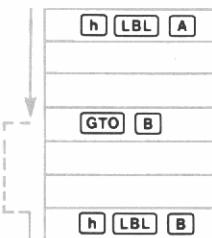
ABSCHNITT 5: PROGRAMMVERZWEIGUNGEN, ENTSCHEIDUNGEN UND FLAGS

Sie haben bereits erfahren, wie die nichtspeicherbare Operation **GTO** vom Tastenfeld aus verwendet werden kann, um zu einer beliebigen Programmzeile im belegten Programmspeicher zu verzweigen.

Sie können die **GTO**-Anweisung (Sprungbefehl) auch innerhalb eines Programms verwenden. Der **GTO**-Befehl kann aber als Bestandteil eines Programms nur dann im Rechner gespeichert werden, wenn Sie im Anschluß an **GTO** als <Sprungadresse> eine der Marken (**A** oder **B** oder 0 bis 9) eintasten. (Sie können im Anschluß an **GTO** auch drücken – dieser Sonderfall wird an späterer Stelle besprochen.)

Wenn der Rechner zum Beispiel während der Ausführung eines Programms auf die Anweisung **GTO B** trifft, wird die Programmausführung angehalten und der Speicher nach dieser Marke abgesucht. Wenn der Rechner bei diesem Suchvorgang die erste Marke **h LBL B** gefunden hat, setzt er die sequentielle Ausführung des Programms ab dieser Stelle fort.

Auf diese Weise können Sie die Programmausführung mit **GTO**, gefolgt von der Bezeichnung einer der Marken, zu einer beliebigen Stelle verzweigen.



Die Programmausführung verzweigt zum nächsten
h LBL B

Da diese Programmverzweigung auf alle Fälle stattfindet, bezeichnet man sie auch als unbedingter Sprung. Nach Erreichen der **GTO**-Anweisung springt der Rechner zur angegebenen Adresse (Marke) und setzt die Programmausführung ab dieser Stelle fort.

(Sie werden an späterer Stelle erfahren, wie Sie **GTO** auch in Verbindung mit einem Vergleichsbefehl zur Programmierung bedingter Programmverzweigungen verwenden können, die nur dann ausgeführt werden, wenn bestimmte Voraussetzungen erfüllt sind.)

Die **GTO**-Anweisung wird häufig zur Programmierung sogenannter «Programmschleifen» verwendet. Das folgende Programm verwendet eine solche Programmschleife zur Berechnung der Quadratwurzeln aufeinanderfolgender ganzer Zahlen, wobei es bei 1 anfängt. Damit fährt das Programm so lange fort, bis Sie vom Tastenfeld aus **RS** drücken (oder ein Rechner-Überlauf) eintritt.

Zum Eintasten des Programms, schieben Sie als erstes den **PRGM** RUN - Schalter in Stellung PRGM.

Drücken Sie **f CLEAR PRGM**; damit wird der Programmspeicher gelöscht und der Rechner an den Speicheranfang gesetzt (Zeile 000).

Tastenfolge

	Anzeige
h LBL A	001-25,13,11
0	002- 0
STO 1	003- 23 1
h LBL 0	004-25,13, 0
1	005- 1 Addiert 1 zum Inhalt von R ₁
STO + 1	006-23,51, 1 Ruft Inhalt von R ₁ zurück
RCL 1	007- 24 1 Anzeigepause
h PSE	008- 25 74 Anzeige der Quadratwurzel der
f ✓	009- 14 3 augenblicklichen Zahl
h PSE	010- 25 74 Verzweigung nach h LBL 0
GTO 0	011- 22 0
h RTN	012- 25 12

Schalten Sie zur Ausführung des Programms in Stellung RUN zurück und drücken Sie die Taste **A**. Der Rechner beginnt dann mit der Anzeige aufeinanderfolgender ganzer Zahlen und den zugehörigen Quadratwurzeln. Das Programm hält an, wenn Sie vom Tastenfeld aus **RS** drücken oder ein Rechner-Überlauf eintritt.

Wie läuft das Programm ab? Wenn Sie **A** drücken, sucht der Rechner den Programmspeicher nach der ersten **h LBL A**-Anweisung ab, die den Anfang des Programms markiert. Dann beginnt er die automatische Ausführung aufeinanderfolgender Programmzeilen, bis er die **GTO 0**-Anweisung in Zeile 011 erreicht.

Bei diesem Sprungbefehl angelangt, beginnt der Rechner erneut zu suchen, diesmal nach **LBL 0**. Dabei geht der Rechner die Speicherpositionen zyklisch durch und findet schließlich in Zeile 004 die erste Anweisung **h LBL 0**. Ab dieser Stelle setzt er dann die sequentielle Ausführung der Programm-Anweisungen fort. (Beachten Sie, daß im Anschluß an **GTO** die Sprungadresse in Form einer Marke und nicht als Zeilennummer anzugeben ist.)



Da das Programm jedesmal nach **LBL 0** in Zeile 004 verzweigt, wenn der Rechner die **GTO 0**-Anweisung in Zeile 011 ausführt, bewegt sich das Programm laufend innerhalb dieser «Schleife». Dabei erhöht der Rechner ständig den Inhalt des Speicherregisters R₁ und druckt dann jeweils die neue Zahl und ihre Quadratwurzel aus.

Viele der zahlreichen Möglichkeiten Ihres HP-34C werden durch die Verwendung solcher Programmschleifen erst richtig ausgenutzt. Auf diese Weise wird es möglich, Daten laufend auf den neuesten Stand zu bringen und Rechnungen automatisch, schnell und, wenn gewünscht, auch in endlosen Wiederholungen auszuführen.

Die Verwendung unbedingter Sprünge ist keineswegs auf Programmschleifen beschränkt; Sie können damit innerhalb eines Programms ebenso einmalige Verzweigungen zu einer beliebigen Marke programmieren. Der Vorgang ist stets der gleiche: Wenn der Rechner bei der Programmausführung eine **GTO**-Anweisung erreicht, sucht er den Programmspeicher bis zum Auftreten der entsprechenden Marke schrittweise nach unten ab und fährt dann mit der sequentiellen Ausführung der weiteren Programmschritte fort.

Im RUN-Modus können Sie die **GTO**-Anweisung auch dazu verwenden, eine Marke im Programm zu suchen, ohne das Programm dabei zu starten. Wenn Sie **GTO(A, B oder n)** über die Tastatur ausführen, verzweigt der Rechner zu der entsprechenden Marke und hält an. Diese Einrichtung ist besonders praktisch, wenn Sie die mit einer bestimmten Marke beginnenden Programmschritte betrachten oder ändern wollen, anstatt sie als Teil eines Programms auszuführen.

Übungsaufgaben

- Das folgende Programm berechnet, so oft Sie es starten, das Quadrat der Zahl 1 und zeigt diesen Wert an. Schieben Sie den **PRGM** RUN - Schalter in Stellung PRGM, tasten Sie das Programm ein und lassen Sie es anschließend im RUN-Modus einige Male ablaufen (Das Ergebnis wird jedesmal 1.0000 sein.)

Ändern Sie dann das Programm ab, indem Sie hinter **STO 1** in Zeile 003 die Tastenfolge **h LBL 1** einfügen und die **h RTN**-Instruktion in Zeile 10 durch eine **GTO**-Anweisung ersetzen. Auf diese Weise haben Sie eine Programmschleife erzeugt. Der Rechner wird jetzt jeweils eine neue Zahl und ihr Quadrat anzeigen, die Zahl dann um eins erhöhen, anzeigen, das entsprechende Quadrat berechnen und anzeigen usw. Geben Sie als erstes das ursprüngliche (noch nicht abgeänderte) Programm ein, indem Sie den **PRGM** RUN - Schalter in Stellung PRGM schieben, den Programmspeicher löschen und die folgenden Tasten drücken:

Tastenfolge	Anzeige
F CLEAR PRGM	000-
H LBL B	001-25,13,12
0	002- 0
STO 1	003- 23 1
1	004- 1
STO + 1	005-23,51, 1
RCL 1	006- 24 1
H PSE	007- 25 74
G X ²	008- 15 3
H PSE	009- 25 74
H RTN	010- 25 12

Schieben Sie den PRGM-RUN-Schalter in Stellung RUN und prüfen Sie das Programm in seiner ursprünglichen Form.

Machen Sie darauf die vorgeschlagenen Änderungen und führen Sie das Programm abermals aus, um eine Tabelle von Quadratwerten zu erzeugen.

2. Erstellen Sie anhand des nachfolgenden Flußdiagramms ein Programm, das für aufeinanderfolgende Jahre den Endbetrag (FV) berechnet und anzeigt, auf den eine Spareinlage durch die Verzinsung angewachsen ist. Verwenden Sie dabei die Formel

$$FV = PV (1 + i)^n$$

wobei FV = zukünftiger oder Endbetrag der Spareinlage.

PV = Anfangswert (Konstostand zu Beginn).

i = Periodenzinssatz (als dezimaler Wert einzugeben;

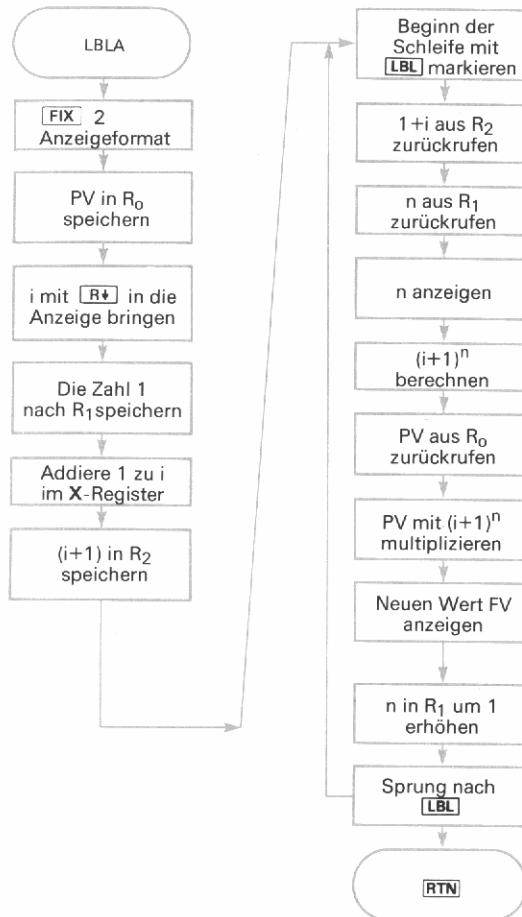
6 % entspricht beispielsweise 0,06).

n = Anzahl der Zinsperioden (in der Regel = Anzahl der Jahre).

Gehen Sie davon aus, daß i vor Starten des Programms in das Y-Register und PV in das angezeigte X-Register eingegeben wird.

Nachdem Sie das Programm erstellt und in den Rechner eingetastet haben, können Sie die Berechnung mit folgenden Ausgangsdaten starten: Jahreszinssatz $i = 6\%$ (als ,06 eingetastet) und Spareinlage (oder Anfangswert, PV) = 1000 DM (Ergebnis: erstes Jahr 1060 DM; zweites Jahr 1123,60 DM; drittes Jahr 1191,02 DM usw.).

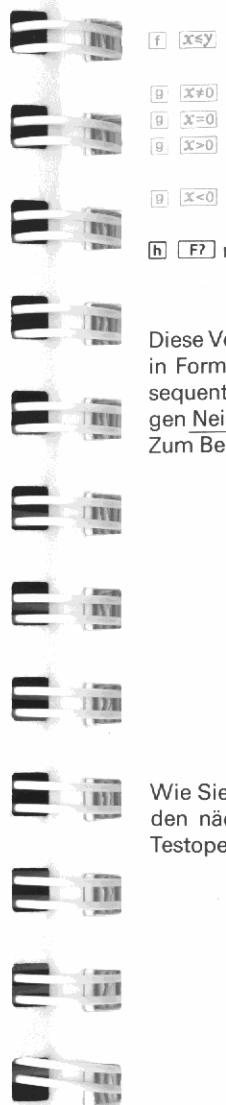
Das Programm bricht ab, wenn Sie R/S (oder eine andere Taste) drücken oder wenn ein Überlauf eintritt. Beobachten Sie, wie Ihre Ersparnisse von Jahr zu Jahr wachsen. Lassen Sie das Programm für unterschiedliche Zinssätze und Anfangswerte ablaufen.



LÖSUNGEN

Untenstehend geben wir eine mögliche Lösung für die 2. Übungsaufgabe

2. **h LBL A**
f FIX 2 Geeignetes, finanzielles Anzeigeformat
STO 0 PV speichern
g R↓ i in das Anzeigeregister zurückrufen
1
STO 1 Den Anfangswert von n (1) speichern
+ 1 zu i addieren
STO 2 (1+i) speichern
h LBL 0
RCL 1 n zurückrufen
h PSE
h yx
RCL 0 PV aus R₀ zurückrufen
x Neuen Wert F^V berechnen
h PSE
STO x 1 FV anzeigen
n in R₁ um 1 erhöhen
GTO 0 Unbedingte Verzweigung



Prüft, ob der Inhalt des **X**-Registers kleiner als oder gleich dem Inhalt des **Y**-Registers ist.

Prüft, ob der Inhalt des **X**-Registers von Null verschieden ist.

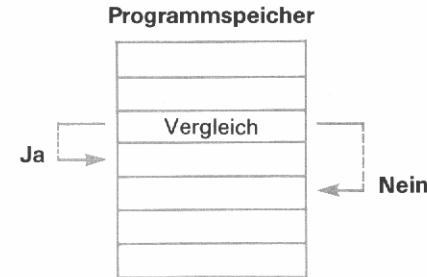
Prüft, ob der Inhalt des **X**-Registers gleich Null ist.

Prüft, ob der Inhalt des **X**-Registers größer als Null (d. h. positiv) ist.

Prüft, ob der Inhalt des **X**-Registers kleiner als Null (d. h. negativ) ist.

Prüft, ob das Flag n gesetzt ist (Flags werden später noch behandelt).

Diese Vergleichsoperationen treten an der entsprechenden Programmstelle in Form einer Frage auf. Ist die Antwort Ja, fährt das Programm mit der sequentiellen Ausführung der Programmschritte fort. Ist die Antwort Nein, überspringt das Programm den nachfolgenden Schritt.
Zum Beispiel:

**VERGLEICHSOOPERATIONEN UND BEDINGTE PROGRAMMVERZWEIGUNG**

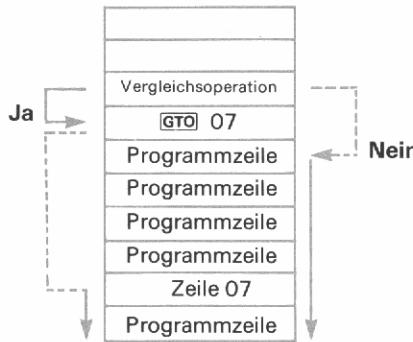
Es kommt häufig vor, daß innerhalb eines Programms eine Entscheidung getroffen werden soll. Ihr HP-34C verwendet Vergleichsoperationen und bedingte Verzweigungen, um Ihre Programmierungsmöglichkeiten zu vergrößern.

Die Vergleichsoperationen des HP-34C sind nützliche Programmieranweisungen; die es Ihrem Rechner ermöglichen, Entscheidungen für Sie zu fällen. Der HP-34C verfügt über die folgenden Vergleichsoperationen:

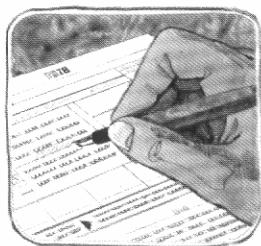
- f X#Y** Prüft, ob die Inhalte von **X**- und **Y**-Register verschieden sind.
f X=Y Prüft, ob die Inhalte von **X**- und **Y**-Register gleich sind.
f X>Y Prüft, ob die Zahl im **X**-Register größer als die Zahl im **Y**-Register ist.

Wie Sie sehen, führt der Rechner im Anschluß an die Vergleichsoperation den nächstfolgenden Programmschritt nur dann aus, wenn die mit der Testoperation gestellte Bedingung erfüllt ist, d. h. die Antwort Ja lautet.

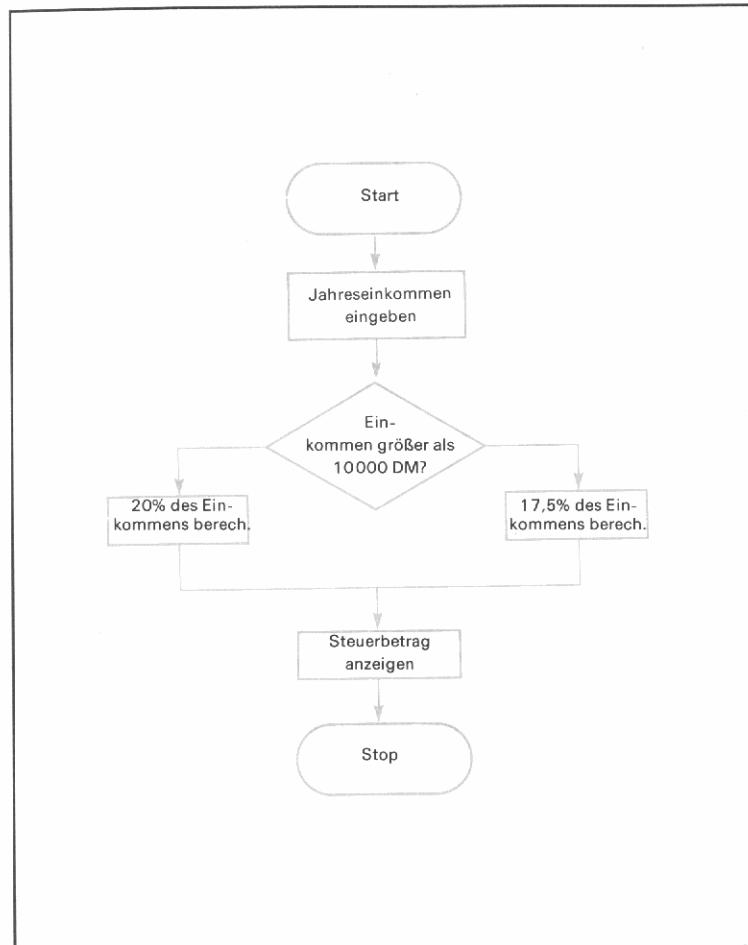
Die auf den Vergleichsbefehl folgende Speicherzeile kann eine beliebige Programmanweisung enthalten. In der Regel wird an dieser Stelle eine Sprunganweisung **GTO** stehen. Auf diese Weise wird die Programmausführung, wenn die gestellte Bedingung erfüllt ist, zu einer anderen Stelle des Programmspeichers verzweigt.



Beispiel: Für die Klienten eines Steuerberaters gelten die Steuersätze 20% bzw. 17,5% für Einkommen über bzw. unter DM 10000,-. Der Steuerberater möchte ein Programm erstellen, das für jeden seiner Klienten den zu zahlenden Steuerbetrag auf einfache Weise errechnet. In seinem Programm wird er bedingte Sprungbefehle benötigen.



Das Flußdiagramm zu diesem Problem kann z. B. folgendermaßen aussehen:



Schieben Sie den PRGM ■■■■■ RUN -Schalter in Stellung PRGM und löschen Sie den Programmspeicher.

Tastenfolge	Anzeige	
f CLEAR PRGM	000-	
h LBL A	001-25,13,11	
EEX	002- 33	
4	003- 4	Die Zahl wird im Y-Register
x ^y	004- 21	gespeichert
f X>Y	005- 14 51	Wenn das Jahreseinkommen
GTO B	006- 22 12	DM 10000 übersteigt, verzweigt das Programm zu Marke B
1	007- 1	
7	008- 7	
•	009- 73	Dieser Teil des Programms
5	010- 5	berücksichtigt 17,5% Steuern
GTO 1	011- 22 1	
h LBL B	012-25,13,12	
2	013- 2	Dieser Teil des Programms
0	014- 0	berechnet 20% Steuern
h LBL 1	015-25,13, 1	
h %	016- 25 11	
h RTN	017- 25 12	

Verwenden Sie das Programm jetzt zur Berechnung der Steuerbeträge, die bei 15.000 DM und 7.500 DM Jahreseinkommen zu zahlen sind:

Schieben sie den PRGM ■■■■■ RUN -Schalter in Stellung RUN, und drücken Sie h RTN um an den Programmanfang zu gelangen.

Tastenfolge	Anzeige
15000 R/S	3.000,0000
7500 R/S	1.312,5000

Um weitere Steuerbeträge zu errechnen, hat der Steuerberater lediglich das Einkommen seiner Klienten einzugeben und A zu drücken. Der Rechner bestimmt automatisch die Einkommensgruppe des Klienten und berechnet den Steuerbetrag.

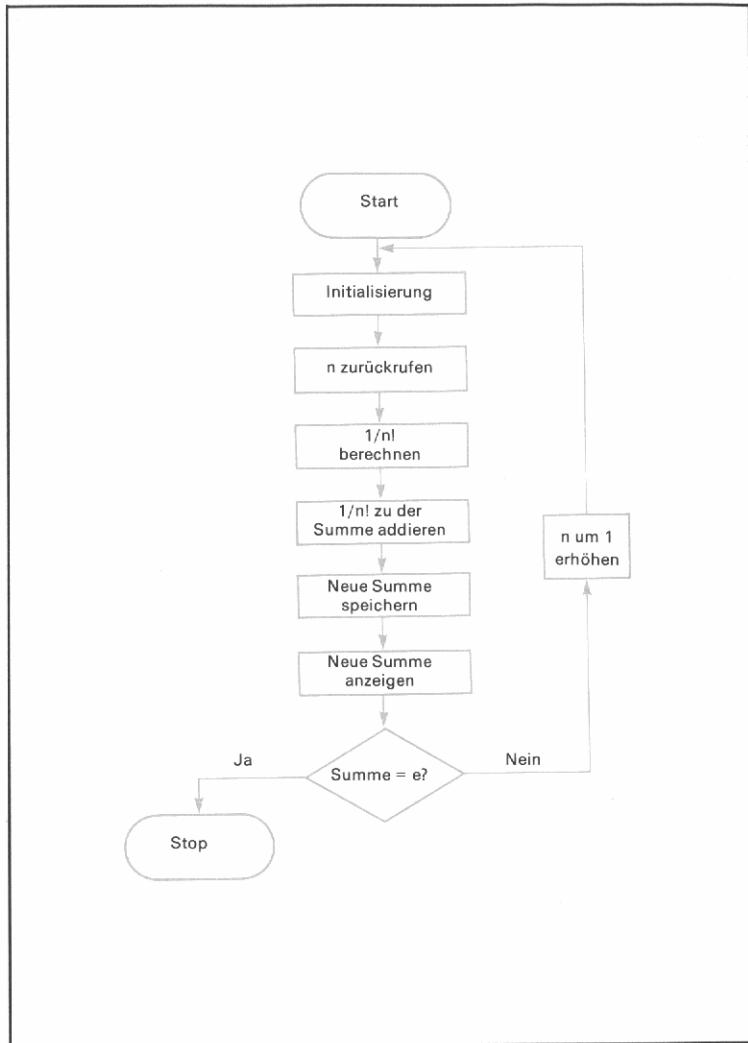
Von großem Nutzen sind bedingte Sprünge auch im Zusammenhang mit Programmschleifen. Die Programmschleifen, denen Sie bis jetzt begegnet sind, waren Endlosschleifen. Der Rechner führt diese Programmschritte immer wieder aus, er bleibt also in dieser Schleife, bis Sie das Programm mit der R/S -Taste anhalten oder ein Überlauf stattfindet.

Sie können die Vergleichsoperationen dazu verwenden, den Rechner zu gegebenem Zeitpunkt wieder aus dieser Programmschleife herausspringen zu lassen. Das kann beispielsweise dann geschehen, wenn der Rechner bereits eine bestimmte Anzahl von Schleifendurchläufen ausgeführt oder einen iterativ berechneten Wert ausreichend genau bestimmt hat.

Beispiel: Was Sie wissen, ist der Wert der Euler'schen Zahl e, das ist die Basis der natürlichen Logarithmen im Innern Ihres HP-34C gespeichert. (Sie können diesen Wert mit der Tastenfolge 1 g ex anzeigen). Das folgende Programm errechnet diese Konstante über die folgende Reihenentwicklung:
 $e = 1 + 1/1! + 1/2! + \dots + 1/n!$

Nach jedem Schleifendurchlauf wird die neue Näherungslösung angezeigt und mit dem im Rechner gespeicherten genauen Wert für e verglichen. Wenn beide Werte gleich sind, verlässt das Programm die Iterationsschleife und hält an.

Ein Flußdiagramm soll zur besseren Verständigung des Programms dienen.



Um das Programm im Rechner zu speichern, schieben Sie den PRGM RUN - Schalter in Stellung PRGM.

Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL	001-25,13,11
RCL 1	002- 24 1
RCL 0	003- 24 0
h x1	004- 25 1
h 1/x	005- 25 2
+	006- 51
f FIX 9	007-14,11, 9
STO 1	008- 23 1
h PSE	009- 25 14
1	010- 1
g ex	011- 15 1
f X=Y	012- 14 71
h RTN	013- 25 12
1	014- 1
STO + 0	015-23,51, 0
GTO A	016- 22 11

Um das Programm zu starten, schieben Sie erst den PRGM RUN - Schalter in Stellung RUN.

Vergewissern Sie sich, daß die Speicherregister gelöscht sind. Drücken Sie dann A, um das Programm zu starten.

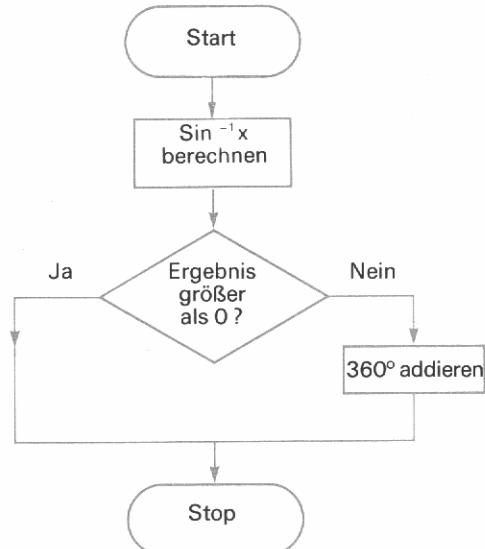
Tastenfolge	Anzeige	
f CLEAR REG	1,31250	löst alle Speicherregister (Die Anzeige des vorangegangenen Beispiels bleibt erhalten.)
A	2.718281828	

Sie können den Programmablauf innerhalb der Iterationsschleife verfolgen, bis der Näherungswert für e gleich dem intern gespeicherten Wert e des Rechners ist.

Wenn die Bedingung $x=y$ schließlich erfüllt ist, verzweigt das Programm entsprechend der nachstehenden Anweisung RTN und hält an.

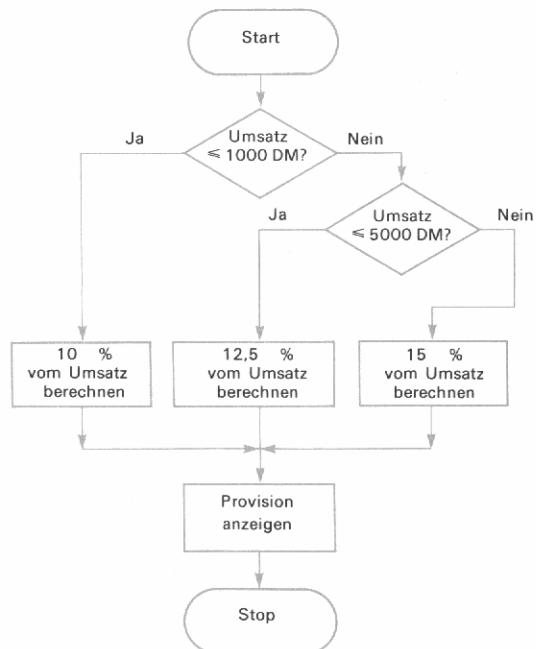
Übungsaufgaben:

1. Erstellen Sie ein Programm, das den Arkussinus SIN^{-1} eines Eingabewertes x berechnet, der zuvor in das angezeigte X-Register eingegeben wurde. Anschließend ist der berechnete Winkel auf sein Vorzeichen zu prüfen und 360° zu addieren, wenn der Winkel nicht bereits größer als Null ist. Damit wird erreicht, daß der von diesem Programm berechnete Winkel stets positiv ist. Beim Erstellen des Programms können Sie sich an das folgende Flußdiagramm halten:



2. Erstellen Sie anhand des folgenden Flußdiagramms ein Programm, mit dem ein Vertreter seine umsatzabhängige Provisionen berechnen kann; bei Verkäufen bis zu 1000 DM werden ihm 10%, für Verkäufe zwischen 1000 DM und 5000 DM 12,5% und für Umsätze über 5000 DM 15% als Provision gewährt. Das Programm soll sowohl den Umsatz als auch die Provision anzeigen.

Tasten Sie das Programm in den Rechner ein. Berechnen Sie dann die Provisionen für folgende Posten: 500 DM, 1500 DM, 5000 DM und 6000 DM (Ergebnisse: 50,00 DM, 187,50 DM, 625,00 DM und 900,00 DM).



Tastenfolge	Anzeige
[CLEAR] [PRGM]	000-
[LBL] A	001-25,13,11
[X<0]	002- 15 41
GTO 0	003- 22 0
[RTN]	004- 25 12
[LBL] 0	005-25,13, 0
3	006- 3
6	007- 6
0	008- 0
+	009- 51
[RTN]	010- 25 12
Bedienungsanweisung: Nach Eintasten des Programms schieben Sie den PRGM [RUN]-Schalter in Stellung RUN und schalten Sie das Anzeigeformat auf FIX 4.	
Geben Sie eine Zahl ein und drücken sie A.	
Tastenfolge	Anzeige
[CLEAR] [PRGM]	000-
[LBL] A	001-25,13,11
EEX	002- 33
3	003- 3
[X>Y]	004- 14 51
GTO 0	005- 22 0
5	006- 5
X	007- 61
X ² Y	008- 21
[X ² Y]	009- 14 41
GTO 1	010- 22 1
1	011- 1
5	012- 5
[%]	013- 25 41
[RTN]	014- 25 12
[LBL] 1	015-25,13, 1
1	016- 1
2	017- 2
•	018- 73
5	019- 3
[%]	020- 25 41
[RTN]	021- 25 12
[LBL] 0	022-25,13, 0

x:y	023-	21
1	024-	1
0	025-	0
h %	026-	25 41
h RTN	027-	25 12

Bedienungsanweisung: Nach Eintasten des Programms schieben Sie den PRGM **RUN**-Schalter in Stellung RUN und schalten Sie das Anzeigeformat auf FIX 2. Geben Sie die Verkaufszahlen in DM ein und drücken Sie **A**.

FLAGS

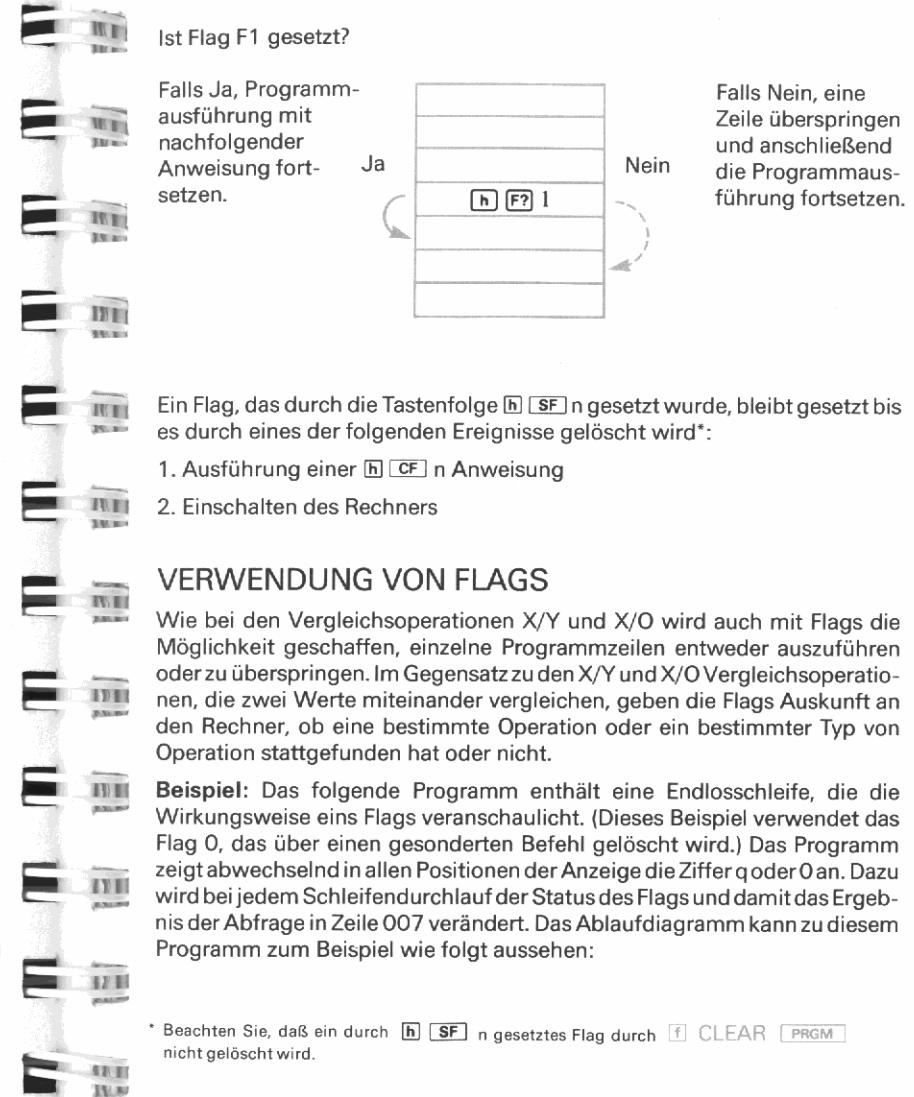
Neben den Vergleichsbefehlen bzw. Verhältnistests (**X=Y**, **X \neq Y** usw.) können Sie auch sogenannte «Flags» für die Programmierung von Verzweigungen oder bedingt auszuführenden Operationen verwenden. Diese Flags sind rechnerinterne Einrichtungen, die wie ein Schalter funktionieren und wahlweise «gesetzt» oder «gelöscht» sein können. Sie können dann im Rahmen eines Programms das Flag mit einer speziellen Tastenfunktion auf seinen Zustand prüfen, d. h. feststellen, ob es gesetzt ist oder nicht. In Abhängigkeit von der Stellung des Flags können dann innerhalb des Programms Entscheidungen getroffen und Verzweigungen ausgeführt werden. Ihr HP-34C verfügt über 4 solcher Flags, die von 0 bis 3 durchnummieriert sind. Wenn Sie eines dieser Flags «setzen» (d. h. einschalten) wollen, müssen Sie **SF** (Flag setzen) und anschließend die entsprechende Zifferntaste (0, 1, 2, 3) drücken. So wird beispielsweise Flag 3 mit der folgenden Tastenfolge gesetzt:

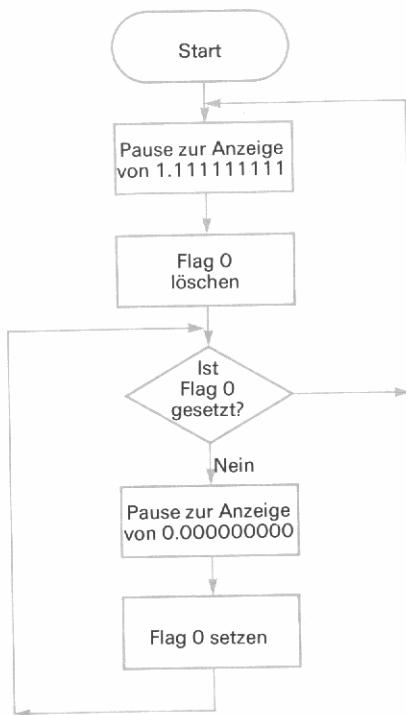
H SF 3

Die Anweisung **CF** (Flag löschen) wird zum Löschen der Flags verwendet. Um Flag 3 zu löschen, drücken Sie:

H CF 3

Sie können bei der Verwendung von Flags mit der Anweisung **F?** (ist Flag gesetzt?), gefolgt von einer der Zifferntasten (0, 1, 2, 3), Bedingungen programmieren, die in Abhängigkeit vom Zustand des entsprechenden Flags erfüllt oder nicht erfüllt sind. Wenn eines der Flags mit der Tastenfolge **H F? n** geprüft wird, führt der Rechner den nachfolgenden Programmbefehl aus, wenn das Flag gesetzt ist (d. h. die Antwort auf die Testfrage Ja ist). Ist das entsprechende Flag dagegen gelöscht und damit die Bedingung nicht erfüllt (die Antwort auf die Testfrage also Nein), überspringt der Rechner die nachfolgende Anweisung, bevor er mit der Ausführung der weiteren Programmschritte fortfährt.





Das Programm geht davon aus, daß Null in das Speicherregister R_0 und die Zahl 1.11111111 in das Register R_1 gespeichert wurde. Schalten Sie den PRGM ■■■ RUN -Schalter in Stellung PRGM.

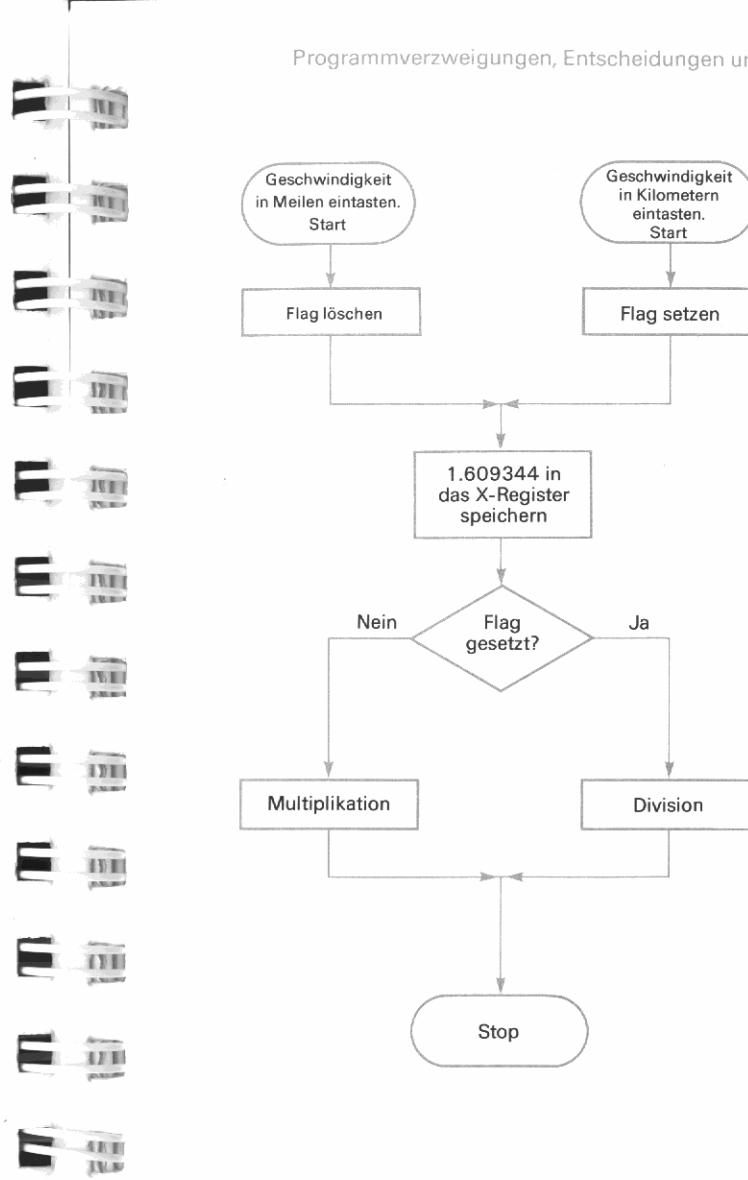
Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL A	001-25,13,11
RCL 1	002- 24 1 Rückruf und Anzeige von 1,11111111 aus Register R_1
h PSE	003- 25 74
h CF 0	004-25,61, 0 Löscht Flag 0
h LBL B	005-25,13,12
h F? 0	006-25,71, 0 Prüft Flag 0
GTO A	007- 22 11 Falls gesetzt (logisch <wahr>), Sprung nach LBL A
RCL 0	008- 24 0 Andernfalls: Rückruf und
h PSE	009- 25 74 Anzeige von 0 aus Register R_0 , Setzen von Flag 0 und
h SF 0	010-25,51, 0 Sprung nach LBL B
GTO B	011- 22 12
h RTN	012- 25 12
Schalten Sie jetzt in den RUN-Modus um, besetzen Sie die verschiedenen Register mit den entsprechenden Werten und starten Sie das Programm: Schieben Sie den PRGM ■■■ RUN -Schalter in Stellung RUN.	
Tastenfolge	Anzeige
f FIX 9	0.000000000
0 STO 0	0.000000000
1.11111111	1.11111111
STO 1	1.11111111
A	1.11111111 Die beiden Zahlen werden abwechselnd angezeigt
Sie können das Programm jederzeit dadurch anhalten, daß Sie auf dem Tastenfeld RVS (oder eine beliebige andere Taste) drücken.	
Wirkungsweise: Wenn Sie Null nach R_0 und die Zahl 1.11111111 in das Register R_1 gespeichert haben, können Sie das Programm mit A starten. Die Programmausführung wird anschließend von h PSE in Zeile 003 kurzfristig zur Anzeige der Zahl aus Register R_1 unterbrochen. Dann löscht h CF 0 in Zeile 004 das Flag 0 (wobei sich nichts ändert, da dieses Flag bereits zu Beginn des Programms gelöscht war).	

Da die auf **LBL A** folgende Routine kein zugehöriges **RTN** besitzt, läuft die Programmausführung über die **LBL B** -Anweisung in Zeile 005 zu dem Test, **h F7 0**, in Zeile 006 weiter. An dieser Stelle wird die Frage gestellt (ist Flag 0 gesetzt (logisch «wahr»)?). Da das Flag bereits gelöscht ist, lautet die Antwort Nein und der Rechner überspringt den nachfolgenden Programmschritt; als nächstes wird demnach die Anweisung **RCL 0** in Zeile 008 ausgeführt. Dieser Programmschritt in Zeile 008 und **h PSE** in 009 verursachen eine Programmpause und die Anzeige Null aus Register R₀. Danach wird 0 durch **h SF 0** in Zeile 010 gesetzt und durch **GTO B** in Zeile 011 ein Sprung nach **LBL B** bewirkt.

Da das Flag 0 jetzt gesetzt ist, ist die Bedingung **h F7 0** (ist Flag 0 gesetzt?) erfüllt, d. h. die Antwort lautet jetzt Ja. Daher führt der Rechner diesmal den **GTO A** -Befehl in Zeile 007 aus, der unmittelbar auf diese Abfrage folgt und zuvor übersprungen wurde. Nachdem das Programm ein weiteres Mal angehalten und Null angezeigt hat, wird das Flag gelöscht, und der Vorgang spielt sich von neuem ab. Auf diese Weise zeigt der Rechner so lange beide Zahlen abwechselnd an, bis Sie die Ausführung des Programms über das Tastenfeld unterbrechen.

