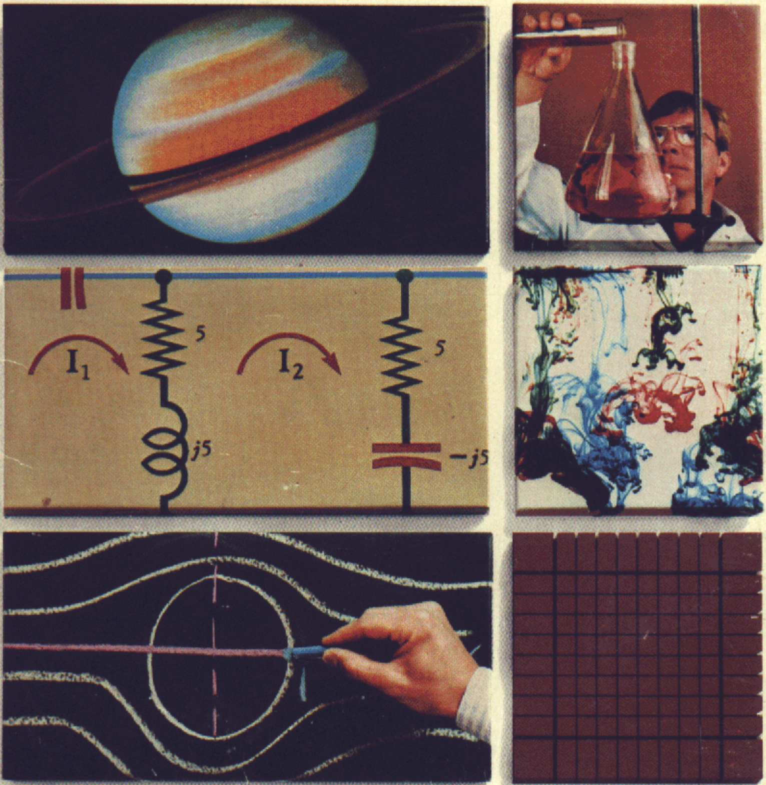
HEWLETT-PACKARD

HP-15C

BENUTZERHANDBUCH



HEWLETT PACKARD

HP-15C

Benutzerhandbuch

Juli 1982

00015-90002

Printed in SGP

Einleitung

Gleichgültig ob Sie ein Neuling unter den Benutzern von HP Rechnern sind, oder schon Erfahrung haben, werden Sie den HP-15C in der Welt der Taschenrechner als unerreicht einstufen. Neben dem Permanentspeicher und einem niedrigen Energieverbrauch, umfaßt das Leistungsspektrum des HP-15C folgende Eigenschaften:

- 448 Bytes zur Programmspeicherung (ein oder zwei Bytes pro Anweisung) und leistungsvolle Programmierung (bedingte und unbedingte Verzweigungen, Unterprogramme, Flags, Programmeditierung.
- Vier Funktionen der höheren Mathematik: Berechnungen mit komplexen Zahlen, Matrizenrechnungen, Berechnungen von Nullstellen und numerische Integration.
- Direkte und indirekte Speicherung in bis zu 67 Registern.
- Lange Batterielebensdauer.

Dieses Handbuch ist für jeden Benutzer geschrieben, gleich welche Erfahrung im Umgang mit Rechnern er mitbringt. Der Teil I, «Grundlagen des HP-15C», beinhaltet alle elementaren Funktionen und erklärt ihre Verwendung. Jeder Abschnitt im zweiten Teil, Programmierung, ist dreifach unterteilt: das Handwerkszeug, Beispiele und zusätzliche Informationen. Diese Aufteilung soll das Auffinden der benötigten Information erleichtern. Teil III «Funktionen der höheren Mathematik» beschreibt die vier fortgeschrittenen mathematischen Operationen* Ihres HP-15C.

Bevor Sie mit diesen Abschnitten beginnen, können Sie einige Erfahrung in der Handhabung und der Programmierung des HP-15C gewinnen, indem Sie den Einleitungsabschnitt, «Der HP-15C: Ein Problemlöser» (siehe Seite 12), durcharbeiten.

Die verschiedenen Anhänge erläutern zusätzliche Einzelheiten der Rechneroperationen und bringen Informationen über die Gewährleistung und den Service. Die Abschnitte, «Index Funktionstasten» und «Index «Programmtasten», am Ende dieses Handbuches können als eine kurze Erläuterung jeder Funktionstaste verwendet werden, und als handliches Seitenzahlenverzeichnis für tiefergehende Erklärungen dienen.

---

* Sollten Sie schon mit HP Rechnern vertraut sein, brauchen Sie sicher Teil I und II nicht zu lesen, bevor Sie sich in den Teil III, «Funktionen der höheren Mathematik» durcharbeiten. Der Gebrauch der Tasten [SOLVE] und [∫] verlangt eine gewisse Kenntnis der Programmierung des HP-15C.

Ihr Hewlett-Packard Händler hält außerdem ein Ergänzungshandbuch, *«HP-15C – Fortgeschrittene Funktionen»*, für Sie bereit, das Anwendungen und technische Beschreibungen zu den Bereichen Integration, Nullstellenbestimmung, komplexe Zahlen und Matrizenrechnungen, enthält.

Inhaltsverzeichnis

Der HP-15C: Ein Problemlöser

Der fortgeschritten programmierbare, wissenschaftliche Taschenrechner HP-15C ist ein leistungsfähiger und handlicher Problemlöser. Sein Permanentspeicher erhält Daten und Programmanweisungen solange, bis Sie entscheiden, die gespeicherte Information zu löschen. Trotz seiner anspruchsvollen Verwendungsfähigkeiten, ist Erfahrung in der Programmierung oder die Kenntnis einer Programmiersprache für die Bedienung des HP-15C nicht notwendig.

Eine wichtige neue Eigenschaft Ihres HP-15C ist sein extrem niedriger Energieverbrauch. Dies ist verantwortlich für sein geringes Gewicht, sein kompaktes Design und macht ein Ladegerät entbehrlich. Der Energieverbrauch des HP-15C ist so niedrig, daß die Durchschnittslebensdauer einer normalen Batterie im HP-15C sechs bis zwölf Monate beträgt. Zusätzlich warnt Sie eine Spannungsabfallanzeige, bevor der Rechner seine Funktion einstellt.

Der HP-15C hilft Ihnen außerdem Energie zu sparen, indem er seine Anzeige automatisch abschaltet, wenn einige Minuten keine Eingabe erfolgt ist. Trotzdem gehen keinerlei Informationen verloren – jede eingegebene Information bleibt durch den Permanentspeicher erhalten.

Eine kurze Einführung der Taste [ENTER]

Ihr Hewlett-Packard Taschenrecher verwendet eine einzigartige Operationslogik, die durch die Taste [ENTER] verkörpert wird, und sich wesentlich von der in anderen Rechnern verwendeten Operationslogik unterscheidet. Sie werden erkennen, daß sich verschachtelte und komplizierte Berechnungen durch die Verwendung der Taste [ENTER] leicht und schnell durchführen lassen. Wir wollen Sie im folgenden mit der Arbeitsweise dieser Taste vertraut machen.

Lassen Sie uns, z.B., die arithmetischen Funktionen betrachten. Zuerst müssen wir eine Zahl in den Taschenrechner eingeben. Ist Ihr Rechner überhaupt eingeschaltet? Wenn nicht drücken Sie die Taste-[ON]. Ist die Anzeige (das Display) gelöscht? Um nur Nullen in der Anzeige erscheinen zu lassen, können Sie [g] [CLx] drücken, d.h. zuerst die Taste [g] und dann die Taste [←]*.

---

* Falls Sie noch nie zuvor einen HP Taschenrechner bedient haben, werden Sie bemerken, daß die meisten der Tasten dreifach belegt sind, d.h. drei Bezeichnungen haben. Um die Primärfunktion zu verwenden, die in weiß auf die Taste gedruckt ist, müssen Sie nur die Taste selbst drücken. Für die in gold und blau aufgedruckten Funktionen, müssen Sie zuvor die goldfarbene Taste [f] oder die blaue Taste [g] drücken.

Zur Ausführung einer arithmetische Berechnung geben Sie zuerst die erste Zahl ein, drücken die Taste [ENTER], um die erste Zahl von der zweiten zu trennen und geben danach die zweite Zahl ein. Anschließend können Sie eine der Tasten [+], [-], [×] oder [÷] drücken. Das Ergebnis erscheint sofort, nachdem Sie eine numerische Funktionstaste gedrückt haben.

Sämtliche in diesem Handbuch illustrierten Anzeigen haben das Format [FIX] 4 (der Dezimalpunkt ist so gesetzt, daß vier Stellen rechts des Dezimalpunkts erscheinen), sofern anderes nicht explizit vermerkt wird. Erscheinen keine vier Stellen rechts des Dezimalpunkts, nachdem Sie Ihren Rechner eingeschaltet haben, drücken Sie [f] [FIX] 4.

Manuelle Lösungen

Es ist nicht notwendig, den Rechner zwischen einzelnen Problemstellungen zu löschen. Sollten Sie bei der Eingabe einer Ziffer einer Zahl einen Fehler gemacht haben, drücken Sie [←] und geben die korrekte Ziffer ein.

Problem	Tastenfolge	Anzeige
$9-6=3$	9 [ENTER] 6 [-]	**3.0000**
$9\times 6=54$	9 [ENTER] 6 [×]	**54.0000**
$9\div 6=1.5$	9 [ENTER] 6 [÷]	**1.5000**
$9^6=531{,}441$	9 [ENTER] 6 [y^x]	**531,441.0000**

Beachten Sie, daß in den vier Beispielen:

- beide Zahlen im Rechner sind, bevor die Funktionstaste gedrückt wird.
- [ENTER] zur Trennung der beiden nacheinander eingegebenen Zahlen verwendet wird.
- Das Auslösen einer numerischen Funktionstaste, in diesem Fall [+], [-], [×], oder [y^x], die sofortige Ausführung der Funktion und die Anzeige des Ergebnisses verursacht.

Um die enge Beziehung zwischen einer manuellen und einer programmierten Problemlösung zu verdeutlichen, soll zunächst die Lösung eines Problems manuell, d.h. über das Tastenfeld, berechnet werden. Danach werden wir ein Programm zur Lösung desselben und ähnlicher Probleme benutzen.

Vernachlässigt man den Luftwiderstand, so ist die Fallzeit eines Körpers zum Boden durch die Formel

$$t = \sqrt{\frac{2h}{g}}$$

leicht zu berechnen, wobei

t = die Zeit in Sekunden,
h = die Höhe in Meter, aus welcher der Körper zu Boden fällt und
g = die Gravitationskonstante, 9.81 m/sec² angeben.

Beispiel: Berechnen Sie die Fallzeit eines Steines von der Spitze des Eiffelturms (300.51 Meter) zum Boden.

Tastenfolge	Anzeige	
300.51 [ENTER]	**300.51**	Eingabe der Höhe h.
2 [×]	**601.0200**	Berechnung von $2h$.
9.8 [÷]	**61.3286**	Berechnung von $2h/g$.
[√x]	**7.8313**	Fallzeit in Sekunden.

Programmierte Lösungen

Angenommen Sie wollen die Fallzeit für verschiedene Höhen berechnen. Die einfachste Methode ist ein Programm zu schreiben, das die konstanten Teile der Berechnung enthält und sie für die variablen Daten bereitstellt.

Schreiben des Programms. Das Programm besteht aus der gleichen Tastenfolge wie bei der manuellen Lösung des Problems. Zwei zusätzliche Anweisungen, ein Label und ein Rücksprung (return) werden benötigt, um den Anfang und das Ende des Programms zu definieren. Des weiteren muß das Programm die Eingabe neuer Daten ermöglichen.

Laden des Programms. Um ein Programm für das obige Beispiel in den HP-15C zu laden, müssen Sie die folgenden Tasten in der angegebenen Reihenfolge drücken. (In der Anzeige erscheinen Informationen, die für Sie erst später von Interesse sind und im Moment ignoriert werden können.)

Tastenfolge	Anzeige	
g P/R	000-	Der Rechner wird in den Programm-Modus geschaltet (**PRGM** Statusanzeige erscheint).
f CLEAR PRGM	000-	Löschen des Programmspeiches (dieser Schritt kann ausgelassen werden).
f LBL A	001- 42,21,11	Label A definiert den Programmanfang.
2	002- 2	Eingabe wie bei der manuellen Lösung des Problems.
×	003- 20	
9	004- 9	
·	005- 48	
8	006- 8	
÷	007- 10	
√x	008- 11	
g RTN	009- 43 32	«Return» definiert das Ende des Programms.
g P/R	7.8313	Schaltet den Rechner wieder auf Run-Modus (**PRGM** Statusanzeige erlischt).

Ausführen des Programms. Geben Sie die folgenden Werte ein, um das Programm zu starten.

Tastenfolge	Anzeige	
300.51	300.51	Höhe des Eiffelturms.
f A	7.8313	Die zuvor manuell berechnete Fallzeit.
1050 f A	14.6385	Die Fallzeit in Sekunden eines Steines aus einer Höhe von 1050 Metern.

Mit dem geladenen Programm läßt sich nun sehr schnell die Fallzeit eines Objektes aus verschiedenen Höhen berechnen. Sie brauchen lediglich die gewünschte Höhe einzugeben und [f] [A] zu drücken. Berechnen Sie beispielsweise die Fallzeit für folgende Höhen: 100 m, 2 m, 275 m und 2000 m.

Die Ergebnisse sind: 4.5175 s, 0.6389 s, 7.4915 s und 20.2031 s.

Dieses Programm war relativ einfach. Sie werden viele Aspekte und Einzelheiten der Programmierung in Teil II kennenlernen. Doch zuerst blättern Sie zu Teil I um, und werfen Sie einen tiefergehenden Blick auf einige Grundzüge der Bedienung des Rechners.

Teil I
Grundlagen des HP-15C

Abschnitt 1

Der Einstieg

Ein- und Ausschalten des Rechners

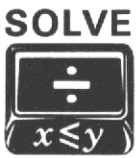
Mit der Taste [ON] wird die Anzeige des HP-15C ein- und ausgeschaltet*. Um Energie zu sparen, schaltet sich der Rechner automatisch ab, wenn für einige Minuten keine Eingabe erfolgt ist.

Bedienung des Tastenfelds

Primär- und Alternativfunktionen

Fast alle Tasten Ihres HP-15C beinhalten eine Primär- und zwei Alternativfunktionen. Die Primärfunktion wird durch den Text auf der Oberseite der Taste beschrieben. Die beiden Alternativfunktionen werden durch die in blau gedruckten Symbole auf der Vorderseite der Taste, bzw. durch die in goldfarben aufgetragenen Symbole oberhalb der Taste, angezeigt.