Übungsaufgabe

Eine Meile entspricht 1,609344 Kilometer. Erstellen Sie anhand des nachstehenden Flußdiagramms ein Programm, das die Eingabe von Entfernungswerten wahlweise in Meilen (markieren Sie diese Routine mit **LBL A**) oder Kilometer (markieren Sie die Routine mit **LBL B**) ermöglicht und diesen Wert dann jeweils in die andere Einheit umrechnet. Verwenden Sie ein Flag und ein Unterprogramm dazu, den angegebenen Wert entweder mit der Umrechnungskonstante zu multiplizieren oder ihn durch diese Zahl zu dividieren. (Hinweis: **h Vx X** hat die gleiche Wirkung wie **A**). Schalten Sie das Anzeigeformat auf FIX 4 und rechnen Sie dann mit Hilfe des Programms 26 Meilen in Kilometer und 1500 Meter (1,5 Kilometer) in Meilen um (Ergebnisse: 41.8429 Kilometer; 0,9321 Meilen)





ABSCHNITT 6: UNTERPROGRAMME



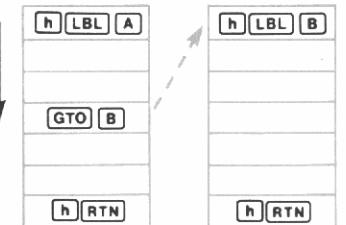
Es kommt häufig vor, daß sich innerhalb eines Programms eine bestimmte Tastenfolge mehrmals wiederholt. Wenn es sich dabei um identische aufeinanderfolgende Programmschritte handelt, kann dieser Teil als **«Unterprogramm»** ausgeführt werden. Ein solches Unterprogramm wird durch **GSB** (go to subroutine = Sprung zum Unterprogramm), gefolgt von einer der Marken (**A** **B** oder 0 bis 9) **aufgerufen**. Sie können zur Ansteuerung eines Unterprogramms auch die Tastenfolge **↓ GSB ↓** verwenden; die Bedeutung von **↓** wird an späterer Stelle ausführlich besprochen.



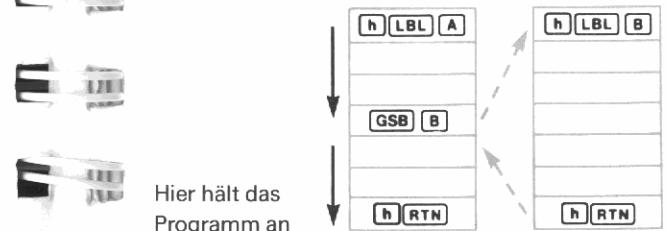
Die **GSB**-Anweisung bewirkt ebenso wie **GTO**, daß die Ausführung des Programms zu der im Anschluß an **GSB** bezeichneten Marke verzweigt. Der Unterschied zu **GTO** besteht darin, daß der Rechner, nach Ausführung des mit dieser Marke gekennzeichneten Unterprogramms, beim nächsten **RTN** nicht anhält, sondern in das Hauptprogramm zurück springt und die Ausführung des Programms ab der Anweisung fortsetzt, die auf den **GSB**-Befehl folgt. Das nachstehende Diagramm macht die unterschiedliche Wirkung von **GTO** und **GSB** deutlich.



Programmverzweigung



Unterprogramme



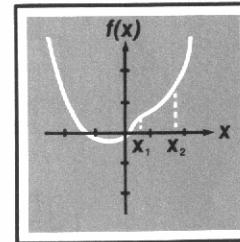
Betrachten wir zunächst einmal die linke Skizze. Nach Drücken von **A** beginnt der Rechner mit der Ausführung aufeinanderfolgender Programmschritte. Bei Erreichen der Anweisung **GTO B** wird die Programmausführung unterbrochen und der Speicher auf die Marke **LBL B** abgesucht. Ab dieser Stelle setzt der Rechner die sequentielle Ausführung aufeinanderfolgender Programmschritte fort. Wenn in der Folge die erste **RTN**-Anweisung auftritt, springt der Rechner nach Zeile 000 zurück und hält an. Tritt im Verlauf der Programmausführung die Anweisung **GSB B** (Sprung zum Unterprogramm B) auf, sucht der Rechner ebenfalls den Programmspeicher auf das erste Auftreten von **LBL B** ab und setzt ab dieser Stelle die Ausführung der einzelnen Programmschritte fort.

Wenn der Rechner jetzt in der Folge auf **RTN** (Zurück!) trifft, bricht er die Programmausführung nicht ab, sondern setzt sie im Hauptprogramm mit der nächsten, auf **GSB B** folgenden, Anweisung fort.

Wie Sie sehen, besteht der einzige Unterschied zwischen dem Unterprogramm und einer normalen Programmverzweigung im anschließenden Rücksprung zum Hauptprogramm nach Ausführung von **RTN**. Im Anschluß an **GTO** springt der Rechner nach **RTN** nach Zeile 000 zurück und hält an; in der Folge von **GSB** verursacht **RTN** den Rücksprung in das Hauptprogramm, wo der Rechner die sequentielle Ausführung der Programmschritte bis zum nächsten **RTN** (oder **RS**) fortsetzt.

Beispiel: Schreiben Sie ein Programm, das die mittlere Steigung eines Graphen $f(x)$ zwischen den Punkten X_1 und X_2 berechnet.

Die Funktion lautet $f(x) = x^2 - \ln(x^2 + e^{-x})$.



Lösung: Die mittlere Steigung von $f(x)$ wird folgendermaßen berechnet:

$$\frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{[X_2^2 - \ln(X_2^2 + e^{-X_2})] - [X_1^2 - \ln(X_1^2 + e^{-X_1})]}{x_2 - x_1}$$

Beachten Sie, daß die Lösung eine zweifache Berechnung des Ausdrucks $x^2 - \ln(x^2 + e^{-x})$ erforderlich macht.



Das folgende Programm sieht die Eingabe der Werte für X_1 und X_2 vor und berechnet die mittlere Steigung, wenn **A** gedrückt wird.



H LBL A



STO 0



X:Y



STO - 0



ENTER



CHS



g e^x



h LST X



g x²



+



f LN



CHS



x:Y



g x²



+



RCL 0



÷



h RTN

Berechnet $f(X_1)$

Diese beiden Programmteile sind identisch
Berechnet $f(X_2)$

X:Y
ENTER
CHS
g e^x
h LST X
g x²
+
f LN
CHS
x:Y
g x²
+
RCL 0
÷
h RTN



Da die Routine zur Berechnung von $f(X_1)$ einen Großteil der Programmschritte umfaßt, die auch zur Berechnung von $f(X_2)$ verwendet werden, ist es sinnvoll, für diesen Teil beider Routinen ein Unterprogramm vorzusehen. Auf diese Weise kann ein Teil des Speicherplatzes von beiden Programmen gemeinsam genutzt werden. Die Routinen zur Berechnung von $f(X_1)$ und $f(X_2)$ können beide das gleiche Unterprogramm aufrufen:



```

001 h LBL A
002 STO 0
003 x2y
004 STO - 0
005 GSB 0
006 CHS ←
007
008 GSB 0
009 - ←
010 RCL 0
011 +
012 h RTN
013 h LBL 0
014 ENTER+
015 CHS
016 g ex
017 h LST X
018 g x2
019 +
020 f LN
021 CHS
022 x2y
023 g x2
024 +
025 h RTN

```

Das so abgeänderte Programm wird, wenn Sie **A** drücken, mit der Anweisung **h LBL A** in Zeile 000 gestartet, wobei die Werte X_1 und X_2 vorher im Y-Register und im angezeigten X-Register stehen sollten.

Wenn das Programm anschließend die Anweisung **GSB** in Zeile 005 erreicht, wird die Programmausführung mit **h LBL 0** in Zeile 013 und der Berechnung von $f(X_1)$ fortgesetzt.

Wenn wir beispielsweise Werte von 2 für X_1 und 3 für X_2 verwendet hätten, würde sich der Stack-Inhalt bei der Berechnung von $f(X_1)$ folgendermaßen ändern:

001	002	003	004	005
2	2	3	3	3
3	3	2	2	2
h LBL A	STO 0	x₂y	STO - 0	GSB 0
(x_1 in Y., x_2 in X.)	(x_2 in R ₀)	(x_1 in X., x_2 in Y.)	($x_2 - x_1$ in R ₀)	

013	014	015	017
3	3	3	3
2	2	2	0.1353
h LBL 0	ENTER+	CHS	g e^x
(Anfang Unterprogramm)	(-x ₁)	(e ^{-x₁})	(Ruft -x ₁ zurück)

018	019	020	021	022
3	3	3	3	3
2	3	3	3	3
0.1353	2	2	2	-1.4196
4	4.1353	1.4196	-1.4196	2
g x²	+	f LN	CHS	x₂y
(x_1^2)	($x_1^2 + e^{-x_1}$)	($\ln(x_1^2 + e^{-x_1})$)	($-\ln(x_1^2 + e^{-x_1})$)	

023	024	025
3	3	3
3	3	3
-1.4196	3	3
4	2.5804	2.5804
g x²	+	h RTN
(x_1^2)	($f(x_1)$)	(Rücksprung ins Hauptprogramm)

006	007	008	013	014
3	3	3	3	3
3	3	-2.5804	-2.5804	-2.5804
3	-2.5804	-2.5804	-2.5804	3
-2.5804	3	3	3	3
CHS	x₂y	GSB 0	h LBL 0	ENTER+
(-f(x ₁))	(-f(x ₁) nach Y.)		(Anfang Unterprogramm)	

Wenn der Rechner die **RTN**-Anweisung in Zeile 025 ausführt, erfolgt ein Rücksprung in das Hauptprogramm, wo die Ausführung mit der ersten Anweisung nach der **GSB**-Anweisung fortgesetzt wird. Wenn das Programm die Anweisung **GSB 0** in Zeile 8 erreicht, wird wiederum die Programmausführung mit **h LBL 0** in Zeile 013 fortgesetzt. Wir führen das Beispiel anhand des Stack-Inhalts fort.

015	016	017	018	019
3	3	-2.5804	-2.5804	-2.5804
-2.5804	-2.5804	3	3	-2
3	3	0.0498		3
-3	0.0498	-3	9	9.0498

CHS
 $(-x_2)$

g e^x
 (e^{-x_2})

h LST X
(Ruft $-x_2$ zurück)

g x²
 (x_2^2)

+
 $(x_2^2 + e^{-x_2})$

020	021	022	023	024
-2.5804	-2.5804	-2.5804	-2.5804	-2.5804
-2.5804	-2.5804	-2.5804	-2.5804	-2.5804
3	3	-2.2027	-2.2027	-2.5804
2.2027	-2.2027	3	9	6.7973

f LN
 $(\ln(x_2^2 + e^{-x_2}))$

CHS
 $(-x_2)$

xx:y
 (x_2)

g x²
 (x_2^2)

+
 $(f(x_2))$

025

-2.5804
-2.5804
-2.5804
6.7973

h RTN
(Rücksprung ins
Hauptprogramm)

Nachdem der Rechner das mit **LBL 0** beginnende Unterprogramm ein zweites Mal durchlaufen hat, erfolgt nach der **RTN**-Anweisung in Zeile 025 ein Rücksprung in das Hauptprogramm, wo das Programm mit der ersten Anweisung fortgesetzt wird, die auf die letzte **GSB 0**-Anweisung folgt. $f(x_1)$ steht im **X**-Register; $f(x_2)$ steht im **Y-Z**- und **T**-Register.



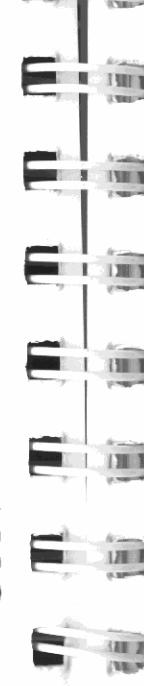
009	010	011	012
-2.5804	-2.5804	-2.5804	-2.5804
-2.5804	-2.5804	-2.5804	-2.5804
-2.5804	4.2168	-2.5804	-2.5804
4.2168	1	4.2168	4.2168

+
 $(f(x_2) - f(x_1))$

RCL 0
 $(x_2 - x_1)$

+
 $(f(x_2) - f(x_1)) \div (x_2 - x_1))$
h RTN

Programmende. Der Rechner springt nach Zeile 000 und hält an. Die mittlere Steigung von $f(X)$ zwischen X_1 und X_2 steht in der Anzeige. Weitere Werte von $f(x_1)$ in den **Y-Z**- und **T**-Registern werden nicht benötigt.



Tasten Sie jetzt das Programm ein und versuchen Sie die folgenden Aufgaben. Schieben Sie den **PRGM** **RUN**-Schalter in Stellung **PRGM**.

Tastenfolge

Anzeige

f CLEAR PRGM	000-
h LBL A	001-25,13,11
STO 0	002- 23 0
xx:y	003- 21
STO - 0	004-23,41, 0
GSB 0	005- 13 0
CHS	006- 32
xx:y	007- 21
GSB 0	008- 13 0
+	009- 51
RCL 0	010- 24 0
÷	011- 71
h RTN	012- 25 12
h LBL 0	013-25,13, 0
ENTER+	014- 31
CHS	015- 32
g e^x	016- 15 1
h LST X	017- 25 0
g x²	018- 15 3

+	019-	51
f LN	020-	14 1
CHS	021-	32
x ² y	022-	21
g x ²	023-	15 3
+	024-	51
h RTN	025-	25 12

Schieben Sie den PRGM ■■■ RUN -Schalter in Stellung RUN. Berechnen Sie jetzt die mittlere Steigung von f(X) für die folgenden Wertpaare:

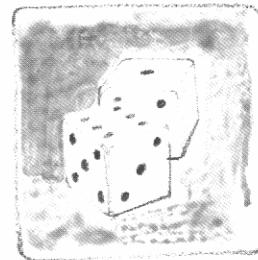
(0, 0.5), (0.55, 1.15), (1.25, 1.75).

Ergebnisse: 0,8097; 0,6623; 1,8804.

ANWENDUNGSBEISPIELE FÜR UNTERPROGRAMME

Unterprogramme stellen eine wesentliche Erweiterung der Programmierungsmöglichkeiten Ihres HP-34C dar. Eine solche, von verschiedenen Teilen des Hauptprogramms verwendete Tastenfolge kann beispielsweise eine Programmschleife beinhalten oder aber selbst Bestandteil einer Programmschleife sein. Ein weiterer häufig angewandter und Speicherplatz sparer Trick besteht darin, die gleiche Routine einmal als Unterprogramm und zum anderen als Bestandteil des Hauptprogramms zu verwenden.

Beispiel: Das folgende Programm simuliert das Werfen zweier Spielwürfel, wobei zuerst die Augenzahl des ersten Würfels (eine ganze Zahl von 1 bis 6) und dann die des zweiten Würfels (ebenfalls eine ganze Zahl von 1 bis 6) während einer Programmpause angezeigt werden.



Den ‹Kern› des Programms bildet ein Zufallszahlen-Generator (genauer: ein Pseudo-Zufallszahlen-Generator), der zuerst als Unterprogramm und dann als Bestandteil des Hauptprogramms eingesetzt wird. Wenn Sie zu Beginn einen ‹Anfangswert› eingeben und dann A drücken, wird die Augenzahl für den ersten Würfel erzeugt, indem die Routine h LBL 2 als Unterprogramm verwendet wird.



Bei der anschließenden Erzeugung der Ziffer für die Augenzahl des zweiten Würfels wird die gleiche Routine als Bestandteil des Hauptprogramms verwendet.



Zum Eintasten des Programms schieben Sie den PRGM ■■■ RUN -Schalter in Stellung PRGM.



Tastenfolge Anzeige

f CLEAR PRGM	000
GTO 1	001- 22 1
h LBL A	002- 25,13,11
STO 0	003- 23 0
h LBL 1	004- 25,13, 1
0	005- 0
STO 1	006- 23 1
GSB 2	007- 13 2

h LBL 2	008- 25,13, 2
RCL 0	009- 24 0
9	010- 9
9	011- 9
7	012- 7

X	013- 61
h FRAC	014- 25 33
STO 0	015- 23 0
6	016- 6
X	017- 61
1	018- 1
+	019- 51
h INT	020- 25 32
f FIX 0	021- 14,11, 0
h PSE	022- 25 74
STO + 1	023- 23,51, 1
RCL 1	024- 24 1
h RTN	025- 25 12



Die mit h LBL 2 beginnende Routine wird erst als Unterprogramm ausgeführt.

Jetzt wird die Routine als Teil des Hauptprogramms ausgeführt.

Das Programm verzweigt nach Zeile 008 wenn LBL 2 als Unterprogramm ausgeführt wird. Nachdem LBL 2 als Teil des Hauptprogramms ausgeführt worden ist, verzweigt der Rechner nach Zeile 000.

Schieben Sie jetzt den PRGM ■■■■■ RUN -Schalter in Stellung RUN und lassen Sie mit Ihrem HP-34C die Würfel rollen. Dazu ist als erstes ein sogenannter «Anfangswert» (eine beliebige Zahl zwischen 0 und 1) vorzugeben und anschließend A zu drücken. Der Rechner zeigt dann zuerst die Augenzahl des ersten und dann des zweiten Würfels an. Um ein zweites Mal zu würfeln, drücken Sie R/S.

Das Programm verwendet den jeweils letzten Wert als «Anfangswert» für den nächsten Wurf.

Wenn sie wollen, können Sie dieses Würfelprogramm dazu verwenden, mit Ihren Freunden um die Wette zu knobeln. Wenn Sie beim ersten «Wurf» 7 oder 11 Augen erhalten, haben Sie gewonnen; haben Sie dagegen eine andere Augenzahl gewürfelt, müssen Sie so lange weiterspielen (Sie drücken wiederholt R/S, bis Sie erneut diese Punktzahl erreichen (und gewinnen) oder die Augenzahl 7 oder 11 erhalten (und verlieren). Zur Ausführung des Programms:

Tastenfolge	Anzeige
2315478	0,2315478
A	10,
R/S	8,
R/S	5,
R/S	7,

Der „Anfangswert“
Ihrer Augenzahl ist 10.
Sie haben die vorgegebene
Augenzahl verfehlt.
Wieder kein Glück gehabt.
Jetzt haben Sie verloren.

Versuchen Sie es noch einmal mit der letzten Zahl als «Anfangswert».

Tastenfolge	Anzeige
R/S	8,
R/S	8,

Ihre Augenzahl ist 8.
Herzlichen Glückwunsch!
Sie haben gewonnen.

Bevor Sie fortfahren, schalten Sie das Anzeigeformat auf FIX 4.

Tastenfolge	Anzeige
f FIX 4	8,0000

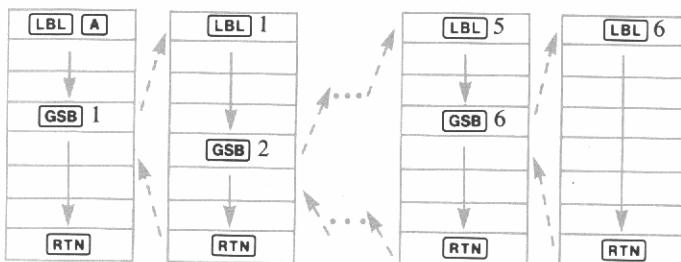
GRENZEN BEI DER VERWENDUNG VON UNTERPROGRAMMEN

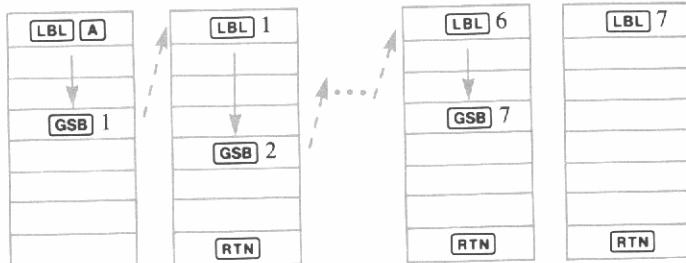
Ein Unterprogramm kann ein zweites Unterprogramm aufrufen, das dann wiederum seinerseits ein Unterprogramm verwenden kann. Die Verschachtelung solcher Unterprogramme ist lediglich durch die maximale Anzahl von Rücksprungbefehlen eingeschränkt, die sich der HP-34C intern «merken» kann.

Der HP-34C kann jederzeit die Rücksprungadressen für 6 Unterprogramme speichern. Das nachstehende Diagramm veranschaulicht diesen Vorgang.

Der Rechner berücksichtigt 6 anstehende Rücksprungadressen.

Hauptprogramm



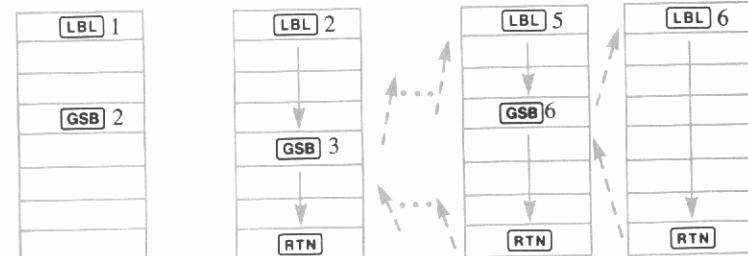
Hauptprogramm

Der Rechner hält mit der Fehlermeldung ERROR 8 an.

Der Rechner kann **RTN** selbstverständlich beliebig oft ausführen, wenn es nicht Bestandteil eines Unterprogramms ist (Verzweigung nach Zeile 000 und Anhalten). Wenn Sie aber **GTO** oder **GSB** gefolgt von **A**, **B** oder 0 bis 9 über die Tastatur ausführen, werden alle schwelbenden **RTN**-Anweisungen im Rechner aufgehoben.

Drücken Sie **GSB 2**

Hauptprogramm Programmausführung beginnt hier



ENDE

Programmausführung wird angehalten und der Rechner kehrt nach Zeile 000 zurück.



Beachten Sie, daß im PRGM-Modus bei schrittweiser Ausführung eines Programms mit Unterprogrammen der Ablauf derselbe wie in einem laufenden Programm ist.

VERWENDUNG VON **h RTN** AM ENDE DES BELEGten PROGRAMMSPEICHERS

Die Programmierbeispiele in Ihrem HP-34C Bedienungshandbuch enthalten eine **h RTN**-Anweisung als letzte Zeile des belegten Programmspeichers. Dies soll sowohl das Ende der Programme kennzeichnen als auch die Wirkungsweise von **RTN** als Programmschritt veranschaulichen. Sie können **h RTN** jedoch weglassen, wenn es als letzte Anweisung im belegten Programmspeicher auftaucht, ohne die Programmausführung dadurch zu beeinflussen. Warum? Wenn die letzte im Programmspeicher stehende Anweisung nicht **h RTN** ist, verhält sich die Programmausführung als ob eine **h RTN**-Anweisung unmittelbar nach der letzten von Ihnen eingetasteten Anweisung stehen würde.

Mit anderen Worten, wenn der Rechner bei der Programmausführung als letzte Anweisung im belegten Programmspeicher keine **h RTN**-Anweisung findet:

1. springt er auf die erste Zeile, die unmittelbar nach der letzten **GSB**-Anweisung steht, wenn er sich in einem Unterprogramm befindet,
2. springt er nach Zeile 000 zurück und hält an, wenn er sich nicht in einem Unterprogramm befindet.

Wenn die letzte Zeile des belegten Programmspeichers eine **GSB**-Anweisung enthält, führt der Rechner das angegebene Unterprogramm aus, kehrt dann nach Zeile 000 zurück und hält an.

Beachten Sie, daß der Rechner den Programmspeicher immer vorwärts (d. h. in Richtung aufsteigender Programmzeilen-Nummern) nach der angegebenen Marke absucht, wenn er auf eine **GTO**- oder **GSB**-Anweisung trifft. Diese Eigenschaft ermöglicht es Ihnen häufig ein Programm so zu schreiben, daß Sie eine bestimmte Marke mehr als einmal verwenden können.

Beispiel:

Das folgende Programm, das den Wert des Aufdrucks $\sqrt{x^2 + y^2 + z^2 + t^2}$ berechnet, verwendet **LBL A** zur Kennzeichnung des Programmanfangs und auch des Unterprogramms innerhalb des Programms. Das Programm wird ausgeführt, indem Sie die Variablen x, y, z und t zum Stack-Inhalt machen und **A** drücken.

Schieben Sie den PRGM -Schalter in Stellung PRGM.

Tastenfolge Anzeige

f CLEAR PRGM	000-
h LBL A	001-25,13,11
g X ²	002- 15 3
GSB A	003- 13 11
GSB A	004- 13 11
GSB A	005- 13 11
f √X	006- 14 3
h RTN	007- 25 12
h LBL A	008-25,13,11
x:y	009- 21
g X ²	010- 15 3
+	011- 51
h RTN	012- 25 12

Schieben Sie den PRGM -Schalter in Stellung RUN und tasten Sie die folgenden Variablen ein:

x = 4,3 y = 7,9 z = 1,3 t = 8,0

Tastenfolge Anzeige

8 ENTER	8,0000
1,3 ENTER	1,3000
7,9 ENTER 4,3 A	12,1074



ABSCHNITT 7: FORTGESCHRITTENE PROGRAMMIERUNG



VERWENDUNG DES I-REGISTERS



Das I-Register ermöglicht eine Fülle weiterer Programmietechniken und ist als eine der wichtigsten Voraussetzungen für die herausragende Leistungsfähigkeit Ihres HP-34C anzusehen. Abgesehen davon, daß es als Register für einfaches Abspeichern und Zurückrufen von Daten dient, kann es in Verbindung mit anderen Operationen für folgende Funktionen verwendet werden:



- Inkrementieren oder Dekrementieren des augenblicklichen Inhalts von I um einen vorgegebenen Wert für Schleifensteuerung oder andere Funktionen.



- Indirekte Steuerung der Registeradressen bei **STO**, **RCL** und Register-Arithmetik-Operationen.



- Indirekte Steuerung der Adresse bei **GTO** und **GSB**.



- Indirekte Steuerung des Anzeigeformats in den **FIX**-, **SCI**- und **ENG**-Modi.



SPEICHERN EINER ZAHL IM I-REGISTER



Mit der Tastenfolge **STO** f I können Sie jederzeit die im X-Register angezeigte Zahl in das I-Register speichern. Um beispielsweise die Zahl 7 in das I-Register zu schreiben:



Vergewissern Sie sich, daß der PRGM -Schalter in Stellung RUN steht.



Tastenfolge Anzeige

7 STO f I 7,0000



Um eine im I-Register gespeicherte Zahl in das X-Register zurückzurufen, genügt es, wenn Sie einfach **RCL** f I drücken.



Tastenfolge Anzeige

CLX 0,0000
RCL f I 7,0000

Inhalt des I-Registers

AUSTAUSCH DER INHALTE VON X UND I

Sie können mit der Tastenfolge **f X↔I** jederzeit den Inhalt des angezeigten X-Registers mit dem des I-Registers austauschen. Die Wirkungsweise dieser Operation ähnelt der von **X↔Y**.

Tasten Sie beispielsweise die Zahl 2 in das angezeigte X-Register und tauschen Sie dann dessen Inhalt gegen den des I-Registers aus.

Tastenfolge	Anzeige	
2	2,	
f X↔I	7.0000	Inhalte von X- und I-Register wurden ausgetauscht.

Als Sie **X↔I** gedrückt haben, wurden die Inhalte der beiden Register wie folgt ausgetauscht:

Vorher	Nachher
0.0000	0.0000
0.0000	0.0000
7.0000	7.0000
2.0000	2.0000
→ Anzeige →	7.0000
I 7.0000	I 2.0000

Um die Inhalte des X- und I-Registers wieder in ihre ursprüngliche Position zu bringen:

Tastenfolge	Anzeige
f X↔I	2.00

INKREMENTIEREN UND DEMENTIEREN DES I-REGISTERS

Eine andere Möglichkeit der Änderung des I-Registerinhalts, die besonders innerhalb eines Programms von Bedeutung ist, besteht in der Verwendung der Funktionen **ISG** (Inkrement und Sprung wenn größer) und **DSE** (Dekrement und Sprung, wenn gleich oder kleiner). Beide Funktionen enthalten interne Zähler, die eine Steuerung des Programmschleifen und der sequentiellen Adressierung, die wir später behandeln werden, ermöglichen.



Diese beiden Funktionen verwenden eine Zahl, die bei der Steuerung der Programmschleifen auf besondere Art interpretiert wird. Die Zahl, die im I-Register gespeichert wird, hat folgendes Format:



wobei

nnnnn.xyyy

± nnnnn

xxx

yy

die Laufvariable
der Endwert
die Schrittweite

ist.



Der **nnnnn**-Teil der Zahl entspricht dem Anfangswert der Laufvariablen, mit der gezählt wird, wie oft eine Schleife durchlaufen wird. Wenn kein **nnnnn**-Wert angegeben ist, fängt der HP-34C mit Null an zu zählen. Der **nnnnn**-Wert kann eine beliebige ein- bis fünfstellige Zahl sein.



Der **xxx**-Teil der Zahl entspricht dem Endwert der Laufvariablen, bei der der Rechner zu zählen aufhört. Der **xxx**-Wert muß als dreistellige Zahl angegeben werden (z. B. muß ein **xxx**-Wert von 10 als 010 angegeben werden). Wenn kein **xxx**-Wert angegeben ist, hört der HP-34C bei Null auf zu zählen.



Der **yy**-Teil der Zahl entspricht dem Inkrement, mit dem gezählt wird. Der **nnnnn**-Wert (der Laufvariablen) wird um den **yy**-Wert erhöht oder erniedrigt. Wenn kein **yy**-Wert angegeben ist, nimmt der HP-34C an, daß um eins inkrementiert oder dekrementiert werden soll (**yy** = 01). Der **yy**-Wert muß als zweistellige Zahl angegeben werden (z. B. 02, 03, 55).



INKREMENT UND SPRUNG WENN GRÖSSER



Jedesmal, wenn **ISG** ausgeführt wird, wird **nnnnn** erhöht. Dann wird geprüft, ob **nnnnn** größer als **xxx** ist. Wenn dies der Fall ist, überspringt der HP-34C die nächste Programmzeile.



Wenn Sie also die Schleifensteuerungszahl 100.2001 im I-Register gespeichert haben und die Anweisung **ISG** ausführen, würde der Rechner bei 100 anfangen zu zählen.



Die Laufvariable würde hochgezählt werden bis sie größer als 200 ist, bei jeder Ausführung der Anweisung würde sie um eins inkrementiert werden.



Inhalt des I-Registers = 100.20001



Folgendes geschieht bei der Ausführung von **ISG** :

Das Zählen der Laufvariablen beginnt bei 100.

Die Laufvariable wird um eins erhöht.

Die Laufvariable wird mit 200 verglichen.

Wenn die Schleife einmal durchlaufen wird, steht 101.20001 in I. Wenn die Schleife 10 mal durchlaufen wird, steht 110.20001 in I. Bei jeder Ausführung von **ISG** wird nach der Inkrementierung überprüft, ob die Laufvariable größer als 200 ist. Sobald sie 200 überschreitet, überspringt der Rechner die folgende Programmzeile.

Wie nützlich das Überspringen einer folgenden Zeile ist, werden Sie gleich erfahren.

DEKREMENT UND SPRUNG WENN GLEICH

Jedesmal, wenn **DSE** ausgeführt wird, wird erst **nnnnn** um **yy** dekrementiert. Dann wird geprüft, ob **nnnnn** gleich (oder kleiner als) **xxx** ist. Wenn das der Fall ist, überspringt der HP-34C die nächste Programmzeile.

Wenn Sie also 100.01001 im I-Register gespeichert haben und die Anweisung **DSE** ausführen, würde der Rechner bei 100 anfangen zu zählen. Die Laufvariable würde verringert werden bis sie gleich (oder weniger als) 10 ist und bei jeder Ausführung der Anweisung würde sie um eins dekrementiert werden.

Inhalt des I-Registers = 100.01001

Folgendes geschieht, wenn **DSE** ausgeführt wird:

- Das Zählen der Laufvariablen beginnt bei 100
- Die Laufvariable wird um eins verringert
- Die Laufvariable wird mit 10 verglichen

Wenn die Schleife einmal durchlaufen wird, steht 99.01001 in I. Wenn die Schleife 10 mal durchlaufen wird, steht 90.01001 in I. Bei jeder Ausführung von **DSE** wird nach der Dekrementierung überprüft, ob die Laufvariable kleiner oder gleich 10 ist. Sobald sie 10 erreicht oder unterschreitet, überspringt der Rechner die folgende Programmzeile.

Beispiel: Das folgende Programm soll die Wirkung von **ISG** veranschaulichen. Es enthält eine Programmschleife, in deren Verlauf der augenblickliche Inhalt von I angezeigt wird. Er wird mit **ISG** gesteuert, wie oft die Schleife zu durchlaufen ist und welcher Wert zu quadrieren ist. Das Programm erzeugt die Quadrate ganzer, gerader Zahlen von 2 bis 50.



Tastenfolge

Schieben Sie den **PRGM** **RUN** -Schalter in Stellung **PRGM**.



Tastenfolge

f CLEAR PRGM
h LBL A
f FIX 5
2



Anzeige

000- Schieben Sie den **PRGM** **RUN** -Schalter in Stellung **PRGM**.

Anzeige

001-25,13,11
002-14,11, 5
003- 2

Die Programmärme
Die Schleifensteuerungszahl
(Der Anfangswert ist 2, die Schrittweite 2, der Endwert 50).



004- 73
005- 0
006- 5
007- 0
008- 0
009- 2
010-23,14,23

Die Schleifensteuerungszahl wird im I-Register gespeichert.
Anfang der Schleife.

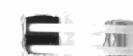


011-25,13, 1
012-24,14,23
013- 25 32
014- 25 14
015- 15 3
016- 25 74
017- 15 24

Zurückrufen der Zahl in I.
Der ganzzahlige Teil der Zahl.
Anzeige der Ganzzahl.



Die Zahl zum Quadrat erhoben:
Anzeige des Quadrats.
I wird um 2 erhöht;
Abfrage, ob die Laufvariable den Endwert 50 überschritten hat.



018- 22 1
019- 25 12

Wenn die Laufvariable größer als der Endwert ist, wird die nächste Zeile übersprungen (Rücksprung nach 1). Das Programm hält an.



Starten Sie jetzt das Programm.

Schieben Sie den **PRGM** **RUN** -Schalter in Stellung **RUN** und drücken Sie **A**.



Tastenfolge

A



Anzeige

2,00000
4,00000
4,00000
16,00000
50,00000
2.500.00000

Beim Ablauf des Programms zeigt der Rechner erst die zu quadrierende Zahl und dann die Zahl zum Quadrat an.
Wenn die Laufvariable 50 übersteigt, hält das Programm an.

Folgendes findet statt, wenn Sie das obenstehende Programm laufen lassen.

1. Mit **LBL A** beginnend wird die Zahl 2.05002 als Schleifensteuerungszahl im I-Register gespeichert. Die Zahl hat das übliche Format, d. h.

nnnnn	xxx	yy
(0000) 2	005	02
Laufvariable	Endwert	Schrittweite

2. Mit **LBL 1** beginnend wird folgendes ausgeführt:

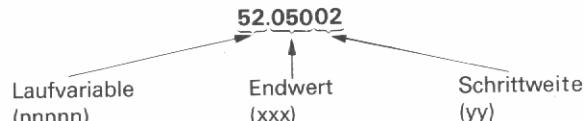
Nach dem 2 und 4 (2 zum Quadrat) angezeigt worden sind, wird die Laufvariable in I, 00002 (nnnnn) um die Schrittweite 02 (yy) inkrementiert. Die neue Zahl im I-Register ist

nnnnn	xxx	yy
(0000) 4	050	02
Laufvariable	Endwert	Schrittweite

Die neue Laufvariable wird mit dem Endwert (050) verglichen. Da sie den Endwert noch nicht überschritten hat, fährt der Rechner mit dem nächsten Programmschritt, GTO 1, fort und der Ablauf wird mit der neuen Zahl wiederholt.

3. Nachdem 25 gradzahlig (2 – 50) Zahlen angezeigt, quadriert und wieder angezeigt worden sind, überschreitet die Laufvariable schließlich den Endwert 50. Der Rechner überspringt jetzt einen Programmschritt hinter dem **GTO** in Zeile 17. Die **GTO 1** -Anweisung in Zeile 18 wird daher übergangen und die **RTN**-Anweisung in Zeile 19 wird ausgeführt. Der Rechner kehrt daher nach Zeile 000 zurück und hält dort an.

Drücken Sie **RCL f I**. Der in die Anzeige zurückgerufene Inhalt des I-Registers sollte wie folgt aussehen:



Jetzt wollen wir ein zweites Programm hinzufügen, das die **DSE** -Funktion Ihres HP-34C verwendet. Beachten Sie, daß hier wie im Falle von **ISG** die Steuerungszahl das Format nnnnn.xyyy hat. Allerdings wird hier die Laufvariable nicht inkrementiert sondern dekrementiert.



Beispiel: Die Insel Manhattan wurde im Jahre 1624 für 24 \$ verkauft. Das nachfolgende Programm berechnet, wie dieser Betrag von Jahr zu Jahr gewachsen wäre, wenn er statt dessen auf ein Konto eingezahlt worden wäre, das diese Einlage mit 6% p.a. verzinst hätte.

Als erstes wird die Anzahl der Jahre, über die Sie diese Entwicklung zu verfolgen wünschen, in das I-Register gespeichert. Dann wird die **DSE**-Anweisung dazu verwendet, die Anzahl der Schleifendurchläufe zu kontrollieren.



Um das Programm einzutasten, schieben Sie den **PRGM ■■■■■ RUN** -Schalter in Stellung PRGM.



Nach der Ausführung von **RTN** in Zeile 019 in dem vorangegangenen Beispiel ist der Rechner auf Zeile 000 zurückgesprungen. Um das folgende Programm anzuhängen, drücken Sie **h BST** (oder **GTO 0 019**) um wieder auf Zeile 019 zu gelangen. Da wir das eben gespeicherte Programm für weitere Zwecke erhalten wollen, beginnen wir das neue Programm bei Zeile 019.



Tastenfolge
h BST

Anzeige

019-

25 12

Letzte Zeile des vorherigen Programms.



020-25,13,12

Neue Programmarke.



021-14,11, 2

022-23,14,23

Die Schleifensteuerungszahl nnnnn.xyyy wird im I-Register gespeichert.



023-

1

021-

6

Das erste Jahr



025-

2

026-

4



027-

51

Das letzte Jahr



028-

23

0 Das letzte Jahr wird gespeichert

029-

2

030-

4

Der anfängliche Betrag



031-25,13, 2

Schleifenanfang

032-

1

033-

0

Berechnung des jährlichen Zuwachses

034-

6

035-

25 41



9 DSE

036- 15 23 Die Laufvariable nnnnn wird dekrementiert und mit dem Endwert xxx verglichen.



Die **DSE**-Anweisung bewirkt, daß der Inhalt vom **I**-Register um 1 dekrementiert wird. Bis der Inhalt von **I** Null ist, wird die Programmausführung bei **LBL** 2 fortgesetzt und die Schleife abermals durchlaufen.

GTO 2

037- 22 2 Wenn nnnnn > xxx, Sprung nach **LBL** 2.



Wenn der Inhalt von **I** Null ist (**nnnnn** = **xxx**), umgeht der Rechner die **GTO** 2-Anweisung in Zeile 37 und setzt die Programmausführung mit **RCL** 0 in Zeile 38 fort. Nacheinander erscheinen die letzte Jahreszahl und der Endbetrag,

RCL 0

038- 21 0 Wenn nnnnn ≤ xxx, Anzeige des
039- 25 74 letzten Jahres und des Endbetrages.
090- 21 Das Programm hält an.
041- 25 12



worauf das Programm anhält.

Schieben Sie den **PRGM RUN**-Schalter in Stellung RUN.

Tasten Sie jetzt die Anzahl der Jahre (die Schleifensteuerungszahl) ein, für die Sie die Berechnung des Betrages durchführen wollen. Drücken Sie **B** um den Eingabewert im **I**-Register zu speichern und das Programm auszuführen.



GÜLTIGKEITSGRENZEN VON ISG UND DSE



Beachten Sie, daß mit den Anweisungen **ISG** und **DSE** jede im HP-34C darstellbare Zahl inkrementiert oder dekrementiert werden kann. Wenn dies jedoch mit Laufvariablen vorgenommen wird, die mehr als 5 Vorkommastellen umfassen, wirkt das sich auf den Dezimalteil der Schleifensteuerungszahl aus.



Beispielsweise wird die Zahl 99950.50055 nach Inkrementierung durch **ISG** in 10000.5006 umgeändert. Die ursprüngliche Zahl wurde um 55 inkrementiert. Da aber die neue Zahl 100005.50055 nicht vollständig dargestellt werden kann, wird der Dezimalteil der Zahl gerundet. Die nächste Schrittweite ist jetzt 60 und nicht 55. Wenn die Zahl 999945.5006 erreicht ist, wird sie durch **ISG** in 1000 005.50 umgeändert, wobei der Dezimalteil der Zahl wieder gerundet wird. Jetzt existiert kein Wert für Schrittweite, so daß von jetzt ab um 1 und nicht 50 inkrementiert wird.



ÜBUNGSAUFGABE



1. Erstellen Sie ein Programm, das mit Hilfe der **ISG**-Anweisung von Null bis zu einer vorgegebenen Grenze hochzählt und dann unter Verwendung der **DSE**-Funktion wieder bis Null zählt.



Das folgende Flußdiagramm wird Ihnen die Programmierung erleichtern.



Tastenfolge Anzeige

5	5, 32,12	Schleifensteuerungszahl; nnnnn = 5,xxx = 000,yy = 00 (interne Voreinstellung 01). Nach 5 Jahren (also 1629) ist der Betrag auf \$ 32,12 angewachsen.
15	15, 57,52	Schleifensteuerungszahl; nnnnn = 15,xxx = 000,yy = 00 (interne Voreinstellung 01). Nach 15 Jahren (also 1639) ist der Betrag auf \$ 57,52 angewachsen.

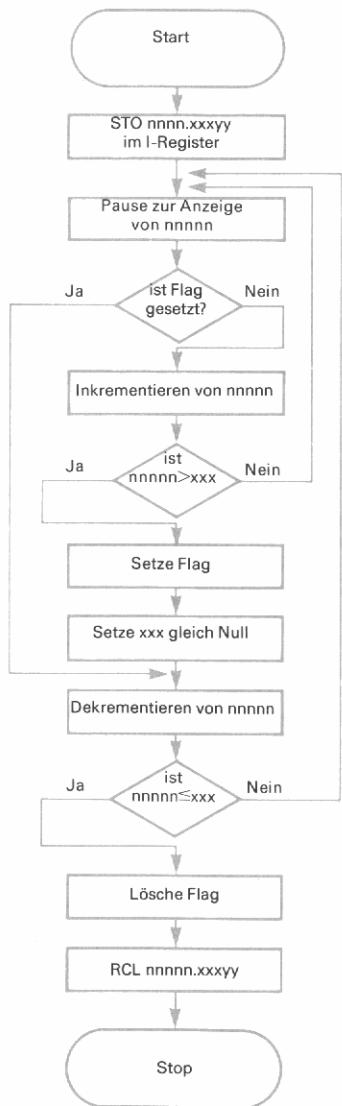
Wirkungsweise des Programms: Wenn Sie die Anzahl der Jahre eintasten, wird dieser Wert als Schleifensteuerungszahl nnnnn,xxxxyy im **I**-Register gespeichert.

nnnnn	xxx	yy
(0000) 5	000	00
Laufvariable	Endwert	Schrittweite

(Beachten Sie, daß Endwert und Schrittweite nicht eingegeben werden müssen, wenn sie Null bzw. Eins sind).

Die Schleifensteuerungszahl wird zum ersten Jahr addiert. Diese Summe ist das letzte Jahr und wird zur späteren Verwendung in **R₀** gespeichert. Dann wird der Anfangsbetrag eingetastet.

Bei jedem Schleifendurchlauf wird der Betrag um 6 % erhöht.



VERWENDUNG DES I-REGISTERS ZUR KONTROLLE DER ANZEIGE, DER SPEICHERREGISTER UND DER PROGRAMMSTEUERUNG.

Sie haben gesehen, wie die im I-Register gespeicherte Zahl **STO**, **X_I**, **ISG** und **DSE** verändert werden kann. Sie können aber auch den in I gespeicherten Wert zur Kontrolle der Anzeige, der Speicherregister und Programmsteuerung verwenden. Wir wollen uns erst einen Überblick über diese Operationen verschaffen und dann jede Funktion im einzelnen betrachten.



Die Anzahl der angezeigten Stellen nach dem Komma wird durch die Zahl im I-Register festgelegt.

Der Inhalt des angezeigten X-Registers wird mit dem Inhalt des verfügbaren Registers vertauscht, das durch den Absolutwert der Zahl im I-Register adressiert wird.

Der Inhalt des angezeigten I-Registers wird in das Register gespeichert, das durch den Absolutwert der augenblicklichen Zahl im I-Register adressiert wird.

Der Inhalt des Registers, das durch den Absolutwert der augenblicklichen Zahl im I-Register adressiert wird, wird zurückgerufen.

Eine Register-Arithmetik-Operation wird mit dem Register durchgeführt, das durch den Absolutwert der augenblicklichen Zahl im I-Register adressiert wird.

Die Ausführung des augenblicklichen Programms wird mit der nächsten durch die positive Zahl im I-Register angegebene Marke fortgesetzt.

Die Ausführung des augenblicklichen Programms wird mit der durch den Absolutwert der negativen Zahl im I-Register angegebenen belegten Zeilennummer fortgesetzt.

Die Ausführung des augenblicklichen Programms wird mit der nächsten durch die positive Zahl im I-Register angegebenen Marke fortgesetzt. Wie bei normalen Unterprogrammen, verzweigt die Programmausführung nach der auf die **GSB**-Anweisung folgende Zeile, wenn eine **RTN**-Anweisung ausgeführt wird.

GSB f l

Die Ausführung des augenblicklichen Programm wird mit der durch den Absolutwert der negativen Zahl im I-Register angegebenen belegten Zeilennummer fortgesetzt. Der weitere Ablauf entspricht dem normalen Unterprogramme.

Wenn das I-Register eine für die oben aufgeführten Operationen ungeeignete Zahl enthält, erscheint eine Fehlermeldung in der Anzeige. Beachten Sie außerdem, daß der Rechner zur Kontrolle der Anzeige, der Speicherregister und der Programmsteuerung nur den ganzzahligen Teil der im I-Register enthaltenen Zahl verwendet. Wenn beispielsweise 12,99041276 im I-Register steht, bleibt dieser Wert voll erhalten, wenn auch für die oben genannten Operationen nur die Zahl 12 verwendet wird.

Sicherlich erkennen Sie bereits jetzt die nahezu uneingeschränkten Verwendungsmöglichkeiten für das I-Register in Verbindung mit den genannten indirekt adressierbaren Operationen.

Wir wollen uns jetzt eingehender mit diesen Operationen befassen

STEUERUNG DES ANZEIGEFORMATS MIT DEM I-REGISTER

Sie können die im I-Register gespeicherte Zahl zusammen mit der Tastenfunktion **DSP 1** zur Steuerung der Anzahl von Dezimalstellen verwenden, mit denen der Rechner Zahlenwerte anzeigt. Wenn die Anweisung **H** **DSP 1** ausgeführt wird, runden der Rechner die Anzeige derart, daß die Zahl den angezeigten Dezimalstellen dem augenblicklich gespeicherten Wert im I-Register entspricht. (Dabei werden alle Zahlenwerte lediglich gerundet angezeigt; intern stellt der Rechner alle Daten mit voller Genauigkeit als 10stellige Mantisse mit 2stelligem Zehnerexponenten dar).

Die Tastenfolge **h** **DSP 1** ist insbesondere als Anweisung im Rahmen eines gespeicherten Programms von großem Wert, kann aber auch manuell über das Tastenfeld ausgeführt werden. Zum Beispiel:

Schieben Sie den PRGM ■■■ RUN -Schalter in Stellung RUN.

Tastenfolge	Anzeige	
CLX  FIX 1	0,0000	Löscht die Anzeige. Standardformat FIX 4 .
STO   9.123456789	0,0000 9,123456789	Null wird in das I-Register gespeichert.
h DSP I	9,	[FIX] -Formatanzeige wird durch den Inhalt des I-Registers gesteuert.
g ISG	9,	Inkrementiert den Inhalt des I-Registers auf 1.
h DSP I	9,1	
g ISG	9,1	
h DSP I	9,12	

Beispiel: Das nachfolgende Programm zeigt zu jedem möglichen Anzeigeformat je ein Beispiel an. Es verwendet eine Unterprogrammschleife mit `DSE` zur automatischen Änderung der Anzahl anzuseigender Dezimalstellen.

Zum Eintasten des Programms schieben Sie den PRGM ■■■■■ RUN -Schalter in Stellung PRGM.

Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL A	001-25,13,11
9	002- 9
STO f 1	003-23,11,23
h LBL 0	004-25,13, 0
h DSP 1	005- 25 11
RCL f 1	006-24,14,32
h PSE	007- 25 74
g DSE	008- 15 23
GTO 0	009- 22 0
g x>0	010- 15 51
GTO 0	011- 22 0
h RTN	012- 25 12

Lassen Sie den HP-34C jetzt ein Beispiel aller möglichen Anzeigeformate darstellen:

Schieben Sie den PRGM ■■■ RUN -Schalter in Stellung RUN.

Tastenfolge

A	9,0000000000
	8,0000000000
	7,00000000
	6,00000000
	5,00000000
	4,00000000
	3,00000000
	2,00000000
	1,00000000
	0,00000000
	0,00000000

Um das wissenschaftliche oder technische Format zu demonstrieren, ersetzen Sie die 9 in Zeile 02 durch eine 6 und schalten Sie den Rechner in den SCI- oder ENG-Modus, indem Sie **f SCI** oder **f ENG** und eine Zahl von 0 bis 7 drücken. Wenn Sie **f SCI** oder **f ENG** 8 oder 9 im PRGM-Modus drücken, ergibt sich automatisch ein **f SCI** 7 oder **f ENG** 7 im Programm-speicher.

Drücken Sie dann A wie in dem vorhergehenden Beispiel.

Schieben Sie den PRGM RUN -Schalter in Stellung PRGM.

Tastenfolge

GTO	002	002-	09
h	DEL	001-25,13,11	
6		002-	6

Schieben Sie den PRGM RUN -Schalter in Stellung RUN.

Tastenfolge

f SCI	4	0,0000	00	Standard SCI-Anzeigeformat
oder				
f ENG	4	0,0000	00	Standard ENG-Anzeigeformat

	A	6,00000000 0
		5,00000000 00
		4,00000000 00
		3,00000000 00
		2,00000000 00
		1,00000000 00
		0,00000000 00

Wenn eine beliebige Zahl kleiner als 0 im I-Register gespeichert ist, ergibt **h DSP 1** dieselbe Anzahl von Ziffern in der Anzeige die auf **h DSP 1** mit 0 im I-Register* erfolgt wäre. Wenn die Zahl im I-Register größer als 9 ist, ergibt **h DSP 1** dieselbe Anzahl von Ziffern in der Anzeige, die auf **h DSP 1** mit 9 im I-Register erfolgt wäre. Beachten Sie, daß im SCI- oder ENG-Anzeigeformat jede Zahl größer als 6 im I-Register eine Anzeige von maximal 6 Ziffern mit einem zweistelligen Exponenten rechts vom Dezimalteil ergibt. (Beachten Sie auch, daß **SCI** oder **ENG** 7 die Anzeige um eine Stelle mehr als **SCI** oder **ENG** 6 rundet.)

Drücken Sie jetzt die folgenden Tasten:

Tastenfolge	Anzeige
f FIX 4	0,0000
1.999999999	1,999999999

Standardanzeige FIX 4
Anzeige wird entsprechend der letzten Formatangabe gerundet. Nur der Ganzteil der Zahl im I-Register wird verwendet.

STO f 1	2,0000
h DSP 1	2,0
.9852	0,9852
STO f 1	1,0

Anzeige wird entsprechend der letzten Formatausgaben gerundet.
Ein Wert kleiner als 1 ergibt dieselbe Anzeige wie ein Wert 0.

* Nur während der Ausführung der **f y**-Funktion wird eine Zahl von -6 bis +9 im I-Register von **DSP 1** als automatischer Parameter bei **f y**-Berechnungen verwendet (mehr über **f y** in Abschnitt 9).

[h] DSP I	1,	
19	19,	
STO f i	19,	
[h] DSP I	19,00000000	Da zwei Ziffern links vom Dezimalpunkt stehen, ergibt ein Zahl größer als 9 dieselbe Anzeige wie ein Wert von 8 oder 9.
CHS STO f i	-19,00000000	Speichert eine negative Zahl im I-Register
[h] DSP I	-19,	Eine negative Zahl im I-Register ergibt dieselbe Anzeige, wie eine positive Zahl kleiner als 1.
f SCI 4	-1,9000	Standardanzeige SCI 4
1.1111119 ENTER	1,1111 00	
7 STO f i X:Y	1,1111 00	
[h] DSP I	1,111111 00	Die Anzeige wird auf 7 Stellen nach dem Komma gerundet.
6 STO f i X:Y	1,111111	
[h] DSP I	1,111112 00	Die Anzeige wird auf 6 Stellen nach dem Komma gerundet.

AUSTAUSCH DER INHALTE VON X UND (i).

Mit **X:(i)** können Sie den Inhalt des angezeigten X-Registers mit einem beliebigen verfügbaren Speicherregister austauschen, dessen Adresse durch den Absolutwert einer im I-Register gespeicherten Zahl n ($-20 \leq n \leq 20$) gegeben ist. Mit den Ganzzahlen 0 bis ± 9 werden die Primärspeicherregister R_0 bis R_9 adressiert. Mit den Ganzzahlen ± 10 bis ± 19 werden die Speicherregister R_{10} bis R_{19} adressiert. Wenn die Zahl ± 20 im I-Register steht, adressiert sich das I-Register mit **X:(i)** selbst.



Die folgende Tabelle macht diese Adressen etwas deutlicher.



Adresse

Adresse



R_0

R_{10}



R_1

R_{11}



R_2

R_{12}



R_3

R_{13}



R_4

R_{14}



R_5

R_{15}



R_6

R_{16}



R_7

R_{17}



R_8

R_{18}



R_9

R_{19}



I

20

Bevor Sie weitermachen, schalten Sie das Anzeigeformat auf FIX 4 und löschen Sie das angezeigte X-Register und alle Speicherregister.



Tastenfolge

Anzeige



f FIX 4 CLX

0,0000



f CLEAR REG

0,0000



Versuchen Sie in den folgenden Beispielen mit Hilfe von **X:(i)** die Zahl 1,234 in die Register R_3 , R_5 und I zu speichern.



Tastenfolge

Anzeige



3 f X:(i)

0,0000



1.2345 h X:(i)

0,0000



RCL 3

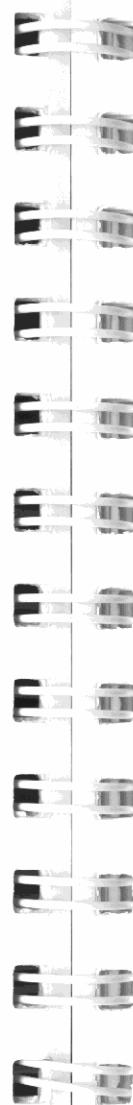
1,2345

Tauscht die Inhalte des angezeigten X-Registers und des I-Registers aus.

Tauscht die Inhalte des angezeigten X-Registers und R_3 unter Verwendung der im I-Register gespeicherten Adresse 3 aus.

Ruft den Inhalt von R_3 zurück.

Tastenfolge	Anzeige	
15	3,0000	Tauscht die Inhalte des angezeigten X-Registers und I aus.
	1,2345	Tauscht die Inhalte des angezeigten X-Registers und des Y-Registers aus.
	0,0000	Tauscht die Inhalte des angezeigten X-Registers und R ₅ unter Verwendung der im I-Register gespeicherten Adresse 15 aus.
5 	1,2345 1,2345 -15,3974 0,0000	Ruft den Inhalt von R ₅ zurück. Löscht den Inhalt aller Speicherregister. Tauscht die Inhalte des angezeigten X-Registers und I aus.
	1,2345	Tauscht die Inhalte des angezeigten X-Registers und des Y-Registers.
	0,0000	Tauscht die Inhalte des angezeigten X-Registers und R ₅ unter Verwendung des Absolutwertes der im I-Register gespeicherten Adresse -15,3974 aus.
20	1,2345 -15,3974	Tauscht die Inhalte des angezeigten X-Registers und I aus.
	1,2345	Tauscht die Inhalte des angezeigten X-Registers und des Y-Registers aus.
	20,0000	Tauscht die Inhalte des angezeigten X-Registers und I unter Verwendung der im I-Register gespeicherten Adresse aus. Ruft den Inhalt von I zurück. Löscht den Inhalt aller Speicherregister.
 	1,2345 1,2345	Löscht alle Stack-Register.
	0,0000	



INDIREKTES SPEICHERN UND ZURÜCKRUFEN VON DATEN.

Wie bei können Sie die im I-Register gespeicherte Zahl auch zur indirekten Adressierung der 21 Speicherplätze bei **STO** - und **RCL** -Operationen verwenden. Wenn Sie **STO** drücken, wird der Inhalt des angezeigten X-Registers in dasjenige Speicherregister geschrieben, dessen Adresse augenblicklich im I-Register steht. **RCL** adressiert die Speicherregister in gleicher Weise, ebenso die Operationen für die Speicherregister-Arithmetik **STO** , **STO** , **STO** und **STO** . (Wenn Sie die Wirkungsweise der normalen Register-Arithmetik-Operationen inzwischen vergessen haben, können Sie die Einzelheiten im Abschnitt 4 des "Allgemeinen Handbuchs für technisch-wissenschaftliche Rechner" nachschlagen.)

Wenn Sie **STO** , **RCL** oder eine der Register-Arithmetik-Operationen in Verbindung mit verwenden, kann das I-Register beliebige positive oder negative Werte von 0 bis 20 wie auch bei enthalten.

Sie können die Wirkungsweise von **STO** und **RCL** leicht erkennen, wenn sie diese Operationen vom Tastenfeld aus manuell in Verbindung mit dem I-Register zur Adressierung der verschiedenen Speicherregister ausführen. Überzeugen Sie sich davon, daß der **PRGM** -Schalter in Stellung RUN steht.

Tastenfolge Anzeige

	0,0000	Alle Speicherregister (einschließlich I-Register) werden gelöscht.
	0,0000	
5 STO 1.23 STO	5,0000 1,2300	Speichert 5 in das I-Register 1,23 wird in das Register gespeichert, dessen Adresse in I steht -d. h. in das Register R ₅ .
19 STO	19,0000	Die Zahl 19 wird in das Y-Register gespeichert.
85083 STO	85.083,0000	Dieser Wert wird in das Speicherregister geschrieben (R ₉), dessen Adresse der augenblicklichen Zahl (19) in I entspricht.

12 STO	12,0000	
77 EEX 43	77,	43 Speichert 12 in das I-Register.
STO	7,7000	44 Der Inhalt des I-Registers (12) bestimmt, in welches der Daten-Speicherregister der Wert $7,7 \times 10^{44}$ gespeichert wird (R_2).

Wenn Sie einen Wert aus einem der Speicherregister zurückrufen wollen, können Sie **RCL** gefolgt von der entsprechenden Ziffern- bzw. Buchstabentaste, verwenden. Sie können mit der entsprechenden Zahl im I-Register den Wert aus einem Daten-Speicherregister auch dadurch zurückrufen, daß Sie einfach (oder **RCL**) drücken. Zum Beispiel:

Tastenfolge	Anzeige	
RCL 5	1,2300	Der Inhalt von Register R_5 wird in das angezeigte X-Register zurückgerufen.
RCL	7,7000	44 Da das I-Register noch die Zahl 12 enthält, ruft diese Operation den Inhalt des durch I adressierten Registers R_2 in die Anzeige zurück.

Wenn der Inhalt des I-Registers geändert wird, ändert sich auch die Adresse des Registers, auf das sich die Operationen **STO** bzw. **RCL** beziehen. Zum Beispiel:

Tastenfolge	Anzeige	
19 STO	19,0000	Inhalt des Speicherregisters R_9 wird in das angezeigte X-Register zurückgerufen.
RCL	85,083,0000	

Tastenfolge	Anzeige	
5 STO	5,0000	Inhalt des Speicherregisters R_5 wird in das angezeigte X-Register zurückgerufen.
RCL	1,2300	

Mit dem Inhalt des I-Register werden auch die Register adressiert, auf die sich die folgenden Register-Arithmetik-Operationen beziehen: **STO** , **STO** , **STO** , **STO** .

Beachten Sie, daß es bei diesen vier Operationen nicht nötig ist, die Präfix-Taste zu verwenden.

Tastenfolge	Anzeige	
1 STO	1,0000	Die Zahl 1 wird zum Inhalt desjenigen Speicher-Registers (R_5) addiert, dessen Adresse augenblicklich in I steht.
RCL	2,2300	Zurückrufen der in R_5 gespeicherten Zahl.
2 STO	2,0000	Der Inhalt von R_5 wird mit 2 multipliziert.
RCL	4,4600	Zurückrufen des neuen Inhalts von R_5
CLX	0,0000	Löschen der Anzeige.
RCL 5	4,4600	Der Inhalt von R_5 wird direkt zurückgerufen.

Anmerkung: In der Programmierung kann die Speicherregister-Arithmetik in Verbindung mit den Registern R_0 bis R_9 sowohl für direkte als auch für indirekte Speicheroperationen verwendet werden. Dagegen kann für das I-Register und R_0 bis R_9 die Speicherregister-Arithmetik nur mit indirekten Speicheroperationen angewandt werden.

Die wirkungsvollste Anwendung des I-Registers ist natürlich die in Verbindung mit **STO** - und **RCL** -Operationen innerhalb eines Programms.

Beispiel: Das folgende Programm verwendet eine Schleife dazu, die Datenspeicher R_0 bis R_9 und R_0 bis R_9 mit ihren eigenen Adressen zu belegen. Die Ausführung des Programms wird während jedes Schleifendurchlaufs kurzzeitig unterbrochen, um den jeweiligen Inhalt des I-Registers anzuzeigen. Wenn I den Wert 20 erreicht, verzweigt die Programmausführung nach Zeile 000 und der Rechner hält an.

Zum Eintasten des Programms, schieben Sie den **PRGM** **RUN** -Schalter in Stellung PRGM.

Tastenfolge	Anzeige	
CLEAR PRGM	000-	
h LBL A	001-25,13,11	
0	002-	
1	003-	
9	004-	
	005-	
	73	
	0	Schleifensteuerungszahl
	1	
	9	

STO
h **LBL** 1
RCL
h **INT**
STO
h **PSE**

g **ISG**

GTO 1
h **RTN**

- | | |
|---------------------|-------------------------------------------------------------------------------|
| 006-23,14,23 | Speichert die Schleifensteuerungszahl |
| 007-25,13, 1 | Augenblicklicher Wert von I |
| 008-24,14,13 | Abspeichern in das durch I adressierte Speicherregister |
| 009- 25,32 | |
| 010-23,14,21 | |
| 011- 25 74 | Zeigt den augenblicklichen Wert von I an. |
| 012- 15 21 | Inhalt des I-Registers wird um 1 erhöht und mit dem Endwert (019) verglichen. |
| 013- 22 1 | Falls $I \leq 19$ wird die Schleife wiederholt |
| 014- 25 12 | Falls $I > 19$ verzweigt das Programm nach Zeile 000 und hält an. |

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung RUN. Nach dem Starten des Programms wird zunächst das I-Register mit Null besetzt. Dann wird im Rahmen einer Programmschleife jeweils der Inhalt des I-Registers (die Schleifensteuerungszahl) zurückgerufen und der ganzzahlige Teil dieser Zahl in das entsprechend adressierte Register geschrieben. Wenn I beispielsweise die Zahl 17,020 enthält, wird diese Zahl zurückgerufen und der ganzzahlige Teil, 17, in das durch die Zahl 17 adressierte indirekte Register (R_{17}) gespeichert.

Der Inhalt des I-Registers wird mit jedem Schleifendurchlauf um 1 verringert. Das Ergebnis dient dann einmal als zu speichernder Wert und zum andern als Adresse für die Anweisung **STO** . Wenn das I-Register schließlich 20 erreicht hat, verläßt der Rechner die Programmschleife und hält an. Zur Ausführung des Programms, schieben Sie den **PRGM** **RUN**-Schalter in Stellung RUN.

Tastenfolge	Anzeige
A	0,0000
	1,0000
	2,0000
.	
.	
19,0000	

Beachten Sie, daß der Inhalt des I-Registers Schritt für Schritt bis zu 20,0190 inkrementiert wurde.



Tastenfolge

RCL

Anzeige

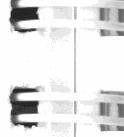
20.0190



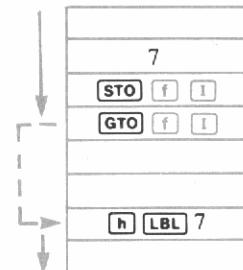
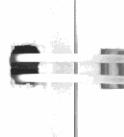
STEUERUNG VON PROGRAMMVERZWEIGUNGEN UND UNTERPROGRAMMEN MIT DEM I-REGISTER



In gleicher Weise, wie Sie die Verwendung der Speicherregister mit **STO** und **RCL** indirekt gesteuert haben, können Sie auch ganze Tastenfolgen, Unterprogramme und sogar vollständige Programme mit Hilfe des I-Registers indirekt adressieren.



Mit **GTO** können Sie die Sprungadresse für eine Programmverzweigung indirekt, d. h. von der in I gespeicherten Zahl abhängig, angeben. Wenn der Rechner bei der Ausführung eines Programms auf **GTO** trifft, verzweigt er zu der nächsten Marke **LBL**, deren Adresse augenblicklich im I-Register steht. Wenn Sie beispielsweise die Zahl 7 in das I-Register speichern und der Rechner dann **GTO** ausführt, springt er zu dem nächsten im Programmspeicher folgenden **LBL** 7, bevor er die Ausführung der gespeicherten Anweisungen fortsetzt.

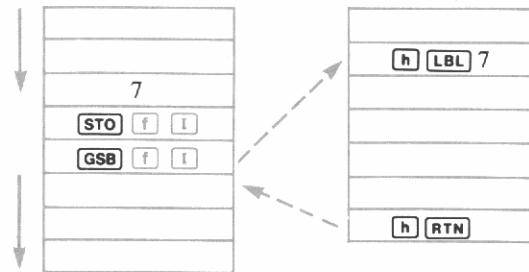


Sie können **GTO** natürlich auch auf dem Tastenfeld von Hand drücken, wenn Sie die Ausführung eines Programms ab einer bestimmten Marke starten wollen.



Sie können mit Hilfe des I-Registers auch Unterprogramme indirekt adressieren. Wenn der Rechner während der Ausführung eines Programms auf die Anweisung **GSB** trifft (oder diese Tasten auf dem Tastenfeld von Hand gedrückt werden), sucht er den Programmspeicher nach der mit dem I-Registerinhalt adressierten Marke ab und führt das nachfolgende Unterprogramm aus. Bei Erreichen des nächsten **RTN** erfolgt ein Rücksprung in das rufende Hauptprogramm. Dort setzt der Rechner die Programmausführung

mit der Anweisung fort, die auf den **GSB** [f1]-Befehl folgt. Wenn Sie beispielsweise die Zahl 7 in das I-Register speichern, veranlaßt **GSB** [f1] die Ausführung des Unterprogramms, das mit **LBL** 7 und **RTN** markiert ist.



Sie können **GSB**   auch von Hand über das Tastenfeld verwenden, wenn Sie die Ausführung eines Programms wünschen, das durch den Inhalt in **L** adressiert ist. In diesem Fall hält der Rechner nach Ausführung dieses Pro-grammteils an.

Die Adressierung erfolgt dabei in gleicher Weise über den Inhalt des I-Registers wie in Zusammenhang mit **GTO** . Wenn das I-Register Null oder eine positive Zahl von 1 bis 9 enthält, adressiert **GTO** die Marken **LBL 0** bis **LBL 9**. Durch eine positive 10 oder 11 wird **LBL A** **LBL B** adressiert. Die nachfolgende Tabelle faßt die verschiedenen Marken und ihre Adressen zusammen.

Inhalt des I-Registers

Das Programm verzweigt mit **GTO** f I
oder **GSB** f I nach:

- | | | | |
|----|-----|-----|---|
| 0 | [h] | LBL | C |
| 1 | [h] | LBL | 1 |
| 2 | [h] | LBL | 2 |
| 3 | [h] | LBL | 3 |
| 4 | [h] | LBL | 4 |
| 5 | [h] | LBL | 5 |
| 6 | [h] | LBL | 6 |
| 7 | [h] | LBL | 7 |
| 8 | [h] | LBL | 8 |
| 9 | [h] | LBL | 9 |
| 10 | [h] | LBL | A |
| 11 | [h] | LBL | B |

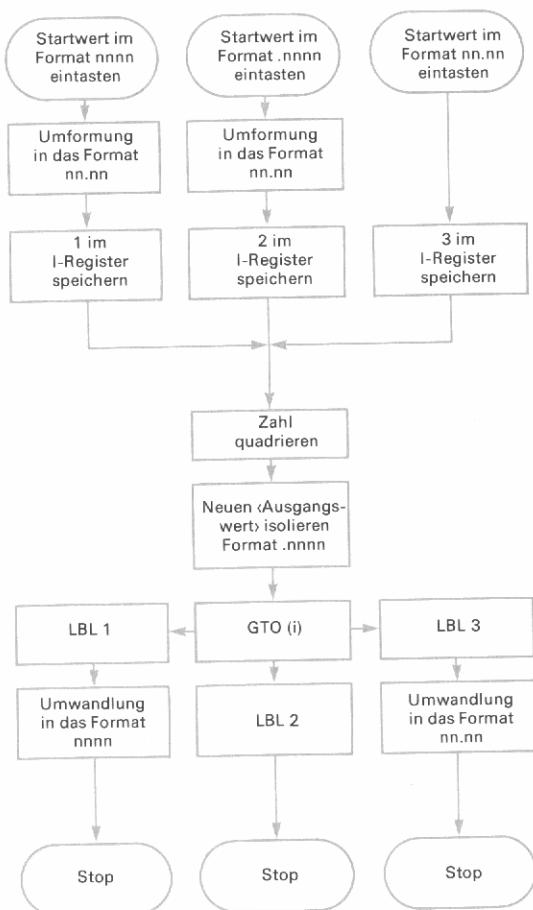


Beachten Sie, daß für die Adressierung von Marken die Zahl im I-Register gleich Null oder eine positive Zahl kleiner als 12 sein muß und daß der Rechner nur den ganzzahligen Teil der Zahl im I-Register in Verbindung mit indirekter Adressierung berücksichtigt.

Beispiel: Eine Möglichkeit zur Erzeugung von Pseudo-Zufallszahlen besteht darin, eine Zahl, (genannt „Startwert“) zu quadrieren, die mittleren Ziffern dieser Zahl herauszuziehen, diese Zahl erneut zu quadrieren usw. Wenn Sie beispielsweise die Zahl 5182 vorgeben, erhalten Sie als Quadrat 26853124. Der Zufallszahlen-Generator könnte dann die vier mittleren Ziffern, 8531, isolieren und diesen Wert erneut quadrieren. Wenn Sie dieses Verfahren im Rahmen einer Programmschleife fortsetzen, können Sie eine Vielzahl von „zufälligen“ Zahlenwerten erhalten. Wie schon angedeutet, sind die so erzeugten Zahlen keine wirklichen Zufallszahlen. Nachdem mehrere dieser „Pseudo-Zufallszahlen“ mit dieser Methode ermittelt worden sind, kann es durchaus sein, daß weitere Zahlen ein systematisches, nicht zufälliges Verhalten aufweisen. Die Kunst, wirkliche Zufallszahlen zu erzeugen, kann im Rahmen dieses Handbuchs nicht behandelt werden.

Das nachfolgende Programm veranschaulicht die Verwendung der **GTO** f
I-Anweisung. Sie können einen vierstelligen Startwert wahlweise in der Form nnnn, .nnnn oder nn.nn eintasten. Dieser Anfangswert wird anschließend im Hauptteil des Programms quadriert und das Quadrat dann beschnitten. Dann wird die sich ergebende 4stellige Zufallszahl in dem gleichen Format angezeigt, in dem Sie den Startwert eingegeben hatten: nnnn, nnnn oder nn nn.

Das Flußdiagramm zu diesem Programm kann zum Beispiel wie folgt aussehen:



Mit Hilfe der **GTO** [I] -Anweisung können Sie wählen, in welcher Weise die erzeugte Zufallszahl im Anschluß an den Hauptteil des Programms umgeformt werden soll.

Sie geben in Abhängigkeit vom Eingabeformat des Startwertes wahlweise 1, 2 oder 3 in das I-Register ein. Dementsprechend wählt das Programm im Anschluß an den Hauptteil die Form aus, in der das Ergebnis anzuseigen ist. Das hier angegebene Programm hält nach jeder Berechnung einer solchen Pseudo-Zufallszahl an. Es ist aber mit einfachen Mitteln möglich, diese Routine zu einer Programmschleife zu schließen, so daß das Verfahren mehrmals durchgeführt wird. Auf diese Weise kann die <Zufälligkeit> erhöht werden, mit der diese Werte aufeinanderfolgen.

Zum Eintasten des vollständigen Programms, schieben Sie den **PRGM** [RUN] -Schalter in Stellung **PRGM**.

Tastenfolge

	Anzeige	
f CLEAR [PRGM]	000-	
h [LBL] 4	001-25,13, 4	Wandelt nnnn in nn.nn um
EEX	002- 33	
2	003- 2	Schreibt 1 für das Ab- speichern nach I in das X-Register
+	004- 71	
1	005- 1	
[GTO] 7	006- 22 7	Wandelt .nnnn in nn.nn um
h [LBL] 5	007-25,13, 5	
EEX	008- 33	
2	009- 2	Schreibt 2 für das Ab- speichern nach I in das X-Register
x	010- 61	
2	011- 2	Schreibt 3 für das Ab- speichern nach I in das X-Register
[GTO] 7	012- 22 7	
h [LBL] 6	013-25,13, 6	
3	014- 3	
h [LBL] 7	015-25,13, 7	Speichert die Adresse für eine später folgende Operation in I
[STO] f [I]	016-23,14,23	
x:y	017- 21	Ruft nn.nn in das X-Register
g [x ²]	018- 15 3	Quadrat nn.nn

EEX	019-	33
2	020-	2
X	021-	61
h INT	022-	25 32
EEX	023-	33
4	024-	4
+/-	025-	71
h FRAC	026-	25 33
GTO f i	027-	22,14,23
h LBL 1	028-	25,13, 1
EEX	029-	33
4	030-	4
X	031-	61
f FIX 0	032-	14,11, 0
h RTN	033-	25 12
h LBL 2	034-	25,13, 2
f FIX 4	035-	14,11, 4
h RTN	036-	25 12
h LBL 3	037-	25,13, 3
EEX	038-	33
2	039-	2
X	040-	61
f FIX 2	041-	14,11, 2
h RTN	042-	25 12

In der vorstehenden Tastenfolge könnten bei Bedarf einige Programmschritte dadurch eingespart werden, daß die mehrfach vorkommende Tastenfolge EEX 2 in den Speicherzeilen 002 – 003, 008 – 009, 019 – 020 und 038 – 039 als Unterprogramm aufgerufen würde.

Da das Programm als Beispiel für die Verwendung von GTO f i gedacht ist, wurde hier aus Gründen der Übersichtlichkeit auf diese Möglichkeiten verzichtet.

Wenn Sie einen 4stelligen Startwert in einem der drei angegebenen Formate eintasten, wird anschließend eine der Adressen 1, 2 oder 3 in das X-Register geladen. Diese Adresse wird in Verbindung mit der GTO f i -Anweisung in Zeile 027 dazu verwendet, die Programmausführung zur entsprechenden Routine zu verzweigen, so daß die errechnete Pseudo-Zufallszahl im gleichen Format angezeigt wird. Führen Sie das Programm jetzt aus; verwenden Sie dabei die Startwerte 5182, 0,5182 und 51,82.

Zum Starten des Programms, schieben Sie den PRGM RUN -Schalter in Stellung RUN.		
Tastenfolge	Anzeige	
5182 GSB 4	8.531,	
,5182 GSB 5	0.8531	Die Zufallszahl wird im jeweiligen Format ausgegeben
51,82 GSB 6	85,31	
Das Programm erzeugt Zufallszahlen in dem gleichen Format, in dem Sie den entsprechenden Startwert eingetastet haben. Normale Zufallszahlen-Generatoren verbessern die Verteilung erzeugter Zufallszahlen dadurch, daß sie Programmschleifen dazu verwenden, die ‹Zufälligkeit› aufeinanderfolgender Werte zu verbessern. Sie können einfach die entsprechende Programm-taste mehrmals hintereinander drücken.		
Tastenfolge	Anzeige	
GSB 6	77,79	
GSB 6	51,28	
GSB 6	29,63	
Sie hätten bei diesem Programm mit geringfügigen Änderungen anstatt GTO f i auch GSB f i verwenden können.		
Übungsaufgabe		
Erstellen Sie unter Verwendung von ISG und STO f i ein Programm, mit dem Sie während aufeinanderfolgender Pausen eine Reihe von Daten eingeben können. Das Programm soll diese Werte in der Reihenfolge der Eingabe in die entsprechenden Register R ₀ bis R ₉ , R. ₀ bis R. ₉ und I speichern. Bei der Programmierung können Sie sich nach dem folgenden Flußdiagramm richten.		



STEUERUNG VON PROGRAMMVERZWEIGUNGEN UND UNTERPROGRAMMEN DURCH ADRESSIERUNG DER ZEILENNUMMER

Wie Sie wissen, wird die Programmausführung nach **GTO f l** oder **GSB f l** so lange unterbrochen, bis der Rechner beim Absuchen des Programmspeichers die Marke findet, die durch die positive Zahl im I-Register angegeben ist. Dann wird die Programmausführung fortgesetzt.

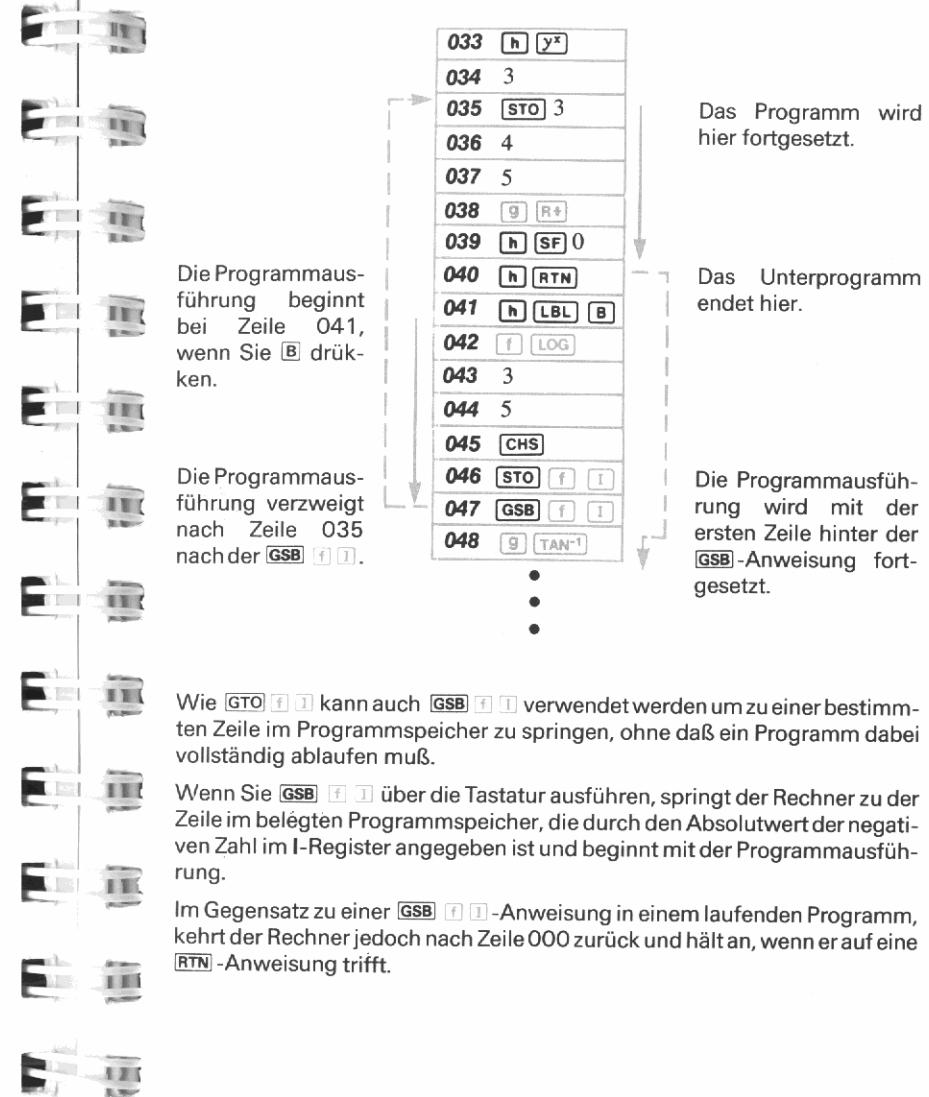
Ist dagegen eine negative Zahl im I-Register gespeichert, sucht der Rechner bei Ausführung von **GTO f l** oder **GSB f l** den Programmspeicher nicht nach einer Marke ab. Die Programmausführung wird statt dessen mit der Zeile im belegten Speicher fortgesetzt, die durch den Absolutwert der Zahl im I-Register angegeben ist. Sie können das Programm auf diese Weise zu jeder beliebigen Position im Programmspeicher übertragen und Programmverzweigungen auch dann noch vorsehen, wenn bereits alle Marken für andere Zwecke verwendet wurden oder wenn Sie nur einen bestimmten Teil eines Programms oder Unterprogramms ausführen wollen.

Sehen Sie sich zum Beispiel den nachstehenden Programmauszug an. Es wird angenommen, daß -35 in das I-Register gespeichert wurde. Wenn jetzt, in Zeile 047, **GTO f l** ausgeführt wird, springt der Rechner sofort zur Zeile 035, wo das Programm fortgesetzt wird.

033 - h y^x	
034 - 3	
035 - STO 3	
036 - 4	
037 - 5	
038 - g R↓	
039 - h SF 0	
040 - h RTN	
041 - h LBL B	
042 - f LOG	
043 - 3	
044 - 5	
045 - CHS	
046 - STO f l	
047 - GTO f l	
048 - g TAN⁻¹	

Der Rechner setzt die Ausführung des Programms nach Befolgen des **GTO f 1**-Befehls so lange fort, bis er auf die nächste **RTN** - oder **RVS**-Anweisung trifft, worauf er anhält. Der Rechner würde nach Drücken der Taste **B** im vorstehenden Programmbeispiel die Anweisungen in den Zeilen 041 bis 047 nacheinander ausführen. Dann würde er im Programmspeicher zurückspringen und die Programmausführung bei Zeile 035 fortsetzen. Anschließend würde er mit den Programmschritten 036, 037 usw. fortfahren, bis er die **RTN**-Anweisung in Zeile 040 erreicht. An dieser Stelle würde das Programm anhalten. Beachten Sie, daß die Ausführung einer **GTO f 1**-Anweisung vom Tastenfeld aus auch einen Sprung zu der angegebenen Zeile zur Folge hätte. Die Programmausführung wird allerdings nicht fortgesetzt, sondern der Rechner hält an dieser Stelle an. Die Anweisung **GSB f 1** bewirkt mit negativen Zahlen mit I-Register ebenfalls einen Sprung zu der Zeile im Programmspeicher, die durch den Absolutwert der negativen Zahl in I angegeben ist. Die darauffolgenden Anweisungen werden aber jetzt als Unterprogramm ausgeführt. Wenn der Rechner die nächste **RTN**-Anweisung erreicht, hält er nicht an, sondern setzt die Ausführung des Programms mit der Anweisung fort, die auf den **GSB f 1**-Befehl folgt (wie bei der normalen Ausführung von Unterprogrammen).

Der folgende Programmausschnitt veranschaulicht die Wirkung von **GSB f 1**. Wenn Sie **B** drücken, wird die Zahl –35 in das I-Register gespeichert. Wenn die Anweisung **GSB f 1** in Zeile 47 ausgeführt wird, springt das Programm zurück nach Zeile 035 und wird fortgesetzt. Wenn anschließend der **RTN**-Befehl (Rücksprungbefehl) in Zeile 040 erreicht wird, setzt der Rechner die Ausführung des Programms mit der Anweisung in Zeile 48 fort.



ABSCHNITT 8: BESTIMMUNG DER LÖSUNGEN EINER GLEICHUNG

In vielen Anwendungen werden Sie eine Gleichung der Form
 $f(x) = 0^*$
 zu lösen haben.

Es gilt hier reelle Werte von x zu finden, die der Gleichung genügen. Ein solcher Wert wird als Lösung der Gleichung $f(x) = 0$ und als Nullstelle der Funktion $f(x)$ bezeichnet.

In vielen Fällen können die Nullstellen einer Gleichung mittels bekannter mathematischer Formeln analytisch bestimmt werden. Das ist jedoch nicht immer möglich. Wenn analytische Methoden nicht ausreichen, werden numerische Verfahren zur Bestimmung der Nullstellen einer Funktion verwendet.

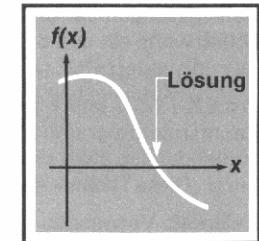
Wenn Sie die **SOLVE**-Taste Ihres HP-34C verwenden, machen Sie sich ein fortgeschrittenes numerisches Verfahren zunutze, mit dem Sie schnell und einfach die Wurzeln einer großen Anzahl unterschiedlicher Gleichungen berechnen können.

VERWENDUNG VON **SOLVE**

Folgende Regeln gelten für die Verwendung von **SOLVE**:

1. Tasten Sie ein Unterprogramm zur Berechnung der gleich Null zu setzenden Funktion $f(x)$ ein. Dieses Unterprogramm muß mit der Anweisung **[LBL]** gefolgt von **0, 1, 2, 3**, oder **[A]** oder **[B]** anfangen. Nach Ausführung des Unterprogramms muß der Wert von $f(x)$ im X-Register stehen.