- Um die Primärfunktion auf der Oberseite der Taste anzuwählen, brauchen Sie nur die Taste selbst zu drücken; hier z.B. [÷].
- Um die in blau oder goldfarben gekennzeichneten Alternativfunktionen anzuwählen, ist zuerst die gleichfarbige Vorwahltaste ([f] oder [g]) und dann die gewünschte Funktionstaste zu drücken, hier z.B.: [f] [SOLVE], [g] [x≤y].



In diesem Handbuch werden Sie verschiedene Konventionen bei der Bezugnahme auf Funktionen entdecken. Die Funktion *selbst* erscheint einfach mit dem Namen der Funktion in einem Rahmen (z.B. «die Funktion [MEM]»). Bei Erläuterungen in Bezug auf die *Verwendung der Taste* hingegen, werden die Vorwahltasten vorangestellt (z.B. «drücken Sie [g] [MEM]»). Den vier, unter der mit «CLEAR» bezeichneten Klammer, goldfarbenen Funktionen wird im Text immer das Wort «CLEAR» vorangestellt, z.B. «die Funktion CLEAR [REG]» oder «drücken Sie [f] CLEAR [PRGM]».

* Die Taste [ON] ist flacher als die anderen Tasten, um ein ungewolltes Drücken zu verhindern.

Beachten Sie, daß nach dem Drücken der Vorwahltasten [f] oder [g], die Statusanzeigen **f** bzw. **g** in der Anzeige erscheinen. Die jeweilige Statusanzeige erlischt, wenn eine Funktionstaste zum Abschluß der Eingabesequenz gedrückt wird.

0.0000 f

Vorwahltasten

Eine Vorwahltaste ist eine Taste, die einer anderen vorangestellt werden muß, um die Tastenfolge zur Vervollständigung einer Funktion zu vervollständigen. Gewisse Funktionen verlangen zwei Teile: eine Vorwahltaste und eine Ziffer oder eine andere Taste. Die Vorwahltasten des HP-15C sind:

[CF]	[ENG]	[FIX]	[GSB]	[$\int_x^y$]	[MATRIX]	[SCI]	[STO]
[DIM]	[f]	[g]	[HYP]	[ISG]	[RCL]	[SF]	[TEST]
[DSE]	[F?]	[GTO]	[HYP^{-1}]	[LBL]	[RESULT]	[SOLVE]	[$x \gtrless$]

Wenn Sie beim Eintasten der Vorwahl für eine Funktion einen Fehler machen, drücken Sie [f] CLEAR [PREFIX], um den Fehler zu annullieren. Da die Taste [PREFIX] auch zur Anzeige der gesamten Mantisse einer Zahl, die sich in der Anzeige befindet, benutzt werden kann, erscheinen nach dem Drücken der Taste [PREFIX] für einen Moment alle Stellen der Zahl.

Vorzeichenwechsel

Das Drücken der Taste [CHS] (*change sign*) ändert das Vorzeichen (positiv oder negativ) der angezeigten Zahl. Um eine negative Zahl einzugeben, müssen Sie nach der Eingabe der Ziffern, die Taste [CHS] drücken.

Eingabe von Exponenten

Die Taste [EEX] (*enter exponent*) ist zu benutzen, wenn ein Exponent als Teil einer Zahl eingegeben werden soll. Zuerst tasten Sie die Mantisse ein, drücken dann die Taste [EEX] und geben danach den Exponenten ein. Im Falle eines negativen Exponenten müssen Sie, nach der Eingabe des Exponenten, die Taste [CHS] drücken*. Üben Sie diese Operation durch die Eingabe des Planck'schen Wirkungsquantum (6.6262×10^{-34} Joule/Sekunden) und multiplizieren Sie diesen Wert mit der Zahl 50.

---

* [CHS] kann auch zwischen [EEX] und dem Exponenten gedrückt werden; dies führt zum gleichen Ergebnis (im Gegensatz zur Mantisse, wo die Zifferneingabe dem [CHS] vorausgehen muß).

Tastenfolge	Anzeige		
6.6262	**6.6262**		
EEX	**6.6262**	**00**	Die beiden Nullen sind die Aufforderung den Exponenten einzugeben.
3	**6.6262**	**03**	(6.6262×10^3).
4	**6.6262**	**34**	(6.6262×10^{34}).
CHS	**6.6262**	**–34**	(6.6262×10^{-34}).
ENTER	**6.6262**	**–34**	Schließt die Zahleneingabe ab.
50 ×	**3.3131**	**–32**	Ergebnis in Joule/Sekunden.

Bemerkung: In den Exponentenbereich der Anzeige eingetastete Ziffern verschwinden beim Auslösen von EEX, bleiben aber intern gespeichert.

Um irreführende Anzeigen zu vermeiden, führt die Funktion EEX keine Operation mit Zahlen durch, die mehr als sieben Vorkommastellen, oder mehr als fünf Nullen rechts des Dezimaltrennzeichens stehen haben. Um solche Zahlen einzugeben, müssen diese in eine Form mit einem geeigneten, höher- oder niederwertigeren Exponenten gebracht werden. Beispielsweise kann die Zahl $123456789.8 \times 10^{23}$ als $1234567.898 \times 10^{25}$, die Zahl $0.00000025 \times 10^{-15}$ als 2.5×10^{-22} dargestellt werden.

Die «CLEAR» Tasten

Löschen bedeutet hier, eine Zahl durch Null zu ersetzen. Die folgende Tabelle zeigt die verschiedenen «Lösch»-Operationen des HP-15C (Fortsetzung auf der nächsten Seite):

Tastenfolge	Wirkung
g CLx	Löscht das X-Register.
←	
Im Run-Modus:	Löscht die letzte Ziffer **oder** die gesamte Anzeige.
Im Programm-Modus:	Löscht die zuletzt eingegebene Anweisung.
f CLEAR Σ	Löscht die Statistik-Speicherregister, die Anzeige und den Stack (siehe Abschnitt 3).

Tastenfolge	Wirkung
[f] CLEAR [PRGM]	
Im Run Modus:	Setzt den Programmspeicher auf die Zeile Null zurück.
Im Programm-Modus:	Löscht den gesamten Programmspeicher.
[f] CLEAR [REG]	Löscht alle Datenspeicherregister.
[f] CLEAR [PREFIX]*	Löscht jede Vorwahl einer erst teilweise eingegebenen Tastenfolge

* Führt auch zur kurzzeitigen Anzeige der Mantisse.

Löschen der Anzeige: [CLx] und [←]

Die Anzeige des HP-15C kann mit zwei verschiedenen Operationen gelöscht werden: [CLx] (*clear x*) und [←] (*Pfeil rückwärts*).

Im Run-Modus:

- [CLx] löscht die Anzeige.
- [←] löscht nur die letzte Ziffer in der Anzeige, wenn die Zahleneingabe noch nicht durch [ENTER] oder eine andere Funktion abgeschlossen wurde. Sie können dann die neue(n) (Ersatz-)Ziffer(n) eingeben. Nach Abschluß der Zifferneingabe wirkt [←] wie [CLx].

Tastenfolge	Anzeige	
12345	**12,345**	Zifferneingabe noch nicht abgeschlossen.
[←]	**1,234**	Löscht nur die letzte Ziffer.
9	**12,349**	
[√x]	**111.1261**	Beendet die Zifferneingabe.
[←]	**0.0000**	Löscht alle Ziffern.

Im Programm-Modus:

- [CLx] ist programmierbar und wird daher als Programmanweisung *gespeichert*. Die sich momentan in der Anzeige befindliche Anweisung wird nicht gelöscht.
- [←] ist *nicht* programmierbar und kann daher zur Programmkorrektur verwendet werden. [←] löscht die gesamte momentan angezeigte Anweisung.

Berechnungen

Funktionen einer Variablen

Als Funktionen einer Variablen bezeichnet man die numerischen Funktionen, die zur Ausführung der Operation nur die Zahl, die sich in der Anzeige befindet, verwenden. Um eine Funktion einer Variablen auszuführen, drücken Sie *nach* Eingabe der Zahl die gewünschte Funktionstaste.

Tastenfolge	Anzeige
45	45
g LOG	1.6532

Funktionen zweier Variablen und die Taste ENTER

Bei einer Funktion zweier Variablen müssen sich zwei Zahlen im Rechner befinden, *bevor* die Funktion ausgeführt werden kann. +, −, × und ÷ sind Beispiele für Funktionen von zwei Variablen.

Abschluß der Zifferneingabe. Wollen Sie zwei Zahlen für die Ausführung einer Operation *eingeben*, braucht der Rechner ein Zeichen dafür, daß die *Zifferneingabe* der ersten Zahl *abgeschlossen* ist. Dies geschieht durch die Taste ENTER. Sie trennt die zwei Eingaben. Befindet sich jedoch eine der beiden Zahlen, als ein Ergebniss einer vorangegangenen Operation bereits im Rechner, brauchen Sie die Taste ENTER nicht. Alle Funktionen, mit Ausnahme der Tasten zur Zifferneingabe selbst*, *beenden die Zifferneingabe*.

Beachten Sie, daß, unabhängig von dem eingegebenen Wert, nach Abschluß der Zifferneingabe (z.B. durch ENTER) immer das Dezimaltrennzeichen und die momentan gesetzte Anzahl von Dezimalstellen angezeigt wird.

Kettenrechnungen. Beachten Sie in den folgenden Berechnungen, daß

- die Taste ENTER *nur* zur Trennung *aufeinanderfolgender* Zahleneingaben verwendet wird,
- die Operationstaste *erst nach* der Eingabe beider Argumente gedrückt wird,
- das Ergebnis einer Operation als Argument in einer anderen Operation verwendet werden kann. Diese Zwischenergebnisse werden automatisch gespeichert, und in umgekehrter Reihenfolge der Eingabe (first in/ last out Basis) wieder abgerufen. Die nach der Ausführung einer Operation eingetasteten Zahlen, werden als neue Werte behandelt.

* Die Zifferntasten ., CHS, EEX und ←.

Beispiel: Berechnen Sie (9 + 17–4) ÷ 4.

Tastenfolge	**Anzeige**	
9 [ENTER]	**9.0000**	Abschluß der Zahleneingabe.
17 [+]	**26.0000**	(9 + 17).
4 [−]	**22.0000**	(9 + 17–4).
4 [÷]	**5.5000**	(9 + 17–4) ÷ 4.

Auch schwierigere Probleme können in der selben Weise berechnet werden – d.h. durch automatische Speicherung und Rückruf der Zwischenergebnisse. Am einfachsten ist es die Berechnung mit der innersten Klammer zu beginnen und sich nach außen vorzuarbeiten, genau wie Sie es auf dem Papier auch tun würden.

Beispiel: Berechnen Sie (6 + 7) × (9–3).

Tastenfolgen	**Anzeige**	
6 [ENTER]	**6.0000**	Berechnen Sie zuerst das Zwischenergebnis (6 + 7).
7 [+]	**13.0000**	
9 [ENTER]	**9.0000**	Berechnen Sie dann das Zwischenergebnis (9–3).
3 [−]	**6.0000**	
[×]	**78.0000**	Schließlich multiplizieren Sie die Zwischenergebnisse (13 und 6) miteinander, um das Endergebnis zu bekommen.

Versuchen Sie sich an den folgenden Problemen. Jedesmal, wenn Sie während der Berechnung die Taste [ENTER] drücken, sichern Sie sich die vorangegangene Zahl für die nächsten Operationen.

$$(16 \times 38) - (13 \times 11) = 465.0000$$

$$4 \times (17 - 12) \div (10 - 5) = 4.0000$$

$$23^2 - (13 \times 9) + 1/7 = 412.1429$$

$$\sqrt{[(5.4 \times 0.8) \div (12.5 - 0.7^2)]} = 0.5998$$

Abschnitt 2

Numerische Funktionen

Dieser Abschnitt behandelt die numerischen Funktionen des HP-15C (nicht eingeschlossen sind hier die statistischen Funktionen und die höheren Funktionen des Rechners). Die nichtnumerischen Funktionen werden getrennt diskutiert (Zahleneingabe in Abschnitt 1, Stackmanipulationen in Abschnitt 3 und Steuerung der Anzeige Abschnitt 5).

Die numerischen Funktionen des HP-15C werden in der selben Art und Weise verwendet, gleichgültig ob sie manuell oder in einem Programm ausgeführt werden. Einige der Funktionen (wie z.B. [ABS]) sind sogar in erster Linie bei der Programmierung von Interesse.

Erinnern Sie sich daran, daß alle numerischen Funktionen ebenso wie alle anderen Funktionen (mit Ausnahme der Zifferneingabetasten), automatisch eine Zahleneingabe abschließen. Das bedeutet, daß alle numerischen Funktionen kein [ENTER] vorausgehen oder nachfolgen muß.

Pi

Nach dem Drücken von [g] [π] werden die ersten zehn Ziffern der Kreiskonstanten in den Rechner übernommen. [π] braucht von anderen Zahlen nicht durch [ENTER] getrennt zu werden.

Funktionen zur Zahlenmanipulation

Die Funktionen der Zahlenmanipulation wirken auf die Zahl, die sich augenblicklich in der Anzeige (dem X-Register) befindet.

Ganzzahliger Anteil. Durch Drücken von [g] [INT] wird die Zahl in der Anzeige durch die ihr am nächsten liegende ganze Zahl (gleicher oder geringerer Wertigkeit) ersetzt.

Gebrochener Anteil. Durch Drücken von [f] [FRAC] wird die Zahl in der Anzeige durch ihren dezimalen Anteil ersetzt, d.h. alle Ziffern links des Dezimaltrennzeichens werden durch Null ersetzt.

Rundung. Durch Drücken von [g] [RND] wird die intern gespeicherte Mantisse der momentan angezeigten Zahl auf die Anzahl von Stellen gerundet, die durch das aktuelle [FIX], [SCI] oder [ENG] Anzeigeformat festgelegt ist.

Absolutwert. Durch Drücken von [g] [ABS] wird die Zahl in der Anzeige durch ihren Absolutwert ersetzt.

Tastenfolge	**Anzeige**	
123.4567 [g] [INT]	**123.0000**	
[g] [LSTx] [CHS] [g] [INT]	**–123.0000**	Verändert das Vorzeichen aber nicht die Ziffern.
[g] [LSTx] [f] [FRAC]	**–0.4567**	
1.23456789 [CHS]	**–1.2346**	
[g] [RND]		
[f] CLEAR [PREFIX]	**1234600000**	Kurzzeitiges Anzeigen aller Ziffern der Mantisse.
(loslassen)	**–1.2346**	
[g] [ABS]	**1.2346**	

Funktionen einer Variablen

Die mathematischen Funktionen einer Variablen auf die Zahl in der Anzeige (X-Register).