*An sich kann jede Gleichung mit einer Veränderlichen in dieser Form ausgedrückt werden. So können beispielsweise die Gleichungen $f(x) = a$ und $f(x) = g(x)$ als $f(x)-a = 0$ bzw. $f(x)-g(x) = 0$ ausgedrückt werden.



2. Geben Sie zwei Anfangswerte der gewünschten Lösung in die X- und Y-Register ein. Diese Schätzwerte teilen den Rechner lediglich in etwa den Bereich von x mit, in dem er anfänglich eine Lösung der Gleichung $f(x) = 0$ suchen soll.

3. Drücken Sie gefolgt von der Marke mit der Ihr Unterprogramm beginnt.

Der Rechner sucht darauf nach der gewünschten Nullstelle der Funktion und zeigt das Ergebnis an. Wenn die Funktion, die Sie untersuchen, mehr als eine Nullstelle besitzt, wird die Routine dennoch beendet, sobald die erste Nullstelle gefunden ist. Um weitere Lösungen zu bestimmen, geben Sie neue Schätzwerte ein und drücken nochmal .

Unmittelbar bevor Ihr Unterprogramm aufruft, wird ein Wert von x in die X-, Y-, Z- und T-Register des Stacks geladen. Mit diesem Wert wird in Ihrem Unterprogramm $f(x)$ berechnet. Da der x-Wert in jedem Stack-Register vorhanden ist, steht diese Zahl Ihrem Unterprogramm jederzeit zur Verfügung. (Diese Technik ist auf Seite 76 ausführlich beschrieben.)

Beispiel: Verwenden Sie um die Werte von x zu bestimmen, für die $f(x) = x^2 - 3x - 10 = 0$ gilt.

Wenn Sie das Horner-Schema (siehe Seite 79) verwenden, können Sie die Funktion $f(x)$ auf folgende Weise ausdrücken:

$$f(x) = (x-3)x-10$$

Schieben Sie den -Schalter in Stellung RUN und tasten Sie das Unterprogramm zur Berechnung von f(x) ein.

**Tastenfolge**

CLEAR
 LBL 0
3

1
0

Anzeige

000-	Löschen des Programmspeichers.
001-25,13, 0	Die -Anweisung steht am Anfang
002- 3	des Unterprogrammes.
003- 41	
004- 61	
005- 1	
006- 0	
007- 41	
008- 25 12	



Schieben Sie jetzt den -Schalter in Stellung RUN. Geben Sie zwei Schätzwerte in die X- und Y-Register ein. Versuchen Sie es mit den Werten 0 und 10, um eine eventuelle positive Wurzel zu ermitteln.

**Tastenfolge**

0
10

Anzeige *

0,0000	Anfangswerte
10,	



Jetzt können Sie die gewünschte Wurzel ermitteln, indem Sie 0 drücken. Das Ergebnis erscheint jedoch nicht sofort in der Anzeige. Der HP-34C verwendet einen iterativen Algorithmus** zur Berechnung der Wurzel. Der Algorithmus bildet wiederholt Momentwerte der Funktion die zur Analyse Ihrer Funktion benötigt werden. Der Vorgang kann sich zwölfmal oder häufiger wiederholen, wobei jedesmal das von Ihnen programmierte Unterprogramm durchlaufen wird. Die Berechnung einer Wurzel dauert im allgemeinen 30 Sekunden bis 2 Minuten, doch manchmal reicht auch diese Zeit nicht aus.



Drücken Sie 0 und lassen Sie sich eine der leistungsfähigsten Eigenschaften Ihres HP-34C demonstrieren:

**Tastenfolge**

SOLVE 0

Anzeige

5,000	Die gewünschte Lösung
-------	-----------------------



* Drücken Sie 4 damit die Anzeige mit der in diesem Beispiel verwendeten Anzeige übereinstimmt. Das Anzeigeformat beeinflusst in keiner Weise die -Funktion.



** Ein Algorithmus ist ein schrittweises Verfahren, das zur Lösung mathematischer Probleme angewandt werden kann. Ein iterativer Algorithmus enthält einen Teil, der zur Lösung des Problems wiederholt ausgeführt werden kann.

Nachdem die Wurzel ermittelt und das Ergebnis angezeigt ist, können Sie im Stack überprüfen, ob der angezeigte Wert tatsächlich eine Nullstelle von $f(x)$ ist.

Sie wissen schon, daß im angezeigten X-Register die gewünschte Lösung steht. Das Y-Register enthält den vorletzten Schätzwert der Wurzel, der sehr nahe der angezeigten Lösung im X-Register liegen sollte. Das Z-Register enthält den Wert der Funktion, wenn für x die errechnete Wurzel eingesetzt wird.

Tastenfolge Anzeige

5,0000

Ein vorheriger Schätzwert der Nullstelle.

0,0000

Wert der Funktion, wenn für x die Nullstelle eingesetzt wird.
Hier gilt $f(x) = 0$.

Quadratische Gleichungen, wie die in unserem Beispiel, besitzen zwei Nullstellen. Mit zwei neuen Anfangswerten können Sie die Lage der zweiten Nullstelle ermitteln. Versuchen Sie es mit Schätzwerten von 0 und -10, um eine negative Wurzel zu finden.

Tastenfolge Anzeige

0

0,0000

Die anfänglichen Schätzwerte.

10

-10,

Die zweite Wurzel.

-2,0000

Ein vorheriger Schätzwert der zweiten Wurzel.

0,0000

Der Wert der Funktion wenn für x die Wurzel eingesetzt wird.



Sie haben jetzt die zwei Nullstellen der Gleichung $f(x) = 0$ gefunden. Beachten Sie, daß es für diese quadratische Gleichung auch eine analytische Lösung gibt – die Ergebnisse wären allerdings die gleichen gewesen, die Sie mit bekommen haben.



Die Anwendbarkeit und Leistungsfähigkeit der -Funktion zeigt sich erst richtig, wenn Sie die Nullstellen einer Gleichung bestimmen wollen, für die es keine analytische Lösung gibt.



Beispiel: Lokalmeister Max Zimmermann wirft seinen Hammer mit einer Geschwindigkeit von 50 Meter/Sekunde in die Luft. Wenn die Höhe des Hammers über dem Erdboden durch den Ausdruck



$$h = 5000 (1 - e^{-t/20}) - 200t$$



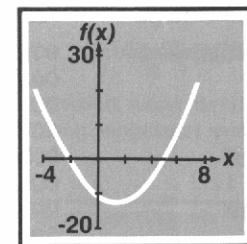
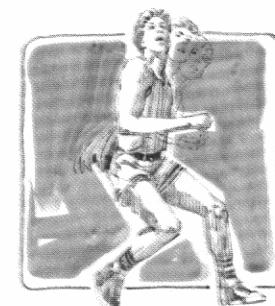
gegeben ist, wie lange dauert es bis der Hammer wieder zur Erde fällt? In dieser Gleichung bedeutet h die Höhe in Metern und t die Zeit in Sekunden.



Lösung: Das gewünschte Ergebnis ist der positive Wert von t für den $h = 0$ gilt.



Schieben Sie den -Schalter in Stellung PRGM und tasten Sie das Unterprogramm zur Berechnung der Höhe ein.

Der Graph $f(x) = x^2 - 3x - 10$ 

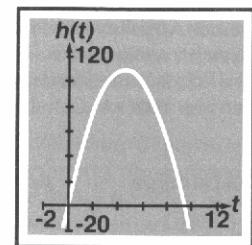
Tastenfolge	Anzeige	
[h] [LBL] [A]	001-25,13,11	Das Unterprogramm beginnt mit einer [LBL]-Anweisung.
2	002-	2
0	003-	0
÷	004-	71
[CHS]	005-	32
[g] [e^x]	006-	15 1
[CHS]	007-	32
1	008-	1
+	009-	51
5	010-	5
0	011-	0
0	012-	0
0	013-	0
[x]	014-	61
[xy]	015-	21 Der t-Wert wird in das X-Register gespeichert.
2	016-	2
0	017-	0
0	018-	0
[x]	019-	61
-	020-	41
[h] [RTN]	021-	25 12

Schieben Sie jetzt den PRGM ■ RUN -Schalter in Stellung RUN. Tasten Sie darauf die zwei anfänglichen Schätzwerte für die Zeit (in diesem Fall 5 und 6 Sekunden) ein und führen Sie SOLVE aus.

Tastenfolge	Anzeige	
5 [ENTER]	5,0000	Die anfänglichen Schätzwerte.
6,		
[f] SOLVE [A]	9,2843	Die gewünschte Lösung.

Überprüfen Sie die Lösung, indem Sie den Inhalt der Y- und Z-Register untersuchen.

Tastenfolge	Anzeige	
[g] [R ^f]	9,2843	Ein vorheriger Schätzwert der Lösung.
[g] [R ^f]	0,0000	Der Wert der Funktion, wenn für t die gefundene Lösung eingesetzt wird. Hier gilt h = 0.
Der Hammer fällt nach 9.2843 Sekunden wieder zu Boden – ein gewaltiger Wurf.	$h(t)$	



WENN KEINE LÖSUNG GEFUNDEN WIRD

Sie haben gesehen, wie die **SOLVE**-Funktion die Lösung einer Gleichung der Form $f(x) = 0$ berechnet und anzeigt. Es ist jedoch möglich, daß es für die Lösung einer Gleichung keine reellen Nullstellen gibt (d. h. es gibt keinen reellen Wert x der der Gleichung genügt). In einem solchen Fall kann der Rechner keine Lösung finden und meldet sich statt dessen mit der Fehleranzeige **ERROR 6**.

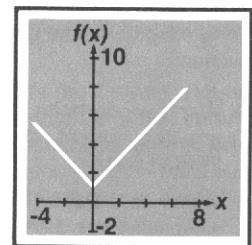
Beispiel: Für die Gleichung

$$|x| = -1$$

 gibt es keine reelle Lösung, da der Absolutwert der Funktion niemals negativ sein kann. Wir schreiben die Gleichung erst in der gewünschten Form

$$|x| + 1 = 0$$

und versuchen dann mit Hilfe von **SOLVE** eine Lösung zu finden. Schieben Sie den **PRGM** -Schalter in Stellung **RUN** und tasten Sie das Unterprogramm zur Berechnung der Funktion wie üblich ein.



Tastenfolge**h LBL 1****Anzeige****001–25,13, 1** Das Unterprogramm beginnt mit einer **LBL**-Anweisung.**h ABS****002– 25 34****1****003– 1****+****004– 51****h RTN****005– 25 12**

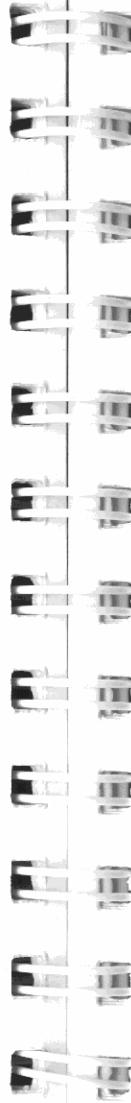
Da der Absolutwert einer Funktion für Argumente nahe Null ein Minimum erreicht, wollen wir die ersten Schätzwerte um Null wählen, beispielsweise 1 und -1. Dann versuchen wir eine Nullstelle zu ermitteln. Erst schieben Sie den **PRGM**  **RUN**-Schalter in Stellung **RUN**:

Tastenfolge**1 ENTER****Anzeige****1,0000** Die anfänglichen Schätzwerte.**1 CHS****-1,** Error 6 Diese Anzeige bedeutet, daß keine Lösung gefunden wurde.**f SOLVE 1**

Wie Sie sehen, hat der HP-34C die Suche nach einer Lösung der Gleichung $f(x) = 0$ abgebrochen, sobald er feststellte, daß wenigstens in der Nähe des anfänglich angegebenen Bereichs keine Wurzel existierte. Die Meldung **ERROR 6** bedeutet nicht, daß eine illegale Operation ausgeführt wurde; vielmehr teilt der Rechner Ihnen mit, daß **SOLVE** keine Wurzel in dem von Ihnen angegebenen Bereich finden konnte.

Wenn der HP-34C die Suche nach Wurzel abbricht und eine Fehlermeldung ausgibt, ist die Ursache bei einer der folgenden vier Bedingungen zu suchen:

- Wenn wiederholte Iterationen einen konstanten Wert für die Funktion ergeben, bricht die Funktionsausführung mit der Fehlermeldung **ERROR 6** ab.
- Wenn zahlreiche Stützwerte erkennen lassen, daß das Minimum des Absolutwertes der Funktion in dem untersuchten Bereich nicht gleich Null ist, wird die Ausführung mit der Anzeige **ERROR 6** unterbrochen.
- Wenn innerhalb des Unterprogramms ein unerlaubtes Argument in einer mathematischen Operation verwendet wird, wird die Ausführung mit der Anzeige **ERROR 0** unterbrochen.



- Wenn der Absolutwert eines Ergebnisses einer Rechnung größer als $9.999999999 \times 10^{99}$ ist, wird die Ausführung unterbrochen. Der Rechner zeigt in jeder Position der Anzeige eine 9 und das entsprechende Vorzeichen (oder Error 1, wenn der Überlauf in einem Speicherregister aufgetreten ist) an.

Im Falle eines konstanten Funktionswertes gibt es keine Anzeichen dafür, daß die Funktion gegen Null geht. Das kann eintreten, wenn die 10 wesentlichen Ziffern eines Funktionswertes konstant sind (wenn die Kurve einer Funktion eine horizontale Asymptote ungleich Null bildet) oder wenn die Funktion im Vergleich zu dem durch die x-Werte angegebenen Bereich lokal einen längeren, gestreckten Verlauf aufweist.

Wenn der gefundene Absolutwert einer Funktion ein Minimum ungleich Null erreicht, hat die Routine eine logische Folge von stetig abnehmenden Absolutwerten der Funktion gebildet. Es wurde jedoch kein Wert von x gefunden für den der Graph der Funktion die x-Achse berührt oder geschnitten hat.

Die beiden letzten Fälle weisen eher auf mögliche Unzulänglichkeiten des Unterprogramms als auf bei der Anwendung der Lösungsroutine zu beachtende Einschränkungen hin. Unerlaubte Operationen können durch Angabe von anfänglichen Schätzwerten vermieden werden, die den Suchvorgang in einem Bereich konzentrieren, in dem ein Ergebnis dieser Art nicht vorkommen kann.

Da die **SOLVE**-Routine aber alles andere als starr ist, wird die Funktion u. U. innerhalb eines großen Bereichs untersucht. Es ist ein guter Brauch, im Unterprogramm eventuelle unerlaubte Argumente vor einer Operation abzufragen und entsprechend zu korrigieren (indem Sie beispielsweise **ABS** vor **\sqrt{x}** einfügen). Es kann auch nützlich sein, Variable maßstäblich zu verändern um große Zahlen zu vermeiden.

Die erfolgreiche Auffindung einer Nullstelle mit Hilfe der **SOLVE**-Routine hängt maßgeblich von der zu untersuchenden Funktion und den anfänglichen Schätzwerten ab. Das bloße Vorhandensein einer Nullstelle ist keine Garantie, daß Sie mit einer ungenauen Anwendung der **SOLVE**-Taste auch gefunden wird. Wenn die Funktion eine horizontale Asymptote ungleich Null aufweist oder ein lokales Minimum ungleich Null besitzt, kann die **SOLVE**-Routine eine Lösung der Gleichung $f(x) = 0$ nur dann finden, wenn die Anfangswerte den Suchvorgang nicht auf diese unergiebigen Bereiche beschränken – vorrausgesetzt, daß eine Wurzel überhaupt existiert.

WAHL DER ANFÄNGLICHEN SCHÄTZWERTE

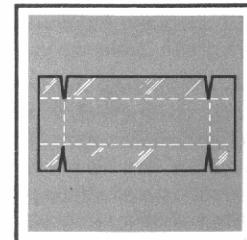
Wenn Sie mit Hilfe von **SOLVE** die Lösung einer Gleichung finden wollen, wählen Sie mit den beiden anfänglichen Schätzwerten die Variable x mit der die Routine den Suchvorgang beginnt. Im allgemeinen wird damit die Wahrscheinlichkeit, daß Sie eine gesuchte Wurzel auch finden, durch Ihre Kenntnis der zur Analyse vorliegenden Funktion bestimmt. Die Berechnung einer Wurzel wird durch realistische und intelligente Schätzwerte wesentlich erleichtert.

Die anfänglichen Schätzwerte können auf unterschiedliche Weise gewählt werden:

Wenn die Variable x auf einen Bereich beschränkt ist, in dem eine Lösung denkbar ist, sollten die anfänglichen Schätzwerte sinnvollerweise in diesem Bereich liegen. Häufig hat eine Gleichung, die ein reelles Problem beschreibt, nicht nur die gewünschte Lösung, sondern auch weitere Lösungen, die jedoch keine physikalische Bedeutung haben. Dies trifft dann zu, wenn die in Frage stehende Funktion nur innerhalb gewisser Grenzen definiert ist. Diese Einschränkung sollten Sie erkennen, damit Sie die Resultate entsprechend interpretieren können.

Wenn Sie wissen, wie sich die Funktion $f(x)$ für unterschiedliche Werte von x verhält, sind Sie in der Lage, Anfangswerte in der näheren Umgebung einer Nullstelle der Funktion anzugeben. Außerdem können Sie auch die Bereiche von x vermeiden, in denen die Funktion entweder konstant verläuft oder einem Minimalwert ungleich Null aufweist.

Beispiel: Aus einem Eisenblech, 8 dm lang und 4 dm breit, soll ein offener Behälter mit einem Volumen von 7,5 dm³ geformt werden. Wie muß das Blech abgekantet werden? (Ein hoher Behälter ist einem niedrigen vorzuziehen.)



Lösung: Gesucht wird die Höhe des Behälters (d. h. die Länge des Blechs, die entlang der vier Seiten umgebogen werden muß), die das gewünschte Volumen ergibt. Wenn x die Höhe ist (die Länge des umzulegenden Blechs) dann ist die Länge des Blechs $(8 - 2x)$ und die Breite $(4 - 2x)$.

Volumen wird durch die Formel

$$V = (8 - 2x)(4 - 2x)x$$

gegeben.

Zerlegt man diesen Ausdruck und wendet dann das Horner-Schema (Seite 79) an, ergibt sich die Formel

$$V = 4(x - 6)x + 8x$$

Da $V = 7,5$ vorgegeben ist, benötigen wir Werte von x für die

$$f(x) = 4((x - 6)x + 8)x - 7,5 = 0$$

ist.

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung **PRGM** und tasten Sie das Unterprogramm zur Berechnung von $f(x)$ ein.

Tastenfolge Anzeige

h LBL 3 001-25,13, 3

Das Unterprogramm beginnt mit einer **LBL**-Anweisung.

6 002- 6

– 003- 41

× 004- 61

8 005- 8

+ 006- 51

× 007- 61

4 008- 4

× 009- 61

7 010- 7

• 011- 73

5 012- 5

– 013- 41

h RTN 014- 25 12

Es ist zu erwarten, daß entweder ein hoher, schmaler oder ein kurzer, flacher Behälter das gewünschte Volumen hat. Da ein hoher Behälter vorgezogen wird, beginnen wir mit größeren Schätzwerten für die Höhe. Eine Höhe von mehr als 2 dm ist physikalisch jedoch nicht möglich (weil das Blech nur eine Breite von 4 dm hat).

Anfangswerte von 1 und 2 dm sind daher angemessen.

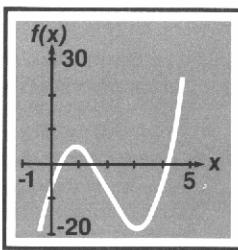
Schieben Sie den PRGM ■■■■■ RUN -Schalter in Stellung RUN und ermitteln Sie die gewünschte Höhe.

Tastenfolge	Anzeige	
1 [ENTER]	1,0000	Anfängliche Schätzwerte
2	2,	
[f] [SOLVE] 3	1,5000	Die gewünschte Höhe
[g] [R ²]	1,5000	Der vorletzte Schätzwert
[g] [R ²]	0,0000	Der Wert von f(x) für x = 1,5

Mit der Höhe von 1,5 dm ergibt sich ein Behälter 5 dm lang, 1 dm breit und 1,5 dm hoch.

Wenn Sie die obere Beschränkung für die Höhe ignorieren und anfängliche Schätz-werte von 3 und 4 dm (also weniger als die Breite des Blechs) verwenden, erhalten Sie eine Höhe von 4,2026 dm – ein offensichtlich sinnloses Ergebnis. Wenn Sie dagegen Schätz-werte von 0 und 1 dm verwenden, beträgt die Höhe 0,2974 dm – ein Behälter mit einer unerwünschten flachen Form.

Der Graph von $f(x) = (8-2x)(4-2x)x-7,5$



Wenn Sie das Verhalten der Funktion untersuchen wollen, können Sie ganz einfach die als Unterprogramm enthaltene Funktion für einige Werte von x berechnen. Dazu tasten Sie den Wert von x in das X-Register und drücken dann [ENTER] [ENTER] [ENTER], um den Wert auf den Stack zu laden. Den Funktionswert berechnen Sie sodann, indem Sie [A], [B] oder [GSB] gefolgt von der der Marke entsprechenden Ziffer drücken.

Die Ergebniswerte können als Kurve graphisch dargestellt werden. Dieser Vorgang ist dann besonders nützlich, wenn Ihnen über den Verlauf der Funktion nichts bekannt ist. Eine anscheinend einfache Funktion mag als Kurve relativ große Variationen aufweisen, die Sie gar nicht erwartet haben. Eine Nullstelle in der Umgebung einer solchen lokalen Variation ist vielleicht schwer zu finden, wenn die anfänglichen Schätz-werte nicht in der Nähe der Nullstellen liegen.

Wenn Sie sich keine Vorstellung hinsichtlich der Art der Funktion oder der Lage der von Ihnen gewünschten Nullstelle machen können, bleibt Ihnen dennoch die empirisch-praktische Methode, zu einem Ergebnis zu gelangen. Die erfolgreiche Lösung hängt zum Teil von der Funktion selber ab. Die empirisch-praktische Methode ist häufig – wenn auch nicht immer – erfolgreich.

- Wenn Sie zwei verhältnismäßige große negative und positive Anfangswerte angeben und die Funktion keine horizontale Asymptote hat, ermittelt die Routine eine Nullstelle, die entweder die größte positive oder die größte negative Nullstelle sein kann (wenn die Funktion nicht oszilliert wie es bei trigonometrischen Funktionen der Fall ist).
- Wenn Sie bereits eine Nullstelle der Funktion bestimmt haben, können Sie die Funktion auf weitere Nullstellen untersuchen, in dem Sie von den bekannten Nullstellen weit entfernte Schätz-werte verwenden.
- Viele Funktionen verhalten sich auf bestimmte Weise, wenn ihre Argumente gegen Null streben. Untersuchen Sie Ihre Funktion hinsichtlich der Werte von x, für die beliebige Argumente gleich Null sind, und verwenden Sie anfängliche Schätz-werte, die mit diesen Werten übereinstimmen und nahe bei ihnen liegen.

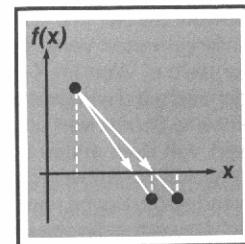
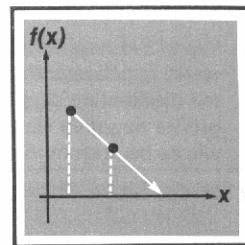
Obwohl Sie die [SOLVE]-Funktion üblicherweise mit zwei unterschiedlichen Anfangswerten versorgen, können Sie [SOLVE] durchaus auch mit ein und demselben Wert im X- und Y-Register verwenden. Wenn die Schätz-werte identisch sind wird im allgemeinen ein zweiter Schätz-wert erzeugt. Wenn der einzelne Schätz-wert ungleich Null ist, unterscheidet sich der zweite Schätz-wert von dem ersten um eine Stelle in der siebenten signifikanten Ziffer. Wenn Ihr Schätz-wert gleich Null ist, wird 1×10^{-7} als zweiter Schätz-wert verwendet. Die Suche nach einer Nullstelle wird dann ganz normal wie mit zwei Schätz-werten durchgeführt.

DIE ARBEITSWEISE VON **SOLVE**

Sie werden **SOLVE** am wirksamsten benutzen können, wenn Sie einen Einblick in den verwendeten Algorithmus gewinnen.

Im Verlauf des Suchvorganges nach einer Nullstelle einer Funktion verwendet der Algorithmus die zwei oder drei zuletzt berechneten Funktionswerte, um eine Näherungsrechnung für das Kurvenbild der Funktion durchzuführen. Anhand dieses Kurvenbildes macht der Algorithmus eine intelligente "Voraussage" eines neuen Schätzwertes, an dem die Kurve die X-Achse schneiden könnte. Das Unterprogramm zur Berechnung der Funktion wird ausgeführt, wobei der Wert der Funktion für diesen neuen Schätzwert ermittelt wird. Dieser Vorgang wiederholt sich im Verlauf des **SOLVE**-Algorithmus.

Wenn zwei Schätzwerte Funktionswerte mit unterschiedlichen Vorzeichen ergeben, erkennt der Algorithmus, daß die Kurve in dem Intervall zwischen diesen Schätzwerten die X-Achse wenigstens einmal schneidet. Das Intervall wird systematisch verkleinert, bis eine Nullstelle gefunden ist.

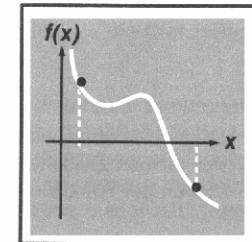


Eine Nullstelle gilt als gefunden, wenn der berechnete Funktionswert entweder gleich Null ist oder wenn aufeinanderfolgende Schätzwerte Funktionswerte zur Folge haben, die unterschiedliche Vorzeichen haben und sich in der letzten (zehnten) signifikanten Ziffer um weniger als zwei oder drei Einerstellen unterscheiden. In diesem Fall wird die Ausführung unterbrochen und der Schätzwert angezeigt. Wir haben schon früher besprochen (siehe Seite 180) wie Situationen während des Iterationsprozesses entstehen können, die auf das Fehlen eines Funktionswertes gleich Null hindeuten. Dies tritt ein, wenn es nicht möglich ist, auf logische Weise einen neuen Schätzwert vorauszusagen, der einen Funktionswert näher Null ergeben würde. In diesen Fällen erscheint die Fehleranzeige **ERROR 6**.

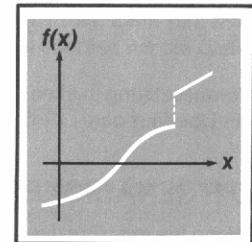
Sie sollten beachten, daß Ihre zuerst eingegebenen Anfangswerte zur Voraussage der nächsten Schätzwerte verwendet werden. Sorgfältig ausgewählte Schätzwerte erleichtern das Finden einer Lösung, weil schon die ersten Schätzwerte viel bessere Voraussagen erlauben.

Der **SOLVE**-Algorithmus wird immer eine vorhandene Nullstelle ermitteln, wenn einer der vier folgenden Bedingungen erfüllt sind:

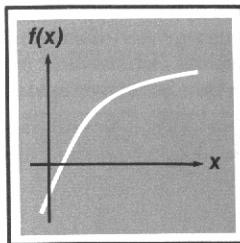
- Zwei Schätzwerte ergeben Funktionswerte mit unterschiedlichen Vorzeichen.



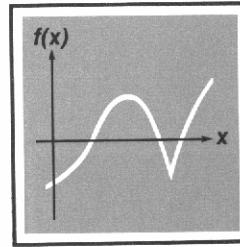
- Die Funktion ist monoton, d. h. daß die Funktion für ansteigende Werte von x entweder immer fällt oder immer wächst.



- Das Kurvenbild der Funktion ist entweder überall konvex oder überall konkav.



- Lokale Maxima und lokale Minima treten nur einzeln zwischen benachbarten Nullstellen der Funktion auf.



Voraussetzung in allen Fällen ist, daß weder eine unerlaubte Operation noch ein Überlauf den **SOLVE**-Algorithmus unterbrechen.

DIE GENAUIGKEIT DER LÖSUNG

Wenn Sie mit Hilfe der **SOLVE**-Funktion die Lösung einer Gleichung ermitteln, wird die Nullstelle genau berechnet. Die angezeigte Wurzel ergibt entweder genau den Funktionswert $f(x)$ Null oder besteht aus einer zehnstelligen Zahl, die praktisch dem Schnittpunkt der Funktion auf der X-Achse entspricht. Jede solche Wurzel hat eine Genauigkeit von zwei oder drei Einheiten in der zehnten signifikanten Stelle.

In den meisten Fällen ist die errechnete Wurzel ein genauer Wert für die theoretische (unendlich genaue) Lösung der Gleichung. Unter gewissen Umständen führt jedoch die endliche Genauigkeit des Rechners zu Ergebnissen, die sich nicht mit den erwarteten theoretischen Werten vereinbaren lassen.

Wenn eine Berechnung ein Resultat ergibt, dessen Absolutwert kleiner als $1.000.000.000 \times 10^{-99}$ ist, wird es gleich Null gesetzt. Dies wird mit Unterlauf bezeichnet.

Wenn im Verlauf Ihres Unterprogrammes ein Unterlauf für einen gewissen Bereich von x eintritt und dieser den Funktionswert beeinflußt, kann die Wurzel in diesem Bereich eine Ungenauigkeit aufweisen. So hat die Gleichung

$$x^4 = 0$$

beispielsweise eine Wurzel bei $x = 0$. Wegen Unterlaufs berechnet der HP-34C eine Wurzel mit einem Wert von 1,5060 – 25 (für anfängliche Schätzwerte von 1 und 2). Als weiteres Beispiel wollen wir die Gleichung

$$1/x^2 = 0$$

betrachten, die keine reelle Wurzel besitzt. Wegen Unterlaufs berechnet **SOLVE** eine Wurzel mit einem Wert von 3,170 49 (für anfängliche Schätzwerte von 10 und 20). In beiden Beispielen wurde ein Wert von x gefunden, für den der Funktionswert gleich Null ist. Wenn Sie die durch Unterlauf entstandene Wirkung verstehen, können Sie Ergebnisse dieser Art ohne weiteres interpretieren.

Die Genauigkeit eines berechneten Wertes kann manchmal auch durch Rundungsfehler nachteilig beeinflußt werden, wenn eine unendlich genaue Zahl in der zehnten signifikanten Stelle gerundet wird.

Wenn Ihr Unterprogramm eine übermäßig hohe Genauigkeit benötigt, um die Funktion für einen Bereich von x richtig zu berechnen, kann das mit **SOLVE** ermittelte Resultat ungenau sein. Als Beispiel nehmen wir die Gleichung

$$|x^2 - 5| = 0$$

die eine Nullstelle bei $x = \sqrt{5}$ besitzt.

Da keine zehnstellige Zahl genau gleich $\sqrt{5}$ ist, wird in diesem Fall die Meldung ERROR 6 (für jeden beliebigen Anfangswert) ausgegeben, da die Funktion niemals gleich Null ist, noch je ihr Vorzeichen wechselt. Dagegen hat die Gleichung

$$[|x| + 1] + 10^{15}^2 = 10^{30}$$

keine Nullstellen, weil der Ausdruck links des Gleichheitszeichens immer größer als 10^{30} ist. Wegen Rundungsfehler ergibt die Berechnung von

$$f(x) = [|x| + 1] + 10^{15}^2 - 10^{30}$$

aber eine Nullstelle bei 1 für Schätzwerte von 1 und 2. Indem Sie Situationen erkennen, in denen Rundungsfehler bei der Ausführung von **SOLVE** eine Rolle spielen können, werden Sie in der Lage sein, die Ergebnisse dennoch richtig zu interpretieren und notfalls die Funktion so zu programmieren, daß die Wirkung von Rundungsfehlern auf ein Minimum beschränkt bleibt.

Für eine Vielzahl von praktischen Anwendungen bestehen die Parameter einer Gleichung – und manchmal die Gleichung selber – nur aus Annäherungen.

Jeder physikalische Parameter ist mit einer eigenen Ungenauigkeit behaftet. Wenn ein physikalischer Ablauf mathematisch dargestellt wird, ergibt dies nur ein Modell des Ablaufs, der insoweit richtig ist wie die zugrundeliegenden Annahmen richtig sind.

Wenn Sie sich dieser und anderer Ungenauigkeiten bewußt sind, können Sie sich dies zunutze machen. Wenn Sie Ihr Unterprogramm derartig gestalten, daß ein Funktionswert von Null erzeugt wird, wenn der errechnete Wert vernachlässigbar klein ist, können Sie viel Rechenzeit bei der Verwendung von **SOLVE** sparen, besonders in Fällen wo die Rechenzeit ohnehin beträchtlich ist.

Beispiel: Hammerwerfer vom Schläge Max Zimmermann können einen Hammer 105 Meter und mehr in die Höhe werfen. In der Tat erreicht Max gewöhnlich eine Höhe von 107 Metern. Wie lange dauert es bis dieser ungeheure Wurf eine Höhe von 107 Metern erreicht? (Dies ist auf Seite 177 beschrieben.)

Lösung: Die gewünschte Lösung besteht aus dem Wert h , für den $t = 107$ ist. Das Unterprogramm des vorherigen Beispiels berechnete die Höhe des Hammers über dem Erdboden. Dieses Unterprogramm kann als Teil eines neuen Funktionsunterprogramms zur Berechnung von

$$f(t) = h(t) - 107$$

verwendet werden.

Schieben Sie den **PRGM ■■■■■ RUN**-Schalter in Stellung RUN und tasten Sie das Unterprogramm zur Berechnung von $f(t)$ ein.

Tastenfolge	Anzeige
h LBL B	001–25,13,12 Das Unterprogramm beginnt mit einer LBL -Anweisung
GSB A	002– 13 11 Berechnung von $h(t)$
1	003– 1
0	004– 0
7	005– 7
h RTN	006– 41 Berechnung von $h(t) = 107$
	007– 25 12
Schieben Sie den PRGM ■■■■■ RUN -Schalter in Stellung RUN. Um herauszufinden, wann der Hammer zum ersten Mal eine Höhe von 107 Meter erreicht, verwenden Sie Anfangswerte von 0 und 1 Sekunden.	
Tastenfolge	Anzeige
0 ENTER	0,0000 { Anfängliche Schätzwerte
1	1,
f SOLVE B	4,1718 Die gewünschte Lösung
g R↓	4,1718 Ein vorheriger Schätzwert der Nullstelle
g R↓	0,0000 Der Wert der Funktion wenn die Nullstelle für t eingesetzt wird
Es dauert 4,1718 Sekunden, bis der Hammer eine Höhe von genau 107 Metern erreicht. (Die Lösung dauert dagegen etwa eine Minute.)	
Jetzt wollen wir annehmen, daß die für $h(t)$ verwendete Funktion nur bis zum nächsten ganzen Meter genau ist. Sie können jetzt ihr Unterprogramm so ändern, daß $f(t) = 0$ gesetzt wird, wenn der berechnete Wert von $f(t)$ kleiner als 0,5 Meter ist.	
Schieben Sie den PRGM ■■■■■ RUN -Schalter in Stellung RUN und führen Sie die folgende Programmänderung durch:	
Tastenfolge	Anzeige
GTO □ 006	006– 41 Die Zeile vor der RTN -Anweisung
h ABS	007– 25 34 Absolutwert von $f(t)$
□	008– 73 Die Genauigkeit
5	009– 5
f X>Y	010– 14 51 Setzt $f(t) = 0$, wenn Absolutwert
CLX	011– 34 \leq Genauigkeit
g X≠0	012– 15 61 $f(t)$ wird zurückgerufen, wenn der
h LSTX	013– 25 0 Wert ungleich Null ist

Schieben Sie den PRGM -Schalter in Stellung RUN und führen Sie [SOLVE] nochmal aus.

Tastenfolge

0 [ENTER]

1 [SOLVE]

[R²]

[R²]

Anzeige

0,0000

1,

4,0681

4,0681

0,0000

Anfängliche Schätzwerte

Die gewünschte Lösung

Ein vorheriger Schätzwert der Wurzel

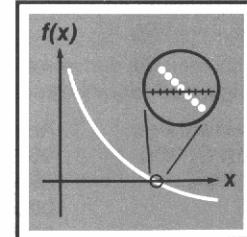
Der Wert der modifizierten Funktion f(t), wenn die Wurzel für t eingesetzt wird.

Nach 4,0681 Sekunden erreicht der Hammer eine Höhe von $107 \pm 0,5$ Meter. Die Lösung, die sich von der vorherigen unterscheidet, ist dennoch korrekt wenn die Unsicherheit bei der Verwendung der Höhengleichung in Betracht gezogen wird. (Und die Rechenzeit ist gegenüber dem ersten Beispiel nur etwa halb so lang.)

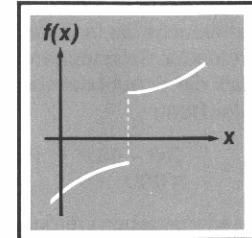
AUSWERTUNG DER ERGEBNISSE

Mit Hilfe der Zahlen die nach der Ausführung von der [SOLVE]-Funktion in den X-, Y- und Z-Registern stehen, können Sie die die bei der Suche nach einer Nullstelle entstandenen Ergebnisse auswerten*. Die Ergebnisse haben auch dann eine Bedeutung, wenn keine Wurzel gefunden wurde.

Wenn mit [SOLVE] die Nullstellen einer angegebenen Gleichung gefunden wurde, stehen die Wurzel und der Funktionswert im X- und Z-Register. Ein Funktionswert von Null ist dabei das erwartete Ergebnis. Jedoch ist ein Funktionswert ungleich Null auch zulässig, da er darauf schließen lässt, daß die Kurve die X-Achse in unendlich kleiner Entfernung von der Nullstelle schneidet. In den meisten dieser Fälle ist der Funktionswert sehr nahe Null.



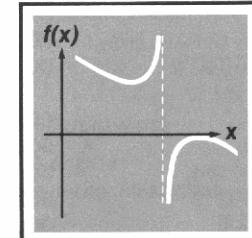
Ein weiterer Fall, in dem [SOLVE] eine Nullstelle mit einem Funktionswert ungleich Null ermittelt, verdient unsere besondere Aufmerksamkeit. Besitzt eine Funktion eine Unstetigkeitsstelle, wobei der Sprung durch die X-Achse geht, gibt [SOLVE] einen an die Unstetigkeitsstelle angrenzenden x-Wert als Wurzel an. Dies ist sinnvoll, da ein großer Wechsel des Funktionswertes zwischen zwei benachbarten x-Werten die Folge eines sehr schnellen kontinuierlichen Verlaufs sein kann. Da der Algorithmus in diesem Fall die Gleichung nicht lösen kann, wird die zu interpretierende Wurzel im X-Register angezeigt.



Eine Funktion kann einen Pol enthalten, wo der Funktionswert gegen ∞ geht. Wenn der Funktionswert an dieser Diskontinuität das Vorzeichen wechselt, kann der entsprechende x-Wert als Nullstelle in Frage kommen, wie dies bei jedem anderen Sprung durch die X-Achse der Fall ist.

Der im Z-Register enthaltene Funktionswert kann hier jedoch unverhältnismäßig groß sein. Liegt die Unstetigkeit bei einem x-Wert, der genau durch eine zehnstellige Zahl dargestellt werden kann, wird die Ausführung des Unterprogramms mit einer Fehler- oder Überlaufmeldung unterbrochen. Die [SOLVE]-Funktion kann in solchen Fällen nicht zu Ende geführt werden.

Durch eine entsprechende Verwendung von Vergleichsoperationen im Unterprogramm kann so ein Abbruch natürlich vermieden werden.



* Die im T-Register enthaltene Zahl ist identisch mit der Zahl, die am Ende Ihres Unterprogramms im Y-Register steht. Im allgemeinen hat diese Zahl keine Bedeutung.

Beispiel: Der Konstrukteur Peter Bauer untersucht die in einem Bauelement auftretenden Belastungen. Dabei stellt er fest, daß die Schubbeanspruchung durch die Gleichung

$$Q = \begin{cases} 3x^3 - 45x^2 + 350 & \text{für } 0 < x < 10 \\ 1000 & \text{für } 10 \leq x \leq 14 \end{cases}$$



wiedergegeben werden kann, wobei Q die Scherkraft in Newton und x die Entfernung von einem Ende des Trägers ist. Schreiben Sie ein Unterprogramm mit dem die Schubbeanspruchung für einen beliebigen Wert von x ermittelt werden kann. Finden Sie mit Hilfe von **SOLVE** die Position, wo die Schubbeanspruchung Null ist.

Lösung: Die Gleichung für die Schubbelastung kann mit Hilfe des Horner-Schemas in eine günstigere Form gebracht werden:

$$Q = (3x - 45)x^2 + 350 \quad \text{für } 0 < x < 10$$

Schieben Sie den **PRGM ■■■■■ RUN**-Schalter in Stellung **PRGM** und tasten Sie das Unterprogramm ein:

Tastenfolge	Anzeige
f CLEAR PRGM	000-
h LBL 2	001-25,13, 2
1	002- 1
0	003- 0 } Abfrage in welchen Bereich x fällt.
f X≤Y	004- 14 41
GTO 9	005- 22 9 Verzweigung, wenn $x \geq 10$
CLX	006- 34
3	007- 3
x	008- 61
4	009- 4
5	010- 5
-	011- 41
x	012- 61
x	013- 61
3	014- 3
5	015- 5
0	016- 0

**Tastenfolge**

+	017-	51
h RTN	018-	25 12
h LBL 9	019-	25,13, 9
EEX	020-	33
3	021-	3
h RTN	022-	25 12

Anzeige

017- 51
018- 25 12
019-25,13, 9
020- 33
021- 3
022- 25 12



Schieben Sie jetzt den **PRGM ■■■■■ RUN**-Schalter in Stellung **RUN**. Untersuchen Sie den Träger auf eine Schubbeanspruchung von Null, in dem Sie mit den Schätzwerten 7 und 14 beginnen.

**Tastenfolge**

7	ENTER	7,0000 }
14		14,
f SOLVE	2	10,0000
g R+ g R+		1,000,000

Anzeige

7,0000 } Die anfänglichen Schätzwerte
14, Eine mögliche Nullstelle
1,000,000 Die Berechnung ist ungleich Null



Der hohe Belastungswert läßt darauf schließen, daß die **SOLVE**-Routine auf eine Unstetigkeitsstelle gestoßen ist. An dieser Stelle wechselt die Schubbeanspruchung sehr schnell das Vorzeichen. Untersuchen Sie jetzt den anderen Teil des Trägers, in dem Sie mit Schätzwerten von 0 und 7 beginnen.

**Tastenfolge**

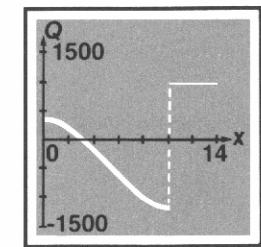
0	ENTER	0,0000 }
7		7,
f SOLVE	2	3,1358
g R+ g R+		2,0000-07

Anzeige

0,0000 } Die anfänglichen Schätzwerte
7, Eine mögliche Nullstelle
3,1358 Die Belastung ist vernachlässigbar.



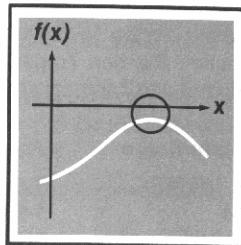
Die Schubbeanspruchung des Trägers ist Null bei 3.1358 Meter und unterliegt einem plötzlichen Wechsel bei 10.0000 Meter.



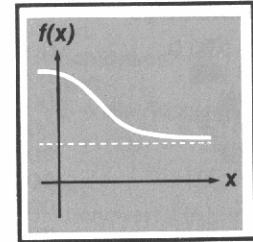
Wenn der Rechner keine Lösung findet und die Meldung ERROR 6 ausgibt, können Sie durch das Drücken einer beliebigen Taste den letzten Schätzwert für die Wurzel im X-Register angezeigt sehen. Außerdem können Sie durch eine Inspektion der Zahlen in den Y- und Z-Registern häufig das Verhalten der Funktion in der Nähe des Schätzwertes erkennen und diese Information verwerthen.

Wenn der Algorithmus die Suche nach einer Wurzel in der Nähe eines lokalen Minimums des Absolutwertes der Funktion abbricht, löschen Sie die Fehlermeldung ERROR 6 und betrachten Sie durch zyklisches Verschieben den Inhalt der Stack-Register. Wenn der Funktionswert im Z-Register sehr nahe Null ist, kann es sein, daß eine Wurzel der Gleichung gefunden wurde – die Zahl im X-Register war also eine zehnstellige Zahl sehr nahe der theoretischen Lösung. Dieses potentielle Minimum können Sie untersuchen, indem Sie den Stack weiterzyklisch verschieben bis die ursprünglichen Schätzwerte wieder in den X- und Y-Registern enthalten sind. Drücken Sie dann nochmal mit diesen Anfangswerten die **SOLVE**-Taste. Wenn es tatsächlich um ein Minimum handelt, wird wieder ERROR 6 angezeigt und die Zahl im X-Register etwa die gleiche wie vorher sein, wenn sie auch möglicherweise jetzt näher dem eigentlichen Minimum liegt.

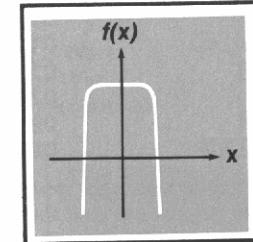
Sie können selbstverständlich **SOLVE** verwenden, um ein lokales Minimum des Absolutwertes einer Funktion zu ermitteln. Sie müssen aber darauf achten, die Suche auf den Bereich, in dem das Minimum liegt, zu beschränken. Bedenken Sie, daß **SOLVE** alles versucht, eine Nullstelle zu finden.



Wenn die Suche nach einer Lösung unterbrochen und ERROR 6 angezeigt wird, weil es sich um eine horizontale Asymptote handelt (und hier der Funktionswert für einen großen Bereich von x praktisch konstant bleibt), unterscheiden sich die Schätzwerte im X- und Y-Register im allgemeinen wesentlich voneinander. Die Zahl im Z-Register ist der Wert einer möglichen Asymptote. Wenn Sie **SOLVE** nochmal mit dem Inhalt der X- und Y-Register als anfängliche Schätzwerte ausführen, kann ERROR 6 wieder eine horizontale Asymptote signalisieren, auch wenn sich der Inhalt der X- und Y-Register diesmal von den vorherigen Werten unterscheidet. Der Inhalt des Z-Registers würde sich dabei nicht geändert haben.



Wenn der Suchvorgang auf einen lokalen, gestreckten Bereich der Funktion beschränkt wird und als Folge die Meldung ERROR 6 erscheint, werden die Schätzwerte im X- und Y-Register entweder sehr nahe beieinander liegen oder sehr klein sein. Führen Sie **SOLVE** nochmal aus, indem Sie als Anfangswerte den Inhalt der X- und Y-Register (oder zwei etwas weiter voneinander liegende Zahlen) verwenden. Wenn der Absolutwert der Funktion nicht ein Minimum oder eine Konstante ist, wird die Suche nach einer Lösung schließlich ausgeweitet, bis ein brauchbares Ergebnis erzielt wird.



Beispiel: Untersuchen Sie die Funktion

$$f(x) = 3 + e^{-|x|} 10 - 2 e^{x^2} e^{-|x|}$$



Schieben Sie erst den **PRGM** ■ **RUN**-Schalter in Stellung RUN und tasten Sie das Unterprogramm zur Berechnung von f(x) ein.

Tastenfolge

	Anzeige
h LBL 0	001–25,13, 0 Das Unterprogramm beginnt mit einer
h ABS	002– 25 34 LBL -Anweisung.
CHS	003– 32
g e^x	004– 15 1
X:Y	005– 21 x steht im X-Register
g X²	006– 13 3
X	007– 61
g e^x	008– 15 1
2	009– 2
X	010– 61
CHS	011– 32
X:Y	012– 21
h ABS	013– 25 34
CHS	014– 32
1	015– 1
0	016– 0
÷	017– 71
g e^x	018– 15 1
+	019– 51
3	020– 3
+	021– 51
h RTN	022– 25 12

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung RUN und verwenden Sie **SOLVE** mit den folgenden – einzelnen – anfänglichen Schätzwerten:
10, 1, 10⁻²⁰.

Tastenfolge

	Anzeige
10 ENTER	10,0000 Der anfängliche Schätzwert steht im X- und Y-Register
f SOLVE 0	Error 6
CLX	455,4335 Der beste x-Wert
g R[±]	48,026,721.85 Der vorherige Schätzwert
g R[±]	1,0000 Der Funktionswert
f R[±] f R[±]	455,4335 Der Stack wird in die Ausgangslage zurückgeführt
f SOLVE 0	Error 6
CLX	48,026,721.85 Der selbe Funktionswert (Eine Asymptote)
g R[±] g R[±]	1,0000 Der anfängliche Schätzwert steht im X- und Y-Register
1 ENTER	1,0000



f SOLVE 0
CLX
g R[±]
g R[±]
f R[±] f R[±]



f SOLVE 0
CLX
g R[±] g R[±]



EEX CHS 20 ENTER **1,0000** **-20**
f SOLVE 0
CLX
g R[±]
g R[±]
f R[±] f R[±]



f SOLVE 0
CLX
g R[±]
g R[±]



In allen drei Fällen sucht **SOLVE** erst eine Lösung in der Richtung, die durch das Kurvenbild in der Umgebung des anfänglichen Schätzwertes bestimmt wird. Mit 10 als Anfangswert fand **SOLVE** die horizontale Asymptote (mit einem Wert von 1.0000). Mit 1 als Schätzwert wurde ein Minimum von 0.3788 bei $x = 2.1213$ gefunden. Und mit 10^{-20} zeigte es sich schließlich, daß die Funktion in dem kleinen Bereich, in dem sie untersucht wurde, mit einem Wert von 2.0000 konstant ist.



Error 6
2,1213
2,1471
0,3788
2,1213

Error 6
2,1213
0,3788

EEX CHS 20 ENTER **1,0000** **-20**
f SOLVE 0
CLX
g R[±]
2,0000
1,0000 **-20**

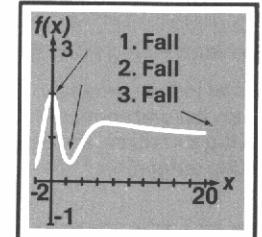
Error 6
1,1250 **-20**
1,5626 **-16**
2,0000

Der beste x-Wert
Der vorherige Schätzwert
Der Funktionswert
Der Stack wird in die Ausgangslage zurückgeführt

Der selbe x-Wert
Der selbe Funktionswert (Ein Minimum)
Der anfängliche Schätzwert steht im X- und Y-Register

Der beste x-Wert
Der vorherige Schätzwert
Der Funktionswert
Der Stack wird in die Ausgangslage zurückgeführt

Ein weiterer x-Wert
Der vorherige Wert
Der selbe Funktionswert

Der Graph von $f(x)$ **VERWENDUNG VON **SOLVE** IN PROGRAMMEN**

Die **SOLVE**-Routine kann als Teil eines Programms verwendet werden. Achten Sie darauf, daß die X- und Y-Register vor dem Aufruf der **SOLVE**-Routine die anfänglichen Schätzwerte enthalten. Wenn die **SOLVE**-Routine beendet ist, steht der x-Wert im X-Register und der entsprechende Funk-

tonswert im Z-Register. Wenn der x-Wert einer Nullstelle entspricht, wird das Programm mit der nächsten Zeile fortgesetzt. Wenn der x-Wert keine Nullstelle ist, wird die nächste Zeile übersprungen. Die **SOLVE**-Anweisung überprüft also, ob der x-Wert eine Nullstelle ist und setzt das Programm nach der "DO-IF-TRUE"-Regel fort. Das Programm kann somit Fälle berücksichtigen, in denen keine Lösung gefunden wird, in dem es neue Schätzwerte wählt oder einen Parameter der Funktion ändert.

Wird **SOLVE** innerhalb eines Programms verwendet, wird eine der sechs möglichen anstehenden Rücksprungadressen belegt. Da **SOLVE** selber ein Unterprogramm aufruft, dürfen höchstens vier andere Rücksprünge anstehen. Wird **SOLVE** dagegen über die Tastatur verwendet, wird keine Rücksprungadresse benötigt, so daß den von **SOLVE** aufgerufenen Unterprogrammen fünf Unterprogrammebenen zur Verfügung stehen. Beachten Sie, daß die Fehlermeldung ERROR 8 erscheint, wenn sechs Rücksprünge anstehen und ein weiteres Unterprogramm aufgerufen wird (siehe Seite 135).

EINSCHRÄNKUNG BEI DER VERWENDUNG VON **SOLVE**

Bei der Verwendung von **SOLVE** ist die Einschränkung zu beachten, daß diese Funktion nicht rekursiv verwendet werden kann. Das heißt, daß **SOLVE** nicht in einem Unterprogramm stehen kann, das bei der Ausführung von **SOLVE** aufgerufen wird. In diesem Fall wird das Programm mit der Fehlermeldung ERROR 5 unterbrochen.

Es ist jedoch möglich, **SOLVE** in Verbindung mit der **∫**-Funktion zu verwenden und somit die fortschrittlichen Eigenschaften beider Tasten gleichzeitig zu nutzen. Im Anhang A wird ein Beispiel einer kombinierten Anwendung aufgeführt.

ZUSÄTZLICHE INFORMATIONEN

Im Anhang A werden die fortgeschrittene Verwendung von **SOLVE** sowie weitere Techniken und Anwendungen in Verbindung mit **SOLVE** beschrieben. Dazu gehören:

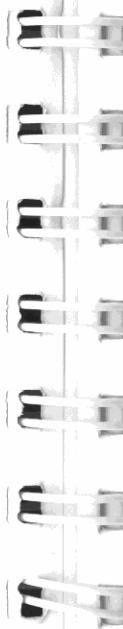
- Verwendung von **SOLVE** mit Polynomen
- Bestimmung mehrerer Nullstellen
- Ermittlung lokaler Extremwerte einer Funktion
- Verkürzung der Rechenzeit
- Verwendung von **SOLVE** mit **∫**



ABSCHNITT 9: NUMERISCHE INTEGRATION

Viele Aufgaben, die in der Mathematik, der Wissenschaft und im Ingenieurwesen zu lösen sind, benötigen die Berechnung des bestimmten Integrals einer Funktion. Das Integral einer Funktion $f(x)$ mit unterer Integrationsgrenze a und oberer Integrationsgrenze b wird mathematisch geschrieben als:

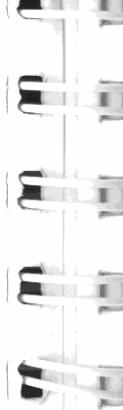
$$I = \int_a^b f(x) dx.$$



Die Größe I ist gleich der Fläche, die von der Kurve $f(x)$, der x-Achse und den Ordinaten a und b begrenzt wird.

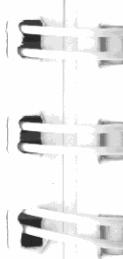


Wenn sich ein Integral nur schwierig oder gar nicht analytisch bestimmen läßt, kann es mit numerischen Methoden berechnet werden. Bis jetzt war man dazu auf mehr oder weniger aufwendige Computerprogramme angewiesen. Mit dem HP-34C können Sie aber jetzt numerische Integration leicht mit der **∫** (Integrations)-Taste durchführen.

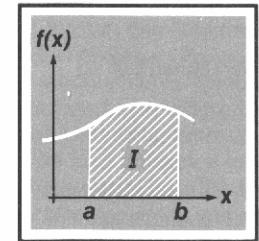


VERWENDUNG VON **∫**

Die folgenden einfachen Regeln sind dabei zu befolgen:



1. Tasten Sie das Unterprogramm der zu integrierenden Funktion $f(x)$ ein. Dieses Unterprogramm muß mit einer **LBL**-Anweisung gefolgt von 0, 1, 2, 3, **A** oder **B** beginnen und den Wert von $f(x)$ in das **X**-Register schreiben.
2. Geben Sie die untere Integrationsgrenze in das **X**-Register ein und drücken Sie dann **ENTER+**, damit sie in das **Y**-Register angehoben wird.
3. Geben Sie die obere Integrationsgrenze in das **X**-Register ein.
4. Drücken Sie **f** **∫** gefolgt von der für das Unterprogramm verwendeten Marke.



Beispiel: In gewissen Zweigen der Physik und der Ingenieurwissenschaften wird die Berechnung der Besselschen Funktion benötigt. Die Besselsche Funktion 0-ter Ordnung hat folgende Form:

$$J_0(x) = \frac{1}{\pi} \int_0^{\pi} \cos(x \sin \theta) d\theta.$$

Berechnen Sie

$$J_0(1) = \frac{1}{\pi} \int_0^{\pi} \cos(\sin \theta) d\theta.$$

Schieben Sie erst den PRGM ■■■ RUN -Schalter in Stellung RUN und tasten Sie das Unterprogramm zur Berechnung der Funktion $f(0) = \cos(\sin 0)$.

Tastenfolge Anzeige

f CLEAR PRGM	000-	Löschen des Programmspeichers
h LBL 0	001-25,13, 9	Das Unterprogramm beginnt mit einer LBL -Anweisung. Das Unterprogramm setzt den Wert 0 im X-Register voraus.
f SIN	002- 14 7	Berechnung von Sin 0
f COS	004- 25 12	Berechnung von cos (sin 0)
h RTN	004- 25 12	

Schieben Sie darauf den PRGM ■■■ RUN -Schalter wieder in Stellung RUN und geben Sie die untere und obere Integrationsgrenze in das Y- X-Register ein. Außerdem muß für den Rechner zur Ausführung der trigonometrischen Funktion der Winkel-Modus Bogenmaß gewählt werden.

Tastenfolge Anzeige

0 ENTER	0,0000	Geben Sie die untere Integrationsgrenze, 0, in das Y-Register ein.
h T	3,1416	Geben Sie obere Grenze, π , in das X-Register ein.
g RAD	3,1416	Bogenmaß für die trigonometrische Funktion.

Jetzt können Sie **f** **0** zur Berechnung des Integrals drücken. Wie bei der Verwendung von **SOLVE** erscheint das Ergebnis nicht sofort in der Anzeige, wie es bei anderen Funktionen der Fall ist. Zur Berechnung des Integrals verwendet der HP-34C einen ausgeklügelten iterativen Algorithmus. Dieser Algorithmus berechnet die zu integrierende Funktion $f(x)$ für viele Werte x innerhalb des Integrationsintervalls. Dabei wird für jeden Wert x das für diesen Zweck geschriebene Unterprogramm durchlaufen, um den Funktionswert zu ermitteln. Sie erinnern sich vielleicht, daß Sie bei einigen Programmen und Unterprogrammen, die früher in diesem Handbuch durchgenommen wurden, mehrere Sekunden auf eine Antwort warten mußten. Das scheint nicht lang zu sein, aber wenn der Rechner ein Unterprogramm viele Male durchlaufen muß – wie es der Fall ist, wenn Sie **0** drücken – dann können Sie das Ergebnis nicht sofort erwarten. Die Rechenzeit für ein Integral dürfte im allgemeinen zwischen 30 Sekunden und 2 Minuten liegen, obwohl Sie in einzelnen Fällen noch länger sein kann. Später werden wir erfahren, wie wir diese Zeit verkürzen können. Jetzt wollen wir aber **f** **0** drücken und dann eine Pause einlegen (oder schon weiterlesen) und den HP-34C den strapaziösen Teil überlassen.

Tastenfolge Anzeige

$$\text{f } 0 \quad 2,4040 \quad = \int_0^{\pi} \cos(\sin \theta) d\theta.$$

Im allgemeinen muß das Ergebnis noch mit eventuellen Konstanten, die außerhalb des Integrationszeichen stehen, multipliziert werden. In diesem Fall muß das Integral mit $1/\pi$ multipliziert werden, um J_0 zu bekommen:

Tastenfolge Anzeige

$$\begin{array}{l} h \text{ T} \\ + \end{array} \quad \begin{array}{l} 3,1416 \\ 0,7652 \end{array} \quad J_0(1).$$

Bevor das Unterprogramm zur Berechnung von $f(x)$ aufgerufen wird, speichert die **0**-Routine – genau wie **SOLVE** – den Wert x in das Y- Z- und T- Register. Da jedes Stack-Register diesen Wert enthält, kann Ihr Unterprogramm mit diesem Wert die Berechnung durchführen, ohne ihn aus einem Speicherregister zurückrufen zu müssen. Die Unterprogramme in den zwei folgenden Beispielen machen sich diese Eigenschaften zunutze. (Wie diese Eigenschaft auch bei Polynomen verwendet werden kann, wird auf Seite 79 beschrieben.)

Anmerkung: Es kann vorkommen, daß Sie ein Unterprogramm, das Sie für die -Routine entwickelt haben, nur zur Benutzung des Funktionswertes für einen beliebigen x-Wert verwenden wollen. Wenn dieses Unterprogramm den x-Wert mehr als einmal vom Stack holt, müssen Sie den Stack vor der Ausführung des Unterprogramms erst manuell mittels   laden. Außerdem speichert der Rechner den x-Wert in jedes Stack-Register, so daß der vorherige Inhalt des Stacks mit dem x-Wert überschrieben wird. Wenn der Stack also Zwischenergebnisse enthält, die Sie nach der Berechnung des Integrals benötigen, müssen Sie diese Zahlen zur späteren Verwendung in einem Speicherregister zwischenspeichern.

Beispiel: Die Besselsche Funktion 1-ter Ordnung erster Gattung hat die Form:

$$J_1(x) = \frac{1}{\pi} \int_0^{\pi} \cos(\theta - x \sin \theta) d\theta.$$

Berechnen Sie

$$J_1(1) = \frac{1}{\pi} \int_0^{\pi} \cos(\theta - \sin \theta) d\theta.$$

Schieben Sie erst den  -Schalter in Stellung PRGM und tasten Sie das Unterprogramm zur Berechnung der Funktion $f(0)=\cos(0-\sin 0)$ ein.

Tastenfolge	Anzeige
  1	001–25,13, 1 Das Unterprogramm beginnt mit einer  -Anweisung
 	002– 14 7 Berechnung von $\sin 0$
	003– 41 Da der  -Algorithmus vor der Ausführung des Unterprogrammes den Wert 0 in das Y-Register speichert, wird hier mit der  -Operation ($0 - \sin 0$) berechnet.
 	004– 14 8 Berechnung von $\cos(0 - \sin 0)$.
 	005– 25 12

  Schieben Sie jetzt den  -Schalter in Stellung RUN und geben Sie die untere und obere Integrationsgrenze in das Y- und X-Register ein. Achten Sie darauf, daß der Winkel-Modus Bogenmaß eingeschaltet ist und drücken Sie dann   1, um das Integral zu berechnen. Dieses Integral müssen Sie schließlich noch mit $1/\pi$ multiplizieren um $J_1(1)$ zu erhalten.

Tastenfolge	Anzeige	
0 	0,0000	Geben Sie die untere Integrationsgrenze in das Y-Register ein.
 	3,1416	Geben Sie die obere Integrationsgrenze in das X-Register ein.
 RAD	3,1416	Achten Sie darauf, daß der Winkel-Modus Bogenmaß eingeschaltet ist. (Dieser Schritt ist nicht erforderlich, wenn Sie den Rechner inzwischen wieder ausgeschaltet noch den Winkel-Modus seit dem letzten Umschalten auf Bogenmaß verändert haben.)
  1	1,3825	$= \int_0^{\pi} \cos(\theta - \sin \theta) d\theta.$
  	0,4401	$J_1(1).$

  **Beispiel:** In der Nachrichtentechnik wird für manche Zwecke (z. B. die Stromstoßübertragung in idealisierten Netzwerken) ein Integral der folgenden Formel (auch Integralsinus genannt) benötigt:

$$Si(t) = \int_0^t \frac{\sin x}{x} dx.$$

  Berechnen Sie $Si(2)$.

Schieben Sie erst den PRGM [RUN]-Schalter in Stellung PRGM und tasten Sie das Unterprogramm zur Berechnung von $f(x) = (\sin x)/x$ ein.*

Tastenfolge	Anzeige
[h LBL 2	001–25,13, 2 Das Unterprogramm beginnt mit einer LBL -Anweisung.
f SIN	002– 14 7 Berechnung von $\sin x$
X^y	003– 21 Dadurch der Algorithmus den x-Wert vor der Ausführung des Unterprogrammes in das Y-Register speichert, bewirkt hier die X^y -Operation, daß der x-Wert in das X-Register und ein x in das Y-Register gespeichert wird.
[[÷] RTN	004– 71 005– 25 12 Division von $\sin x$ durch x.

Schieben Sie jetzt den PRGM  RUN -Schalter in Stellung RUN und geben Sie die Integrationsgrenzen in das X- und Y-Register ein. Achten Sie darauf, daß der Winkel-Modus Bogenmaß eingeschaltet ist und drücken Sie dann  2 um das Integral zu berechnen.

Tastenfolge	Anzeige	
0 [ENTER]	0,0000	Geben Sie die untere Integrationsgrenze in das Y-Register ein.
2	2,	Geben Sie die obere Integrationsgrenze in das X-Register ein.
[9] RAD	2,0000	Schalten Sie den Winkel-Modus Bogenmaß ein. (Dieser Schritt ist nicht notwendig, wenn Sie den Rechner inzwischen weder ausgeschaltet noch den Winkel-Modus seit dem letzten Umschalten auf Bogenmaß verändert haben.)
[f] [2] 2	1,6054	Si(2).

GENAUIGKEIT VON

Die Genauigkeit des Integrals einer Funktion hängt von der Genauigkeit der Funktion selber ab. Daher wird auch die Genauigkeit eines mit  berechneten Integrals durch die Genauigkeit der mit Ihrem Unterprogramm berechneten Funktion begrenzt.* Diese Genauigkeit können Sie durch die Anzahl der Ziffern, die der Genauigkeit der Funktionswerte entsprechen, über das AnzeigefORMAT angeben.** Wenn Sie weniger Ziffern angeben, wird das Ergebnis schneller berechnet.*** Dafür nimmt der Rechner aber an, daß die Funktion nur eine Genauigkeit besitzt, die durch die Anzahl der Stellen des Anzeigeformats angegeben ist. Wir werden Ihnen zeigen, wie Sie die Genauigkeit eines berechneten Integrals ermitteln können, nachdem wir einige Bemerkungen über das AnzeigefORMAT gemacht haben.

- * Es ist möglich, daß die Integrale von Funktionen mit gewissen Charakteristika (wie scharfe Zacken oder sehr schnelle Oszillationen) falsch berechnet werden. Dies ist jedoch sehr unwahrscheinlich. Im Anhang B werden die Eigenschaften von Funktionen, die Schwierigkeiten verursachen können, und Techniken, die in diesen Fällen Abhilfe schaffen können, beschrieben.
- ** Die Genauigkeit einer berechneten Funktion hängt von mehreren Faktoren ab; dazu zählen z. B. die Genauigkeit der in der Funktion verwendeten empirischen Konstanten und Rundungsfehler bei der Bestimmung der Funktionswerte. Dies wird in Anhang B ausführlich besprochen.
- *** Der Grund hierfür wird auch im Anhang B behandelt.

*Der Versuch, die Funktion $f(x)/x$ für $x = 0,1$ der unteren Integrationsgrenze zu berechnen, würde die Fehlermeldung ERROR 0 zur Folge haben, weil Division durch Null zum Programmab- bruch führt. Das Integral würde nicht berechnet werden. Der -Algorithmus berechnet jedoch normalerweise keine Funktion an ihren Integrationsgrenzen, so daß der Rechner ein Integral bestimmen kann, auch wenn die Funktion an diesen Punkten nicht definiert ist. Nur wenn das Integra- tionsintervall sehr klein ist, oder die Anzahl der Stützstellen sehr groß ist, wird die Funktion an den Integrationsgrenzen bestimmt.

Sie werden sich erinnern, daß der HP-34C über drei Arten von Anzeigeformaten verfügt: **FIX**, **SCI** und **ENG**.

Im allgemeinen spielt es keine große Rolle welches Anzeigeformat verwendet wird, da für viele Integrale das Ergebnis mit jedem der Formate identisch ist (vorausgesetzt, daß in Abhängigkeit des Absolutwertes der Funktion die Anzahl der Stellen in der Anzeige richtig gewählt wurde). Weil es praktischer ist, bei der Bestimmung von Integralen das **SCI**-Anzeigeformat zu verwenden, wird im weiteren Verlauf dieses Handbuchs dieses Format verwendet.

Anmerkung: Erinnern Sie sich, daß Sie die Anzahl der angezeigten Ziffern des einmal mit **SCI**, **ENG** und **FIX** gewählten Anzeigeformats ändern können, indem Sie eine Zahl im **I** Register speichern und dann **H** **DSP I** drücken, wie in Abschnitt 7 beschrieben wurde. Diese Eigenschaft ist häufig besonders nützlich, wenn **DISP** als Teil eines Programms ausgeführt wird, um in besonderen Fällen von entscheidender Bedeutung, wie wir im Anhang B unter "Berechnung von Integralen mit maximaler Genauigkeit" zeigen werden.

Weil die Genauigkeit jedes Integrals durch die Genauigkeit der Funktion begrenzt wird (die durch das Anzeigeformat festgesetzt ist), kann der Rechner nicht den genauen sondern nur einen angenäherten Wert des Integrals bestimmen. Der HP-34C speichert die Fehlerabschätzung^{*} der Näherung des Integrals in das **Y**-Register und schreibt den Näherungswert in das **X**-Register. Um die Genauigkeit einer Näherung zu überprüfen, holen Sie sich die Fehlerabschätzung mit **[x,y]** in die Anzeige.

Beispiel: Wählen Sie das Anzeigeformat von SCI 2 und berechnen Sie das Integral $J_1(1)$ des Beispiels auf Seite 204.

- * Kein Algorithmus, der in der numerischen Integration verwendet wird, kann den genauen Unterschied zwischen der Näherung und dem tatsächlichen Integral feststellen. Der vom HP-34C verwendete Algorithmus bestimmt aber eine obere Grenze für diese Differenz, die der Fehlerabschätzung der Näherung entspricht. Wenn beispielsweise das Integral von $\sin(2x)$ gleich 1.6045 ± 0.0001 ist, so ist die Näherung des Integrals 1.6054 und die Fehlerabschätzung 0.0001. Damit wissen wir zwar nicht wie groß der Unterschied zwischen dem tatsächlichen Integral und der Näherung ist, aber wir wissen, daß diese Differenz nicht größer als 0.0001 ist.

Wenn die Fehlerabschätzung größer als gewünscht ist, können sie verkleinern, indem Sie eine größere Anzahl von Ziffern über das Anzeigeformat festlegen und die Berechnung wiederholen.

Wenn Sie eine Näherung wiederholen wollen, erspart Ihnen der HP-34C die Mühe, die Integrationsgrenzen wieder in das **X**- und **Y**-Register zu laden. Nach der Berechnung eines Integrals stehen nicht nur das Ergebnis und die Fehlerabschätzung im **X**-Register, sondern die unteren und oberen Integrationsgrenzen stehen auch im **T**- und **Z**-Register. Um diese Grenzen für eine erneute Berechnung des Integrals wieder in das **X**- und **Y**-Register zu laden, Drücken Sie einfach    

Beispiel: Das Integral $J_1(1)$ soll jetzt in der vierten und nicht der zweiten Stelle nach dem Komma genau sein.

Tastenfolge	Anzeige	
  4    	1,8826 -03 3,1416 00	Wählen Sie das Anzeigeformat  4. Vertauschen Sie die Stackinhalte so lange zyklisch, bis die obere Integrationsgrenze wieder im X -Register steht.
  1 	1,3825 00 1,7091 -05	Das angenäherte Integral mit  4. Die Fehlerabschätzung der Näherung mit  4.

Die Fehlerabschätzung lässt erkennen, daß die Näherung wenigstens bis auf vier Stellen nach dem Komma genau ist. Beachten Sie, daß die Fehlerabschätzung bei  4 etwa um ein Prozent der Fehlerabschätzung bei  2 beträgt. Im allgemeinen wird die Fehlerabschätzung bei jeder mit  ausführten Näherung für jede zusätzliche Ziffer im Anzeigeformat um einen Faktor 10 verringert.

In dem Beispiel war aus der Fehlerabschätzung zu schließen, daß die Näherung nur bis auf vier Stellen nach dem Komma genau ist. Wenn wir aber kurzzeitig alle zehn Stellen des Ergebnisses anzeigen und diese Zahl mit dem tatsächlichen Wert des Integrals (eigentlich eine Näherung, die auf eine ausreichende Anzahl von Nachkommastellen genau ist) vergleichen, werden wir finden, daß unser Ergebnis in der Tat genauer ist, als es die Fehlerabschätzung vermuten läßt.

	Tastenfolge	Anzeige	
	 x:y	1,3825 00	Die Näherung wird in das X -Register zurückgerufen.
	 h MANT	1382459676	Die zehnstellige Zahl des Ergebnisses.
			Das Integral beträgt auf acht Stellen nach dem Komma genau 1 38245969. Das Ergebnis der  -Routine ist also auf sieben und nicht nur vier Stellen nach dem Komma genau. Da die Fehlerabschätzung durchweg sehr vorsichtig wird, werden die durch den Rechner bestimmten Näherungen im allgemeinen eine höhere Genauigkeit aufweisen, als die Fehlerabschätzung andeutet. Im Normalfall läßt sich jedoch nicht feststellen, wie genau eine Näherung ist; wir können nur sicher sein, daß die Differenz zwischen den Ergebnissen und dem tatsächlichen Wert des Integrals nicht größer als die Zahl im Y -Register ist.
			Mit der Genauigkeit und Fehlerabschätzung der mit  ermittelten Näherungswerte werden wir uns im Anhang B noch eingehend befassen.
			
			
			
			
			VERWENDUNG VON  IN PROGRAMMEN
			Bei der Verwendung von  ist die Einschränkung zu beachten, daß diese Funktion nicht rekursiv verwendet werden kann. Das heißt, daß  nicht in einem Unterprogramm stehen kann, das bei der Ausführung von  aufgerufen wird. Eine Berechnung von Mehrfachintegralen ist folglich nicht möglich. In diesem Fall wird das Programm mit der Fehlermeldung ERROR 5 anhalten. Es ist jedoch möglich,  in Verbindung mit der  -Funktion zu verwenden. Im Anhang A wird ein Beispiel einer kombinierten Anwendung aufgeführt.
			
			Wird  innerhalb eines Programms verwendet, wird eine der sechs möglichen anstehenden Rücksprungadressen belegt. Da  selber ein Unterprogramm aufruft, dürfen höchstens vier andere Rücksprünge anstehen. Wird  dagegen über die Tastatur verwendet, wird keine Rücksprungadresse benötigt, so daß den von  aufgerufenen Unterprogrammen fünf anstehende Rücksprünge zu Verfügung stehen. Beachten Sie, daß die Fehlermeldung ERROR 8 erscheint, wenn sechs Rücksprünge anstehen und ein weiteres Unterprogramm aufgerufen wird. (siehe Seite 135)!
			

ZUSÄTZLICHE INFORMATION

Der in diesem Abschnitt behandelte Stoff lässt Sie in vielen Anwendungsbereichen erfolgreich einsetzen. In Anhang B werden die fortgeschrittenen Verwendung von sowie weitere Techniken und Anwendungen in Verbindung mit beschrieben. Dazu gehören:

- Die Arbeitsweise von
- Genauigkeit, Fehlerabschätzung und Rechenzeit
- Genauigkeit der zu integrierenden Funktion
- Fehlerabschätzung und AnzeigefORMAT
- Integralberechnung mit maximaler Genauigkeit
- Der augenblickliche Näherungswert eines Integrals
- Faktoren, die zu falschen Ergebnissen führen können
- Faktoren, die die Rechenzeit verlängern.



ANHANG A: FORTGESCHRITTENE VERWENDUNG VON



In Abschnitt 8 wurden die für eine erfolgreiche Verwendung der Routine benötigten Grundbegriffe erläutert. Dieser Anhang befasst sich nun eingehender mit der -Funktion.



VERWENDUNG VON MIT POLYNOMEN

In vielen praktischen Anwendungen lassen sich physikalische Verfahren oder kompliziertere mathematische Ausdrücke durch Funktionen darstellen, die als Polynome bezeichnet werden. Polynome sind leicht verständlich und lassen sich so gestalten, daß eine Vielfalt mathematischer Charakteristika wiedergegeben werden können.



Ein Polynom vom Grade n lässt sich folgendermaßen darstellen:

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$



Es gibt dabei höchstens n reelle Werte, für die die Funktion gleich Null ist.



BESTIMMUNG MEHRERER NULLSTELLEN

Die Anzahl der positiven Wurzeln dieser Funktion ist nicht größer als die Anzahl der Vorzeichenwechsel in der Folge der Koeffizienten der Funktion. In gleicher Weise kann die maximale Anzahl negativer Wurzeln festgestellt werden, in dem x durch $-x$ in der ursprünglichen Gleichung ersetzt wird. Wenn die tatsächliche Anzahl von positiven oder negativen Wurzeln kleiner als die maximale Anzahl ist, kann sie sich von dieser nur um eine gerade Zahl unterscheiden (Descartesche Regel).



Als Beispiel betrachten wir das Polynom 3-ten Grades:

$$f(x) = x^3 - 3x^2 - 6x + 8$$



Diese Gleichung hat höchstens drei reelle Nullstellen. Außerdem existieren höchstens zwei positive reelle Nullstellen (man beachte den Vorzeichenwechsel zwischen dem ersten und zweiten und zwischen dem dritten und vierten Glied) und höchstens eine negative reelle Wurzel (nach Anzahl der Vorzeichenwechsel im Falle der Gleichung $f(-x) = -x^3 - 3x^2 + 6x + 8$).

Polynome lassen sich leichter programmieren, wenn sie unter Verwendung von verschachtelter Multiplikation in eine andere Form gebracht werden. Diese Form wird als Horner-Schema bezeichnet. So läßt sich die Funktion des vorhergehenden Beispiels folgendermaßen darstellen:

$$f(x) = [(x-3)x-6]x+8$$

In dieser Form läßt sich die Funktion leichter programmieren und schneller berechnen als in der ursprünglichen Form, besonders daß der x-Wert in allen vier Stack-Registern enthalten ist. (Diese Technik ist auf Seite 79 beschrieben).

Beispiel: Den Winter 1978 verbringt der Polarforscher Jean-Claude Coulerre ganz allein in seinem Camp im eisigen Norden. Er beginnt den südlichen Horizont täglich nach den ersten Sonnenstrahlen abzusuchen. Dabei weiß er, daß die Sonne nicht vor Anfang März, wenn sie eine Deklination von $5^\circ 28' S$ erreicht, sichtbar sein wird. An welchem Tag und zu welcher Tageszeit wird die Nachtwache des frierenden Forschers endlich belohnt?

Lösung: Der Zeitpunkt, an dem die Sonne im März eine Deklination von $5^\circ 28' S$ erreicht, kann über die Lösung der folgenden Gleichung ermittelt werden:

$$D = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0$$

wobei

D die Deklination in Grad
t die Zeit in Tagen und

$$a_4 = 4.2725 \times 10^{-8}$$

$$a_3 = 1.9931 \times 10^{-5}$$

$$a_2 = 1.0229 \times 10^{-3}$$

$$a_1 = 3.7680 \times 10^{-1}$$

$$a_0 = -8.1806.$$

ist.

Für den März, 1978 gilt die Einschränkung für t: $1 \leq t < 32$.

Wandeln Sie erst $5^\circ 18' S$ in Dezimalgrad um, indem Sie 5,18 **CHS** **9** **→H** drücken. Das Ergebnis ist -5.3000 Anzeigeformat **FIX** 4. (Südliche Breitengrade werden bei der Berechnung als negative Zahlen dargestellt).

Die Lösung von Coulerres Problem ist der Wert von t, der der Gleichung

$$-5.3000 = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t + a_0.$$

genügt.

Die **SOLVE**-Routine benötigt die Gleichung in der Form

$$0 = a_4 t^4 + a_3 t^3 + a_2 t^2 + a_1 t - 2.8806$$

wobei das letzte, konstante Glied jetzt die Deklination beinhaltet. Unter Verwendung des Horner-Schemas sieht die gleich Null zu setzende Funktion folgendermaßen aus:

$$f(t) = (((a_4 t + a_3)t + a_2)t + a_1)t - 2.8806$$

Damit das Unterprogramm kurz bleibt, werden die Konstanten in Registern gespeichert, die dem jeweiligen Exponenten von t entsprechen. Schieben Sie den **PRGM** **RUN**-Schalter in Stellung **RUN** und tasten Sie das Unterprogramm ein.

Tastenfolge Anzeige

f CLEAR PRGM	000-	Löschen des Programmspeichers.	
h LBL A	001-25,13,11	Den Anfang bildet eine LBL -Anweisung.	

RCL 4	002-	24	4
X	003-		61
RCL 3	004-	24	3
+	005-		51
X	006-		61
RCL 2	007-	24	2
+	008-		51
X	009-		61
RCL 1	010-	24	1
+	011-		51
X	012-		61
RCL 0	013-	24	0
+	014-		51
h RTN	015-	25	12

Schieben Sie jetzt den -Schalter in Stellung RUN und geben Sie die fünf Koeffizienten ein:

Tastenfolge **Anzeige***

4.2725 8	4,2725	-08	Der Koeffizient von t^4 .
4	4	4,2725	-08
1.9931			
5	-1,9931	-05	Der Koeffizient von t^3 .
3	-1,9931	-05	
1.0229 3	1,0229	-03	Der Koeffizient von t^2 .
2	0,0010		
3.7680 1	3,7680	-01	Der Koeffizient von t^1 .
1	0,3768		
2.8806 0	-2,8806		Die Konstante.

Da die gesuchte Lösung zwischen 1 und 32 liegen sollte, sind diese zwei Zahlen als anfängliche Schätzwerte zu verwenden. Drücken Sie anschließend um die Nullstelle zu berechnen.

Tastenfolge **Anzeige**

1	1,0000	Die anfänglichen Schätzwerte.
32	32,	Die berechnete Lösung.
	7,5137	Der vorherige Schätzwert ist der gleiche.
	7,5137	Der Funktionswert.
	0,0000	
	7,5137	Der Stack ist wieder in seiner Ausgangsposition.

Der Tag war der 7. März. Wandeln Sie den Dezimalteil der Zahl in Dezimalstunden und anschließend in Stunden, Minuten und Sekunden um.

* Drücken Sie 4 damit die Anzeige mit der in diesem Beispiel verwendeten Anzeige übereinstimmt.



Tastenfolge

24

Anzeige

0,5137
12,3293
12,1945

Dezimalteil des Tages
Dezimale Stunden
Stunden, Minuten, Sekunden



Der Forscher Coulerre sah die Sonne erstmalig am 7. März um $12^h 19^m 45^s$ (koordinierte Weltzeit).



Wenn Sie Coulerres Funktion untersuchen, werden Sie feststellen, daß sie bis zu vier reelle Nullstellen – drei positive und eine negative – haben kann. Versuchen Sie weitere positive Lösungen zu finden, indem Sie mit größeren positiven Schätzwerten verwenden.



Tastenfolge

1000

Anzeige

1.100,
Error 6

Zwei größere, positive Schätzwerte
Keine Lösung gefunden.



278,4497

Der letzte untersuchte Schätzwert.
Ein vorheriger Schätzwert.



276,7942

Der Funktionswert ist ungleich Null.
Der Stack wird in die Ausgangsstellung gebracht.



7,8948

Wieder keine Lösung gefunden.
Ungefähr der gleiche Schätzwert.



278,4497

Ein vorheriger Schätzwert.
Der gleiche Funktionswert.



Sie haben ein positives lokales Minimum anstatt einer Nullstelle gefunden. Probieren Sie jetzt die negative Wurzel zu ermitteln.



Tastenfolge

1000
1100

Anzeige

-1.000,0000
-1.100,

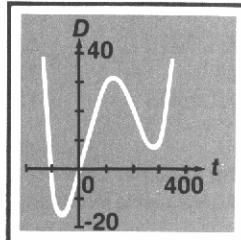
Zwei größere negative Schätzwerte.
Die negative Nullstelle.



-108,9441

Der gleiche vorherige Schätzwert.
Der Funktionswert.

Weiteres Suchen ist unnötig – Sie haben alle vorhandenen Nullstellen gefunden. Die negative Lösung hat keine Bedeutung, da sie außerhalb des Gültigkeitsbereichs der Annäherung für die Deklination liegt. Das Kurvenbild der Funktion bestätigt die gefundenen Resultate.



SUCHE NACH MEHREREN NULLSTELLEN

Viele Gleichungen, mit denen Sie sich befassen werden, haben mehr als eine Nullstelle. Aus diesem Grund wollen wir einige Techniken erläutern, mit deren Hilfe Sie mehrere Nullstellen einer Gleichung bestimmen können.

Die einfachste Methode mehrere Nullstellen zu finden, besteht darin, die jeweiligen Bereichsgrenzen, in denen eine Nullstelle vermutet wird, wiederholt einzugeben. Die eingegebenen Schätzwerte legen den Bereich fest, in denen anfänglich nach Lösungen gesucht wird. Im Abschnitt 8 "Bestimmung der Lösungen einer Gleichung" wurde diese Methode durchweg verwendet. Häufig lassen sich die Nullstellen auf diese Weise ermitteln.

Eine etwas fortgeschrittene Methode wird mit der Abspaltung der Nullstellen bezeichnet. Diese Technik ist angebracht, wenn die Gleichung Charakteristika aufweist, die es **SOLVE** schwierig machen, alle Wurzeln zu finden. Beim Abspalten von Nullstellen werden Wurzeln aus der Gleichung eliminiert. Die Gleichung wird daher so geändert, daß eine schon gefundene Nullstelle keine Lösung mehr ist, daß aber die restlichen Nullstellen weiterhin Lösungen bleiben.

Wenn eine Funktion $f(x)$ eine Nullstelle bei $x = a$ besitzt, dann geht eine neue Funktion $\frac{f(x)}{(x - a)}$ in diesem Bereich nicht gegen Null (sofern a eine einfache Nullstelle von $f(x) = 0$ ist). Anhand dieser Information können Sie eine bekannte Nullstelle abspalten. Ihr Unterprogramm muß lediglich um einige Programmzeilen verlängert werden. Der bekannte Wert der Nullstelle wird (auf zehn signifikante Stellen genau) vom x-Wert subtrahiert und der Funktionswert durch diese Differenz dividiert. In vielen Fällen wird es sich um eine einfache Nullstelle handeln, so daß **SOLVE** für die neue Funktion Lösungen in einem anderen Bereich suchen wird.