Allgemeine Funktionen

Reziprokwert. Durch Drücken von [1/x] wird der Reziprokwert der Zahl in der Anzeige berechnet.

Fakultät und Gammafunktion. Durch Drücken von [f] [x!] wird die Fakultät des angezeigten Wertes berechnet, sofern dies eine ganze Zahl x ist, die die Bedingung $0 \leqslant x \leqslant 69$ erfüllt.

Die Taste [x!] kann auch zur Berechnung der Gammafunktion, $\Gamma(x)$, die in der höheren Mathematik und der Statistik verwendet wird, benutzt werden. Durch Drücken von [x!] wird $\Gamma(x+1)$ berechnet. Um den Funktionswert eines bestimmten Arguments zu berechnen, ist von dem Argument der Wert eins zu subtrahieren. Bei der Gammafunktion ist x nicht auf positive ganze Zahlen beschränkt.

Quadratwurzel. Durch Drücken von [g] [x^2] wird das Quadrat der Zahl in der Anzeige berechnet.

Tastenfolge	**Anzeige**	
25 [1/x]	**0.0400**	
8 [f] [x!]	**40,320.0000**	Berechnung von 8! oder Γ (9).
3.9 [$\sqrt{x}$]	**1.9748**	
12.3 [g] [x^2]	**151.2900**	

Trigonometrische Operationen

Trigonometrische Modi. Die trigonometrischen Funktionen arbeiten in einem von Ihnen zu wählenden Modus. Durch die Wahl eines bestimmten Modus werden bereits sich im Rechner befindliche Zahlen nicht in den neuen Modus konvertiert. Die Spezifikation eines trigonometrischen Modus beinhaltet lediglich, daß dem Rechner mitgeteilt wird, welche Maßeinheit (Altgrad, Bogenmaß oder Neugrad) bei der Ausführung einer trigonometrischen Funktion benutzt werden soll.

Durch Drücken von [g] [DEG] wird der trigonometrische Modus «dezimale Altgrad» gewählt. Die Gradangabe erfolgt in *dezimaler* Form, nicht in der Minuten/Sekunden Angabe. Im Display erscheint keine Statusanzeige.

Durch Drücken von [g] [RAD] wird der trigonometrische Modus «Radiant» (*Bogenmaß*) gewählt (die Statusanzeige **RAD** erscheint im Display). Im Komplex-Modus setzen alle Funktionen (mit Ausnahme von [→P] und [→R]) voraus, daß alle Werte im Bogenmaß angegeben sind, ungeachtet, ob die Statusanzeige **RAD** im Display erscheint oder nicht.

Durch Drücken von [g] [GRD] wird der trigonometrische Modus «dezimale Neugrad» (360 Altgrad = 400 Neugrad) gewählt. Die Statusanzeige **GRAD** erscheint in der Anzeige.

Der Permanentspeicher erhält den zuletzt gewählten trigonometrischen Modus. Im Einschaltzustand (nach dem allerersten Einschalten oder einem Löschen des Permanentspeichers) befindet sich der Rechner im [DEG] Modus.

Trigonometrische Funktionen. Gegeben sei der Wert x in der Anzeige (X-Register) des Rechners.

Tastenfolge	Funktion
[SIN]	Sinus x
[g] [SIN⁻¹]	Arcussinus x
[COS]	Cosinus x
[g] [COS⁻¹]	Arcuscosinus x
[TAN]	Tangens x
[g] [TAN⁻¹]	Arcustangens x

Vergewissern Sie sich vor der Ausführung einer trigonometrischen Funktion, daß der Rechner auf den gewünschten Modus eingestellt ist.

Zeit- und Winkelkonvertierung

Zahlen, die eine Zeit (Stunden) oder einen Winkel (Grad) darstellen, können vom HP-15C von einem Dezimal- in ein Minuten/Sekunden Format konvertiert werden:

Stunden.Dezimalstunden (H.h) ⟷	Stunden.Minuten Sekunden Dezimalsekunden (H.MMSSss)
Grad.Dezimalgrad (D.d) ⟷	Grad.Minuten Sekunden Dezimalsekunden (D.MMSSss)

Stunden (oder Grad)/Minuten/Sekunden Konvertierung. Durch Drücken von f →H.MS wird die Zahl in der Anzeige vom Dezimalstunden (oder Dezimalgrad-) Format in ein Stunden (oder Grad)/Minuten/Sekunden Format konvertiert.

Drücken Sie beispielsweise f →H.MS, um

$\underbrace{1.2\,3\,4\,5}_{\text{Stunden}}$ in $\underbrace{1}_{\text{Stunden}}\,.\,\underbrace{1\,4}_{\text{Minuten}}\,\underbrace{0\,4}_{\text{Sekunden}}$

umzuwandeln. Durch Drücken von f PREFIX werden alle Dezimalstellen angezeigt.

$1\,1\,4\,0\,4\,2\,0\,0\,0\,\underbrace{0}_{\text{bis zu dem hunderttausendsten Teil einer Sekunde.}}$

Dezimalstunden- (oder Dezimalgrad) Konvertierung. Durch Drücken von g →H wird die Zahl im angezeigten X-Register aus einem Stunden (oder Grad)/Minuten/Sekunden/Dezimalsekunden-Format in ein Dezimalstunden (oder Dezimalgrad) Format konvertiert.

Altgrad/Bogenmaß (Radiant) Konvertierung

Die Funktionen →DEG und →RAD dienen zur Konvertierung von Winkeln aus Altgrad in Radiant und umgekehrt (D.d ⟷ R.r). Die Gradangabe muß im Dezimal- und nicht im Minuten/Sekunden Format erfolgen.

Tastenfolge	Anzeige	
40.5 f →RAD	**0.7069**	Radiant.
g →DEG	**40.5000**	40.5 Grad (als Dezimalzahl).

Logarithmische Funktionen

Natürlicher Logarithmus. Durch das Drücken von [g] [LN] wird der natürliche Logarithmus der Zahl in der Anzeige berechnet; d.h. der Logarithmus zur Basis *e*.

Natürliche Exponentialfunktion. Durch Drücken von [e^x] wird der Wert der *e*-Funktion mit der Zahl in der Anzeige als Argument berechnet: d.h. es wird *e* hoch dem Inhalt der Anzeige gebildet.

Dekadischer Logarithmus. Durch Drücken von [g] [LOG] wird der dekadische Logarithmus der Zahl in der Anzeige berechnet, d.h. der Logarithmus zur Basis 10 des Wertes in der Anzeige.

Dekadische Exponentialfunktion. Durch Drücken von [10^x] wird der Wert der dekadischen Exponentialfunktion mit der Zahl in der Anzeige als Argument berechnet; d.h. die Zahl 10 wird mit dem Inhalt der Anzeige potenziert.

Tastenfolge	Anzeige	
45 [g] [LN]	**3.8067**	Natürlicher Logarithmus von 45.
3.4012 [e^x]	**30.0001**	Natürliche Exponentialfunktion von 3.4012.
12.4578 [g] [LOG]	**1.0954**	Dekadischer Logarithmus von 12.4578.
3.1354 [10^x]	**1,365.8405**	Dekadische Exponentialfunktion von 3.1354.

Hyperbolische Funktionen

Der HP-15C verfügt über die folgenden hyperbolischen Funktionen:

Tastenfolge	Funktion
[f] [HYP] [SIN]	Sinushyperbolicus
[g] [HYP^{-1}] [SIN]	Inverser Sinushyperbolicus
[f] [HYP] [COS]	Cosinushyperbolicus
[g] [HYP^{-1}] [COS]	Inverser Cosinushyperbolicus
[f] [HYP] [TAN]	Tangenshyperbolicus
[g] [HYP^{-1}] [TAN]	Inverser Tangenshyperbolicus

Funktionen zweier Variablen

Die mathematischen Funktionen zweier Variablen des HP-15C benutzen zur Berechnung eines Resultates die beiden zuletzt eingegebenen Werte. Wenn Sie die beiden Zahlen eingeben, beachten Sie, daß die Werte durch [ENTER] – oder jede andere, die Ziffereingabe abschließende Funktion (z.B. [g] [INT] oder [1/x]) – getrennt werden müssen.

Der zuerst eingegebene Wert wird als y-Wert bezeichnet, da er in das Y-Register geladen wird. Der als zweites eingegebene Wert befindet sich in der Anzeige, dem X-Register, und wird daher x-Wert genannt.

Die vier arithmetischen Funktionen [+], [-], [×] und [÷] sind die elementaren Beispiele für Funktionen zweier Variablen. Weitere Funktionen werden im folgenden aufgeführt.

Potenzrechnung

Durch Drücken von [y^x] wird der y-Wert mit dem x-Wert potenziert. Die Basis y ist vor dem Exponenten einzugeben.

Zu berechnen	Tastenfolge	Anzeige
$2^{1.4}$	2 [ENTER] 1.4 [y^x]	**2.6390**
$2^{-1.4}$	2 [ENTER] 1.4 [CHS] [y^x]	**0.3789**
$(-2)^3$	2 [CHS] [ENTER] 3 [y^x]	**-8.0000**
$\sqrt[3]{2}$ or $2^{1/3}$	2 [ENTER] 3 [1/x] [y^x]	**1.2599**

Prozentrechnung

Die Prozentfunktionen [%] und [Δ%] erhalten den Wert der spezifizierten Basiszahl im Y-Register. Wie in den folgenden Beispielen gezeigt wird, erlaubt Ihnen dies aufeinanderfolgende Berechnungen mit derselben Basiszahl schnell auszuführen, da die Basiszahl nicht immer wieder neu eingegeben zu werden braucht.

Prozentualer Anteil. Die Funktion [%] berechnet einen bestimmten prozentualen Anteil einer vorgegebenen Zahl (der Basiszahl).

Lassen Sie uns als Beispiel die dreizehnprozentige Mehrwertsteuer und den Endpreis eines Artikels mit einem Nettopreis von 15.76 DM berechnen:

Tastenfolge	Anzeige	
15.76 [ENTER]	**15.7600**	Eingabe der Basiszahl (der Nettopreis).
3 [g] [%]	**2.0488**	Berechnung der 13% von 15.76.
[+]	**17.8088**	Endpreis des Artikels.

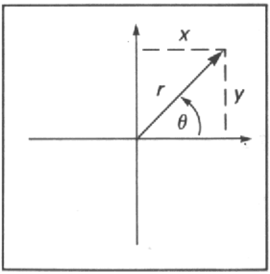
Prozentuale Differenz. Die Funktion [Δ%] berechnet die prozentuale Differenz zweier Zahlen. Das Ergebnis stellt den relativen Zuwachs (eine positive Zahl) oder die relative Abnahme (eine negative Zahl), der als zweites eingegebenen Zahl im Vergleich zur anderen dar.

Nehmen wir als Beispiel an, daß der Artikel zu 15.76 DM im letzten Jahr nur 14.12 DM gekostet hat. Was ist der relative Größenunderschied des letztjährigen Preises, zu dem in diesem Jahr?

Tastenfolge	Anzeige	
15.76 [ENTER]	**15.7600**	Diesjähriger Preis (Basiszahl).
14.12 [g] [Δ%]	**–10.4061**	Der *letztjährige* Preis war 10.41% *billiger* als der *diesjährige*.

Transformation von Polar- und Rechteckskoordinaten

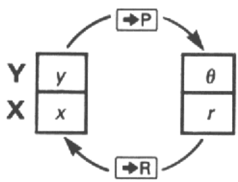
Die beiden Funktionen [→P] und [→R] dienen zur Umwandlung von Polar- in Rechteckskoordinaten und umgekehrt. Der Winkel θ kann in Abhängigkeit von dem jeweiligen trigonometrischen Modus in dezimalen Altgrad, Radiant oder Neugrad angegeben werden. θ wird wie in der Abbildung rechts gemessen.



Transformation in Polarkoordinaten. Durch Drücken von [g] [→P] (*polar*) werden die Rechteckskoordinaten (x,y) in die Polarkoordinaten (Betrag r, Winkel θ) umgewandelt.

Zur Ausführung der Funktion muß zuerst der *y*- und danach der *x*-Wert eingegeben werden. Nach der Ausführung von g →P erscheint *r* in der Anzeige. Durch Drücken von $x \gtrless y$ (*x exchange y*) wird der *y*-Wert aus dem Y-Register in die Anzeige (X-Register) übernommen. θ wird als ein Wert zwischen –180° und 180° Altgrad, zwischen –π und πRadiant oder –200° und 200° Neugrad angegeben.

Transformation in Rechteckskoordinaten. Durch Drücken von g →R (*rectangular*) werden gegebene Polarkoordinaten (Betrag *r* und Winkel θ) in Rechteckskoordinaten umgewandelt. Der Winkel θ muß zuerst eingegeben werden, und danach der Betrag *r*. Nach dem Ausführen von g →R erscheint der *x*-Wert in der Anzeige; durch Drücken von $x \gtrless y$ wird der *y*-Wert angezeigt.



Tastenfolge	Anzeige	
g DEG		Rechner wird in den Altgrad Modus geschaltet.
5 ENTER	**5.0000**	*y*-Wert.
10	**10**	*x*-Wert.
g →P	**11.1803**	*r*.
$x \gtrless y$	**26.5651**	θ; Umwandlung von Rechteckskoordinaten in Polarkoordinaten.
30 ENTER	**30.0000**	θ.
12	**12**	*r*.
f →R	**10.3923**	*x*-Wert.
$x \gtrless y$	**6.0000**	*y*-Wert. Umwandlung von Polarkoordinaten in Rechteckskoordinaten.

Abschnitt 3

Automatischer Speicherstack, LAST X und Datenspeicherung

Automatischer Speicherstack und Stackmanipulation

Die HP Operationslogik basiert auf einem mathematischen System, das unter dem Namen «Polnische Notation» bekannt ist, und von dem polnischen Logiker Jan Lukasiewicz (1878-1956) entwickelt wurde. Die herkömmliche mathematische Notation setzt die algebraische Operatoren *zwischen* die relevanten Zahlen oder Variablen, wenn ein algebraischer Ausdruck berechnet werden soll. Die Lukasiewicz Notation stellt die Operatoren *vor* die Variablen. Für die optimale Ausnutzung beim Gebrauch eines Taschenrechners, verwendet HP eine Konvention, bei der das Ausführen (Eingabe) der Operatoren nach der Angabe (Eingabe) der Variablen erfolgt. Daher der Name «Umgekehrte Polnische Notation» (*Reverse Polish Notation*) oder UPN.

Der HP-15C verwendet UPN, um komplizierte Berechnungen übersichtlich, d.h. ohne Verwendung von Klammern und Zeichensetzung zu lösen. Dabei werden Zwischenergebnisse automatisch gespeichert und wieder abgerufen. Dieses System funktioniert durch den automatischen Speicherstack und die Taste [ENTER], die zusammen die Anzahl der benötigten Tastenfolgen auf ein Mindestmaß reduzieren.

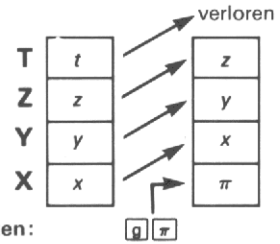
Register des automatischen Speicherstacks

T	0.0000	
Z	0.0000	
Y	0.0000	
X	0.0000	Wird immer angezeigt.

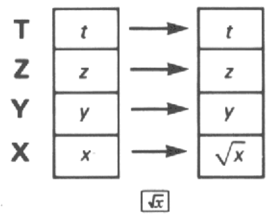
Wenn sich der HP-15C im Run-Modus befindet (die Statusanzeige **PRGM** ist aus), wird im Display der Wert des X-Registers angezeigt.

Jede neu eingegebene Zahl und jedes Ergebnis der Ausführung einer numerischen Funktion wird im angezeigten X-Register abgelegt. Durch die Ausführung einer Funktion oder die Eingabe einer Zahl werden die sich im Stack befindlichen Zahlen nach oben (Stack Lift) oder nach unten (Stack Drop) verschoben oder verbleiben in dem ursprünglichen Register. Welche Veränderung im Stack stattfindet, hängt von der unmittelbar zuvor oder momentan ausgeführten Funktion ab. Die Zahlen im Stack sind auf der Basis «last-in, first-out» (d.h. in umgekehrter Reihenfolge der Eingabe) verfügbar. Die folgenden Stackillustrationen verdeutlichen die drei Arten der Stackveränderung. Nehmen Sie an, daß *x, y, z* und *t* Zahlen darstellen, die sich im Stack befinden.

Stack Lift

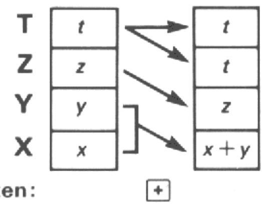


Kein Stack Lift oder Stack Drop



Tasten:

Stack Drop



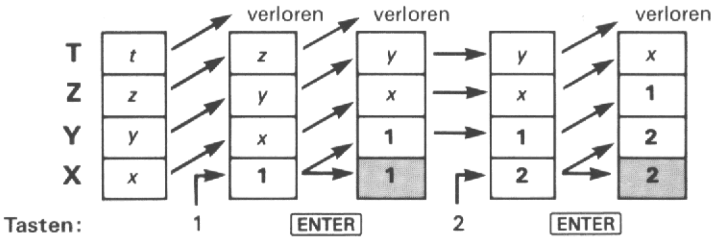
Tasten:

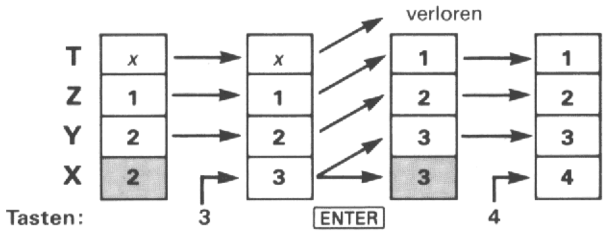
Beachten Sie, daß die Zahl im T-Register dort verbleibt, wenn sich der Stack nach unten verschiebt. Dies erlaubt die mehrfache Verwendung dieser Zahl als arithmetische Konstante.

Funktionen zur Stackmanipulation

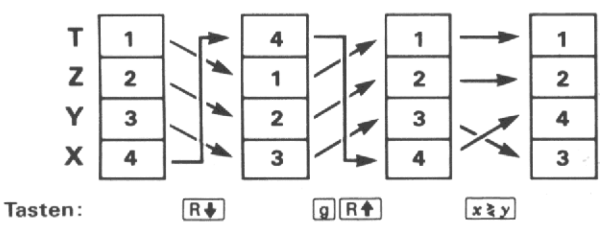
ENTER. Die Taste ENTER trennt zwei nacheinander eingegebene Zahlen. Dies geschieht durch Anheben des Stackinhalts und Kopieren der Zahl im angezeigten X-Register in das Y-Register. Die nachfolgends eingegebene Zahl überschreibt dann die Zahl im X-Register, ohne daß dabei der Stack angehoben wird. Der Stack soll beispielsweise mit den Zahlen 1, 2, 3, 4 geladen werden.

(Die schattierten Felder deuten dabei an, daß der vorige Inhalt überschrieben wird.)



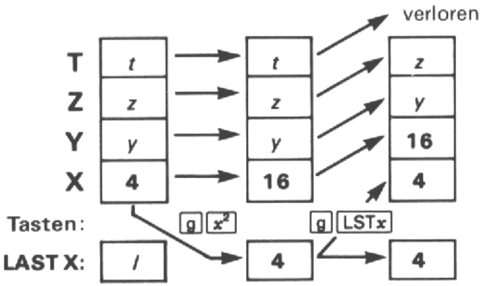


R↓ (***roll down***), R↑ (***roll up***) und $x \gtrless y$ (***X exchange Y***). Die Tasten R↓ und R↑ verschieben zyklisch den Inhalt des Stackregisters um ein Register nach oben oder nach unten. Dabei gehen keine Werte verloren. $x \gtrless y$ vertauscht die Zahlen im X- und Y-Register. Wenn der Stack mit den Zahlen 1, 2, 3, 4 geladen ist, hat das Drücken von R↓ und R↑ die folgende Wirkung:



Das LAST X-Register und die Taste LSTx

Das LAST X-Register ist ein separates Speicherregister zum Retten des Werts, der *vor der Ausführung* einer numerischen Funktion in der Anzeige war*. Durch Drücken von g LSTx (*LAST X*) wird eine Kopie des momentanen LAST X-Registerinhaltes in die Anzeige übernommen. Zum Beispiel:



Die Taste LSTx erlaubt Ihnen die mehrfache Verwendung einer Zahl, ohne sie jedesmal neu eingeben zu müssen (siehe Abschnitt «Arithmetische Berechnungen mit Konstanten», Seite 39). Ebenso wird die Korrektur einer irrtümlicherweise ausgeführten Funktion oder einer falschen Zahleneingabe vereinfacht.

Nehmen wir an, daß Sie in einer Kettenrechnung den falschen Teiler eingegeben haben:

Tastenfolge	Anzeige	
287 ENTER	287.0000	
12.9 ÷	22.2481	Der falsche Teiler!
g LSTx	12.9000	Aus dem LAST X-Register wird der letzte Wert des X-Registers (der falsche Teiler) vor der Ausführung von ÷ zurückgerufen.

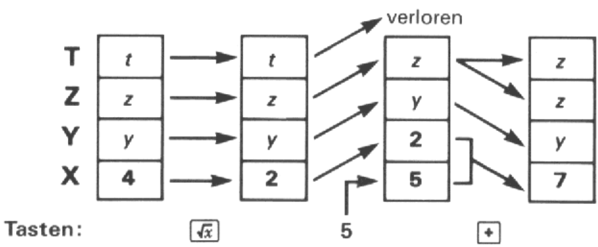
* Dies ist nicht der Fall bei den Operationen $\bar{x}$, s und L.R., die nicht den Inhalt des X-Registers verwenden bzw. erhalten. Diese Funktionen führen Berechnungen mit den Daten in den Statistikregistern (R_2 bis R_7) durch. Eine Liste mit allen Funktionen, die x in das LAST X-Register retten, finden Sie im Anhang B.

Tastenfolge	Anzeige	
[×]	287.0000	Umkehren der Funktion, die das falsche Ergebnis lieferte.
13.9 [÷]	20.6475	Das korrekte Ergebnis.

Rechnerfunktionen und Stack

Wenn Sie zwei Zahlen nacheinander eingeben wollen, müssen Sie zur Trennung der beiden Zahlen [ENTER] drücken. Wenn Sie jedoch nur eine Zahl, unmittelbar von einer Funktion (einschließlich Manipulationen wie [R↓]) gefolgt, eingeben wollen, brauchen Sie [ENTER] nicht einzufügen. Woran liegt das? Die Ausführung der meisten Funktionen des HP-15C hat die folgende zusätzliche Wirkung:

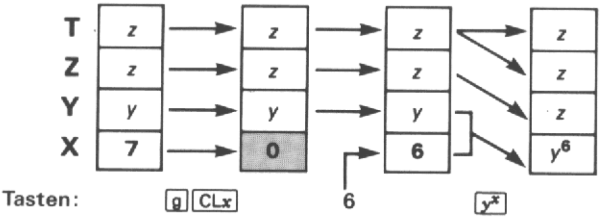
- Der automatische Speicherstack wird für einen Stack Lift freigegeben, d.h. der Stack wird automatisch angehoben, wenn die *nächste* Zahl eingegeben oder in die Anzeige zurückgerufen wird.
- Die Zahleneingabe wird beendet, d.h. daß die nächste Zahl zu einer neuen Eingabe gehört.



Es gibt vier Funktionen – [ENTER], [CLx], [Σ+] und [Σ-] –, nach deren Ausführung, der Stack *gesperrt* ist*. Bei diesen Operationen wird der Stack *nicht* nach oben verschoben, wenn die *nächste* Zahl eingegeben oder zurückgerufen wird. Nach der Ausführung einer dieser Funktionen überschreibt die neu eingegebene Zahl den Inhalt der Anzeige, anstatt den Stack anzuheben. (Beachten Sie, daß der Stack direkt *nach* dem Drücken von [ENTER] angehoben wird, und nicht bei der Eingabe der *nächsten* Zahl.)

* Ebenso sperrt die Taste [←] den Stack Lift, *wenn die Zahleneingabe bereits abgeschlossen ist.* [←] löscht in diesem Fall die gesamte Anzeige, wie die Taste [CLx]. Ansonsten ist [←] neutral. Eine weitere Diskussion des Stacks finden Sie in Anhang B.

In den meisten Anwendungen sind die hier genannten Aspekte so natürlich in den Rechenprozeß eingegliedert, daß Sie Ihnen gar nicht auffallen werden.

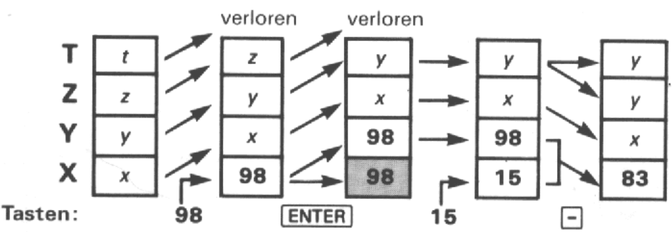


Reihenfolge der Eingabe und die Taste ENTER

Ein wichtiger Aspekt beim Rechnen mit Funktionen zweier Variablen ist die Positionierung der Zahlen im Stack. Vor der Ausführung einer arithmetischen Funktion sollten die Zahlen so ausgerichtet sein, wie man sie auch auf dem Papier anordnen würde. Zum Beispiel:

$$\begin{array}{r} 98 \\ \underline{-15} \end{array} \qquad \begin{array}{r} 98 \\ \underline{+15} \end{array} \qquad \begin{array}{r} 98 \\ \underline{\times 15} \end{array} \qquad \frac{98}{15}$$

Die erste (oben stehende) Zahl würde sich im Y-Register, die zweite (darunter stehende) Zahl im X-Register befinden. Beim Ausführen der mathematischen Operation wird der Stack nach unten verschoben und das Resultat im X-Register abgelegt. Das folgende Diagramm zeigt den Ablauf einer Subtraktion im Rechner:



Die gleiche Zahlenanordnung würde verwendet werden, um 15 zu 98 zu addieren, 98 mit 15 zu multiplizieren oder 98 durch 15 zu dividieren.

Verschachtelte Berechnungen

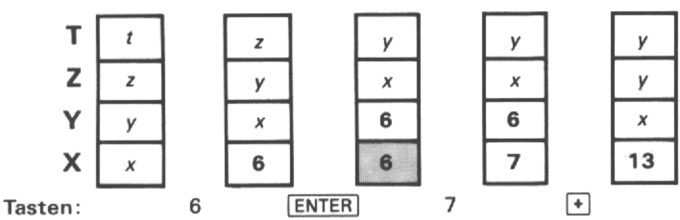
Der automatische Stack Lift und Stack Drop ermöglicht das Lösen komplizierterer und ineinandergeschachtelter Berechnungen ohne Klammerung und Abspeicherung von Zwischenergebnissen. Die verschachtelte Berechnung wird dabei einfach in eine Folge von Operationen einer und zweier Variablen umgesetzt.

Die vier Stackregister genügen für beinahe jede komplizierte Berechnung, die Ihnen begegnen wird. In der Regel ist es am sinnvollsten, die Berechnung mit der Zahl oder Klammer zu beginnen, die sich am weitesten «innen» befindet, und sich dann nach «außen» vorzuarbeiten (genauso wie Sie auf dem Papier diese Rechnung auch ausführen würden). Wenn Sie sich nicht an diese Vorgehensweise halten, kann es nötig sein, Zwischenergebnisse in einem Speicherregister abzulegen. Lassen Sie uns folgendes Beispiel betrachten:

$$3\,[4+5\,(6+7)] \quad :$$

Tastenfolge	**Anzeige**	
6 ENTER 7 +	**13.0000**	Zwischenergebnis von (6 + 7).
5 ×	**65.0000**	Zwischenergebnis von 5 (6 + 7).
4 +	**69.0000**	Zwischenergebnis von [4 + 5 (6 + 7)].
3 ×	**207.0000**	Endergebnis 3 [4 + 5 (6 + 7)].

Die folgende Abbildung illustriert die Stackumordnung in diesem Beispiel. Der Stack verschiebt sich nach der Ausführung einer Funktion zweier Variablen automatisch nach unten und wird angehoben, wenn eine neue Zahl eingetastet wird. (Zur Vereinfachung werden wir von nun an in diesem Handbuch keine Pfeile mehr zwischen den Stackabbildungen einzeichnen.)