Es kann jedoch auch eine vielfache Nullstelle auftreten. In diesem Fall schneidet der Graph $f(x)$ nicht nur die X-Achse, sondern es sind auch die Steigung des Graphen (und eventuell weitere höhere Ableitungen) an dieser Stelle gleich Null. Wenn in Ihrer Gleichung eine vielfache Nullstelle bekannt ist, wird diese durch eine Division durch den oben beschriebenen Faktor nicht abgespalten.

So hat die Gleichung

$$f(x) = x(x - a)^3 = 0$$

beispielsweise eine vielfache Nullstelle bei $x = a$ (mit einer Vielfachheit von 3). Diese Nullstelle wird durch ein Dividieren der Funktion durch $(x - a)$ nicht abgespalten. Vielmehr muß hier durch $(x - a)^3$ dividiert werden.

Beispiel: Ermitteln Sie mit Hilfe der Abspaltung von Nullen die Nullstellen der Gleichung

$$60x^4 - 944x^3 + 3003x^2 + 6171x - 2890 = 0.$$

Unter Verwendung des Horner-Schemas kann die Gleichung in folgende Form gebracht werden:

$$((60x - 944)x + 3003)x + 6171)x - 2890 = 0.$$

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung **PRGM**. Tasten Sie das Unterprogramm zur Berechnung des Polynoms ein.

Tastenfolge	Anzeige
H LBL 2	001–25,13, 2
6	002– 6
0	003– 0
X	004– 61
9	005– 9
4	006– 4
4	007– 4
–	008– 41
X	009– 61
3	010– 3
0	011– 0
0	012– 0

Tastenfolge	Anzeige
3	013- 3
+	014- 51
×	015- 61
6	016- 6
1	017- 1
7	018- 7
1	019- 1
+	020- 51
×	021- 61
2	022- 2
8	023- 8
9	024- 9
0	025- 0
-	026- 41
[h] RTN	027- 25 12

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung RUN. Geben Sie zwei große negative Zahlen als Schätzwerte (z. B. -10 und -20) ein und ermitteln Sie mit Hilfe von **SOLVE** die größte negative Nullstelle.

Tastenfolge	Anzeige
10 CHS ENTER	-10,0000
20 CHS	-20
[f] SOLVE 2	-1,6667
STO 0	-1,6667
[g] R [g] R	4,0000 -06

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung PRGM. Fügen Sie die zur Abspaltung der gefundenen Nullstellen nötige Programmzeilen in Ihr Unterprogramm ein.

Tastenfolge	Anzeige
GTO [] .026	026- 41
X [y]	027- 21
RCL 0	028- 24 0
-	029- 41
[+]	030- 71

	Tastenfolge	Anzeige	Schieben Sie den PRGM RUN -Schalter in Stellung RUN. Verwenden Sie die gleichen anfänglichen Schätzwerte wie beim ersten Mal.
	10 CHS ENTER	-10,0000	Die gleichen anfänglichen Schätzwerte.
	20 CHS	-20,	Die zweite Nullstelle.
	[f] SOLVE 2	0,4000	Die Nullstelle wird zur Abspaltung gespeichert.
	STO 1	0,4000	Der Funktionswert nach Abspaltung der Nullstelle.
	[g] R [g] R	0,0000	
	Tastenfolge	Anzeige	Schieben Sie den PRGM RUN -Schalter in Stellung PRGM und führen Sie die zur Abspaltung der zweiten Nullstelle nötigen Änderungen durch.
	030- 71	71	Die Zeile vor der RTN -Anweisung.
	031- 21	21	x wird in das X-Register geladen.
	032- 24 1	24 1	Abspaltung der zweiten Nullstelle.
	033- 41	41	
	034- 71	71	
	Tastenfolge	Anzeige	Schieben Sie den PRGM RUN -Schalter in Stellung RUN. Verwenden Sie wieder die gleichen anfänglichen Schätzwerte.
	10 CHS ENTER	-10,0000	Die gleichen anfänglichen Schätz- Schätzwerte.
	20 CHS	-20,	Die dritte Nullstelle.
	[f] SOLVE 2	8,4999	Die Nullstelle wird zur Abspaltung gespeichert.
	STO 2	8,4999	
	[g] R [g] R	-1,0929 -07	Der Funktionswert nach Abspaltung der Nullstelle ist nahe Null.
	Tastenfolge	Anzeige	

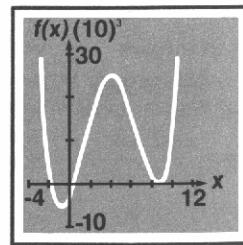
Schieben Sie wieder den **PRGM** **RUN**-Schalter in Stellung PRGM und führen Sie die zur Abspaltung der dritten Nullstelle nötigen Änderungen durch.

Tastenfolge	Anzeige
GTO 0 .034	034- 71 Die Zahl vor der RTN -Anweisung.
x:y	035- 21 x wird in das X-Register geladen.
RCL 2	036- 24 2 } Abspaltung der dritten Nullstelle.
–	037- 41 }
±	038- 71 }

Schieben Sie den **PRGM** **RUN**-Schalter in Stellung RUN und ermitteln Sie die vierte Nullstelle.

Tastenfolge	Anzeige
10 CHS ENTER	-10,0000 }
20 CHS	-20,
f SOLVE 2	8,5001
STO 3	8,5001
g R g R	-0,0009
	Der Funktionswert nach Abspaltung der Nullstelle ist nahe Null.

Für dieses Polynom 4-ten Grades haben Sie vier Nullstellen gefunden. Dabei haben Sie jedesmal die gleichen anfänglichen Schätzwerte verwendet. Die zwei letzten sehr nahe beieinander liegenden Lösungen sind tatsächlich eine Nullstelle (mit einer Vielfachheit von 2). Daher wurde diese Nullstelle nicht eliminiert, als Sie die Abspaltung durchführen wollten. (Durch Rundungsfehler ergeben sich für die ursprüngliche Funktion kleine positive und negative Werte für x im Bereich von 8,4999 bis 8,5001. Wenn x = 8,5 gesetzt wird, ist der Funktionswert genau Null.)



Im allgemeinen wird Ihnen nicht die Vielfachheit einer Nullstelle, die sie abspalten wollen, im Voraus bekannt sein. Wenn Sie versucht haben, eine Nullstelle abzuspalten und **SOLVE** diese Wurzel dennoch ein weiteres Mal findet, können Sie auf unterschiedliche Weise fortfahren:

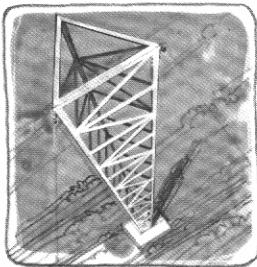
- Verwenden Sie andere anfängliche Schätzwerte mit der Funktion, aus der Sie eine Nullstelle abgespalten haben, um eine andere zu ermitteln.
- Wiederholen Sie die Abspaltung der Nullstelle, um die mehrfache Nullstelle zu eliminieren.
Wenn die Vielfachheit einer Nullstelle nicht bekannt ist, müssen Sie diesen Vorgang vielleicht mehrmals wiederholen.
- Untersuchen Sie die Funktion, aus der Sie eine Nullstelle abgespalten haben, in deren Nähe. Wenn das Kurvenbild an dieser Stelle parallel zur X-Achse verläuft, deutet das entweder auf eine weitere Nullstelle oder eine größere Vielfachheit hin.
- Analysieren Sie die ursprüngliche Funktion. Es ist vielleicht möglich, das Verhalten der Funktion in der Umgebung der bekannten Wurzel zu bestimmen. (So lässt sich mittels der Taylorschen Reihe die Vielfachheit einer Nullstelle feststellen.)

BESTIMMUNG VON LOKALEN EXTREMWERTEN EINER FUNKTION

Verwendung der Ableitung

Zur Bestimmung lokaler Maxima und Minima einer Funktion wird üblicherweise die Ableitung einer Funktion verwendet. Durch die Ableitung wird die Steigung einer Funktion beschrieben. Eine Funktion besitzt einen möglichen Extremwert dort wo die Ableitung gleich Null ist. (Wenn es auch seltener bei vernünftigen Funktionen vorkommt, können Extremwerte auch dort existieren, wo die Ableitung gegen ∞ geht oder nicht definiert ist.) Wenn Sie die Ableitung einer Funktion in geschlossener Form ausdrücken können, lässt sich mit **SOLVE** berechnen, wo die Ableitung verschwindet – womit ein Maximum der Funktion gefunden ist.

Beispiel: Der Radiotechniker Felix Leiter benötigt für den Entwurf eines vertikalen Antennenturms den Winkel, unter dem die relative Feldstärke vom Turm aus den größten negativen Wert besitzt. Die durch den Antennenmast erzeugte relative Feldstärke lässt sich durch



$$E = \frac{\cos(2\pi h \cos 0) - \cos(2\pi h)}{[1 - \cos(2\pi h)] \sin 0}$$

ausdrücken.

Hier ist
E die relative Feldstärke
h die Höhe der Antenne in Wellenlängen
O der Winkel von der Vertikalen im Bogenmaß.

Die Höhe der Antenne wird in diesem Entwurf mit 0,6 Wellenlängen festgesetzt.

Lösung: Die Feldstärke E wird nach O differenziert. Der gesuchte Winkel O ist der, für den die Ableitung verschwindet.

Um das Programm und die Rechenzeit kurz zu halten, speichern Sie die folgenden Konstanten im Speicherregister:

$R0 = 2\pi h$ und wird in R_0 gespeichert.

$R1 = \cos(2\pi h)$ und wird in R_1 gespeichert.

$R2 = 1/(1-\cos(2\pi h))$ und wird in R_2 gespeichert.

Die Differentiation der Feldstärke E nach O ergibt folgenden Ausdruck:

$$\frac{dE}{dO} = R2 \left[R0 \sin(R0 \cos 0) - \frac{\cos(R0 \cos 0) - R1}{\sin 0 \tan 0} \right]$$

- 3 Schieben Sie den **PRGM** **RUN**-Schalter in Stellung **PRGM** und tasten Sie das Unterprogramm zur Berechnung der Ableitung ein.
- 3 Tastenfolge Anzeige

f CLEAR	PRGM	000-
h LBL	0	001- 25,13, 0
f COS		002- 14 8
RCL	0	003- 24 0
x		004- 61
f COS		005- 14 8
RCL	1	006- 24 1
-		007- 41
x²y		008- 21
f SIN		009- 14 7
÷		010- 71
x²y		011- 21
f TAN		012- 14 9
÷		013- 71
CHS		014- 32
x²y		015- 21
f COS		016- 14 8
RCL	0	017- 24 0
x		018- 61
f SIN		019- 14 7
RCL	0	020- 24- 0
x		021- 61
+		022- 51
RCL	2	023- 24 2
x		024- 61
h RTN		025- 25 12

- 3 Schieben Sie jetzt den **PRGM** **RUN**-Schalter in Stellung **RUN**. Berechnen und speichern Sie im Winkel-Modus Bogenmaß die drei Konstanten.
- 3 Tastenfolge Anzeige

9 **RAD** 0,0000 Schalten Sie auf den Winkel-Modus Bogenmaß um (die Anzeige wird als gelöscht vorausgesetzt).

2	H	T	X	6,2832
.6	X	STO	0	3,7699
f	COS	STO	1	-0,8090
CHS	1	+		1,8090
H	Vx	STO	2	0,5528

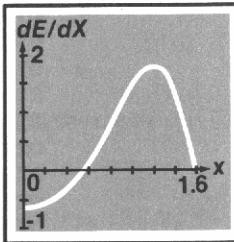
Die Konstante R 0.
Die Konstante R 1.
Die Konstante R 2.

Die relative Feldstärke erreicht ein Maximum bei einem Winkel von 90° (im rechten Winkel zur Antenne). Um das Minimum zu finden, verwenden Sie anfängliche Schätzwerte näher Null, wie z. B. die äquivalenten Werte in Bogenmaß von 10° und 60° .

Tastenfolge Anzeige

10	g	→R	0,1745	{ Die anfänglichen Schätzwerte.
60	g	→R	1,0473	
f	SOLVE	0	0,4899	
g	R↓	g	R↓	Bei diesem Winkel ist die Steigung gleich Null.
			-5,5279	
			-10	
f	R↑	f	R↑	Die Steigung, wenn für 0 der gefundene Winkel eingesetzt wird.
f	→D		0,4899	Der Stack wird umgeordnet.
			28,0680	Der Winkel in Grad.

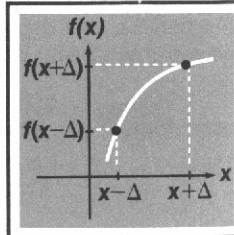
Die relative Feldstärke erreicht den größten negativen Wert bei einem Winkel von $28,0680^\circ$ von der Vertikalen.



Verwendung einer angenäherten Steigung.

Die Ableitung einer Funktion kann auch durch eine Näherung bestimmt werden. Wenn Sie Momentswerte einer Funktion für zwei dicht bei x liegenden Werte (nämlich $x + \Delta$ und $x - \Delta$) bilden, können Sie die Steigung des Kurvenbildes folgendermaßen berechnen:

$$s = \frac{f(x+\Delta) - f(x-\Delta)}{2 \Delta}$$



Die Genauigkeit der Steigung hängt sowohl von dem Inkrement Δ wie auch der Art der Funktion ab. Kleine Werte von Δ ergeben eine bessere Annäherung für die Ableitung. Sehr dicht beieinanderliegende x-Werte können zu Rundungsfehlern führen. Ein x-Wert (zwischen x_1 und x_2) für den die Steigung gleich Null ist, deutet auf einen lokalen Extremwert der Funktion hin.

Beispiel: Lösen sie das vorherige Problem, ohne die Gleichung für die Ableitung $dE/d\theta$ zu verwenden.

Schieben Sie den **PRGM** **RUN** -Schalter in Stellung **PRGM** und tasten Sie zwei Unterprogramme ein. Mit einem Unterprogramm wird die Ableitung der Feldstärke geschätzt und mit dem anderen wird die Funktion der Feldstärke E berechnet. In dem folgenden Unterprogramm wird die Steigung zwischen $\theta+0,001$ und $\theta-0,001$ (dies entspricht etwa einem Bereich von $0,1^\circ$) berechnet.

Tastenfolge Anzeige

h	LBL	A	001-25,13,11
	EEX		002- 33
	CHS		003- 32
	3		004- 3
	+		005- 51
	ENTER+		006- 31
	GSB	B	007- 13 12
	x:y		008- 21
	EEX		009- 33
	CHS		010- 32
	3		011- 3
	-		012- 41
	ENTER+		013- 31
	GSB	B	014- 13 12
	-		015- 41
	2		016- 2
	EEX		017- 33
	CHS		018- 32
	3		019- 3
	+		020- 71
	h	RTN	021- 25 12
	h	LBL	022-25,13,12
	f	COS	023- 14 8
	RCL	0	024- 24 0
	x		025- 61
	f	COS	026- 14 8

Berechnung von E für $\theta+0,001$

Berechnung von E für $\theta-0,001$

RCL 1	027-	24	1
-	028-		41
X ² Y	029-		21
SIN	030-	14	7
+	031-		71
RCL 2	032-	24	2
X	033-		61
RTN	034-	25	12

Schieben Sie den PRGM RUN -Schalter in Stellung RUN. Der Winkel-Modus Bogenmaß und die drei Konstanten in R_0 , R_1 und R_2 des vorherigen Beispiels werden weiter benötigt. Geben Sie die gleichen Anfangswerte wie vorhin ein und drücken Sie

Tastenfolge Anzeige

10 g R	0,1745	} Die anfänglichen Schätzwerte.
60 g R	1,0472	
f SOLVE A	0,4899	
g R R	0,0000	Bei diesem Winkel ist die Steigung gleich Null.
f R f R	0,4899	Die Steigung, wenn für 0 der gefundene Winkel eingesetzt wird. Der Stack kehrt in seine Ausgangsstellung zurück.
ENTER ENTER B	-0,2043	Die minimale Feldstärke wird mittels des Funktions-Unterprogramms berechnet.
	0,4899	Der 0-Wert wird zurückgerufen.
f	28,0679	Der Winkel in Grad.

Diese Annäherung der Ableitung ergibt eine minimale Feldstärke von -0,2043 bei einem Winkel von 28,0679°. (Dieses Ergebnis unterscheidet sich von der vorher gefundenen Lösung um 0,0001°.)

ITERATIVE BERECHNUNG

Eine dritte Methode läßt sich anwenden, wenn es nicht günstig ist, die Ableitung zu berechnen. Diese Methode dauert länger, weil die **SOLVE** -Funktion wiederholt verwendet werden muß. Um lokale Extremwerte einer Funktion $f(x)$ zu finden, wird eine neue Funktion

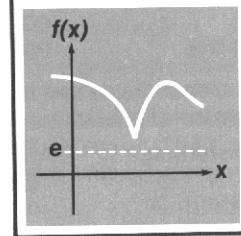
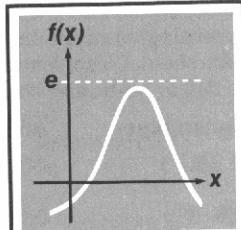
$$g(x) = f(x) - e$$

definiert, wobei e jenseits des abgeschätzten Extremwertes von $f(x)$ liegt und sich von diesen nur geringfügig unterscheidet. Bei einer richtigen Wahl von e geht $g(x)$ gegen Null in der Nähe des Extremwertes von $f(x)$, ohne gleich Null zu werden. Untersuchen sie die Funktion $g(x)$ mittels **SOLVE** in der Nähe des Extremwertes. Das gewünschte Ergebnis ist ERROR 6.

- Wenn ERROR 6 in der Anzeige erscheint, enthält das X-Register den x-Wert in der Nähe des Extremwertes. Die Zahl im Z-Register gibt Aufschluß darüber, wie weit e von dem Extremwert von $f(x)$ entfernt ist. Der Wert e wird jetzt geändert, so daß er näher am Extremwert liegt, ohne diesem gleich zu sein.
- Die geänderte Funktion $g(x)$ wird in der Nähe des gefundenen x-Wertes mit **SOLVE** untersucht. Dieser Vorgang wird so lange wiederholt, bis aufeinanderfolgende x-Werte sich nicht wesentlich von einander unterscheiden.
- Wenn eine Nullstelle von $g(x)$ gefunden wird, liegt entweder der Wert e nicht jenseits des Extremwertes der Funktion oder **SOLVE** hat einen anderen Bereich gefunden, in dem $f(x)$ gleich e ist. Der Wert e muß in der Nähe, aber jenseits, des Extremwertes von $f(x)$ liegen. Ändern Sie e entsprechend und führen Sie **SOLVE** nochmals aus. Vielleicht ist es auch möglich diese Nullstelle durch eine entsprechende Modifikation von $g(x)$ zu eliminieren.

Beispiel: Lösen Sie das vorhergehende Beispiel ohne die Ableitung der relativen Feldstärke E zu berechnen.

Lösung: Das Unterprogramm zur Berechnung von E und die Konstanten wurden bereits im vorausgegangenen Beispiel eingegeben.



Schieben Sie den **PRGM** -Schalter in Stellung **PRGM**. Tasten Sie ein Unterprogramm ein, das einen geschätzten Extremwert von der Feldstärke E subtrahiert. Dieser Extremwert sollte in einem Register gespeichert werden, so daß er bei Bedarf leicht geändert werden kann.

Tastenfolge Anzeige

h LBL 1	001–25,13, 1	Das Unterprogramm beginnt mit einer (LBL)-Anweisung.
GSB B	002– 13 12	Berechnung von E.
RCL 9	003– 24 9	Der geschätzte Extremwert wird
-	004– 41	abgezogen.
h RTN	005– 25 12	

Schieben Sie den **PRGM** -Schalter in Stellung **RUN**. Ermitteln Sie die minimale Feldstärke, indem Sie manuell einige Werte der Funktion bilden.

Tastenfolge	Anzeige
10 g R	0,1745
ENTER+ B	-0,1029
30 g R	0,5236
ENTER+ B	-0,2028
50 g R	0,8727
ENTER+ B	0,0405

Funktionswerte bei
10°, 30°, 50°, ...

Verwenden Sie anhand dieser Ergebnisse einen geschätzten Extremwert von -0,25 und führen Sie **SOLVE** mit anfänglichen Schätzwerten von 10° und 30° (in Bogenmaß) aus.

Tastenfolge	Anzeige
.25 CHS STO 9	-0,2500
.2 ENTER+	0,2000
.6	0,6
f SOLVE 1	Error 6
CLX STO 4	0,4849
g R+ STO 5	0,4698
g R+ 9	0,0457
9 X	0,0411
STO + 9	0,0411
RCL 4	0,4849
ENTER+ ENTER+ B	-0,2043

Der Extremwert wird gespeichert.
Die anfänglichen Schätzwerte.
Keine Nullstelle gefunden.
Der letzte Schätzwert von θ wird gespeichert.
Der vorletzte Schätzwert von θ wird gespeichert.
Abstand zum Extremwert.
Abstand wird mit dem Faktor 0,9 multipliziert und zu dem Extremwert addiert.
Der Schätzwert θ wird zurückgerufen.
Berechnung der Feldstärke E.

CLX	0,0000	Der zweite Schätzwert θ wird zurückgerufen. Der erste bleibt im Y-Register erhalten.
RCL 5	0,4698	Keine Nullstelle gefunden. Der Schätzwert θ .
f SOLVE 1	Error 6	Der vorherige Schätzwert θ .
CLX	0,4898	Der Schätzwert θ wird zurückgerufen.
x_y	0,4893	Berechnung der Feldstärke E.
x_y	0,4898	Der Wert θ wird zurückgerufen.
ENTER+ ENTER+ B	-0,2043	Umwandlung in dezimale Grad.
x_y	0,4898	
f ↵D	28,0660	
		Mit der zweiten Iteration werden zwei Schätzwerte von θ gefunden, die sich in der vierten Stelle nach dem Komma unterscheiden. Die Feldstärken Es sind auf vier Stellen nach dem Komma genau. Die Ergebnisse lassen, wenn wir an dieser Stelle abbrechen, auf eine minimale Feldstärke von -0,2043 bei einem Winkel von 28,0660° schließen. (Dieser Winkel unterscheidet sich von dem vorherigen Resultat um 0,002°.)
		BEGRENZUNG DER RECHENZEIT
		Zuweilen wollen Sie die für den SOLVE -Algorithmus benötigte Rechenzeit begrenzen. In diesem Fall bieten sich zwei Techniken an – zählen der Iterationen und Festlegen der Genauigkeit.
		ZÄHLEN DER ITERATIONEN
		Wenn mit SOLVE die Lösung einer Gleichung gesucht wird, werden im Durchschnitt wenigsten zwölfmal Stützwerte der Funktion gebildet. Zuweilen benötigt SOLVE aber mehr als hundert Stützwerte. Da für jeden neuen Schätzwert Ihr Unterprogramm durchlaufen wird, kann dieses die Anzahl der Iterationen zählen und auch begrenzen. Dies läßt sich mittels einer ISG -Anweisung einfach durchführen, indem im I-Register die Anzahl der durchgeführten Iterationen summiert werden. Wenn Sie vor der Verwendung der SOLVE -Funktion eine geeignete Zahl in das I-Register laden, kann der SOLVE -Algorithmus unterbrochen werden, wenn dieser Grenzwert erreicht ist. (Die ISG -Anweisung ist auf Seite 140 beschrieben.)

FESTLEGEN EINER TOLERANZGRENZE

Sie können die zur Lösung einer Gleichung benötigte Rechenzeit verkürzen, indem Sie eine der Funktionen angemessene Genauigkeit festlegen. Das Ergebnis Ihres Unterprogramms wird in einen Funktionswert von Null abgeändert, wenn der berechnete Funktionswert die angegebene Toleranzgrenze unterschreitet. Dadurch wird der **SOLVE**-Algorithmus abgebrochen und die Näherung der Nullstelle angezeigt.

Die Genauigkeit die Sie festlegen, sollte einem Wert entsprechen, der entweder praktisch vernachlässigbar klein ist, oder der die Genauigkeit der Berechnungen wiederspiegelt. Auf diese Weise wird durch die Verkürzung der Rechenzeit eine Genauigkeit des Resultats vermieden, die bei dem Problem gar nicht gerechtfertigt wäre. (Beispiele dieser Methode sind im folgenden und auf Seite 190 gegeben.)

VERWENDUNG VON **SOLVE** MIT

Beispiel: Im Falle eines phasenmodulierten Funksignals ist die Amplitude des Trägersignals proportional zu $J_0(x)$, der Besselschen Funktion 0-ter Ordnung erster Gattung, wobei x der Modulationsindex ist. Was ist der kleinste Modulationsindex für den das Trägersignal unterdrückt wird, d. h., die Amplitude gleich Null ist?

Lösung: Der gesuchte Index ist der Wert von x für den

$$J_0(x) = \int_0^{\pi} \frac{\cos(x \sin \theta)}{\pi} d\theta = 0 \quad \text{gilt}$$

Mit **SOLVE** lässt sich dieser Wert finden. Die Funktion $J_0(x)$ wird mittels **plot** berechnet:

Die Fehlerabschätzung, die bei der Näherung von $J_0(x)$ entsteht, ist im Y-Register enthalten. Wenn der Absolutwert von $J_0(x)$ diese Fehlerabschätzung unterschreitet, kann $J_0(x)$ als Null betrachtet werden. Wenn Sie sich dieser Information bedienen, vermeiden Sie, daß mit **SOLVE** ein Ergebnis mit ungerechtfertigter Genauigkeit erzielt wird.

Schieben Sie den PRGM [RUN]-Schalter in Stellung PRGM. Tasten Sie ein Unterprogramm zur Berechnung von $J_0(x)$ und ein weiteres zur Berechnung der zu integrierenden Funktion ein.

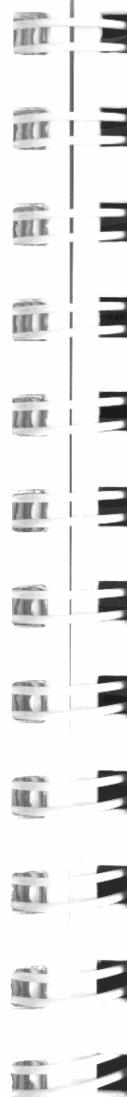
Tastenfolge	Anzeige
f CLEAR PRGM	000– Löschen des Programmspeichers.
h LBL A	001–25,13,11 Das Unterprogramm beginnt mit einer LBL -Anweisung.
STO 0	002– 23 0 Das Argument x wird gespeichert.
0	003– 0 } Die Integrationsgrenzen.
h T	004– 25 73
f J 3	005–14,72, 3 Berechnung von $J_0(x)$.
h ABS	006– 25 34
f X^y	007– 14 41 f(x) wird gleich Null gesetzt.
CLX	008– 34 } wenn $J_0(x) \leq$ Fehlerabschätzung ist.
g X#0	009– 15 61 Der Wert von $J_0(x)$ wird
h LST x	010– 25 0 } zurückgerufen, wenn der Wert
h RTN	011– 25 12 ungleich Null ist.
h LBL 3	012–25,13, 3
f SIN	013– 14 7
RCL 0	014– 24 0
x	015– 61 Berechnung der Funktion, die zu
f COS	016– 14 8 integrieren ist.
h T	017– 25 73
±	018– 25 12
h RTN	019– 25 12

Um die gewünschte Lösung schneller zu finden, sollten Sie anfänglich für die Integration das Anzeigeformat **SCI** 0 festlegen. Sobald ein angenäherter Wert gefunden ist, können Sie die Genauigkeit erhöhen (indem Sie **SCI** 3 angeben). Jetzt wird unter Verwendung der genaueren Funktion **SOLVE** auf die Lösung der Gleichung angesetzt. Auf diese Weise wird vermieden, daß mit hoher Genauigkeit in einem Bereich von x integriert wird, der die Nullstelle gar nicht enthält. Das führt zu einer wesentlichen Zeittersparnis.

Schieben Sie den **PRGM** ■■■■■ **RUN**-Schalter in Stellung **RUN** und führen Sie die folgenden Schritte aus. Bedenken Sie, daß **SOLVE** viele male Stützwerte Ihrer Funktion bildet, und daß **■■■■■** häufig eine Minute oder länger zur Berechnung eines Integrals benötigt. Aus diesen Gründen wird der Rechner bei den folgenden **SOLVE**-Operationen 3 bis 6 Minuten beschäftigt sein.

Tastenfolge	Anzeige	
f SCI 0	0,	00 Angabe der Genauigkeit. Die Anzeige wird als gelöscht vorausgesetzt.
g RAD 0 ENTER+	0, 0,	00 Die anfänglichen Schätzpunkte bewirken eine Suche in der Nähe Null.
1	1,	
f SOLVE A	2,	00 Die gesuchte Nullstelle.
f SCI 3	2,480	00 Angabe einer erhöhten Genauigkeit.
2.4 ENTER+	2,400	00 Die anfänglichen Schätzpunkte liegen in der Nähe der ersten Näherung.
2.5	2,5	
f SOLVE A	2,405	00 Die gewünschte Nullstelle.
f FIX 4	2,4049	Anzeigeformat FIX 4. $J_0(x)$ ist kleiner als die Fehlerabschätzung.
g R↓ g R↓	0,0000	
g R↓	0,0001	Fehlerabschätzung von $\text{R}^{\frac{1}{2}}$

Ein Modulationsindex von 2,4049 bewirkt, daß die Amplitude des Trägersignals zu 99,99% unterdrückt wird. (Das heißt, daß die Amplitude kleiner als 0,0001 des Maximums beträgt).



ANHANG B: DIE ARBEITSWEISE VON $\int_{\text{X}}^{\text{Y}}$

Der $\text{R}^{\frac{1}{2}}$ -Algorithmus berechnet das Integral einer Funktion $f(x)$, indem er einen gewichteten Mittelwert der Funktionswerte für eine große Anzahl von x -Werten, die als Stützstellen bezeichnet werden, innerhalb des Integrationsintervalls bildet.

Die Genauigkeit des Resultats eines derartigen Prozesses hängt von der Anzahl der Stützstellen ab: je größer die Anzahl dieser Punkte, desto größer die Genauigkeit. Wenn $f(x)$ für eine unendliche Anzahl von Stützstellen berechnet würde, könnte der Algorithmus ein genaues Ergebnis berechnen falls man Ungenauigkeit der berechneten Funktion $f(x)$ außer Betracht läßt.

Die Berechnung der Funktion für eine unendliche Anzahl Stützstellen würde allerdings sehr lange, in der Tat ewig dauern. Glücklicherweise ist dies aber nicht notwendig, da die maximale Genauigkeit des zu berechnenden Integrals durch die Genauigkeit der berechneten Funktionswerte begrenzt ist. Schon mit einer endlichen Anzahl von Stützstellen kann der Algorithmus ein Integral so genau berechnen, wie es die mit $f(x)$ verbundenen Fehler rechtfertigen.

Der $\text{R}^{\frac{1}{2}}$ -Algorithmus beginnt mit nur wenigen Stützstellen, die eine Näherung mit einer relativ geringen Genauigkeit ergeben. Wenn die Genauigkeit dieser Näherung nicht der möglichen Genauigkeit von $f(x)$ entspricht, wird der Algorithmus mit einer größeren Anzahl von Punkten wiederholt. Die Iterationen werden so lange durchgeführt, wobei jedesmal eine etwa doppelte Anzahl von Stützstellen berücksichtigt wird, bis die Näherung so genau ist, wie es der in $f(x)$ enthaltene Fehler rechtfertigt. Die mit dem Endresultat verbundene Fehlerschätzung wird von dem Anzeigeformat abgeleitet, das ein Maß für die Ungenauigkeit der Funktion darstellt. Der Zusammenhang zwischen dem Anzeigeformat, dem Fehler einer Funktion und der Fehlerabschätzung der Annäherung des Integrals werden noch später in diesem Anhang behandelt. Nach jeder Iteration vergleicht der Algorithmus die berechnete Näherung mit den Näherungswerten, die während der zwei vorangegangenen Iterationen ermittelt wurden. Wenn die Differenz zwischen einem dieser drei Näherungswerte und den beiden anderen kleiner als die Fehlerabschätzung des Endresultats ist, wird der Algorithmus beendet. Der letzte Näherungswert und die Fehlerabschätzung werden dabei in das X- und Y-Register geladen.

Da der -Algorithmus so aufgebaut ist, daß es höchst unwahrscheinlich ist, daß der Fehler in jeder von drei aufeinanderfolgenden Näherungen – also der Unterschied zwischen dem tatsächlichen Integral und der Näherung – jedesmal geringer ist als die Differenz der Näherungswerte untereinander, wird der Fehler der letzten Näherung geringer als die Fehlerabschätzung sein.* Obwohl wir also nicht den Fehler der Näherung kennen, können wir uns dennoch darauf verlassen, daß der Fehler kleiner als die angezeigte Fehlerabschätzung der Näherung ist. Diese Fehlerabschätzung bildet somit eine obere Grenze der Differenz zwischen dem berechneten Näherungswert und dem eigentlichen Integral.

GENAUIGKEIT, FEHLERABSCHÄTZUNG UND RECHENZEIT

Die Genauigkeit einer -Näherung wird nicht unbedingt erhöht, wenn Sie die Anzahl der Stellen im Anzeigeformat nur um eins erhöhen. In gleicher Weise kann sich die Rechenzeit verändern, wenn Sie das Anzeigeformat wechseln, obwohl dies nicht der Fall sein muß.

Beispiel: Die Besselsche Funktion 4-ter Ordnung erster Gattung läßt sich folgendermaßen ausdrücken:

$$J_4(x) = \frac{1}{\pi} \int_0^{\pi} \cos(4\theta - x \sin \theta) d\theta.$$

Berechnen Sie das Integral $J_4(1) = \frac{1}{\pi} \int_0^{\pi} \cos(4\theta - \sin \theta) d\theta$.



Schieben Sie erst den -Schalter in Stellung PRGM und tasten Sie das Unterprogramm zur Berechnung von $f(\theta) = \cos(4\theta - \sin\theta)$ ein.



Tastenfolge	Anzeige
	000-
0	001-25,13, 0
4	002- 4
	003- 61
	004- 21
	005- 14 7
-	006- 41
	007- 14 8
	008- 25 12

Schieben Sie jetzt den -Schalter in Stellung RUN und geben Sie die Integrationsgrenzen in die X- und Y-Register ein. Wählen Sie den Winkel-Modus Bogenmaß und das Anzeigeformat 2. Dann drücken Sie 0 um das Integral zu berechnen.



Tastenfolge	Anzeige
0	0,0000
	3,1416
2	3,14
0	7,79 -03
	1,45 -03

Geben Sie die untere Integrationsgrenze in das Y-Register ein.
Geben Sie die obere Integrationsgrenze in das X-Register ein.
Achten Sie darauf, daß der Winkel-Modus Bogenmaß eingeschaltet ist.
(Dieser Schritt ist nicht erforderlich, wenn Sie den Rechner inzwischen weder ausgeschaltet, noch den Winkel-Modus seit dem letzten Umschalten auf Bogenmaß verändert haben.)

Wählen Sie das Anzeigeformat 2.
Das angenäherte Integral mit 2.
Die Fehlerabschätzung mit 2.

*Voraussetzung ist, daß es sich um eine ausreichend glatte Funktion handelt. Mit diesem Aspekt werden wir uns noch eingehend befassen.

Die Fehlerabschätzung läßt vermuten, daß keine der angezeigten Ziffern des Ergebnisses als genau betrachtet werden kann. Tatsächlich ist die Näherung, wie es schon beim letzten Ergebnis in Abschnitt 9 der Fall war, sehr viel genauer als es die Fehlerabschätzung andeutet.

Tastenfolge	Anzeige
x^y	7,79 -03 Der Näherungswert steht im X-Register.
h MANT	7785820888 Alle zehn Stellen der Näherung mit SCI 2 werden angezeigt.

Das Integral ist auf fünf Stellen nach dem Komma genau $7,7805 \times 10^{-3}$. Der Fehler bei dieser Näherung ist also etwa $(7,7805) \times 10^{-3} = 5,3 \times 10^{-6}$. Dieser Fehler ist beträchtlich kleiner als die Fehlerabschätzung $1,45 \times 10^{-3}$. Die Fehlerabschätzung stellt nur eine obere Grenze des Fehlers bei der Näherung dar; der tatsächliche Fehler wird im allgemeinen kleiner sein.

Jetzt wollen wir das Integral mit SCI 3 berechnen und die Ergebnisse mit den mit SCI 2 durchgeföhrten Berechnungen vergleichen.

Tastenfolge	Anzeige
f SCI 3	7,786 -03 Das Anzeigeformat SCI 3.
9 R_↓ 9 R_↑	3,142 00 Der Stack wird zyklisch nach unten verschoben bis die obere Integrationsgrenze im X-Register steht.
f 0	7,786 -03 Die Näherung des Integrals mit SCI 3
x^y	1,448 -04 Die Fehlerabschätzung mit SCI 3
x^y	7,786 -03 Der Näherungswert steht in der Anzeige.
h MANT	7785820888 Alle zehn Stellen der Näherung werden mit SCI 3 angezeigt.

Alle zehn Ziffern der Näherung mit SCI 2 und SCI 3 sind identisch; die Genauigkeit der Näherung mit SCI 3 ist nicht höher als mit SCI 2, obwohl die Fehlerabschätzung mit SCI 3 geringer als mit SCI 2 ist. Wie erklärt sich das? Bedenken Sie, daß die Genauigkeit einer Näherung in erster Linie von der Anzahl der Stützstellen abhängt, an denen die Funktion f(x) bestimmt wurde. Der -Algorithmus wird mit zunehmender Zahl von Stützstellen wiederholt ausgeführt, bis der Unterschied zwischen drei aufeinanderfolgenden Näherungen kleiner als die Fehlerabschätzung ist, die von dem Anzeigeformat abgeleitet wird. Nach einer beliebigen Iteration kann es durchaus sein, daß die Differenz unter den Näherungswerten schon so klein ist, daß sie auch unter der um einen Faktor 10 reduzierten Fehler schranke liegen würde.

In solchen Fällen berücksichtigt der Algorithmus keine zusätzliche Stützstellen, wenn Sie die Genauigkeit um eine weitere Stelle im Anzeigeformat erhöhen, und das Ergebnis identisch mit einer Näherung ist, die mit einer größeren Ungenauigkeit berechnet wurde.

Vielleicht haben Sie bei der Berechnung dieser zwei Integrale bemerkt, daß die Rechenzeit mit SCI 3 genauso lang wie mit SCI 2 war. Der Grund hierfür ist, daß die zur Berechnung eines Integrals benötigte Zeit von der Anzahl der Stützstellen abhängt, die zur Bestimmung einer Näherung mit ausreichender Genauigkeit erforderlich sind. Für die Näherung mit SCI 3 verwendete der Algorithmus nicht mehr Stützstellen als mit SCI 2, so daß die Rechenzeit in diesem Fall nicht länger wurde.

Es passiert jedoch häufig, daß eine zusätzliche Stelle im Anzeigeformat zur Folge hat, daß die Funktion für eine größere Anzahl von Stützstellen bestimmt werden muß, womit sich die Rechenzeit verlängert. Wir wollen dasselbe Integral jetzt mit SCI 4 berechnen.

Tastenfolge	Anzeige
f SCI 4	7,7858 -03 Das Anzeigeformat SCI 4.
9 R_↓ 9 R_↑	3,1416 00 Der Stack wird zyklisch nach unten verschoben bis die obere Integrationsgrenze im X-Register steht.
f 0	7,7807 -03 Die Näherung des Integrals mit SCI 4.

Diese Näherung dauerte etwa doppelt so lange wie die Berechnungen mit **SCI** 3 oder **SCI** 2. Hier mußte der Algorithmus etwa zweimal so viele Funktionswerte berechnen, um eine ausreichende Genauigkeit zu erzielen. Aber unsere Geduld wurde auch belohnt: Das Ergebnis ist fast um zwei zusätzliche Stellen nach dem Komma genauer, als die beiden Resultate, die mit nur der Hälfte der Stützstellen berechnet wurde.

Die oben ausgeführten Beispiele zeigen, daß die Wahl des Anzeigeformats sich zwar auf die Genauigkeit des Näherungswertes eines Integrals auswirken kann aber dies nicht der Fall sein muß. Ob sich die Genauigkeit verändert, hängt von der jeweiligen Funktion ab und läßt sich im allgemeinen nur durch Probieren feststellen.

Hinzu kommt, daß eine größere Genauigkeit auf Kosten der Rechenzeit geht, die dann etwa zweimal so lang ist.

Diese Wechselwirkung zwischen Genauigkeit und Rechenzeit müssen Sie berücksichtigen, wenn Sie die Genauigkeit erhöhen, um ein eventuell besseres Resultat zu erreichen.

Anmerkung: Die für die Berechnung eines Integrals erforderliche Zeit hängt nicht nur von der im Anzeigeformat festgelegten Stellen, sondern in gewisser Weise auch von den Integrationsgrenzen ab. Wenn die Berechnung eines Integrals übermäßig lange dauert, es kann daran liegen, daß das Integrationsintervall im Verhältnis zu gewissen Eigenschaften der zu integrierenden Funktion zu groß ist. Im allgemeinen können Sie jedoch die Wirkung des Integrationsintervalls auf die Rechenzeit außer Betracht lassen. Im Verlauf dieses Anhangs werden wir uns noch mit Situationen, in denen die Integrationsgrenzen die Rechenzeit übermäßig verlängern, und mit Techniken, die in diesen Fällen Abhilfe schaffen können, befassen.

DIE GENAUIGKEIT DER ZU INTEGRIERENDEN FUNKTION

Die Genauigkeit des mit  bestimmten Integrals hängt von der Genauigkeit der mit ihrem Unterprogramm berechneten Funktion ab. Diese Genauigkeit, die Sie mit dem Anzeigeformat festlegen, hängt in erster Linie von drei Faktoren ab:

1. Die Genauigkeit der empirischen Konstanten in der Funktion.



2. Die Genauigkeit, mit der die Funktion v. V. einen physikalischen Zusammenhang wiedergibt.
3. Das Ausmaß der Rundungsfehler, die bei den internen Berechnungen entstehen.

FUNKTIONEN, DIE EINEN PHYSIKALISCHEN ZUSAMMENHANG WIEDERGESEN

Die Funktionen, die wir bis jetzt in Abschnitt 9 und in diesem Anhang integriert haben – $\cos(\sin \theta)$, $\cos(\theta - \sin \theta)$, $\cos(4\theta - \sin \theta)$, und $(\sin x)/x$ – sind Beispiele reiner mathematischer Funktionen. In diesem Zusammenhang bedeutet dies, daß die Funktionen keine empirischen Konstanten enthalten, und daß weder die Variablen noch die Integrationsgrenzen tatsächliche physikalische Größen darstellen. Für solche Funktionen können Sie jede beliebige Zahl (bis zu 9) im Anzeigeformat festlegen, um ein Integral mit der gewünschten Genauigkeit zu erhalten.*

Sie haben nur die unvermeidliche Wechselwirkung zwischen Genauigkeit und Rechenzeit zu berücksichtigen.