T	*y*	*y*	*y*	*y*
Z	*y*	*x*	*y*	*x*
Y	*x*	**13**	*x*	**65**
X	**13**	**5**	**65**	**4**

Tasten: 5 [×] 4

T	*y*	*y*	*y*	*y*
Z	*x*	*y*	*x*	*y*
Y	**65**	*x*	**69**	*x*
X	**4**	**69**	**3**	**207**

Tasten: [+] 3 [×]

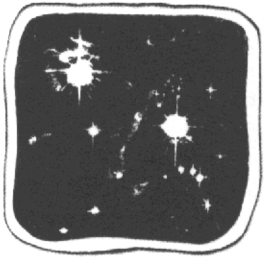
Arithmetische Berechnungen mit Konstanten

Sie können (ohne die Verwendung eines Speicherregisters) den Speicherstack auf drei Arten so manipulieren, daß Berechnungen mit einer Konstanten wiederholt durchgeführt werden können:

1. Verwenden Sie das LAST X Register.
2. Laden Sie den Stack mit einer Konstanten und geben Sie dann für jede arithmetische Operation den zweiten benötigten Operanden einzeln ein. (Löschen Sie jedesmal die Anzeige [das X-Register], wenn Sie die Zahl, die Sie zusammen mit der Konstanten in der arithmetischen Operation benutzen wollen, verändern wollen.)
3. Laden Sie den Stack mit einer Konstanten und führen Sie die arithmetische Operation unter Verwendung der Konstanten und der Zahl im X-Register durch (d.h. Sie geben keine neue Zahl in das X-Register ein und benutzen stattdessen das X-Register als *Akkumulator*).

LAST X. Verwenden Sie Ihre Konstante im X-Register (d.h. geben Sie diese als zweites ein), so daß sie immer in das LAST X-Register gerettet wird. Durch Drücken von [g] [LSTx] wird die Konstante in das X-Register (die Anzeige) zurückgeladen. Dieser Vorgang kann beliebig oft wiederholt werden.

Beispiel: Zwei Nachbarsterne der Erde sind Rigil Centaurus (4.3 Lichtjahre entfernt) und Sirius (8.7 Lichtjahre entfernt). Berechnen Sie unter der Verwendung der Lichtgeschwindigkeit, *c* (3.0×10^8 Meter/Sekunde oder 9.5×10^{15} Meter/Jahr), die Entfernung zu diesen Sternen in Metern. (Das folgende Stackdiagramm zeigt nur eine Dezimalstelle.)



T	*t*	*z*	*y*	*y*
Z	*z*	*y*	*x*	*x*
Y	*y*	*x*	**4.3**	**4.3**
X	*x*	**4.3**	**4.3**	**9.5 15**

Tasten: 4.3 ENTER 9.5 EEX 15

LAST X:	*l*	*l*	*l*	*l*

T	*y*	*y*	*y*	*x*
Z	*x*	*y*	*x*	**4.1 16**
Y	**4.3**	*x*	**4.1 16**	**8.7**
X	**9.5 15**	**4.1 16**	**8.7**	**9.5 15**

Tasten: × 8.7 g LSTx

LAST X:	*l*	**9.5 15**	**9.5 15**	**9.5 15**

T	*x*	*x*
Z	**4.1 16**	*x*
Y	**8.7**	**4.1 16**
X	**9.5 15**	**8.3 16**

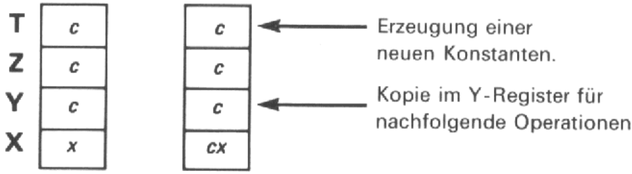
(Entfernung zu Rigil Centaurus beträgt 4.1×10^{16} Meter.)

(Entfernung zu Sirius beträgt 8.3×10^{16} Meter.)

Tasten: ×

LAST X:	**9.5 15**	**9.5 15**

Laden des Stacks mit einer Konstanten. Da der Inhalt des T-Registers bei einem Stack Drop reproduziert wird, kann diese Zahl als eine Konstante bei arithmetischen Operationen verwendet werden.



Tasten: [×]

Der Stack wird mit einer Konstanten geladen, indem Sie diese in die Anzeige tasten und dreimal die Taste [ENTER] drücken. Geben Sie dann das erste Argument ein und führen Sie die Operation aus. Der Stack verschiebt sich nach unten, eine Kopie der Konstanten «fällt» in das Y-Register, und eine neue Kopie der Konstanten wird im T-Register erzeugt.

Wenn wie im vorangegangenen Beispiel sich die Variablen ändern, müssen Sie die Anzeige vor der Eingabe einer neuen Variablen löschen. Dies sperrt den Stack, so daß nur das arithmetische Ergebnis überschrieben wird, und der Rest des Stacks mit der Konstanten belegt bleibt.

Wenn Sie *keine* verschiedenen Argumente haben (d.h. die Operation wird mit einem *kumulierenden* Wert ausgeführt), brauchen Sie die Anzeige *nicht* vor jeder Ausführung der Operation zu löschen, sondern Sie wiederholen einfach die arithmetische Funktion so oft wie nötig.

Beispiel: Ein Bakteriologe testet eine Kultur von Mikroorganismen, deren Population im Durchschnitt jeden Tag um 15% wächst (d.h die Wachstumsrate beträgt 1.15). Wie groß ist die Bakterienpopulation am Ende eines jeden von fünf aufeinanderfolgenden Tagen, wenn die Kultur mit 1.000 angesetzt wird.

Tastenfolge	Anzeige	
1.15	**1.15**	Wachstumsrate.
[ENTER] [ENTER]		
[ENTER]	**1.15000**	Auffüllen des Stacks.
1000	**1,0000**	Anfängliche Größe der Kultur.
[×]	**1,150.0000**	Population am Ende des ersten Tages.

Tastenfolge	Anzeige	
[×]	**1,322.5000**	Zweiter Tag.
[×]	**1,520.8750**	Dritter Tag.
[×]	**1,749.0063**	Vierter Tag.

Speicherregister-Operationen

Das Speichern oder Zurückrufen von Zahlen sind Operationen, bei denen der Inhalt der Anzeige (X-Register) in ein Datenspeicherregister geladen wird, bzw. der Inhalt eines Speicher-Registers in das X-Register übertragen wird. Nach dem allerersten Einschalten oder einem Löschen des Permanentspeichers hat der HP-15C 21 direkt adressierbare Speicherregister: R_0 bis R_9 ,$R_{.0}$ bis $R_{.9}$ und das Indexregister (R_I) (beachten Sie das Registerdiagramm auf der Innenseite des Rückumschlags). Sechs Register, R_2 bis R_7, werden auch für statistische Berechnungen verwendet.

Die Anzahl der verfügbaren Datenspeicherregister kann erhöht oder erniedrigt werden. Die dazu verwendete Funktion [DIM] wird in Anhang C, «Speicheraufteilung», erläutert. Die Register mit den niedrigsten Adressen werden zuletzt umgewandelt, so daß es sinnvoll ist, Daten immer zuerst in diese Register zu speichern.

Speichern und Zurückrufen von Zahlen

[STO] (*store*). Wenn auf diese Funktion eine Speicherregisteradresse folgt (0 bis 9, .0 bis .9*), wird die Zahl der Anzeige in das angegebene Speicherregister kopiert. Dabei wird der alte Inhalt des Registers überschrieben.

[RCL] (*recall*). Auf die gleiche Weise können Sie Daten eines bestimmten Speicherregisters zurückrufen, indem Sie [RCL] und die gewünschte Registeradresse drücken. Dies bringt eine *Kopie* der gewünschten Daten in die Anzeige; der Inhalt des Registers bleibt jedoch unverändert.

[x≷] (*X exchange Y*). Gefolgt von 0 bis .9*, tauscht diese Funktion den Inhalt des X-Registers und des adressierten Datenspeicherregisters aus. Dies ist vor allem hilfreich, wenn Sie den Inhalt eines Speicherregisters kontrollieren wollen, ohne den Stack zu verändern.

* Alle Speicherregisteroperationen können auch mit dem Indexregister ausgeführt werden (durch [I] oder [(i)]); dies wird in Abschnitt 10 und in Zusammenhang mit Matrizen in Abschnitt 12 erläutert.

Die soeben erläuterten Operationen beinhalten einen Stack Lift, so daß die im X-Register verbliebene Zahl für nachfolgende Berechnungen verwendet werden kann. Adressieren Sie in einer Operation ein nicht vorhandenes Register, erscheint in der Anzeige die Meldung **Error 3**.

Beispiel: Der Frühling kommt und Sie wollen über 24 Krokusse, die Sie in Ihrem Garten gepflanzt haben, Buch führen. Speichern Sie die Anzahl der Krokusse, die am ersten Tag geblüht haben, und addieren Sie die Anzahl neuer Blüten des zweiten Tages.

Tastenfolge	**Anzeige**	
3 [STO] 0	**3.0000**	Speichern der Anzahl der Blüten des ersten Tages.

Nehmen wir an, Sie schalten Ihren Rechner aus und am nächsten Tag erst wieder an.

[RCL] 0	**3.0000**	Rückruf der Zahl von Krokussen, die am Vortag geblüht haben.
5 [+]	**8.0000**	Addition hinzugekommener Krokusblüten, um die Gesamtzahl von Krokusblüten zu berechnen.

Löschen von Datenspeicherregistern

Durch Drücken von [f] CLEAR [REG] (*clear register*) wird der Inhalt *aller* Datenspeicherregister durch Null ersetzt. (Die Inhalte der Stackregister und des LAST X Registers bleiben unverändert.) Um ein einzelnes Datenspeicherregister zu löschen, speichern Sie Null in das betreffende Register. Das Löschen des Permanentspeichers löscht alle Register und den Stack.

Speicher- und Rückruf-Arithmetik

Speicherarithmetik. Angenommen Sie wollen eine Zahl nicht nur speichern, sondern auch eine arithmetische Operation mit ihr ausführen, und das Ergebnis in dem gleichen Register ablegen. Dies können Sie direkt – ohne die Taste [RCL] zu verwenden – tun, indem Sie wie folgt verfahren:

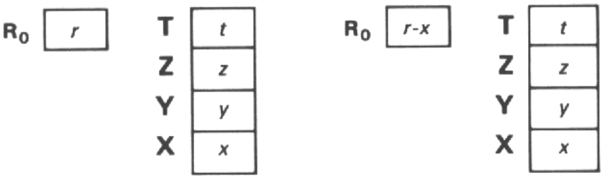
1. Laden Sie das zweite Argument (neben dem im Speicher) in die Anzeige (als ein Ergebnis einer Rechnung, eines Rückrufes oder einer Eingabe).
2. Drücken Sie [STO].

3. Drücken Sie [+], [−], [×] oder [÷].
4. Geben Sie die Registeradresse ein (0 bis 9, .0 bis .9). (Sie können auch das Indexregister verwenden – siehe Abschnitt 10.)

Die neue Zahl in dem Register wird wie folgt bestimmt:

Bei Speicherarithmetik,

$$\begin{matrix}\text{neuer}\\ \text{Registerinhalt}\end{matrix} = \begin{matrix}\text{alter}\\ \text{Registerinhalt}\end{matrix} \left\{\begin{matrix} + \\ - \\ \times \\ \div \end{matrix}\right\} \begin{matrix}\text{Zahl in der}\\ \text{Anzeige}\end{matrix}$$

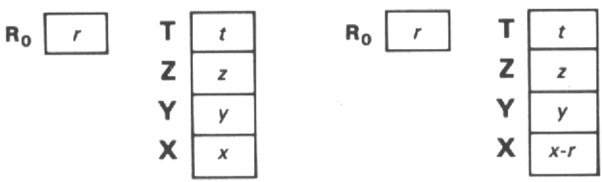


Tasten: [STO] [−] 0

Rückrufarithmetik. Diese Arithmetik erlaubt es Ihnen, arithmetische Operationen mit dem Wert in der Anzeige auszuführen, ohne den Stack anzuheben, d.h. ohne irgendeinen Wert im Y-, Z- oder T-Register zu verlieren. Die Tastenfolge ist die gleiche wie bei der Speicherarithmetik, außer daß anstelle von [STO] die Taste [RCL] verwendet wird.

Bei Rückrufarithmetik,

$$\text{neue Anzeige} = \text{alte Anzeige} \left\{\begin{matrix} + \\ - \\ \times \\ \div \end{matrix}\right\} \text{Registerinhalt}$$



Tasten: [RCL] [−] 0

Beispiel: Führen Sie Ihre laufende Zählung der neuen Krokusblühten für zwei weitere Tage fort:

Tastenfolge	Anzeige	
8 [STO] 0	**8.0000**	Speichern der Zahl der Blüten des zweiten Tages in R_0.
4 [STO] [+] 0	**4.0000**	3. Tag: Addition von vier neuen Blüten zu den schon blühenden.
3 [STO] [+] 0	**3.0000**	4. Tag: Addition von drei neuen Blüten.
24 [RCL] [−] 0	**9.0000**	Subtraktion der in R_0 aufsummierten Blüten (15) von der Gesamtzahl (24): 9 Krokusse haben noch nicht geblüht.
[RCL]	**15.0000**	(Die Zahl in R_0 verändert sich durch die letzte Operation nicht.)

Overflow und Underflow

Würde eine Speicherung oder ein Rückruf innerhalb einer arithmetischen Operation einen Overflow (*Überlauf*) in einem Datenspeicherregister verursachen, wird der Inhalt in dem betreffenden Register durch $\pm 9.999999999 \times 10^{99}$ ersetzt und die Anzeige beginnt zu blinken. Um das Blinken zu beenden (Löschen der Overflow-Bedingung), können Sie entweder [←] oder [ON] oder [g] [CF] 9 drücken.

Im Falle eines Underflow, wird die Zahl in dem betreffenden Register durch Null ersetzt (die Anzeige blinkt nicht). Overflow und Underflow werden auf Seite 61 näher erläutert.

Übungsaufgaben

1. Berechnen Sie den Wert von x in der folgenden Gleichung:

$$x = \sqrt{\frac{8.33\,(4-5.2) \div [(8.33-7.46)\,0.32]}{4.3\,(3.15-2.75)-(1.71)\,(2.01)}}$$

Ergebnis: 4.5728.

Ein möglicher Lösungsweg ist:

4 [ENTER] 5.2 [−] 8.33 [×] [g] [LSTx] 7.46 [−] 0.32 [×] [÷] 3.15 [ENTER] 2.75 [−] 4.3 [×] 1.71 [ENTER] 2.01 [×] [−] [÷] [√x]

2. Benutzen Sie die Konstantenarithmetik, um den Restrückzahlungsbetrag eines Darlehens in Höhe von 1000 DM zu ermitteln, für das bereits sechs Rückzahlungsraten in Höhe von jeweils 100 DM geleistet worden sind. Der Zinssatz (i) pro Zahlungsperiode beträgt 1% (0.01).

   Vorgehensweise: Laden Sie den Stack mit $(i + 1)$, wobei i der Zinssatz ist, und geben Sie den ursprünglichen Restrückzahlungsbetrag ein. Benutzen Sie folgende Formel, um nach jeder Rückzahlungsrate den noch ausstehenden Restrückzahlungsbetrag zu ermitteln:

   neuer Rückzahlungsbetrag =
   ((alter Rückzahlungsbetrag) × $(1 + i)$) – Rückzahlungsrate)

   Der erste Teil der Tastenfolge würde dann lauten:

   1.01 [ENTER] [ENTER] [ENTER] 1000

   Drücken Sie für jede Zahlung die Tastenfolge:

   [×] 100 [−].

   Restrückzahlung nach sechs Zahlungen: 446.32 DM.

3. Speichern Sie im Register R_5 die Zahl 100.

   a) Dividieren Sie den Inhalt von R_5 durch 25.

   b) Subtrahieren Sie 2 vom Inhalt von R_5.

   c) Multiplizieren Sie den Inhalt von R_5 mit 0.75.

   d) Addieren Sie 1.75 zum Inhalt von R_5.

   e) Rufen Sie den Inhalt von R_5 zurück.

   Ergebnis: 3.2500.

Abschnitt 4

Statistische Funktionen

Vor der Verwendung der im folgenden beschriebenen statistischen Funktionen sollten Sie die in Abschnitt 3 erläuterten Speicherstackmanipulationen verstanden haben. Sie werden erkennen, daß die Reihenfolge der Eingabe für die meisten der statistischen Berechnungen von großer Bedeutung ist.

Wahrscheinlichkeitsrechnung

Alle Eingaben für die Berechnung von Permutationen oder Kombinationen sind auf *nichtnegative ganze Zahlen* beschränkt. Geben Sie den y-Wert vor dem x-Wert ein. Diese Funktionen verursachen – wie die arithmetischen Funktionen – einen Stack Drop und das Ergebnis wird in der Anzeige (X-Register) abgelegt.

Permutationen. Durch Drücken von [f] [Px,y] wird die Anzahl aller *verschiedener Möglichkeiten*, y verschiedene Elemente zu Mengen mit x Elementen zusammenzufassen, berechnet. Jedes Element darf in einer Menge nur einmal vorkommen, und Mengen, die die gleichen Elemente in unterschiedlicher Reihenfolge enthalten, werden einzeln mitgezählt. Zur Berechnung von [Px,y] wird folgende Formel benutzt:

$$P_{y,x} = \frac{y!}{(y-x)!}$$

Kombinationen. Durch Drücken von [g] [Cy,x] wird die Anzahl *aller Möglichkeiten*, y verschiedene Elemente zu Mengen mit jeweils x Elementen ohne Beachtung der Reihenfolge zusammenzufassen, berechnet. Jedes Element darf nur einmal vorkommen, und Mengen, die die gleichen Elemente in unterschiedlicher Reihenfolge enthalten, werden *nicht* einzeln gezählt. Zur Berechnung von Kombinationen wird folgende Formel benutzt:

$$C_{y,x} = \frac{y!}{x!\,(y-x)!}$$

Beispiele: Wieviele Möglichkeiten gibt es, aus einem Satz von fünf Bildern drei an eine Wand zu hängen?

Tastenfolge	Anzeige	
5 [ENTER] 3	**3**	Nehmen Sie von fünf (y) Bildern drei (x) gleichzeitig heraus.
[f] [Py,x]	**60.0000**	Es gibt sechzig verschiedene Möglichkeiten.

Wieviel verschiedene Blätter zu vier Karten können aus einem Spiel von 52 Karten gegeben werden?

Tastenfolge	**Anzeige**	
52 ENTER 4	**4**	52 (y) Karten, gegeben werden vier (x) gleichzeitig.
g Cy,x	**270,725.0000**	Anzahl der möglichen verschiedenen Blätter.

Die Ausführung dieser Funktionen kann manchmal mehrere Sekunden in Anspruch nehmen. Während dieser Zeit erscheint die Meldung **running** in der Anzeige. Der maximale Eingabewert für x oder y ist 9,999,999,999.

Zufallszahlengenerator

Durch Drücken von f RAN# (*random number*) wird eine Zufallszahl (als Teil einer gleichverteilten Pseudo-Zufallszahlenfolge) im Bereich $0 \leqslant r < 1$* erzeugt.

Bei der ersten Inbetriebnahme des Rechners (oder nach einem Löschen des Permanentspeichers) benutzt der HP-15C die Null als Startwert zur Erzeugung der Zufallszahlenfolge. Jede neu erzeugte Zufallszahl wird automatisch als Startwert für die nächste Zufallszahlenfolge benutzt. Durch die Eingabe eines neuen Startwertes können Sie eine andere Zufallszahlenfolge beginnen. (Durch wiederholte Verwendung einer Zahl als Zufallszahlenstartwert, wird jedesmal die gleiche Zufallszahlenfolge erzeugt.)

Durch Drücken von STO f RAN# wird die Zahl im X-Register ($0 \leqslant r < 1$) als neuer Startwert für den Zufallszahlengenerator gespeichert. (Ein Wert für r außerhalb dieser Grenzen wird in einen Wert innerhalb der Grenzen konvertiert.)

Durch Drücken von RCL f RAN# wird die augenblicklich als Zufallszahlenstartwert verwendete Zahl in die Anzeige zurückgerufen.

* Die erzeugte Sequenz erfüllt den Spektraltest (D. Knuth, «*Seminumerical Algorithms*», Vol. 2, 1969).

Tastenfolge	Anzeige	
.5764	**0.5764**	Speichert 0.5764 als
[STO] [f] [RAN#]	**0.5764**	Zufallszahlenstartwert. (Das Drücken von [f] kann ausgelassen werden.)
[f] [RAN#]	**0.3422**	Zufallszahlenfolge basierend
[f] [RAN#]	**0.2809**	auf dem obigen Startwert.
[←]	**0.0000**	
[RCL] [f] [RAN#]	**0.2809**	Ruft die zuletzt erzeugte Zufallszahl zurück. (Die Taste [f] kann ausgelassen werden.)

Akkumulation von Statistiken

Der HP-15C führt statistische Operationen mit einer und zwei Variablen aus. Die Daten werden dabei zuerst in das X- und Y-Register eingegeben. Anschließend werden mittels [Σ+] automatisch die Statistiken dieser Daten berechnet und die Ergebnisse in den Speicherregistern R_2 und R_7 gespeichert. Diese Register werden deshalb als *Statistikregister* bezeichnet.

Vor der Akkumulation von Statistiken für einen neuen Datensatz sollten Sie die Statistikregister durch [f] CLEAR [Σ] löschen. (Wenn Sie die Register des Speichers neu aufgeteilt haben, und irgendein Statistikregister nicht mehr existiert, zeigt der Rechner **Error 3** an, sobald Sie versuchen CLEAR [Σ], [Σ+] oder [Σ-] auszuführen. Die Aufteilung des Speichers wird in Anhang C erläutert.)

Zur Berechnung der Statistiksummen einer Variablen, geben Sie jeden Wert (x) einzeln ein und drücken dann die Taste [Σ+].

Zur Berechnung der Statistiken zweier Variablen, geben Sie jedes Datenpaar (x- und y-Wert) wie folgt ein:

1. Geben Sie zuerst den y-Wert in das X-Register ein.
2. Drücken Sie die Taste [ENTER]. Der angezeigte y-Wert wird in das Y-Register kopiert.
3. Geben Sie den x-Wert ein.
4. Drücken Sie die Taste [Σ+]. Es wird die Anzahl der augenblicklich aufsummierten Daten angezeigt. Der x-Wert wird in das LAST X-Register gespeichert und der y-Wert verbleibt im Y-Register. [Σ+] sperrt den Stack Lift; d.h. der Stack wird nicht angehoben, wenn die nächste Zahl eingegeben wird.

Wenn Sie x- und y-Werte verwenden, die sich um einen relativ kleinen Betrag unterscheiden, kann der Rechner s, r, die lineare Regression oder $\hat{y}$ nicht berechnen und meldet stattdessen **Error 2**. Dies können Sie vermeiden, indem Sie Ihre Werte normalisieren; d.h. Sie geben nicht die Werte selbst, sondern stattdessen die Differenz der einzelnen Werte von einem ungefähren Durchschnittswert der Datenfolge, ein. Diese Zahl muß anschließend zum Resultat der jeweiligen Berechnung addiert werden. Wenn z.B. die x-Werte aus der Folge 66599, 666000, und 666001 bestehen, sollten die Daten 1,0 und 1 eingegeben werden; anschließend muß 666000 zu dem jeweiligen Ergebnis einer Berechnung addiert werden.

Der Rechner erstellt die folgenden Statistiken der eingegebene Datenfolge:

Register	Inhalt	
R_2	n	Anzahl der akkumulierten Datenelemente (Datenpaare) (n erscheint auch im angezeigten X-Register).
R_3	Σx	Summe der x-Werte.
R_4	Σx^2	Summe der Quadrate der x-Werte.
R_5	Σy	Summe der y-Werte.
R_6	Σy^2	Summe der Quadrate der y-Werte.
R_7	Σxy	Summe der Produkte der x- und y-Werte.

Sie können jede der statischen Summen in die Anzeige (X-Register) zurückrufen, indem Sie [RCL] und die Nummer des Speicherregisters mit der gewünschten Summe drücken. Entsprechend werden Σx und Σy gleichzeitig in die Anzeige kopiert, wenn Sie [RCL] [Σ+] drücken. (Die Tastenfolge [RCL] [Σ+] hebt den Stack zweifach an, wenn der Stack freigegeben ist, und nur einmal, wenn der Stack gesperrt ist, und gibt anschließend den Stack Lift frei.)

Beispiel: Der Agronom Silas Farmer hat eine neue sehr ertragreiche Reisart entwickelt. Die Ertragsrate seiner neuen Sorte wurde als Funktion der Düngung gemessen. Verwenden Sie die Taste [Σ+], um die folgenden Daten zu akkumulieren und die Werte für Σx, Σx^2, Σy, Σy^2, und Σxy, zu berechnen. Der x-Wert stellt dabei die als Dünger verwendete Stickstoffmenge dar und der y-Wert den Ertrag an Reis.

X	Verwendete Menge Stickstoff (kg pro Hektar*) x	0.00	20.00	40.00	60.00	80.00
Y	Ertrag (Tonnen pro Hektar) y	4.63	4.78	6.61	7.21	7.78
* Ein Hektar entspricht 100 Ar.						

Tastenfolge	**Anzeige**	
[f] CLEAR [Σ]	**0.0000**	Löscht die Statistikregister (R_2 bis R_7 und den Stack).
[f] [FIX] 2	**0.00**	Spezifiziert entsprechend den Daten ein Anzeigeformat mit zwei Dezimalstellen.
4.63 [ENTER]	**4.63**	
0 [Σ+]	**1.00**	Erstes Datenpaar.
4.78 [ENTER]	**4.78**	
20 [Σ+]	**2.00**	Zweites Datenpaar.
6.61 [ENTER]	**6.61**	
40 [Σ+]	**3.00**	Drittes Datenpaar.
7.21 [ENTER]	**7.21**	
60 [Σ+]	**4.00**	Viertes Datenpaar.
7.78 [ENTER]	**7.78**	
80 [Σ+]	**5.00**	Fünftes Datenpaar.
[RCL] 3	**200.00**	Summe der x-Werte, Σx (verwendete Stickstoffmenge in kg).
[RCL] 4	**12,000.00**	Summe der Quadrate der x-Werte, Σx^2.
[RCL] 5	**31.01**	Summe der y-Werte, Gesamtertrag an Reis.
[RCL] 6	**200.49**	Summe der Quadrate der y-Werte, Σy^2.
[RCL] 7	**1,415.00**	Summe der Produkte der x- und y-Werte.

Korrektur fehlerhafter Statistiken

Durch fehlerhafte Eingaben entstandene, inkorrekte Statistiksummen können sehr einfach korrigiert werden. Auch wenn nur ein Wert eines Zahlenpaares (x, y) falsch ist, müssen *beide* Werte entfernt und neu eingegeben werden:

1. Geben Sie das *fehlerhafte* Datenpaar in das X- und Y-Register ein.
2. Drücken Sie die Tasten [g] [Σ-]. Die Daten werden dann aus den entsprechenden Summen entfernt.
3. Geben Sie die korrekten Werte für x und y ein.
4. Drücken Sie die Taste [Σ+].

Sie können auch, wenn das zuletzt eingegebene Datenpaar falsch war, die Taste [Σ+] aber schon gedrückt wurde, [g] [LSTx] [g] [Σ-] drücken, um die falschen Daten zu entfernen*.

Beispiel: Nach der Eingabe der letzten Daten bemerkt Farmer, daß er eine verschmierte Zahl in seinem Laborbuch falsch gelesen hat. Der zweite y-Wert hätte 5.78 anstatt 4.78 lauten sollen. Korrigieren Sie diese Eingabe.

Tastenfolge	Anzeige	
4.78 [ENTER]	**4.78**	Eingabe des zu ersetzenden Datenpaares. Die fehlerhaften
20 [g] [Σ-]	**4.00**	Daten werden aus den Statistiksummen gelöscht, und der Zähler wird auf vier zurückgesetzt.
5.78 [ENTER]	**5.78**	Eingabe der korrekten Daten und Berechnung der Summen mit diesen neuen Daten.
20 [Σ+]	**5.00**	Die Anzahl (n) der akkumulierten Datenpaare ist wieder fünf.

Wir werden diese Statistikdaten in den folgenden Beispielen dieses Abschnittes weiterhin verwenden.

---

* Beachten Sie, daß diese Methode des Entfernens von Daten Rundungsfehler, die in den statistischen Registern entstanden sind, nicht korrigiert bzw. entfernt. Der Unterschied wird jedoch nicht sehr groß sein, solange sich das fehlerhafte Paar betragsmäßig nicht zu sehr vom korrekten Paar unterscheidet. In diesen Fällen ist es ratsam, die Berechnung neu zu beginnen.

Mittelwert

Die Funktion [$\bar{x}$] berechnet das arithmetische Mittel (Mittelwert) der x- und y-Werte, unter Verwendung der in Anhang A erläuterten Formel und der Statistiksummen in den entsprechenden Registern. Wenn Sie [g] [$\bar{x}$] drücken, wird der Inhalt des Stacks angehoben (um zwei Register, wenn der Stack nicht gesperrt ist, andernfalls um ein Register). Der Mittelwert von x ($\bar{x}$) wird in das X-Register kopiert, der Mittelwert von y ($\bar{y}$) gleichzeitig in das Y-Register. Drücken Sie [$x \gtrless y$], um $\bar{y}$ anzeigen zu lassen.

Beispiel: Berechnen Sie für die berichtigten Statistiksummen, die wir schon eingegeben haben, die durchschnittliche Menge an Dünger, $\bar{x}$, und den Durchschnittsertrag.