Wenn Sie dagegen Funktionen integrieren, die einen physikalischen Zusammenhang darstellen, sind weitere Faktoren zu berücksichtigen. Sie sollten sich vor allem fragen, ob die Genauigkeit der Funktion die erwartete Genauigkeit rechtfertigt. Wenn die Funktion beispielsweise Konstanten enthält, die nur auf drei Stellen nach dem Komma genau sind, ist es vielleicht nicht sinnvoll, eine Genauigkeit von mehr als drei Stellen nach dem Komma mit dem Anzeigeformat festzulegen.

Ebenso wichtig, aber feinsinniger und daher öfters übersehen, ist die Tatsache, daß fast jede Funktion, die einen physikalischen Zusammenhang wiedergibt, bis zu einem gewissen Grad von Natur aus ungenau ist, weil sie nur ein mathematisches Modell eines tatsächlichen Prozesses oder Ereignisses darstellt. Ein mathematisches Modell ist selber nur eine Annäherung, die die Wirkung von bekannten oder unbekannten Faktoren unberücksichtigt läßt, die insoweit unwesentlich sind, als die Resultate dennoch verwendbar sind.

Die Normalverteilungs-Funktion

$$\int_{-\infty}^t \frac{e^{-(x-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}} dx,$$

*Vorausgesetzt, daß die Funktion $f(x)$ trotz Rundungsfehler entsprechend der Anzahl Stellen im Anzeigeformat genau berechnet wird.

ist ein Beispiel für ein mathematisches Modell, das zur Informationsgewinnung bei der Datenerhebung von lebenden Organismen, Maßgrößen, durchschnittlichen Temperaturen usw. nutzbringend eingesetzt werden kann.

Die Gleichung

$$C = \frac{C_0}{\sqrt{\pi D t}} \int_{x/2\sqrt{D t}}^{\infty} e^{-y^2/4 D t} dy,$$

beschreibt ein weiteres Modell, mit dem die Lösung der Diffusionsgleichung für Halbleiter gefunden werden kann. Diese mathematischen Ausdrücke werden entweder aus der Theorie abgeleitet oder ergeben sich aus experimentellen Daten.

Damit Sie auch praktisch anwendbar sind, werden gewisse Ausnahmen gemacht, die den Einfluß verhältnismäßig kleiner Faktoren außer Betracht lassen. Wenn beispielsweise die Normalverteilung gewisser Größen berechnet wird, hängt die Genauigkeit der Ergebnisse von der Größe der untersuchten Grundgesamtheit ab. Bei der Diffusionsgleichung wird die Quantenwirkung vernachlässigt. Und wenn bei der Gleichung $s = s_0 - \frac{1}{2} g t^2$, welche die Höhe eines fallenden Körpers beschreibt, die Variation der Erdbeschleunigung g mit der Höhe nicht berücksichtigt wird, wirkt sich das auf die Genauigkeit der Ergebnisse aus.

Mathematische Modelle, die einen tatsächlichen Vorgang beschreiben, können also nur Ergebnisse einer begrenzten Genauigkeit ergeben. Wenn das numerische Resultat einer Modellberechnung nur auf beispielsweise drei Stellen nach dem Komma sein soll, kann der Einfluß von vielen Faktoren und Annahmen vernachlässigt werden. Wenn diese Faktoren und Annahmen jedoch in einem verbesserten Modell berücksichtigt werden, kann sich das auf das Resultat ab der fünften Stelle nach dem Komma auswirken.

Wenn ein Resultat eine anscheinend höhere Genauigkeit aufweist, als in dem Modell enthalten ist, das zur Beschreibung des tatsächlichen Vorgangs oder Ereignisses verwendet wurde, sind Sie an sich nicht berechtigt, das Ergebnis in seinem vollen Umfang anzuwenden.

RUNDUNGSFEHLER BEI INTERNEN RECHNUNGEN

In jeder Rechenmaschine – also auch in Ihrem HP-34C – müssen Ergebnisse auf eine endliche Anzahl Ziffern (10 Ziffern bei Ihrem HP-34C) gerundet werden. Durch diesen Rundungsfehler können sich besonders bei Berechnungen, die mehrere mathematische Operationen beinhalten, Resultate ergeben, die nicht auf zehn Stellen nach dem Komma genau sind. Beachten

Sie, daß Rundungsfehler bei der Berechnung jedes mathematischen Ausdrucks entstehen und nicht nur bei der Berechnung einer Funktion im Zusammenhang mit  auftreten.

Wenn eine Funktion $f(x)$ einen physikalischen Zusammenhang wiedergibt, sind die durch Rundung verursachten Fehler im allgemeinen unbedeutend im Vergleich zu Fehlern, die durch empirische Konstanten usw. entstehen. In der Praxis bestimmt die Erfahrung eher als eine Analyse, inwieweit sich solche Fehler auswirken. Wenn $f(x)$ eine rein mathematische Funktion ist, wird die Genauigkeit nur durch den Rundungsfehler bestimmt. Es kann jedoch im allgemeinen nur mit einer aufwendigen Analyse ermittelt werden, wieviele Ziffern der berechneten Funktion von solchen Fehlern betroffen sind.

In manchen Fällen können Rundungsfehler zu merkwürdigen Resultaten führen, besonders bei der Bestimmung von Integralen, die mathematisch äquivalent sind und sich nur durch eine Transformation der Variablen unterscheiden. Eine genaue Untersuchung dieser Fälle, die kaum bei praktischen Anwendungen auftreten werden, kann nicht im Rahmen dieses Handbuchs durchgeführt werden.

FEHLERSCHRANKE UND ANZEIGEFORMAT

Als Folge von Rundungsfehlern kann das Unterprogramm, das Sie zur Berechnung von $f(x)$ geschrieben haben, $f(x)$ nicht genau berechnen. Vielmehr wird

$$\hat{f}(x) = f(x) \pm \delta_1(x),$$

berechnet, wobei $\delta_1(x)$ der durch Rundung verursachte Fehler ist.

Wenn $f(x)$ einen physikalischen Zusammenhang wiedergibt, werden Sie nicht $f(x)$ sondern

$$F(x) = f(x) \pm \delta_2(x),$$

integrieren, wobei der Fehler $\delta_2(x)$ dadurch entsteht, daß $f(x)$ ein Modell eines tatsächlichen physikalischen Vorgangs darstellt.

Da $f(x) = \hat{f}(x) + \varepsilon_1(x) + \varepsilon_2(x)$ gilt
lautet die Funktion, die Sie integrieren wollen,

$$F(x) = \hat{f}(x) \pm \delta_1(x) \pm \delta_2(x)$$

oder $F(x) = \hat{f}(x) \pm \delta(x),$

wobei $\varepsilon(x)$ den in Verbindung mit $f(x)$ entstandenen Gesamtfehler darstellt.

Das gesuchte Integral ist daher

$$\begin{aligned} \int_a^b F(x) dx &= \int_a^b [\hat{f}(x) \pm \delta(x)] dx \\ &= \int_a^b \hat{f}(x) dx \pm \int_a^b \delta(x) dx \\ &= I \pm \Delta \end{aligned}$$

Hier ist I die Näherung für $\int_a^b F(x) dx$ und Δ der in Verbindung mit der Näherung entstandenen Fehler: Der -Algorithmus speichert I in das X-Register und die Zahl Δ in das Y-Register.

Der Fehler $\varepsilon(x)$ der mit Ihrem Unterprogramm berechneten Funktion $f(x)$ wird folgendermaßen ermittelt. Wir wollen annehmen, daß die Funktionswerte auf drei signifikante Stellen genau sind und werden daher das Anzeigeformat 2 wählen. Die Anzeige enthält von der Mantisse dann nur die genauen Stellen der Funktionswerte, z. B. **1,23** –04. Da die Zahl in der Anzeige entsprechend dem Anzeigeformat gerundet wird, heißt es, daß der Fehler der Funktionswerte $\pm 0,005 \times 10^{-4} = \pm 0,5 \times 10^{-5} \times 10^{-4} = \pm 0,5 \times 10^{-6}$ ist. Wenn also n eine Ganzzahl ist und das Anzeigeformat n oder n gewählt wird, heißt es, daß der Fehler der Funktionswerte

$$\begin{aligned} \delta(x) &= 0,5 \times 10^{-n} \times 10^{m(x)} \\ &= 0,5 \times 10^{-n+m(x)} \end{aligned}$$

ist.

In dieser Formel ist n die im Anzeigeformat verwendete Stellenzahl und $m(x)$ der Exponent des Funktionswertes für x , der erscheinen würde, wenn das Ergebnis im -Format angezeigt würde.

Der Fehler ist proportional zu $10^{m(x)}$ der die Größenordnung des Funktionswertes für x darstellt. Mit den Anzeigeformaten · und wird also ein Fehler relativ zum Absolutwert der Funktion ermittelt.

Ähnlich bedeutet ein mit n angezeigter Funktionswert, daß die Rundung der Anzeige einen Fehler der Größe

$$\delta(x) = 0,5 \times 10^{-n}$$

ergibt.

Da dieser Fehler unabhängig vom Absolutwert der Funktion ist, heißt es, daß ein -Anzeigeformat einen Fehler angibt, der absolut ist.

Für jeden für x gebildeten Funktionswert berechnet der -Algorithmus auch die Ungenauigkeit $\varepsilon(x)$ des Funktionswertes. Dieser Wert wird unter Verwendung der augenblicklich für das Anzeigeformat festgelegten Stellen n und $m(x)$, dem Absolutwert der Funktion für x (im Falle eines - oder -Anzeigeformats) berechnet.

Die Zahl, die als Maß für die Fehlerabschätzung der Näherung des gewünschten Integrals gilt, ist das Integral

$$\begin{aligned} \Delta &= \int_a^b \delta(x) dx \\ &= \int_a^b [0,5 \times 10^{-n+m(x)}] dx. \end{aligned}$$

Dieses Integral wird in ähnlicher Weise wie das Integral der Funktion über die Stützstellenwerte von $f(x)$ berechnet.

Beachten Sie, daß die Näherung bei Verwendung des -Formates $m = 0$ ist und daher die Fehlerabschätzung durch den Ausdruck

$$\Delta = 0,5 \times 10^{-n}(b - a).$$

wiedergeben wird.

Da Δ proportional zu 10^{-n} ist, ändert sich die Fehlerabschätzung eines Integrals um einen Faktor 10 für jede Stelle, die im Anzeigeformat festgelegt wird. Da eine Änderung des Anzeigeformats zu Folge hat, daß die Funktion für eine unterschiedliche Anzahl von Stützstellen bestimmt wird, ändert sich auch der Wert $\varepsilon(x)10^{m(x)}$, so daß im - oder -Anzeigeformat dieser Faktor im allgemeinen nicht genau 10 ist.

Normalerweise wird es nicht nötig sein, die Ungenauigkeit einer Funktion exakt zu bestimmen. (Das würde in vielen Fällen eine komplizierte Analyse erforderlich machen). Es ist im allgemeinen sinnvoller das SCI - oder ENG - Anzeigeformat zu verwenden, wenn die Ungenauigkeit der Funktion als relative Ungenauigkeit berechnet werden kann. Dagegen sollte das FIX -Anzeigeformat verwendet werden, wenn die Ungenauigkeit als absoluter Fehler zu bestimmen ist. Das FIX -Anzeigeformat kann sich als ungeeignet erweisen (und zu seltsamen Ergebnissen führen), wenn Sie eine Funktion integrieren, die entweder extrem kleine Absolutwerte und Fehlerabschätzungen im Integrationsintervall besitzt oder deren Absolutwerte und Fehlerabschätzungen im Integrationsintervall sehr stark variieren. In gleicher Weise mag das SCI -Format ungeeignet (und zu sonderbaren Ergebnissen führen), wenn der Absolutwert der Funktion sehr viel kleiner als die Fehlerabschätzung ist. Wenn das Resultat einer Berechnung seltsam erscheint, ist es vielleicht angebracht, die Berechnung mit einem anderen Anzeigeformat zu wiederholen.

BERECHNUNG VON INTEGRALEN MIT MAXIMALER GENAUIGKEIT

Im SCI - oder ENG -Anzeigeformat können Zahlen mit einer Mantisse von bis zu sieben Stellen angezeigt werden. Ein Anzeigeformat von SCI 8 oder SCI 9 ergibt gewöhnlich die selbe Anzeige wie SCI 7. Die Fehlerabschätzung ist bei SCI 8 oder SCI 9 jedoch kleiner als bei Integralen, die mit SCI 7 berechnet werden. Das gilt gleichermaßen für Integrale, die mit einem ENG -Anzeigeformat berechnet werden.

Das Resultat mit der größtmöglichen Genauigkeit erhalten Sie, wenn Sie den Anzeige-Modus SCI (oder ENG) 9 wählen.* Dies kann entweder direkt durch f SCI 9 oder indirekt durch 9 STO f I h DSP I geschehen, wenn sich der Rechner im RUN-Modus befindet (und der Anzeige-Modus SCI oder ENG schon gewählt worden war). Wenn sich der Rechner dagegen im PRGM-Modus befindet, kann der Anzeige-Modus SCI 8, SCI 9 ENG 8, ENG 9 nicht direkt gewählt werden. Wenn Sie es trotzdem versuchen, erkennen Sie am Tasten-Code, daß die Anweisung SCI 7 oder ENG 7 abgelegt wurde. Die Integrale werden mit einer Fehlerabschätzung berechnet, die von dem Anzeige-



format mit sieben Stellen hergeleitet wird. Um im PRGM-Modus die höchstmögliche Genauigkeit zu erzielen, müssen Sie das Anzeigeformat mit DSP I wählen.*

Das geht folgendermaßen vorstatten: Schieben Sie den PRGM RUN -Schalter in Stellung PRGM und tasten Sie das folgende Programm zur Berechnung des Integrals Si(2) mit maximaler Genauigkeit ein.

Tastenfolge	Anzeige
h LBL A	001-25,13,11
9	002- 9
STO f I	003-23,14,23
h DSP I	004- 25 11
9 R↓	005- 15 22
f β 2	006-14,72, 2
h RTN	007- 25 12
h LBL 2	008-25,13, 2
f SIN	009- 14 7
X2Y	010- 21
±	011- 71
h RTN	012- 25 12

Schieben Sie den PRGM RUN -Schalter wieder in Stellung RUN. Tasten Sie die Integrationsgrenzen in das X- und Y-Register. Drücken Sie dann A um die Berechnung durchzuführen.

- * Wenn das I-Register eine negative Zahl enthält, werden mit h DSP I Ergebnisse angezeigt als ob eine Null im I-Register stehen würde. Der f -Algorithmus berücksichtigt jedoch die Zahl bei der Berechnung der Fehlerabschätzung des Integrals. Die kleinste Zahl, die bei der Berechnung der Fehlerabschätzung einer Näherung verwendet werden kann, ist -6. Wenn der Inhalt des I-Registers kleiner als -6 ist, wird die Berechnung durchgeführt, als ob -6 im I-Register stehen würde.

* Vorausgesetzt, daß f(x) auf 10 Dezimalstellen genau berechnet wird.

Tastenfolge SCI 3**Anzeige**

0,000

00

Wählen Sie das **SCI**-Anzeigeformat. Wenn das Programm (Durch Drücken von **SCI**) ausgeführt wird, verändert sich die Anzahl der mit Anzeigeformat verwendeten Stellen von 3 auf 9. (Es wird vorausgesetzt, daß keine Ergebnisse von vorherigen Rechnungen in der Anzeige stehen.)

0 **ENTER+**

0,000

00

Geben Sie die untere Integrationsgrenze in das **Y**-Register ein.

2

2,

RAD

2,000

00

Geben Sie die obere Integrationsgrenze in das **X**-Register ein. Achten Sie darauf, daß der Winkel-Modus Bogenmaß eingeschaltet ist. (Dieser Schritt ist nicht erforderlich, wenn Sie den Rechner inzwischen weder ausgeschaltet, noch den Winkel-Modus seit dem letzten Umschalten auf Bogenmaß verändert haben).

Tastenfolge**A****Anzeige**

1,605412 00 Das Resultat $\text{Si}(2)$ mit maximaler Genauigkeit.

xy

6,000000 -10 Fehlerabschätzung der Näherung

xy

1,605412 Der Näherungswert steht wieder in der Anzeige.

MANT

1605412977 Alle zehn Ziffern des Ergebnisses.

Da die signifikanteste Ziffer der Fehlerabschätzung in der zehnten Dezimalstelle steht, kann man annehmen, daß das Ergebnis wenigstens auf neun Dezimalstellen genau ist. Tatsächlich stimmen die ersten neun Stellen des Ergebnisses mit dem im mathematischen Tabellen aufgeführten Wert von $\text{Si}(2)$ überein.

DER AUGENBLICKLICHE NÄHERUNGSWERT DES INTEGRALS

Wenn Sie während der Bestimmung eines Integrals **R/S** drücken, wird die Berechnung wie jede andere Programmausführung unterbrochen.

Der Rechner hält bei der augenblicklichen Programmzeile des zur Berechnung der Funktion geschriebenen Unterprogramms an, wobei die Anzeige das Ergebnis der vorangegangenen Befehlsanweisung enthält. Beachten Sie, daß nach der Unterbrechung weder das **X**-Register noch eines der anderen Stack-Register den augenblicklichen Näherungswert des Integrals enthält. Wie bei jedem anderen Programm wird mit **R/S** die Programmausführung ab der augenblicklichen Programmzeile fortgesetzt.

Wenn die Berechnung eines Integrals länger dauert als Ihnen lieb ist, können Sie den Rechner anhalten und den augenblicklichen Näherungswert untersuchen. Sie können auf diese Weise aber nur den Näherungswert und nicht die Fehlerabschätzung erhalten. Der **LAST**-Algorithmus aktualisiert die augenblickliche Näherung und speichert sie jeweils im LAST X-Register ab, nachdem die Funktion für eine neue Stützstelle bestimmt wurde. Um also die augenblickliche Näherung zu erhalten, halten Sie den Rechner einfach an, lassen Sie den Rechner die Funktion, soweit erforderlich, durch schrittweises Ausführen der Anweisungen zu Ende berechnen und die augenblickliche Näherung aktualisieren und rufen Sie dann den Inhalt des LAST X-Registers in die Anzeige.

Beachten Sie, daß während der Aktualisierung die Anzeige nicht aufleuchtet, wie es sonst der Fall ist, wenn der Rechner Ihr Funktionsprogramm durchläuft. Die schrittweise Ausführung mehrerer Programmzeilen können also vermeiden, wenn Sie **R/S** drücken, wenn die Anzeige dunkel ist.

Wir wollen die Schritte zusammenfassen, die Sie ausführen müssen, um den augenblicklichen Näherungswert zu erhalten:

1. Drücken Sie **R/S** um den Rechner anzuhalten, am besten, wenn die Anzeige dunkel ist.
2. Wenn der Rechner mit einer Zahl in der Anzeige anhält, schieben Sie den **PRGM ■■■■■ RUN**-Schalter in Stellung **PRGM**.
 - a. Wenn die Anzeige die Programmzeile mit der Marke Ihres Funktionsunterprogramms enthält, schieben Sie den **PRGM ■■■■■ RUN**-Schalter wieder in Stellung **RUN** und fahren Sie mit Schritt 3 fort.

b. Wenn Sie **R/S** nicht gedrückt haben, während die Anzeige dunkel war, wird die Anzeige jetzt eine andere Programmzeile Ihres Unterprogramms enthalten. Schieben Sie den **PRGM**  **RUN**-Schalter wieder in Stellung RUN und drücken Sie so oft **SST** bis **25 12** rechts (oder **000-links***) in der Anzeige erscheint, während Sie **SST** gedrückt halten. Lassen Sie die Taste los und warten Sie bis eine Zahl in der Anzeige erscheint.

3. Drücken Sie **h** **LST X**. Der augenblickliche Näherungswert wird in der Anzeige erscheinen. Wenn Sie bis zum endgültigen Ergebnis der Näherung fortfahren wollen, drücken Sie **CLX**  **R/S**. Damit wird der augenblickliche x-Wert in den Stack geladen und die Berechnung fortgesetzt.

Als Beispiel wollen wir nochmals $\text{Si}(2)$ berechnen, aber uns den augenblicklichen Näherungswert nach etwa einer Minute ausgeben lassen.

Tastenfolge	Anzeige
g R↓ g R↓	2,000000 00 Der Stack wird zyklisch nach unten verschoben bis die obere Integrationsgrenze im X-Register steht.
A	(blinkende Anzeige) Die Bestimmung des Integrals beginnt.

Halten Sie den Rechner nach ein oder zwei Minuten an und überprüfen Sie den augenblicklichen Näherungswert des Integrals:

Tastenfolge	Anzeige
R/S	6,771087-10 Halten Sie durch Drücken von R/S den Rechner an. Die Anzeige hängt natürlich von dem Zeitpunkt ab, zu dem Sie R/S gedrückt haben.

* Dies kommt vor, wenn Sie keine **RTN**-Anweisung als letzten Befehl in Ihrem Unterprogramm verwendet haben.



Schieben Sie den **PRGM**  **RUN**-Schalter in Stellung PRGM, um sich zu vergewissern, daß der Rechner bei der Marke Ihres Unterprogramms angehalten hat.

Anzeige
001-25,13,2 Marke 2

Da der Rechner tatsächlich bei der Marke Ihres Unterprogramms angehalten hat, können Sie den augenblicklichen Näherungswert aus dem LAST X-Register zurückrufen, nachdem Sie den **PRGM**  **RUN**-Schalter wieder in Stellung RUN gebracht haben.

Tastenfolge	Anzeige
h LST X	1,605412 00 Der augenblickliche Schätzwert des Integrals (Die tatsächliche Zahl hängt wieder davon ab, wann Sie R/S gedrückt haben.)

Um die Berechnung fortzusetzen und das Endergebnis zu erhalten:

Tastenfolge	Anzeige
CLX 	6,771087-01 Der augenblickliche x-Wert steht im X-Register.
R/S	1,605412 00 Das Endergebnis für den Näherungswert.

UMSTÄNDE, DIE ZU FALSCHEN ERGEBNISSEN FÜHREN KÖNNEN

Obwohl der in Ihrem HP-34C implementierte -Algorithmus zu den besten gehört, kann es dennoch passieren, daß er – wie fast jeder in der numerischen Integration verwendete Algorithmus – ein falsches Ergebnis erzeugt.

Es ist allerdings höchst unwahrscheinlich, daß dies eintritt. Der -Algorithmus ist so konzipiert, daß bei fast jeder glatten Funktion richtige Ergebnisse erzeugt werden. Nur bei Funktionen, die ein extrem unregelmäßiges Verhalten aufweisen, besteht überhaupt die Gefahr, daß sie falsche Ergebnisse erhalten.

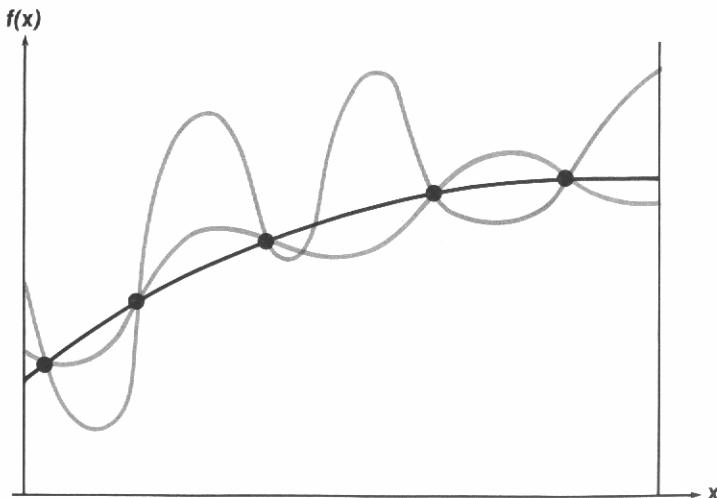
Solche Funktionen kommen jedoch nur selten in Problemen vor, die einen tatsächlichen physikalischen Zusammenhang wiedergeben. In diesen seltenen Fällen, ist man gewöhnlich in der Lage, die Funktion zu erkennen und sie auf ganz einfache Art zu behandeln.

Wir wollen uns jetzt etwas eingehender mit der Arbeitsweise des -Algorithmus befassen, um zu erfahren, wie ein falsches Ergebnis zustande kommen könnte.

Auf diese Weise werden wir die allgemeinen Charakteristika der Funktionen feststellen können, die zu Schwierigkeiten führen können. Schließlich werden wir erfahren, wie die Genauigkeit einer Berechnung zu überprüfen ist, wenn dies wünschenswert erscheinen sollte.

Wie wir schon auf Seite 235 erfahren haben, bildet der -Algorithmus eine Reihe von Stützwerten der Funktion innerhalb des Integrationsintervalls. Es wird ein gewichtetes Mittel der Funktionswerte an den Stützstellen gebildet, aus dem der Algorithmus einen Näherungswert für das Integral ableitet.

Da der Algorithmus die Funktion $f(x)$ nur an Ihren Stützstellen kennt, kann er leider nicht zwischen $f(x)$ und einer beliebigen Funktion, die an den Stützstellen mit $f(x)$ übereinstimmt, unterscheiden. Dieser Sachverhalt wird auf der untenstehenden Abbildung veranschaulicht, die drei von unendlich vielen Funktionen zeigt, die an endlich vielen Stützstellen (über einen Teil des Integrationsintervalls) übereinstimmen.



Bei dieser Anzahl von Stützstellen wird der Algorithmus für jede der abgebildeten Funktionen den selben Näherungswert für das Integral berechnen.

Die tatsächlichen Integrale der schwarzen und goldfarbenen Funktionen sind etwa gleich, so daß die Näherung mehr oder weniger genau ist, wenn $f(x)$ einer dieser Funktionen entspricht. Dagegen hat das tatsächliche Integral der blauen Funktion einen ganz anderen Wert als die beiden anderen Integrale. Der Näherungswert wird also ziemlich ungenau sein, wenn $f(x)$ dieser Funktion entspricht.



Wir wollen einmal annehmen, daß sich bei dieser Anzahl von Stützstellen die Näherung von den vorangegangenen Näherungswerten um weniger als die vom Anzeigeformat abgeleitete Fehlerabschätzung unterscheidet.



Der -Algorithmus wird dann beendet und der augenblickliche Näherungswert gilt als bestmögliche Näherung für das Integral, mit der von Ihnen im voraus zugelassenen Fehlerabschätzung. Für eine Reihe von Funktionen – wie hier die in blau abgebildete – ergibt sich also eine ziemlich ungenaue Näherung, weil die Funktion nur für eine endliche Anzahl von Punkten berechnet wird.



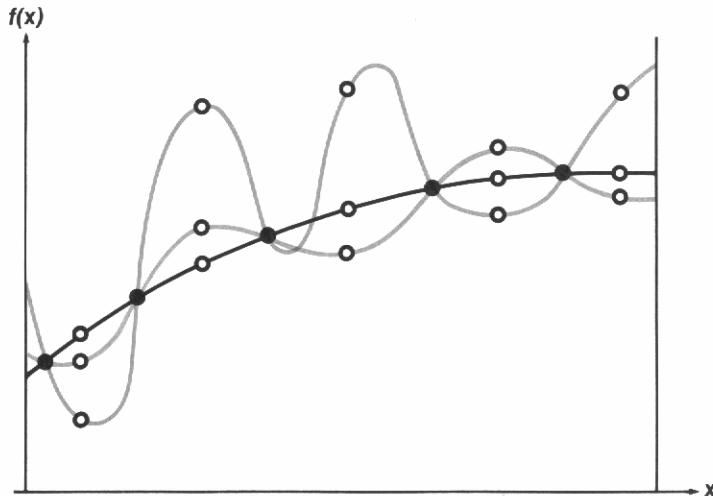
Diese Situation stellt einen extremen Fall der Wechselwirkung (siehe Seite 241) zwischen Genauigkeit und Rechenzeit dar: da Sie nicht bereit sind, unendlich lange auf ein Ergebnis zu warten (also unendlich viele Stützstellen zu berücksichtigen), können Sie nicht absolut sicher sein, daß die berechnete Näherung so genau ist, wie man aus der Fehlerabschätzung schließen könnte.



Im Gegensatz zu der oben beschriebenen Situation nehmen wir jetzt an, daß die Fehlerabschätzung so klein sein muß (durch entsprechende Angabe der Stelle im Anzeigeformat), daß der Näherungswert mit dieser Anzahl von Stützstellen nicht ausreichend genau ist. Der Algorithmus bildet jetzt Stützwerte von $f(x)$ für zusätzliche Punkte.

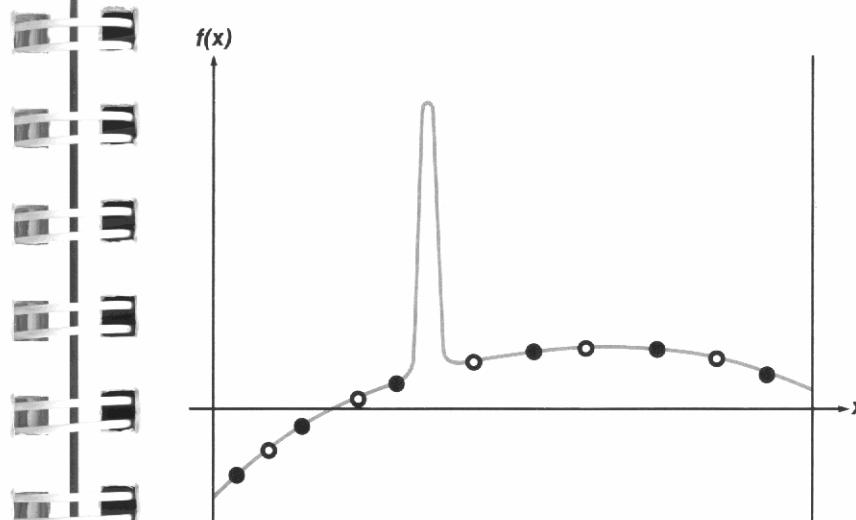


Diese Situation wird in der zweiten Abbildung wiedergegeben, die die drei Funktionen mit den neuen und ursprünglichen Stützstellen zeigt.



Die drei in der Abbildung gezeigten Funktionen haben identische Werte für die anfängliche Menge der Stützstellen, aber die blaue Funktion hat sehr unterschiedliche Werte an den dazugekommenen Punkten. Wenn der Algorithmus jetzt diese neuen Funktionswerte verarbeitet, wird es sich herausstellen, daß der Unterschied zwischen der augenblicklichen Näherung und den vorherigen Näherungswerten viel größer als die erlaubte Fehlerabschätzung ist. Jetzt werden mehr und mehr Punkte in die Rechnung miteinbezogen, bis aufeinanderfolgende Näherungen genügend nahe beieinander liegen. In diesem Fall berechnet der Algorithmus einen genauen Näherungswert, weil Sie sich durch Angabe der maximalen Fehlerabschätzung bereit erklärt haben, die längere Rechenzeit in Kauf zu nehmen.

Aus praktischen Gründen wollen Sie jedoch nicht ewig auf ein Ergebnis warten. (Schon weil es inzwischen nicht mehr gebraucht wird!). Indem Sie den Algorithmus derart einschränken, müssen Sie einsehen, daß die Funktion nicht an unendlich vielen Stützstellen berechnet werden kann. Folglich kann es passieren, daß eine in einer Funktion auftretende scharfe, spitze Zacke durch den Algorithmus übersehen wird. Die dritte Abbildung veranschaulicht eine Situation, in der bis auf eine eindeutige Zacke die Funktion glatt verläuft.



Trotz relativ dicht beieinanderliegender Stützstellen bleibt eine Zacke in der Funktion unentdeckt. Da jedoch aufeinanderfolgende Näherungen eine gute Übereinstimmung aufweisen, wird der Algorithmus mit einem Näherungswert beendet, der sich wesentlich von dem tatsächlichen Wert unterscheidet, weil die Zacke durch den Algorithmus nicht gefunden wurde.

Warum bleibt die Zacke unentdeckt?

Weil Sie im Vergleich zu dem ansonsten glatten Verlauf der Funktion innerhalb des Integrationsintervalls so grundverschieden ist. Bis auf diese Zacke ist die Funktion in dem gezeigten Intervall völlig glatt. (Es kann natürlich durchaus sein, daß diese Funktionen im gesamten Integrationsintervall keineswegs so glatt sind, sondern starke Schwankungen aufweisen. Die Abbildungen zeigen nur einen kleinen ausgedehnten Teil des Integrationsintervalls, der charakteristische Schwankungen als glatterscheinen läßt.) Liegt die Menge der Stützpunkte ausreichend dicht beieinander, kann der Algorithmus sich ein Bild vom allgemeinen Verhalten der Funktion machen. Wenn sich die Zacke nicht so deutlich vom übrigen Teil der Funktion abheben würde, könnte der Algorithmus diese und andere Variationen durchaus entdecken. In diesem Fall wird die Menge der Stützstellen so lange vergrößert, bis sich aufeinanderfolgende Näherungswerte ergeben, die das Vorhandensein von sehr schnellen, aber charakteristischen Schwankungen mit in Betracht ziehen.

Als Beispiel wollen wir die Gleichung

$$\int_0^x xe^{-x} dx.$$

betrachten. Da wir dieses Integral numerisch bestimmen wollen, liegt es nahe, als obere Integrationsgrenze 10^{99} , die größte vom Rechner verstandene Zahl, anzugeben. (Dies wird sich später als Irrtum erweisen). Wir wollen sehen, was passiert.

Schieben Sie den PRGM RUN -Schalter in Stellung PRGM und tasten Sie das Unterprogramm zur Berechnung von $f(x) = xe^{-x}$ ein.

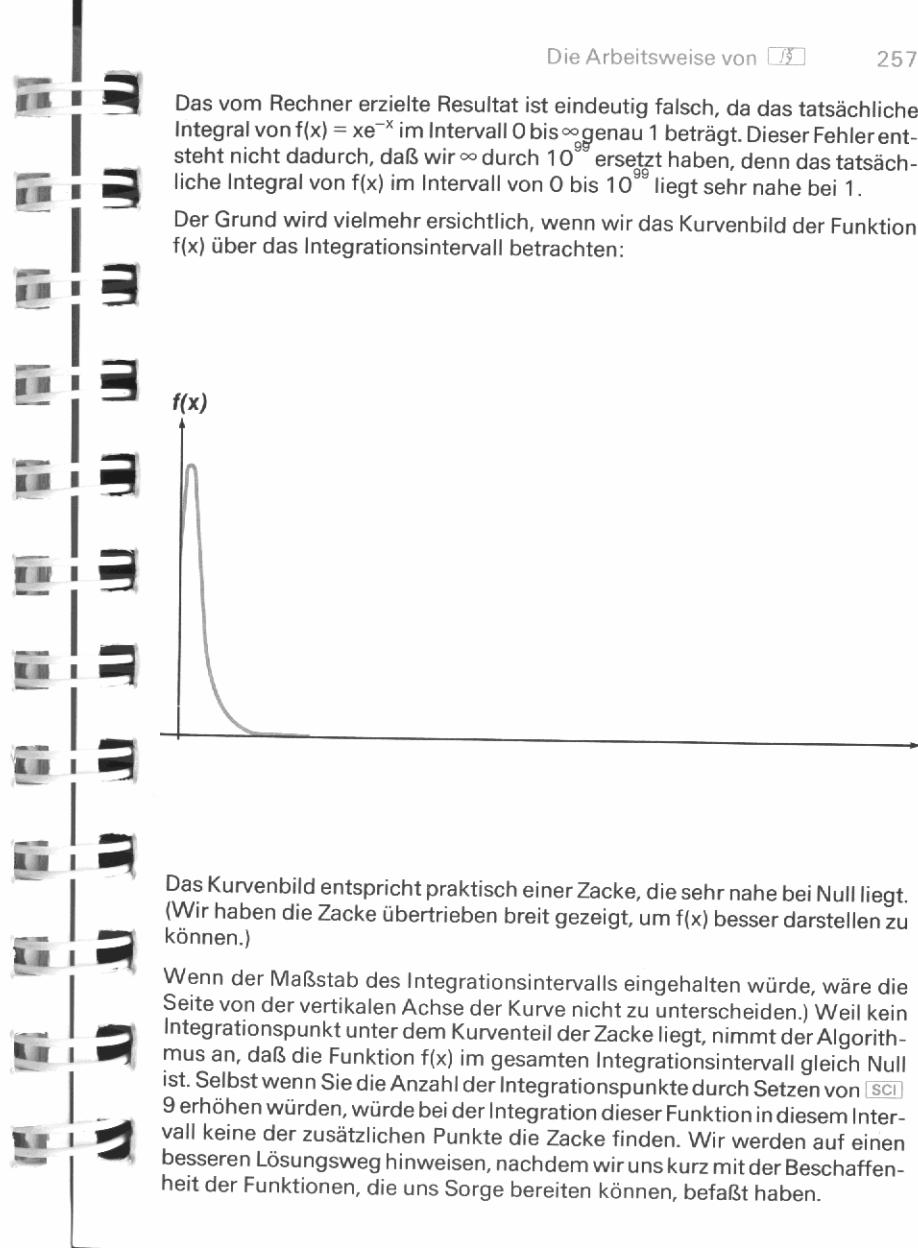
Tastenfolge Anzeige

h	LBL	1	001-25,13, 1
CHS		002-	32
g	e^x	003-	15 1
x		004-	61
h	RTN	005-	25 12

Schieben Sie den PRGM RUN -Schalter wieder in Stellung RUN und geben Sie die Integrationsgrenzen in das X- und Y-Register ein. Wählen Sie das Anzeigeformat SCI 3.

Tastenfolge Anzeige

f	SCI	3	0,000	00	Das Anzeigeformat SCI 3 wird gewählt. (Es wird vorausgesetzt, dass die Anzeige keine vorherigen Ergebnisse enthält.)
0	ENTER		0,000	00	Die untere Integrationsgrenze steht im Y-Register.
EEX	99		1,	99	Die obere Integrationsgrenze steht im X-Register.
f	J	1	0,000	00	Der Näherungswert für das Integral.



Wir haben gesehen, daß der -Algorithmus ein falsches Resultat ermittelt wenn $f(x)$ irgendwo eine Unregelmäßigkeit aufweist, die sonst nicht charakteristisch für die Funktion ist. Glücklicherweise sind Funktionen, die solche Abweichungen enthalten selten genug, so daß Sie kaum unwissentlich eine Funktion dieser zu integrieren haben werden.

Funktionen, die zu falschen Ergebnissen führen können, werden am leichtesten erkannt, wenn man sie vom mathematischen Standpunkt der kompletten Analyse beschreibt.* Man kann aber auch sagen, daß derartige Funktionen daran erkannt werden, wie sehr die Funktion und ihre ersten Ableitungen im Integrationsintervall variieren. Grundsätzlich kann man sagen, je größer die Schwankung der Funktion und ihre Ableitungen und je niedriger die Ordnung der Ableitungen, desto länger braucht der Algorithmus um zu einem Ergebnis zu kommen und desto unzuverlässiger ist das Ergebnis.

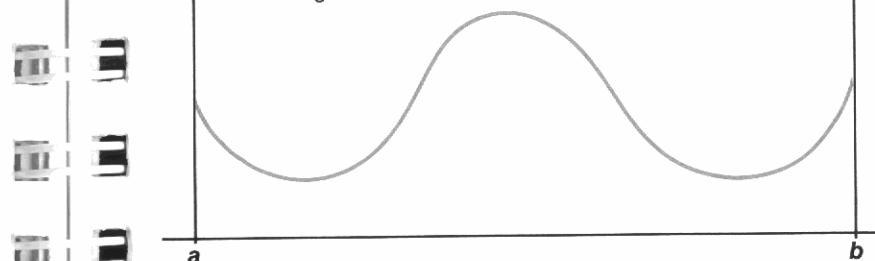
Beachten Sie, daß das Maß der Schwankung der Funktion (und ihrer Ableitung) in Beziehung zur Länge des Integrationsintervalls bestimmt werden muß.

Bei einer festen Anzahl von Stützstellen kann eine Funktion mit drei Unregelmäßigkeiten besser durch ihre Stützwerte wiedergegeben werden, wenn die Unregelmäßigkeiten über das gesamte Intervall weit verteilt sind und nicht an einer Stelle gehäuft auftreten. (Diese Situation wird in der untenstehenden Abbildung gezeigt.) Wenn man eine Unregelmäßigkeit als eine Art von Oszillation der Funktion ansieht, dann gilt das Interesse dem Verhältnis der Periode der Oszillationen zu der Breite des Integrationsintervalls. Je größer dieses Verhältnis, desto eher wird der Algorithmus beendet und desto zuverlässiger sind die erzielten Resultate.

In vielen Fällen werden Sie mit der zu integrierenden Funktion so vertraut sein, daß Sie wissen, ob die Funktion derartige Unregelmäßigkeiten relativ zum Integrationsintervall besitzt. Wenn Sie die Funktion dagegen nicht kennen und Sie Grund haben, den Ergebnissen nicht zu trauen, können Sie sie mit Hilfe Ihres Unterprogramms schnell anhand einiger Punkte graphisch auftragen.

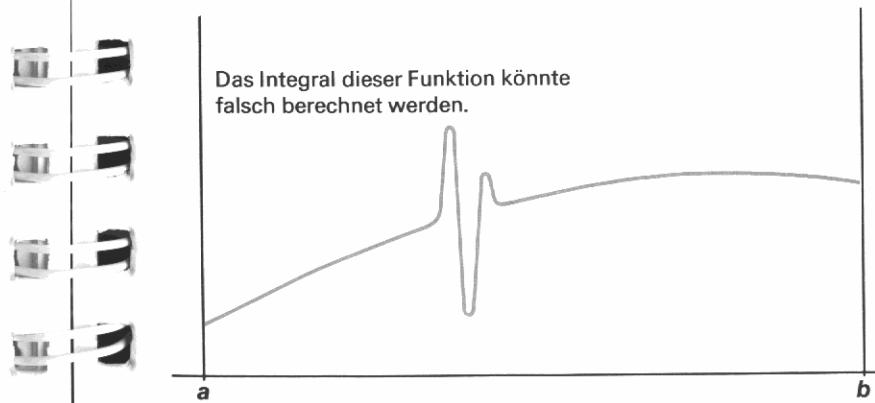
- Die durch den HP-34C errechneten Näherungswerte konvergieren schnell gegen das richtige Ergebnis, vorausgesetzt, daß der als analytische Funktion der komplexen Variable z betrachtete Integrand $f(z)$ keine Singularitäten im oder in der Nähe des Integrationsintervalls besitzt, und daß der Mittelwert in diesem Intervall nicht erheblich kleiner als der Absolutwert der Funktion in der Nähe des Intervalls ist.

Das Integral dieser Funktion wird richtig berechnet.



Wenn Sie aus irgendeinem Grund Bedenken haben sollten, ob der Näherungswert des Integrals richtig berechnet wurde, können Sie das Ergebnis mittels des folgenden Verfahrens überprüfen: Teilen Sie das Integrationsintervall in zwei oder mehr Teilintervalle, integrieren Sie die Funktion über die einzelnen Teilintervalle und summieren Sie die Näherungswerte. Durch diesen Vorgang wird die Funktion für eine ganz neue Menge von Stützstellen berechnet, so daß bis jetzt unentdeckte Zacken viel eher gefunden werden. Wenn das ursprüngliche Resultat richtig war, wird es gleich der Summe der Näherungswerte der Integrale für die einzelnen Teilintervalle sein.