Tastenfolge	**Anzeige**	
[g] [$\bar{x}$]	**40.00**	Durchschnittliche Stickstoffmenge in kg, $\bar{x}$, unter Berücksichtigung aller Messungen.
[$x \gtrless y$]	**6.40**	Durchschnittlicher Ertrag von Reis in Tonnen, $\bar{y}$.

Standardabweichung

Durch Drücken von [g] [s] wird die *Standardabweichung* der akkumulierten statistischen Daten berechnet. Die zur Berechnung von s_y (Standardabweichung der akkumulierten x-Werte) und s_y (Standardabweichung der akkumulierten y-Werte) benutzten Formeln, werden im Anhang A aufgeführt.

Diese Funktion liefert die aus den *Stichprobendaten* resultierende *beste Schätzung* für die Standardabweichung der Grundgesamtheit, und wird deshalb gewöhnlich als Standardabweichung der *Stichprobe* bezeichnet*. Durch Drücken der Tasten [g] [s] wird der Stack angehoben (zweifach, wenn der Stack nicht gesperrt ist, andernfalls einfach), s_x wird in das X-Register geladen und s_y in das Y-Register. Um s_y anzuzeigen, drücken Sie [$x \gtrless y$].

---

* Wenn Ihre Daten nicht nur eine Stichprobe aus einer Grundgesamtheit darstellen, sondern die Grundgesamtheit selbst, ist die Standardabweichung dieser Daten die wahre Standardabweichung der Grundgesamtheit (und wird mit σ bezeichnet). Die Formel der wahren Standardabweichung unterscheidet sich um den Faktor $\sqrt{(n-1)/n}$ von der Formel, die zur Berechnung von [s] verwendet wird. Der Unterschied in beiden Werten ist gering für große n-Werte, und kann für die meisten Anwendungen vernachlässigt werden. Wollen Sie jedoch den exakten Wert der Standardabweichung der Grundgesamtheit berechnen, erhalten Sie das Ergebnis sehr einfach wie folgt: addieren Sie unter Verwendung von [Σ+] den Mittelwert der Daten zu den Daten, bevor Sie [g] [s] drücken. Das Ergebnis ist dann die wahre Standardabweichung der Grundgesamtheit. (Wenn Sie daraufhin irgendwelche Ihrer Daten korrigieren wollen, vergessen Sie nicht zuvor den Mittelwert wieder zu entfernen.)

Beispiel: Berechnen Sie die Standardabweichung zu dem oben berechneten Mittelwert.

Tastenfolge	**Anzeige**	
g s	**31.62**	Standardabweichung des mittleren Stickstoffverbrauches, $\bar{x}$.
$x \gtrless y$	**1.24**	Standardabweichung des mittleren Ertrages, $\bar{y}$.

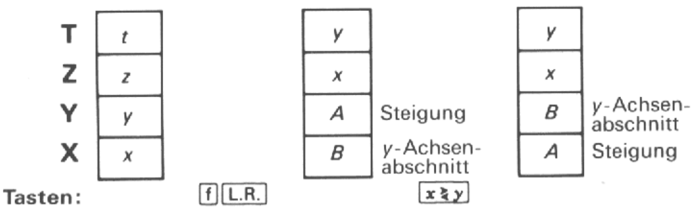
Lineare Regression

Die lineare Regression ist ein statistisches Verfahren zum Auffinden derjenigen Geraden, die die Quadrate der Abstände von zwei oder mehreren Datenpaaren von der Geraden minimiert und damit einen Zusammenhang zwischen den beiden Variablen schafft. Unter Verwendung der Methode der kleinsten Quadrate, wird durch Drücken von f L.R. die Steigung A und der y-Achsenabschnitt B der Geraden

$$y = Ax + B$$

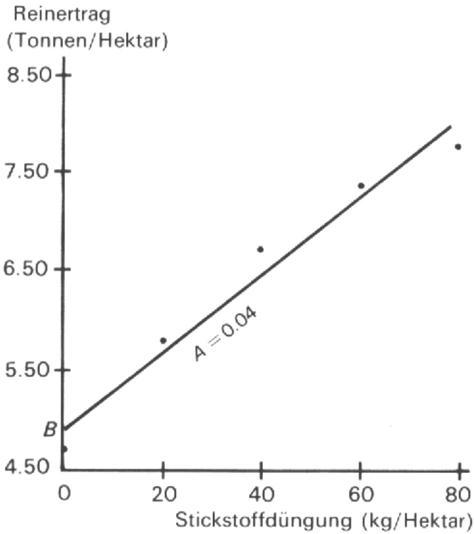
berechnet. Zur Benutzung der Regressionsfunktion ist wie folgt zu verfahren:

1. Akkumulieren Sie die Statistiksummen Ihrer Daten durch Σ+.
2. Drücken Sie die Tasten f L.R.. Der y-Achsenabschnitt B erscheint in der Anzeige (X-Register). Die Steigung A wird gleichzeitig im Y-Register abgelegt.
3. Drücken Sie $x \gtrless y$ zur Anzeige von A (wie bei den Funktionen $\bar{x}$ und s, verursacht L.R. die Anhebung des Stacks um zwei Register, wenn er nicht gesperrt ist, um ein Register andernfalls).



Die Steigung und der y-Achsenabschnitt der Regressionsgeraden werden mit den in Anhang A aufgeführten Gleichungen aus den akkumulierten Daten berechnet.

Beispiel: Finden Sie den *y*-Achsenabschnitt und die Steigung der linearen Approximation der obigen Daten und vergleichen Sie diese mit den gezeichneten Daten in der folgenden Grafik.



Tastenfolge	**Anzeige**	
[f] [L.R.]	**4.86**	*y*-Achsenabschnitt der Geraden.
[$x\gtrless y$]	**0.04**	Steigung der Geraden.

Linearer Schätzwert und Korrelationskoeffizient

Bei der Ausführung der Funktion [f] [$\hat{y}$,r] wird der *lineare Schätzwert* ($\hat{y}$) im angezeigten X-Register und der *Korrelationskoeffizient* (*r*) im Y-Register abgelegt. Drücken Sie [$x\gtrless y$] zur Anzeige von *r*.

Linearer Schätzwert. Aus den Statistiksummen wird nach Eingabe eines bekannten Wertes für *x* durch Auslösen von [f] [$\hat{y}$,r] ein geschätzter Wert für *y*, der mit $\hat{y}$ bezeichnet wird, berechnet.

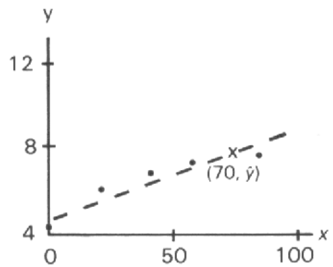
Korrelationskoeffizient. Bei der linearen Regression und der Schätzung wird unterstellt, daß der Zusammenhang zwischen den *x*- und *y*-Werten durch eine lineare Funktion approximiert werden kann. Der Korrelationskoeffizient, *r*, ist ein Maß dafür, wie «eng» gestreut die Datenpunkte um die Gerade liegen.

Der Korrelationskoeffizient (r) nimmt Werte von –1 bis 1 an:

$-1 \leqslant r \leqslant 1$; wobei –1 eine perfekte negative und +1 eine perfekte positive Korrelation darstellt.

Beachten Sie, daß Sie vor der Ausführung der Funktion [f] [ŷ,r] einen Wert für x eingeben müssen. Andernfalls wird der alte Inhalt des X-Registers verwendet, was gewöhnlich zu einem sinnlosen Wert für $\hat{y}$ führt.

Beispiel: Was würde geschehen, wenn 70 kg Stickstoffdünger für das Reisfeld verwendet würden? Sagen Sie aufgrund von Farmer's akkumulierten Statistiksummen den Ertrag voraus. Da der Korrelationskoeffizient automatisch in der Rechnung beinhaltet ist, können Sie anzeigen lassen, wie nahe die Werte an der Geraden liegen. Drücken Sie [x≷y], wenn der y-Schätzwert angezeigt wurde.



Tastenfolge	Anzeige	
70 [f] [ŷ,r]	**7.56**	Vorausgesagter Ertrag in Tonnen pro Hektar.
[x≷y]	**0.99**	Die ursprünglichen Daten liegen beinahe auf einer Geraden.

Andere Anwendungen

Interpolation. Lineare Interpolation von Tabellenwerten, wie z.B. in Thermodynamik- und Statistiktabellen, können mit dem HP-15C durch die Funktion [ŷ,r] sehr einfach ausgeführt werden. Dies ist möglich, da die lineare Interpolation eine lineare Schätzung *ist*: von zwei aufeinanderfolgenden Tabellenwerten wird angenommen, daß Sie auf einer Geraden liegen, und von dem unbekannten, dazwischenliegenden Wert wird angenommen, daß er auf dieselbe Gerade fällt.

Vektorarithmetik. Die statistischen Akkumulationsfunktionen können zur Ausführung von Vektoradditionen und Subtraktionen verwendet werden. Polarkoordinaten müssen vor der Eingabe in Rechteckskoordinaten umgewandelt werden (θ, [ENTER], r, [→R], [Σ+]). Die Ergebnisse werden in den Registern R_3 (Σx) und R_5 (Σy) (durch [RCL] [Σ+]) zurückgerufen und gegebenenfalls wieder in Polarkoordinaten umgewandelt.

Nach der Eingabe des zweiten Vektors ist anschließend die Taste [Σ+] oder [Σ−] zu drücken; je nachdem, ob der zweite Vektor addiert oder subtrahiert werden soll.

Abschnitt 5

Anzeige und Permanentspeicher

Kontrolle der Anzeige

Der HP-15C verfügt über drei Anzeigemodi – FIX, SCI und ENG – durch die mit Hilfe einer zusätzlichen Variablen (0 bis 9) das Anzeigeformat spezifiziert wird. Die folgende Illustration zeigt, in welchem Format die Zahl 123456 in jedem der drei Modi, bei jeweils vier Dezimalstellen, dargestellt wird.

f FIX 4	:	**123,456.0000**
f SCI 4	:	**1.2346 05**
f ENG 4	:	**123.46 03**

Durch den Permanentspeicher bleibt jeder Wechsel im Anzeigeformat solange erhalten, bis der Permanentspeicher gelöscht wird.

Das momentane Anzeigeformat wird wirksam, sobald die Ziffereingabe (z.B. durch ENTER) abgeschlossen wird; bis dahin werden alle Ziffern, die eingegeben werden (bis zu zehn), angezeigt.

Festkommaformat

Das FIX (*fixed decimal*) Format zeigt alle Zahlen mit den von Ihnen gewählten Dezimalstellen an (bis zu neun Stellen sind möglich, je nach Größe des ganzzahligen Anteils). Wenn die Zahl zur Anzeige im Festkommaformat zu groß oder zu klein ist, wird sie automatisch im Gleitkommaformat angezeigt. Im Einschaltzustand ist der HP-15C auf das FIX 4 Format voreingestellt. Ein Festkommaformat wird durch die Tastenfolge f FIX *n* definiert.

Tastenfolge	**Anzeige**	
123.4567895	**123.4567895**	
f FIX 4	**123.4568**	
f FIX 6	**123.456790**	Anzeige der auf sechs Dezimalstellen gerundeten Zahl. (Intern sind jedoch zehn Stellen gespeichert.)
f FIX 4	**123.4568**	Anzeige im üblichen FIX 4 Format.

Wissenschaftliches Anzeigeformat

Im SCI (*scientific*) Format wird die Zahl im X-Register in wissenschaftlicher Notation angezeigt. Die Tastenfolge f SCI *n* spezifiziert die Anzahl von Dezimalstellen, die angezeigt werden sollen. Da der Exponent drei Stellen in Anspruch nimmt, können maximal sechs Dezimalstellen angezeigt werden.

Die Anzeige wird auf die gewählte Anzahl von Dezimalstellen gerundet; wenn Sie jedoch mehr als maximal sechs Dezimalstellen spezifizieren (d.h. [SCI] 7, 8 oder 9) wird die entsprechende *nicht angezeigte* Stelle (siebte, achte oder neunte Stelle) gerundet*.

Mit der vorangegangenen Anzeige ergibt sich:

Tastenfolge	Anzeige	
[f] [SCI] 6	**1.234568 02**	Rundet auf sechs Ziffern und zeigt sechs Dezimalstellen an.
[f] [SCI] 8	**1.234567 02**	Rundet auf acht Dezimalstellen, zeigt aber nur sechs an.

Technisches Anzeigeformat

Bei Wahl von [ENG] (*engineering*) werden alle Zahlen in einem Gleitkommaformat angezeigt, das sich in den folgenden Punkten vom [SCI] Format unterscheidet:

- Im technischen Anzeigeformat ist der Exponent zur Basis 10 immer ein Vielfaches von 3 (z.B. 10^3, 10^6, 10^{12}).
- Die bei der Formatwahl spezifizierte Ziffernanzahl bezieht sich auf die Anzahl der signifikanten Stellen, die nach der *führenden* Ziffer angezeigt werden sollen.

Tastenfolge	Anzeige	
.012345	**0.012345**	
[f] [ENG] 1	**12. –03**	Rundet auf die erste Ziffer nach der führenden Ziffer.
[f] [ENG] 3	**12.35 –03**	
10 [×]	**123.5 –03**	Verschiebt den Dezimalpunkt, um ein Vielfaches von drei im Exponenten zu erhalten.
[f] [FIX] 4	**0.1235**	Anzeige im bisher üblichen [FIX] 4 Format.

* Die Anzeige zeigt keinen Unterschied zwischen [SCI] 7, 8 und 9, solange die gerundete Zahl keine 9 ist. In diesem Fall wird eine 1 zur nächst höheren Dezimalstelle addiert.

Mantissenanzeige

Unabhängig vom aktuellen Anzeigeformat stellt der HP-15C jede Zahl intern als Gleitkommazahl mit einer 10-stelligen Mantisse und einem zweistelligen Exponenten dar. Z.B. wird π intern immer als $3.141592654 \times 10^{00}$ dargestellt, gleich welche Form in der Anzeige erscheint. Zur Anzeige aller zehn Stellen der Mantisse einer Zahl im X-Register müssen Sie f CLEAR PREFIX drücken, und die Taste PREFIX gedrückt halten.