Das Integral dieser Funktion könnte falsch berechnet werden.



FAKTOREN, DIE DIE RECHENZEIT VERLÄNGERN

Im vorangegangenen Beispiel hat der Algorithmus ein falsches Resultat errechnet, weil er die Zacke in der Funktion nicht gefunden hat. Dies passte, weil die Unregelmäßigkeit in der Funktion relativ zum Integrationsintervall zu klein war. Wenn die Breite des Intervalls verkürzt würde, bekämen wir ein korrektes Ergebnis. Der Vorgang würde dann jedoch sehr lange dauern, wenn das Intervall immer noch sehr groß wäre.

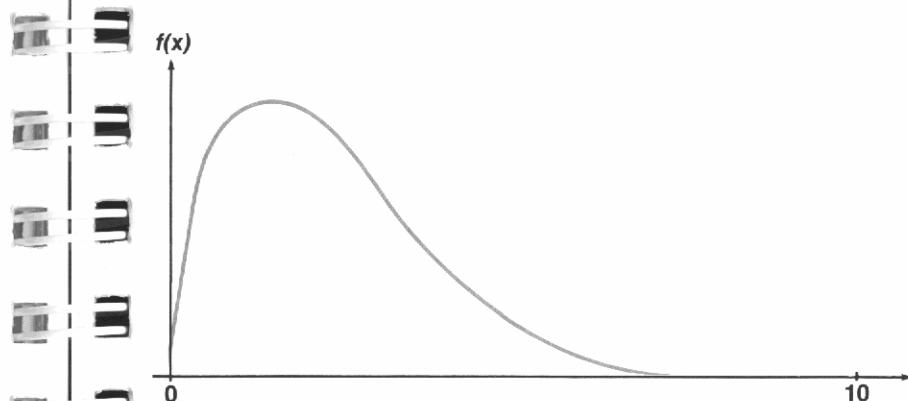
Für Funktionen dieser Art kann die Rechenzeit auch dann über die Maßen verlängert werden, weil das Integrationsintervall im Verhältnis zu gewissen Eigenschaften der zu integrierenden Funktion zu groß ist.

Wir wollen einmal ein Integral betrachten, wo das Integrationsintervall so groß ist, daß die Rechenzeit übermäßig lang ist, ohne daß das Ergebnis wegen der Größe des Intervalls falsch berechnet wird. Beachten Sie, daß bei der Funktion $f(x) = xe^{-x}$ der Anteil der Funktion am Integral für große Werte von x vernachlässigbar klein ist, weil $f(x)$ sehr schnell gegen Null geht, wenn x gegen ∞ geht.

Wir können das Integral also berechnen, indem wir die obere Integrationsgrenze durch eine Zahl kleiner 10^{99} (wie z. B. 10^3) ersetzen.

Tastenfolge	Anzeige	
0 ENTER	0,000 00	Die untere Integrationsgrenze steht im Y-Register.
EEX 3	1, 03	Die obere Integrationsgrenze steht im X-Register.
f [y] 1	1,000 00	Der Näherungswert des Integrals.
[x:y]	1,824 -00	Die Fehlerabschätzung.

Das Ergebnis ist richtig, aber es hat sehr lange gedauert. Warum dies so ist, sehen wir an dem Kurvenbild der Funktion im Integrationsintervall, das dem auf Seite 257 gezeigten Graphen der Funktion zwischen $x = 0$ und $x = 10$ sehr ähnlich ist.



Wenn wir die zwei Kurvenbilder vergleichen, sehen wir, daß die Funktion nur für kleine Werte von x "interessant" ist. Für größere Werte von x wird die Kurve "uninteressant", weil sie ständig und regelmäßig in voraussagbarer Weise gegen Null geht.

Wie wir schon früher besprochen haben, bildet der Algorithmus für eine zunehmende Anzahl von Punkten Stützwerte der Funktion, bis der Unterschied zwischen aufeinanderfolgenden Näherungen genügend klein ist. Mit anderen Worten, die Anzahl der gebildeten Stützwerte nimmt zu, bis der Algorithmus soviel Information über die Funktion gesammelt hat, daß eine weitere Anzahl von Punkten sich nicht wesentlich auf den Näherungswert auswirken würde.

Wenn das Integrationsintervall $(0, 10)$ verwendet wird, befaßt sich der Algorithmus nur dort mit der Funktion, wo sie zwar interessant aber relativ glatt ist, so daß weitere Stützwerte nach wenigen Iterationen keine neue Information über das Verhalten der Funktion liefern.

Daher sind nur wenige Iterationen nötig, bis der Unterschied zwischen aufeinanderfolgenden Näherungen so klein wird, daß der Algorithmus mit einem Ergebnis der gewünschten Genauigkeit beendet wird.

Wenn dagegen ein wie auf Seite 257 gezeigtes Integrationsintervall verwendet wird fallen die meisten Stützstellen in einen Bereich, in dem die Funktion nicht sonderlich variiert.

Die wenigen Stützwerte für kleine Werte von x würden ergeben, daß sich die Funktionswerte von Iteration zu Iteration wesentlich unterscheiden.

Daher müßten zusätzliche Stützstellen mit einbezogen werden, bevor der Unterschied zwischen aufeinanderfolgenden Näherungen klein genug wird.

Damit die Näherung eines Integrals die gleiche Genauigkeit in einen größeren wie in einem kleineren Intervall aufweist, muß die Dichte der Stützstellen in den Bereich, in dem die Funktion interessant ist, jeweils die gleiche sein. Um dies zu erreichen, muß die Anzahl der Stützstellen, die für das größere Intervall benötigt wird, viel größer sein als für das kleine Intervall. Daher werden bei einem größeren Intervall etliche zusätzliche Rechnungen benötigt, um einen Näherungswert mit der gleichen Genauigkeit zu erhalten. Die Berechnung des Integrals dauert daher wesentlich länger.

Weil die Rechenzeit davon abhängt, wie schnell eine gewisse Dichte von Stützstellen in dem Bereich, in dem die Funktion interessant ist, erreicht wird, erhöht sich die Rechenzeit für die Integralbestimmung, wenn das Integrationsintervall zum größten Teil eine Kurvenstrecke enthält, die nicht interessant ist. Glücklicherweise kann man in solchen Fällen Abhilfe schaffen, indem das Problem so formuliert wird, daß die Rechenzeit wesentlich verkürzt wird. Wir wollen zwei Techniken hier erörtern: Unterteilung des Integrationsintervalls und Transformation der Variablen.

UNTERTEILUNG DES INTEGRATIONSPREINTERVALLS

In Bereichen, wo die Steigung von $f(x)$ sehr variiert, sind viele Stützstellen notwendig, um eine Näherung zu erhalten, die sich nur unwesentlich von einer Iteration zur nächsten ändert. Dagegen werden in Bereichen, wo die Steigung mehr oder weniger konstant bleibt, weniger Stützstellen benötigt. Der Grund hierfür ist, daß die an zusätzlichen Punkten berechnete Funktion nicht mehr viel neue Information über die Funktion liefert, so daß der Unterschied zwischen aufeinanderfolgenden Näherungen nicht stark beeinflußt wird. In solchen Bereichen könnten daher Näherungswerte mit vergleichbarer Genauigkeit mit wesentlich weniger Stützstellen erreicht werden. Ein großer Teil der Rechenzeit für die Berechnung der Funktion in diesem Bereich wird verschwendet. Wenn Sie solche Funktionen integrieren wollen, können sie auf folgende Weise Rechenzeit sparen:

1. Teilen Sie das Integrationsintervall in Teilintervalle in denen die Funktion interessant und Teilintervalle, in denen die Funktion uninteressant ist.
2. Integrieren Sie die Funktion in Abschnitten, in denen Sie interessant ist mit der durch das Anzeigeformat gesteuerten Genauigkeit, die Sie im Gesamtergebnis sehen wollen.

3. Bei Abschnitten, in denen die Funktion entweder nicht interessant ist oder nur geringfügig zu dem Ergebnis beiträgt, verwenden Sie eine geringere Genauigkeit, in dem Sie im Anzeigeformat weniger Stellen festlegen.

4. Um das Integral für das gesamte Integrationsintervall zu bekommen, summieren Sie die einzeln erzeugten Näherungswerte mit ihren Fehlerschätzungen. Das läßt sich mit der $\Sigma+$ -Taste bewerkstelligen. Bevor Sie das Integrationsintervall unterteilen, überprüfen Sie, ob eine Berechnung in der Umgebung der oberen (oder unteren) Integrationsgrenze zu einem Unterlauf führt.* Da es keinen Sinn hat, die Funktion für solche Werte von x zu berechnen, die einen Unterlauf ergeben, läßt sich hier die Rechenzeit vielleicht wesentlich verkürzen, wenn die obere Integrationsgrenze verschoben wird.

Beachten Sie, daß wenn Sie einmal ein Unterprogramm zur Berechnung von $f(x)$ eingetastet haben, Sie für jeden Wert von x im X-Register den dazugehörigen Funktionswert berechnen können, indem Sie $\text{ENTER} \downarrow \text{ENTER} \downarrow \text{ENTER} \downarrow \text{GSB}$ gefolgt von der Marke Ihres Unterprogramms drücken.

Wenn der Rechner in der Umgebung der oberen Integrationsgrenze einen Unterlauf erzeugt, können Sie kleinere Werte von x probieren, bis kein Unterlauf eintritt.

Tastenfolge	Anzeige	
EEX 3	1,	03 Der Integrationswert steht im X-Register.
$\text{ENTER} \downarrow \text{ENTER} \downarrow \text{ENTER} \downarrow$	1,000	03 Der Stack wird mit x geladen.
GSB 1	0,000	00 Unterlauf tritt bei der oberen Grenze auf.
300 $\text{ENTER} \downarrow$ $\text{ENTER} \downarrow \text{ENTER} \downarrow$	3,000	02
GSB 1	0,000	00 Der Unterlauf tritt immer noch auf.
200 $\text{ENTER} \downarrow$ $\text{ENTER} \downarrow \text{ENTER} \downarrow$	2,000	02 Der x -Wert wird verringert.
GSB 1	2,768	-85 Bei $x = 200$ tritt der Unterlauf nicht mehr auf.
		• Beachten Sie, daß jedes Resultat, das eine Zahl kleiner als 10^{-99} ergibt, gleich Null gesetzt wird. Dies wird mit Unterlauf bezeichnet.

225 **ENTER+**
ENTER+ **ENTER+**
GSB 1

- 2,250** **02** Ein x-Wert zwischen 200 und 250 wird probiert.
4,324 **-96** Das Ergebnis ist nahe einem Unterlauf.

An dieser Stelle können Sie **SOLVE** verwenden, um den kleinsten x-Wert zu ermitteln, bei dem Unterlauf auftritt.

Tastenfolge **Anzeige**

g R↓ **2,250** **02** Der Stack wird zyklisch verschoben bis der zuletzt gefundene Wert im X- und Y-Register steht.
f SOLVE 1 **2,280** **02** Der kleinste Wert von x, bei dem Unterlauf auftritt, liegt bei etwa 228.

Wir haben festgestellt, daß wir nur zwischen 0 und 228 integrieren müssen. Da die Funktion nur für Werte unter 10 interessant ist, wollen wir an dieser Stelle das Integrationsintervall unterteilen. Das Problem kann jetzt wie folgt formuliert werden:

$$\int_0^{1000} xe^{-x} dx \approx \int_0^{228} xe^{-x} dx = \int_0^{10} xe^{-x} dx + \int_{10}^{228} xe^{-x} dx.$$

Tastenfolge **Anzeige**

0 ENTER+ **0,000** **00** Eingabe der unteren Integrationsgrenze für das erste Teilintervall.
10 **10,**
f S 1 **9,995** **-01** Das Integral für (0,10) mit **SCI 3** Anzeigeformat.
f CLEAR Σ **9,995** **-01** Löschen der Statistik-Register.
f Σ+ **1,000** **00** Der Näherungswert und die Fehlerabschätzung werden in R₁ und R₃ summiert.
x:y **1,841** **-04** Die Fehlerabschätzung der Näherung.

- g R↓ g R↓** **1,000** **04** Der Stack wird zyklisch verschoben, bis die obere Integrationsgrenze im X-Register erscheint.
228 **228,**
f SCI 0 **2,** **02** Eingabe der oberen Integrationsgrenze für das zweite Teilintervall. Die obere Grenze des ersten Teilintervalls wird in das Y-Register angehoben, wo sie als untere Grenze des zweiten Teilintervalls verwendet wird.
f JY 1 **5,** **-04** Anzeigeformat zur schnellen Berechnung im Bereich (10,228). Wenn die Fehlerabschätzung zu hoch ausfällt, können wir die Berechnung mit mehr Stellen in der Anzeige wiederholen.
f SCI 3 **5,328** **-04** Das Integral über (10,228) mit **SCI 0**.
x:y **7,568** **-05** Anzeigeformat **SCI 3**. Überprüfung der Fehlerabschätzung. Sie ist geringer als die Fehlerabschätzung für die Berechnung des ersten Teilintervalls und somit besitzt die mit **SCI 0** berechnete Näherung eine ausreichende Genauigkeit.
x:y **5,328** **-04** Der Näherungswert und die Fehlerabschätzung werden vor der Summierung in den Statistik-Registern wieder in das X- und Y-Register gespeichert.
f Σ+ **2,000** **00** Summierung der Näherungswerte und der Fehlerabschätzungen.
RCL f Σ+ **1,000** **00** Das Integral über den ganzen Bereich (0,228).
x:y **2,598** **-04** Die Fehlerabschätzung des Integrals.
- Die Berechnung des Integrals über zwei Teilbereiche dauerte nur einen Bruchteil der Zeit, die für die Berechnung des Integrals über (0,228) benötigt wurde. Die summierte Fehlerschätzung der beiden Teilergebnisse ist dabei kaum größer als die Fehlerschätzung, die bei der Integration des gesamten Bereichs erzeugt wurde.

TRANSFORMATOREN VON VARIABLEN

In vielen Fällen, in denen sich die Funktion nur langsam innerhalb des Integrationsintervalls verändert, kann die Rechenzeit durch eine geeignete Transformation der Variablen verkürzt werden.

Wir betrachten nochmal das Integral

$$\int_0^{\infty} xe^{-x} dx$$

Es sei $u = e^{-x}$.

Dann gilt $x = -\ln u$

und $dx = -\frac{du}{u}$.

Durch Substitution erhalten wir

$$\begin{aligned}\int_0^{\infty} xe^{-x} dx &= \int_{e^{-\infty}}^{e^{-0}} (-\ln u) (u) \left(\frac{du}{u} \right) \\ &= \int_1^0 \ln u du.\end{aligned}$$

Schieben Sie den **PRGM**  **RUN**-Schalter in Stellung PRGM und tasten Sie das Unterprogramm zur Berechnung der Funktion $f(n) = \ln u$ ein.

Tastenfolge	Anzeige
[h] LBL 3	001-25,13, 3
[f] LN	002- 14 1
[h] RTN	003- 25 12

Schieben Sie den **PRGM**  **RUN**-Schalter wieder in Stellung RUN, geben Sie die Integrationsgrenzen ein und drücken Sie dann **f**  3 um das Integral zu berechnen.

Tastenfolge	Anzeige
1 ENTER	1,000 00
0	0,
f  3	9,998 -01
[x:y]	2,130 -04

Wenn man die Fehlerabschätzung dieses Näherungswertes in Berücksichtigung zieht, so stimmt das Ergebnis mit dem oben gefundenen, ursprünglichen Integral überein. Für die Berechnung wurde jedoch nur ein Bruchteil der Zeit benötigt.

ANHANG C: PFLEGE UND WARTUNG

IHR HEWLETT-PACKARD-RECHNER

Der Rechner ist ein weiteres Beispiel für preisgekröntes Design, überlegene Qualität und Sorgfalt im Detail bei der Entwicklung und Herstellung, die die elektronischen Instrumente von Hewlett-Packard seit mehr als 30 Jahren kennzeichnen. Jeder Hewlett-Packard-Rechner ist ein Stück Präzision, geschaffen von Leuten, die sich der Aufgabe widmen, Ihnen das bestmögliche Produkt zu liefern.

Jeder Rechner wird nach Fertigstellung auf elektrische, mechanische und äußere Mängel untersucht.

NETZBETRIEB

Der Rechner wird mit einem aufladbaren Batteriesatz aus Nickel-Kadmium-Zellen betrieben. Wenn Sie das Gerät erhalten, ist dieser Batteriesatz vielleicht entladen. Sie können den Rechner aber dennoch sofort benutzen, wenn Sie das Netzladegerät verwenden.

Anmerkung: Wenn sich kein Batteriesatz im Gerät befindet, darf das Gerät nicht an das Netz angeschlossen werden.

Das Netzladegerät wird folgendermaßen verwendet:

1. Der Rechner braucht nicht ausgeschaltet zu werden.
2. Stecken Sie den Ladestecker in die rückwärtige Buchse am Rechner, wobei die Verriegelungszunge zur rechten Seite zeigen muß.
3. Stecken Sie den Netzstecker des Ladegeräts in eine Steckdose.

Anmerkung: Es ist völlig normal, daß das Netzladegerät und der Batteriesatz beim Betrieb warm werden.

Vorsicht

Der Rechner kann beschädigt werden, wenn Sie ein anderes als das mitgelieferte Ladegerät verwenden.

Verwenden Sie nur Netzladegeräte des Typs "B". Frühere "A"-Typen werden zwar den Rechner nicht beschädigen, können dagegen aber beim Einsticken den Permanentenspeicher löschen.

BATTERIEBETRIEB

Wenn Sie den Rechner nur im Batteriebetrieb verwenden wollen, entfernen Sie den Ladestecker, indem Sie ihn mit dem Daumen und Zeigefinger zur Entriegelung leicht drücken und herausziehen. (Das Ladegerät kann in der Steckdose bleiben, auch wenn es nicht an den Rechner angeschlossen ist).

Wenn Sie den Rechner netzunabhängig verwenden, können Sie völlig ortsungebunden arbeiten, weil Sie den Rechner praktisch überall hin mitnehmen können. Mit aufgeladenen Batterien ist ein kontinuierlicher Betrieb von etwa 3 Stunden möglich. Indem Sie den Rechner ausschalten, wenn Sie ihn nicht benötigen, reicht dies normalerweise für einen ganzen Arbeitstag aus.

ANZEIGE ABFALLENDER BATTERIESPANNUNG

Wenn Sie netzunabhängig arbeiten und die Batterie nahezu entladen ist, warnt Sie der Rechner durch einen Punkt ganz links in der Anzeige, daß noch bis zu 25 Minuten Batteriebetrieb möglich ist.

• 1,23

Mit dieser Anzeige erscheint das Minuszeichen als unvollständiges Divisionssymbol.

• 1,23

Damit der Rechner wieder eine ausreichende Spannung erhält, muß entweder das Netzladegerät angeschlossen werden oder ein aufgeladener Batteriesatz in den Rechner gesetzt werden.



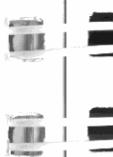
LADEN DER BATTERIE

Die Batterien werden durch das Ladegerät aufgeladen, wenn Sie den Rechner im Netzbetrieb verwenden. Das Gerät kann dabei ein- oder ausgeschaltet sein, solange die Batterie richtig installiert und der Ladestecker angeschlossen ist. Das Laden einer vollständig entladenen Batterie dauert (abhängig von der Netzspannung):

Rechner aus: 5 bis 9 Stunden

Rechner ein: 17 Stunden

Nach einer kürzeren Ladedauer ist nur ein entsprechend kürzerer netzunabhängiger Betrieb möglich. Der Batteriesatz kann nie überladen werden, egal ob der Rechner ein- oder ausgeschaltet ist.

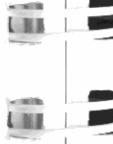


Anmerkung: Das Ladegerät ist eine geschlossene Einheit und kann nicht repariert werden. Senden Sie es an Hewlett-Packard zurück, wenn es schadhaft ist.



DER PERMANENT-SPEICHER

Wenn Sie den Rechner ausschalten, bleiben folgende Informationen erhalten:



- Sämtliche im Rechner gespeicherten Programme.
- Inhalt der Speicherregister.
- Anzeigeformat (FIX, SCI oder ENG und die Anzahl der angezeigten Ziffern).



Wenn der Rechner wieder ausgeschaltet wird, kehrt er nach Zeile 000 (Anfang des Programmspeichers) zurück.



Der Stack-Inhalt, LAST X und der Winkel-Modus (DEG, RAD oder GRAD) bleiben nicht erhalten, wenn Sie den Rechner ausschalten. Alle Flags und anstehenden Rücksprungadressen werden gelöscht.



Der Permanent-Speicher erfordert, daß stets ein Batteriesatz im Rechner eingesetzt ist. Wenn die Anzeige, die auf eine niedrige Spannung hinweist, erscheint, schalten Sie den Rechner sofort aus und schließen Sie das Netzladegerät an oder wechseln Sie die Batterien aus. Wenn die Batterien vollständig entladen werden, kann die ganze Information im Permanent-Speicher verloren gehen.

Wenn Sie das Gerät fallen lassen oder der Strom für den Permanent-Speicher aus anderen Gründen bei ein- oder ausgeschaltetem Rechner unterbrochen wird, kann der Inhalt des Programmspeichers und der Speicherregister verloren gehen. Wenn der Rechner das nächste Mal eingeschaltet wird, erscheint **Pr Error** (Stromunterbrechung) in der Anzeige. (Diese und andere Fehlermeldungen können durch das Drücken einer beliebigen Taste gelöscht werden.)

AUSTAUSCHEN DES BATTERIESATZES

Wenn dies einmal nötig sein sollte, ersetzen Sie den mitgelieferten Batteriesatz durch einen gleichartigen von Hewlett-Packard.

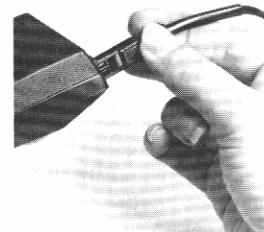
Wegen des Permanent-Speichers sollten die Batterien so schnell wie möglich ausgetauscht werden. Normalerweise stehen Ihnen wenigstens 5 Sekunden zur Verfügung, um die Batterien auszuwechseln. Wenn für längere Zeit kein Batteriesatz im Rechner installiert ist, geht die Information im Permanent-Speicher verloren.

VORSICHT

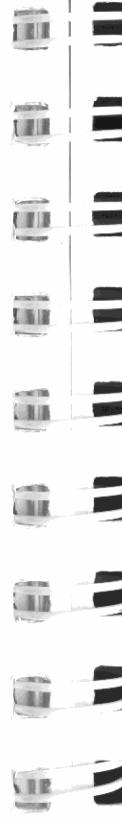
Wenn Sie einen anderen als den Original Hewlett-Packard Batteriesatz in Ihrem Gerät verwenden, kann der Rechner beschädigt werden.

Wenn Sie den Batteriesatz wechseln wollen, verfahren Sie nach folgenden Anweisungen:

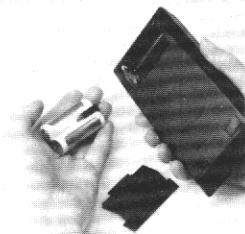
1. Schalten Sie den Rechner aus und ziehen Sie das Ladekabel ab. Dazu ist die federnde Verriegelzungszunge am Ladestecker leicht einzudrücken und der Stecker dann aus der Ladebuchse des Rechners zu ziehen.



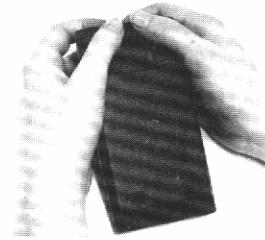
2. Drücken Sie mit dem Daumen auf die schmale Zunge des Batteriefachdeckels und schieben sie diesen dabei gleichzeitig nach oben, bis er aufspringt.



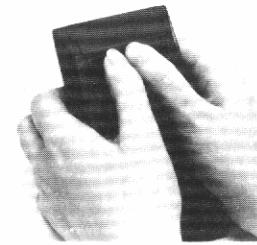
3. Drehen Sie den Rechner nach Abnehmen des Deckels um und lassen Sie den Batteriesatz in Ihre Hand fallen.



4. Setzen Sie die neue Batterie ein (achten Sie auf die richtige Lage der Kontakte) und setzen Sie den Deckel auf.



5. Schieben Sie den Batteriefach-Deckel wieder an seinen Platz.



6. Schalten Sie den Rechner ein und prüfen Sie so das korrekte Einsetzen des Batteriesatzes. Falls die Anzeige dunkel bleibt, ist der Batteriesatz entweder falsch eingesetzt oder entladen.



PFLEGE DES BATTERIESATZES

Auch wenn Sie ihren HP-34C nicht im Batteriebetrieb verwenden, entladen sich die Batterien langsam von selbst. Diese Selbstentladung ist gering und beträgt etwa 1% Kapazitätsverlust pro Tag. Es kann vorkommen, daß die Batterien nach einer Lagerung von 30 Tagen nur noch 50–70% ihrer Kapazität haben und der Rechner sich nicht einschalten läßt. In diesem Fall sollten Sie den Batteriesatz gegen einen geladenen Austausch-Batteriesatz auswechseln oder aber den teilgeladenen Batteriesatz mindestens 14 Stunden lang laden.

Falls sich die Batterien in kurzer Zeit von selbst entladen oder nur eine sehr kurze Betriebszeit zulassen, kann es sein, daß sie defekt sind. Falls die Gewährleistungsfrist von einem Jahr noch nicht abgelaufen ist, senden Sie den Batteriesatz, gemäß den Versandanweisungen, an Hewlett-Packard. Nach Ablauf der Gewährleistungsfrist können Sie mit der Zubehör-Bestellkarte einen neuen Batteriesatz anfordern.

VORSICHT

Versuchen Sie nicht, einen Batteriesatz mit anderen Mitteln zu überladen oder einen alten Batteriesatz ins Feuer zu werfen – die NC-Akkumulatoren können dabei platzen oder giftige Stoffe freisetzen.

TEMPERATURBEREICH

Der Rechner kann im folgenden Temperaturbereich eingesetzt werden.

Betrieb	0° bis 45°	+32° bis 113° F
Laden	+15° bis 40°	+59° bis 104° F
Lagerung	-40° bis 55° C	-40° bis 131° F

KEINE ANZEIGE

Wenn die Anzeige dunkel bleibt oder erlischt, schalten Sie den HP-34C aus und dann wieder ein. Wenn Sie keine Anzeige erhalten, überprüfen Sie die folgenden Punkte:

1. Falls das Ladegerät angeschlossen ist, sollten Sie prüfen, ob die verwendete Steckdose auch unter Spannung steht.
2. Überprüfen Sie, ob vielleicht die Kontakte des Batteriesatzes verschmutzt sind.
3. Tauschen Sie den Batteriesatz, wenn möglich, gegen einen geladenen Reserve-Batteriesatz aus.
4. Wenn die Anzeige noch immer ausbleibt, versuchen Sie, den Rechner (mit eingeschaltetem Batteriesatz) am angeschlossenen Ladegerät zu betreiben.
5. Wenn Sie jetzt noch immer keine Anzeige erhalten, ist der Rechner defekt (siehe Absatz *Gewährleistung*).

GEWÄHRLEISTUNG

Hewlett-Packard gewährleistet, daß der Rechner frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn der Rechner – direkt oder über einen autorisierten Hewlett-Packard-Vertragshändler – an Hewlett-Packard eingeschickt wird. Die Gewährleistungsfrist beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die durch unsachgemäße Bedienung oder Gewalteinwirkung entstanden bzw. auf Reparaturen oder Veränderungen des Rechners durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt.

Die Gewährleistung gilt nur in Verbindung mit entweder

- a) dem von einem Hewlett-Packard-Vertragshändler ausgestellten Kaufbeleg und der vollständig ausgefüllten, von diesem Hewlett-Packard-Vertragshändler unterschriebenen Service-Karte, oder
- b) der Original-Rechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Gewährleistungsregelung unberührt.

Nach Ablauf der Gewährleistungsfrist werden Instandsetzungen gegen Berechnung ausgeführt. Die Gewährleistungsfrist auf Instandsetzungsarbeiten beträgt 180 Tage.

VERSANDANWEISUNGEN

Bei fehlerhaftem Arbeiten des Rechners schicken Sie bitte:

- Den Rechner mit Batterien und Ladegerät
- Den Kaufbeleg, aus dem das Kaufdatum des Gerätes ersichtlich ist
- Die komplett ausgefüllte Service-Karte

direkt – oder über einen autorisierten Hewlett-Packard-Vertragshändler – an die nächstgelegene Hewlett-Packard Service-Niederlassung.

Die Geräte sollten zur Vermeidung von Transportschäden gut verpackt und gegen Verlust ausreichend versichert werden, da Hewlett-Packard hierfür keine Gewährleistung übernimmt.

Die Kosten für die Rücksendung des instandgesetzten Gerätes werden im Fall der Gewährleistung von Hewlett-Packard übernommen.

REPARATURDAUER

Normalerweise erfolgt die Instandsetzung eingesandter Geräte und der Rückversand innerhalb von fünf Werktagen. Dieser Wert ist allerdings als Mittelwert anzusehen. In Abhängigkeit von der Belastung der Service-Abteilung kann im Einzelfall diese Frist von fünf Tagen auch einmal überschritten werden.

SONSTIGES

Service-Verträge werden zu diesem Rechner nicht angeboten. Ausführung und Entwurf des Rechners und der Elektronik sind geistiges Eigentum von Hewlett-Packard; Service-Handbücher können daher an Kunden nicht abgegeben werden.

Sollten weitere servicebezogene Fragen auftreten, so rufen Sie eine der nächsten HP-Niederlassungen an.

TECHNISCHE ÄNDERUNGEN

Hewlett-Packard behält sich technische Änderungen vor. Die Produkte werden auf der Basis der Eigenschaften verkauft, die am Verkaufstag gültig waren. Eine Verpflichtung zur Änderung einmal verkaufter Geräte besteht nicht.

ANHANG D: UNERLAUBTE OPERATIONEN

 Wenn Sie versuchen, eine der folgenden unerlaubten Operationen (wie beispielsweise die Division durch Null) auszuführen, zeigt der Rechner in der Anzeige das Wort **ERROR** und eine Zahl an.

 Die Fehlermeldung kann durch das Drücken einer beliebigen Taste gelöscht werden.

 Die folgenden Operationen zeigen das Wort **Error** und eine Zahl an.

Error 0

	Wenn $x = 0$.
	wenn $y = 0$ und $x \leq 0$.
	wenn $y \leq 0$ und x nicht ganzzahlig.
	wenn $x \leq 0$.
	wenn $x = 0$.
	wenn $x \leq 0$.
	wenn $x \leq 0$.
	wenn $x > 1$.
	COS^{-1} wenn $x > 1$.
	$\text{STO } \pm$ wenn $x = 0$.
	$\Delta\%$ wenn $y = 0$.

Error 1: Speicherregisterüberlauf

Überlauf nach Speicherregister-Arithmetik (nicht bei $\Sigma+$ - und $\Sigma-$ -Operationen). Der Absolutwert der Zahl im Speicherregister würde $9.999999999 \times 10^{99}$ überschreiten.

Error 2: Unerlaubte Speicherregisternummer

Das angesprochene Register ist augenblicklich in Programmspeicher umgewandelt oder existiert nicht.

Error 3: Unerlaubte statistische Operation:

- wenn $n = 0$.
- wenn $n \leq 1$.

Anmerkung: Error 3 wird auch dann angezeigt, wenn Division durch Null oder die Wurzel einer negativen Zahl in einer der untenstehenden Formeln benötigt wird:

$$s_x = \sqrt{\frac{M}{n(n-1)}} \quad s_y = \sqrt{\frac{N}{n(n-1)}} \quad r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M} \quad B = \frac{M \sum y - P \sum x}{n \cdot M}$$

A und B sind die beiden Ergebnisse der [LR] Operation wobei $y = Ax + B$ ist.

$$\hat{y} = \frac{M \sum y + P(n \cdot x - \sum x)}{n \cdot M}$$

wobei:

$$M = n \sum x^2 - (\sum x)^2$$

$$N = n \sum y^2 - (\sum y)^2$$

$$P = n \sum xy - \sum x \sum y$$

Error 4: Unerlaubter Sprung zu einer Zeilennummer oder Marke

Die angesprochene Zeilennummer ist augenblicklich nicht belegt oder existiert nicht (> 210), die Marke existiert nicht oder es wird versucht eine Programmanweisung in nicht vorhandenen Speicher zu laden.

Error 5

Wiederholter Aufruf der **[S]**-oder **[SOLVE]**-Routine, d. h. **[S]** oder **[SOLVE]** innerhalb eines Unterprogramms werden durch ein weiteres **[S]** bzw. **[SOLVE]** aufgerufen.

Error 6

[SOLVE] findet keine Lösung mit den gegebenen Schätzwerten.

Error 7:

Verwendung einer unerlaubten Marke (4–9) in Verbindung mit **[S]** oder **[SOLVE]** oder eine unerlaubte Flag-Bezeichnung (4–9).

Error 8

Unerlaubte Verschachtelung von Unterprogrammen.

Error 9

Die Selbstprüfroutine (**[STO]** **[ENTER]**) hat einen Fehler im Schaltungsaufbau entdeckt. Beachten Sie, daß der Programmspeicher die Speicherregister und das Anzeigeformat durch die Ausführung der Selbstprüfroutine nicht gelöscht werden.

Pr Error

Der Permanent-Speicher ist wegen einer Stromunterbrechung gelöscht worden.

ANHANG E: STACK-LIFT UND LAST X

Ihr Rechner wurde so entworfen, daß er sich auf natürliche, normale Weise bedienen läßt. Beim Erarbeiten der in diesem Handbuch aufgeführten Aufgaben und Beispiele ist Ihnen bestimmt aufgefallen, wie selten Sie sich um die automatischen Stack-Operationen kümmern mußten. In der Hauptsache haben Sie ihre Berechnungen durchgeführt, als wenn Sie sie Schritt für Schritt mit Bleistift und Papier erledigen würden.

Es kann aber vorkommen, und dies besonders, wenn Sie ein Programm erstellen, daß Sie die Auswirkung einer gewissen Operation auf den Stack wissen wollen. Die folgende Erläuterung und Tabelle sollte Ihnen dabei behilflich sein.

ABSCHLUSS EINER ZIFFERNEINGABE

Die meisten auf dem Rechner ausgeführten Operationen, seien Sie als Instruktion innerhalb eines Programms oder über die Tastatur ausgeführt, schließen eine Zifferneingabe ab. Das bedeutet, daß nach diesen Operationen eingegebene Ziffern für den Rechner Teil einer neuen Zahl sind.

STACK-LIFT

Es gibt drei Arten von Operationen auf dem Rechner, wenn man sie in Hin- sicht ihrer Beeinflussung des Stacks betrachtet. Es sind Operationen, die den Stack-Lift unwirksam machen und ihn nicht beeinflussen.

OPERATIONEN, DIE DEN STACK-LIFT UNWIRKSAM MACHEN

Der Rechner enthält nur vier Operationen, die den Stack-Lift unwirksam machen, so daß eine nach dieser Operation eingegebene Zahl lediglich den Inhalt des angezeigten **X**-Registers überschreibt, ohne daß der Stack angehoben wird.

Diese Sonderoperationen sind die folgenden:

ENTER+

CLX

$\Sigma +$

$\Sigma -$

OPERATIONEN, DIE DEN STACK-LIFT WIRKSAM MACHEN

Der größte Teil der Rechenoperationen, so auch mathematische Funktionen mit einem oder zwei Argumenten wie x^2 und \sqrt{x} , machen den Stack-Lift wirksam, so daß eine nach einer dieser Operationen eingegebene Zahl den Stack anhebt.

Beachten Sie, daß durch ein Umschalten vom PRGM-Modus in den RUN-Modus der Stack-Lift wirksam gemacht wird.

OPERATIONEN, DIE DEN STACK NICHT BEEINFLUSSEN

Einige Operationen verhalten sich neutral, d. h. sie verändern nicht den bestehenden Status des Stack-Lifts. Wenn der Stack-Lift z. B. durch das Drücken der **ENTER**-Taste unwirksam ist und dann **f FIX** und eine neue Zahl eingetastet wird, überschreibt diese Zahl den Inhalt des X-Registers, ohne daß der Stack angehoben wird. In ähnlicher Weise wird der Stack aufgehoben, wenn der Stack-Lift nach Drücken z. B. der **x^2** -Taste "eingeschaltet" war und dann eine **GTO 03** Anweisung von einer Ziffernangabe gefolgt wird. Die untenstehende Tabelle führt alle neutralen Operationen auf.

FIX	BST
SCI	SST (Im RUN-Modus kann SST eine
ENG	Operation ausführen, die dann
DEG	– ihrerseits – den Stack-Lift
RAD	ermöglicht).
GRD	
DSP 1	MEM
GTO <input checked="" type="checkbox"/> nnn	CLEAR PREFIX
	CLEAR REG

CLEAR	X
CHS	*
MANT	
R/S	
PSE	

LAST X

Untenstehend wird angegeben, nach welcher Operation der Inhalt des X-Registers im LAST X-Register gespeichert wird.

-	$\Sigma +$	e^x	\sqrt{x}
+	$\Sigma -$	LOG	x^2
\times	%	10^x	$1/x$
\div	$\Delta %$	SIN	y^x
-H.MS	y	SIN^{-1}	$\Rightarrow R$
-H	$x!$	COS	$\Rightarrow P$
ABS	FRAC	COS^{-1}	
-R	INT	TAN	
-D	LN	TAN^{-1}	

* Nur neutral im Zusammenhang mit einer Zifferneingabe.

Serie E Service Information

Sollte Ihr Rechner der Serie E nicht fehlerfrei arbeiten, wenden Sie zuerst die Fehlerselbstkontrolle an. Falls in der Anzeige keine Zahl erscheint, verfahren Sie wie im Anhang A Ihres Bedienungshandbuchs beschrieben. Wenn Sie sich davon überzeugt haben, daß Ihr Rechner defekt ist, schicken Sie ihn, entsprechend den Versandanweisungen im Bedienungshandbuch, an Hewlett-Packard.

Welche Fehler treten auf?

- | | | |
|----------------------------------------|-------------------------------------------------------------------------------|-------------------------------------------------------|
| <input type="checkbox"/> Keine Anzeige | <input type="checkbox"/> Inkorrekte Anzeige nach
der Fehlerselbstkontrolle | <input type="checkbox"/> Tastenfeld/Tasten |
| <input type="checkbox"/> Ladegerät | <input type="checkbox"/> Batterie | <input type="checkbox"/> zeitweise auftretende Fehler |

Beschreiben Sie das Problem:

Gewährleistung?

Ja Nein

Gewährleistungsansprüche können nur durch Kaufbeleg und der umseitig von Ihrem HP-Vertragshändler bei Kaufdatum ausgefüllten Service-Karte oder der Original-Rechnung von Hewlett-Packard geltend gemacht werden.

Für Produkte, die nicht mehr unter die Gewährleistung fallen, wird Hewlett-Packard mit dem Einsenden der Service-Karte autorisiert, die nötigen Reparaturen auszuführen und die anfallenden Kosten dem Eigentümer in Rechnung zu stellen.

Ihr HP-Vertragshändler wird Sie gern über die zur Zeit gültigen Reparaturkosten Ihres Modells informieren.

HP-34C Mitteilungskarte

Beantworten Sie bitte die folgenden Fragen, damit wir unseren Kunden noch besseren Service erweisen können.

1. In welchem Land wurde der Rechner gekauft?

2. Verkaufsdatum:

Monat _____ Jahr _____

3. Wo wurde der Rechner gekauft?

1	Kaufhaus
2	Fachgeschäft
3	Radio-Fotogeschäft
4	Studentische Organisation
5	Hewlett-Packard
6	Andere

4. Wann wurde der Rechner geliefert?

1	Sofort
2	Innerhalb zwei Wochen
3	Später als zwei Wochen

5. Der Rechner wurde gekauft:

1	Privat
2	Von meiner Firma
3	Geschenk

6. Woher kennen Sie den Rechner?

1	Presseartikel
2	Anzeige
3	Prospekt
4	Freund/Kollege
5	Mein Professor benutzt ihn
6	Benutze bereits HP-Rechner
7	Im Fachgeschäft vorgeführt
8	Im Büro vorgeführt

7. Wie fanden Sie...
...die Dokumentation? ...den Inhalt?

1	Hervorragend	11	
2	Gut	12	
3	Zufriedenstellend	13	
4	Mangelhaft	14	
5	Nicht erforderlich	15	

10. Ihre Hauptanwendungsbereiche:

(Bitte mit 1., 2., 3. usw. angeben)

1	Physik
2	Chemie
3	Medizin/Biologie
4	Mathematik
5	Statistik
6	Informatik
7	Elektrotechnik
8	Maschinenbau
9	Hoch-/Tiefbau
10	Technik (allgemein)
11	Kaufmännisch

9 Ingenieurwissenschaften

10 Wirtschaftswissenschaften

11 Sozialwissenschaften

12 Andere

8. Ihre Rechner-Erfahrung?

1	Mein erster Rechner
2	14-Funktionen-Rechner

Technisch-wissenschaftlicher Rechner
Nicht programmierbar Programmierbar

30	HP	50
31	T.I.	51
32	Casio	52
33	Sharp	53
34	Brother	54
38	MBO	58
42	Triumph/Adler	62
37	Commodore	57

11. Ihre Ausbildung bzw.

Ihr augenblickliches Studium?

1	Lehre
2	Gymnasium

Fachschule:

3	Technisch
4	Kaufmännisch
5	Andere

Universität:

6	Naturwissenschaften
7	Medizin
8	Mathematik

12. Ihre Altersgruppe:

1	Unter 14
2	15-19
3	20-25

4 26-35

5 36-50

6 Über 50

13. Ihr Arbeitsgebiet:

1	Studium
11	Erziehung/Ausbildung
12	Forschung/Entwicklung
13	Produktion
14	Organisation/Planung
15	Marketing/Verkauf
16	Buchhaltung/Finanzen
17	EDV
18	Geschäftsleitung
19	Ingenieurberatung
20	Unternehmensberatung
21	Freiberuflich

14. Ihre Stellung:

1	Inhaber/Vorstand
2	Geschäftsführer
3	Abteilungsleiter
4	Gruppenleiter
5	Sachbearbeiter
6	Verkäufer

Vielen Dank, daß Sie sich die Zeit genommen haben, diese Karte zurückzusenden.

Serie EC



Rechner

Service-Karte

Verkaufsinformation

(Vom HP-Vertragshändler auszufüllen)

Modell HP 34C

Serien-Nr.: 2128538464

Kaufdatum: _____

Rechnungs-Nr.: _____

Gekauft bei: W. Lampert OHG

Wilhelm Lampert OHG
Büro und Zeichneramt
Uhlandstr. 34-35 68028
6800 Mannheim

(Vertragshändler Unterschrift/Stempel)

Kundeninformation

Name: _____

Straße: _____

PLZ/Ort: _____

Tel.: _____

Name: _____

Straße: _____

PLZ/Ort: _____

Land: _____

Firma: _____

Stellung: _____

Zur Reparatur sollten Sie Ihren Rechner unter keinen Umständen an die folgende Adresse senden.

**HEWLETT-PACKARD S.A.
P.O. Box
CH-1217 MEYRIN 2
Switzerland**

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please do not make copies of this scan or
make it available on file sharing services.