Tastenfolge	Anzeige
g π	**3.1416**
f CLEAR PREFIX (gehalten)	**3141592654**

Rundungsfehler

Wie Ihnen bereits bekannt ist, speichert der HP-15C intern jede Zahl mit 10 Stellen. Das Resultat jeder Berechnung wird auf zehn Stellen gerundet. Da der HP-15C nur endliche Approximationen (10 Stellen) von Zahlen wie z.B. π und 2/3 (0.666...) verarbeiten kann, tritt in der zehnten Stelle gegebenenfalls ein Rundungsfehler auf. Dieser Fehler kann im Verlauf längerer Rechnungen anwachsen, bleibt aber meist unbedeutend. Die korrekte Abschätzung der Auswirkungen von Rundungsfehlern erfordert Methoden der numerischen Analysis, die den Rahmen dieses Handbuches sprengen würden. Für eine detaillierte Behandlung dieser Methoden empfehlen wir Ihnen das Handbuch «*HP-15C – Fortgeschrittene Funktionen*».

Besondere Anzeigen

Statusanzeigen

Die Anzeige des HP-15C enthält acht Statusanzeigen, die den augenblicklichen Status Ihres Rechners bei verschiedenen Operationen melden. Die Bedeutung und Verwendung dieser Statusanzeigen wird auf den folgenden Seiten diskutiert.

*	Spannungsabfallanzeige, Seite 62.
USER	User-Modus, Seite 79 und 144.
f und **g**	Vorwahltasten der Alternativfunktionen, Seite 18–19.
RAD und **GRAD**	Trigonometrischer Modus, Seite 26.
C	Komplex-Modus, Seite 121.
PRGM	Programm-Modus, Seite 66.

Dezimal- und Zifferntrennzeichen

In der Voreinstellung des HP-15C wird der ganzzahlige Teil einer Zahl durch einen Punkt vom Dezimalteil getrennt. Der ganzzahlige Anteil wird durch Kommata in dreistellige Zifferngruppen gegliedert. Sie können diese Schreibweise umdrehen, um die in Deutschland gültige numerische Konvention zu setzen. Um das zu erreichen, schalten Sie zuerst Ihren Rechner aus. Drücken Sie dann [ON] und [·] gleichzeitig und halten Sie beide Tasten niedergedrückt. Lassen Sie anschließend zuerst die Taste [·] und dann die Taste [ON] los. (Das Wiederholen der Tastensequenz setzt den Rechner in die ursprüngliche Anzeigekonvention zurück.)

Tastenfolge	**Anzeige**
12345.67	**12,345.67**
[ON]/[·]	**12.345,6700**
[ON]/[·]	**12,345.6700**

Fehleranzeige

Wenn Sie versuchen, eine unerlaubte Operation – wie z.B. eine Division durch Null – auszuführen, erscheint im Display eine Fehlermeldung (die Anzeige **Error** gefolgt von einer Zahl). Im Anhang A finden Sie eine vollständige Auflistung aller möglichen Fehlermeldungen und ihrer Ursachen.

Um **Error** aus der Anzeige zu löschen und den Rechner in seinen vorangegangenen Zustand zurückzusetzen, können Sie eine beliebige Taste drücken. Anschließend kann der normale Rechenbetrieb fortgesetzt werden.

Overflow und Underflow

Overflow. Wenn das Ergebnis einer Berechnung in irgendeinem Register eine Zahl größer als $9.999999999 \times 10^{99}$ ist, wird in das entsprechende Register die Zahl $9.999999999 \times 10^{99}$ geladen, und Flag 9, der Overflowflag, gesetzt*. Ein gesetzter Flag 9 bedingt ein Blinken der Anzeige. Tritt innerhalb eines Programms ein Overflow auf, wird das Programm erst zu Ende ausgeführt, bevor die Anzeige zu blinken beginnt.

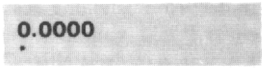
Mit Hilfe der Tasten [←], [ON] und [g] [CF] 9 können Sie Flag 9 löschen und das Blinken der Anzeige beenden.

Underflow. Ist das Ergebnis einer Rechnung in irgendeinem Register eine Zahl kleiner als $1.000000000 \times 10^{-99}$, so wird diese Zahl durch Null ersetzt. Ein Underflow hat keine weiteren Auswirkungen.

* Beachten Sie, daß die Anzeige die letzten drei Ziffern der Mantisse nicht enthält.

Spannungsabfallsanzeige

Erscheint ein blinkender Stern, der ein Absinken der Batteriespannung signalisiert, im linken unteren Eck der Anzeige, besteht noch kein Grund zur Sorge. Sie haben immer noch sehr viel Rechenzeit zur Verfügung: mindestens zehn Minuten, wenn Sie durchgehend ein Programm laufen lassen, und etwa eine Stunde, wenn Sie manuell Berechnungen ausführen. In Anhang F (Seite 259) wird das Auswechseln der Batterien beschrieben.



Der Permanentspeicher

Status

Der Permanentspeicher des HP-15C erhält die folgenden Informationen, auch wenn der Rechner ausgeschaltet wird:

- Alle numerischen Daten, die im Rechner gespeichert sind.
- Alle im Rechner gespeicherten Programme.
- Position des Rechners im Programmspeicher.
- Anzeigemodus und -Format.
- Den trigonometrischen Modus (Altgrad, Bogenmaß, Neugrad).
- Alle anstehenden Unterprogrammrücksprünge («pending returns»).
- Alle Flagzustände (mit Ausnahme von Flag 9, der durch das *manuelle* Ausschalten des Rechners gelöscht wird).
- Den User-Modus (falls gesetzt).
- Den Komplex-Modus (falls gesetzt).

Beim Einschalten «erwacht» der HP-15C immer im Run-Modus.

Schaltet man den HP-15C aus, bleibt der Inhalt des Permanentspeichers sogar während der kurzen Zeit eines Batteriewechsels erhalten. Daten und Programme werden länger erhalten als die anderen Informationen über den Rechnerzustand. Das Auswechseln der Batterien wird in Anhang F beschrieben.

Löschen des Permanentspeichers

Wenn Sie den Permanentspeicher Ihres HP-15C völlig löschen wollen, müssen Sie folgendermaßen vorgehen:

1. Schalten Sie den Rechner aus.
2. Drücken Sie die Taste [ON], halten Sie diese Taste niedergedrückt und drücken Sie gleichzeitig die Taste [-].
3. Lassen Sie zuerst die Taste [ON] und dann die Taste [-] wieder los. Dies wird durch die Tastenfolge [ON]/[-] dargestellt.

Nach einem Löschen des Permanentspeichers erscheint **Pr Error** (*power error*) in der Anzeige. Das Drücken irgendeiner Taste löscht diese Anzeige.

> **Hinweis**: Der Permanentspeicher kann versehentlich gelöscht werden, wenn der Rechner zu Boden fällt oder auf andere Art und Weise mißhandelt wird.

Teil II
Programmierung des HP-15C

Abschnitt 6

Grundlagen der Programmierung

Die folgenden fünf Abschnitte erläutern die Grundlagen der Programmierung Ihres HP-15C. Jeder dieser Abschnitte ist dreifach unterteilt, und zwar in

1. das Handwerkszeug, hier wird die grundlegende Technik erklärt,
2. Beispiele zur Ausführung dieser Technik und
3. zusätzliche Informationen.

Es empfiehlt sich nur so weit zu lesen, wie Sie es für die Handhabung Ihres HP-15C benötigen.

Das Handwerkszeug

Entwerfen eines Programms

Die Programmierung Ihres HP-15C ist sehr einfach und basiert auf der Aufzeichnung der Tastenfolge, wie sie bei einer manuellen Berechnung der Problemstellung verwendet werden würde. (Man nennt dieses Prinzip «Tastenfolgenprogrammierung».) Um jedoch aus einer Folge von Rechenschritten ein Programm zu entwickeln bedarf es zweier zusätzlicher Überlegungen: Sie müssen entscheiden wann und wo die Daten eingegeben werden sollen, das Programm anschließend eingeben und dann abspeichern. Des weiteren können Sie Ihr Programm so aufbauen, daß es während der Ausführung Entscheidungen oder bestimmte Schritte wiederholt. Dies wird durch bedingte und unbedingte Verzweigungen erreicht.

Im Verlaufe der Erläuterungen zu den Grundlagen der Programmierung, werden wir das Beispiel des «Körper im freien Fall» (siehe Seite 14), nochmals aufgreifen.

Eingabe eines Programms

Programm-Modus. Durch Drücken von [g] [P/R] (*program/run*) wird der Rechner in den *Programm-Modus* geschaltet. (Die Statusanzeige **PRGM** erscheint im Display.) Befindet sich der Rechner im Programm-Modus, werden alle Funktionen gespeichert, aber nicht ausgeführt.

Tastenfolge	**Anzeige**	
[g] [P/R]	**000–**	Setzt den Rechner in den Programm-Modus; die Statusanzeige **PRGM** und die Zeilennummer **000** werden angezeigt.

Aufteilung des Programmspeichers. Der der Programmspeicherung zugewiesene Teil des Speichers des HP-15C ist in Zeilen aufgeteilt, die wiederum durch Zeilennummer gekennzeichnet sind. Die Zeile 000 markiert den Anfang des Programmspeichers und kann nicht zur Speicherung einer Programmanweisung verwendet werden. Die erste Zeile, die eine Anweisung enthalten kann, ist die Zeile 001. Andere Zeilen als die Zeile 000, existieren nicht, solange keine Anweisungen darin abgelegt wurden.

Sie *können* ein Programm in jeder existierenden (mit *nnn* bezeichneten) Zeile beginnen. Es ist aber am einfachsten und sichersten, ein unabhängiges Programm (im Gegensatz zu einem Unterprogramm) am Anfang des Programmspeichers zu beginnen. Bei der Eingabe neuer Programmzeilen bleiben die bereits vorhandenen Programmzeilen erhalten und werden im Speicher nach unten verschoben.

Durch das Drücken von [GTO] [CHS] 000 wird sowohl im Programm-Modus als auch im Run-Modus der Rechner auf Zeile 000 gesetzt. Diese Anweisung [GTO] wird dabei nicht als Programmzeile aufgezeichnet. Im *Run-Modus* wird der Rechner auch durch Drücken von [f] CLEAR [PROG] in die Zeile 000 gesetzt, jedoch ohne dabei den Programmspeicher zu löschen.

Durch Drücken von [f] CLEAR [PRGM] im *Programm-Modus*, können Sie den Programmspeicher löschen, d.h. alle Programme im Speicher entfernen und den Rechner auf Zeile 000 setzen.

Anfang eines Programms. Eine *Label*-Anweisung – die Tasten [f] [LBL] gefolgt von einem Buchstaben ([A] bis [E]) oder einer Zahl (0 bis 9 und .0 bis .9) – wird dazu verwendet den Beginn eines Programms oder einer Routine zu definieren. Die Verwendung von Labels erlaubt Ihnen schnell ein ganz bestimmtes Programm oder eine bestimmte Routine aus vielen anderen herauszugreifen und ausführen zu lassen.

Tastenfolge	**Anzeige**	
[f] CLEAR [PRGM]	**000–**	Löschung des Programmspeichers und setzt den Rechner auf die Zeile 000 (Beginn des Programmspeichers).
[f] [LBL] [A]	**001–42,21,11**	

Aufzeichnen eines Programmes. Jede gedrückte Taste – sei es ein Operator oder eine Konstante – wird im Speicher als eine Programmanweisung* abgelegt.

---

*Ausgenommen die *nicht programmierbaren* Funktionen, die auf Seite 80 aufgelistet sind.

Tastenfolge	Anzeige		
2	002–	2	Gibt man h in das X-Register ein, berechnen die Zeilen 002 bis 008 die Fallzeit $\sqrt{\frac{2h}{9.8}}$.
[×]	003–	20	
9	004–	9	
[.]	005–	48	
8	006–	8	
[÷]	007–	10	
[√x]	008–	11	

Ende eines Programms. Es gibt drei Möglichkeiten ein Programm zu beenden:

- Die Anweisung [g] [RTN] (*return*) beendet ein Programm, kehrt zur Zeile 000 zurück und stoppt.
- Die Anweisung [R/S] stoppt ein Programm *ohne* zur Zeile 000 zurückzuspringen.
- Am Ende des Programmspeichers befindet sich automatisch eine [RTN] Anweisung.

Tastenfolge	Anzeige		
[g] [RTN]	009–	42 32	Diese Anweisung ist überflüssig, *falls das Programm das letzte Programm im Speicher ist*.

Anhalten der Programmausführung

Die Verwendung der Tasten [f] [PSE] (*pause*) als Programmanweisung, hat zur Folge, daß die Programmausführung an dieser Stelle *kurzzeitig* angehalten, und ein Zwischenergebnis angezeigt wird. (Um eine längere Pause zu bewirken verwenden Sie [f] [PSE] mehrmals.)

Die Anweisung [R/S] hält das Programm für unbestimmte Zeit an. Das Programm bleibt auf der betreffenden Zeile positioniert. Sie können die Programmausführung fortsetzen, beginnend mit dieser Zeile, indem Sie im Run-Modus die Taste [R/S] drücken.

Ausführen eines Programms

Run-Modus. Wenn Sie Ihr Programm geladen haben, schalten Sie um in den Run-Modus: [g] [P/R]. Die Ausführung des Programms muß im Run-Modus erfolgen.

Tastenfolge	Anzeige	
[g] [P/R]		Run-Modus; die Statusanzeige **PRGM** wird nicht mehr angezeigt. (Die Anzeige hängt von den vorhergehenden Berechnungen ab.)

Die Position des Rechners im Programmspeicher verändert sich nicht, wenn von einem Modus in einen anderen umgeschaltet wird. Nach einem Ausschalten, «erwacht» der Rechner immer im Run-Modus.

Starten des Programms. Drücken Sie im *Run-Modus* die Taste [f] und ein *Buchstaben-Label* oder [GSB] und ein *Ziffern-* (oder Buchstaben-)*Label*. Dies adressiert ein Programm und startet seine Ausführung. In der Anzeige blinkt während der Ausführung die Meldung **running**.

Tastenfolge	Anzeige	
300.51	**300.51**	Geben Sie einen Wert für *h* in das X-Register ein.
[f] [A]	**7.8313**	Das Ergebnis des Programms «A». (Die Fallzeit eines Objektes aus einer Höhe von 300.51 m.)

Fortsetzen der Programmausführung. Durch Drücken von [R/S] wird die Ausführung eines Programmes, das durch eine Anweisung [R/S] unterbrochen wurde, fortgesetzt.

User-Modus. Der User-Modus ist ein Zustand des Rechners, durch den Sie sich bei der Ausführung von *mit Buchstaben benannten* Programmen das Bedienen bestimmter Tasten ersparen können. Durch Drücken von [f] [USER] werden die Primärfunktionen und die durch die Vorwahltasten anzuwählenden Funktionen der mit [A] bis [E] belegten Tasten vertauscht. Sie können also im User-Modus die Ausführung eines Programms starten, indem Sie nur eine einzige Taste drücken (die Tasten [f] oder [GSB] brauchen nicht gedrückt zu werden).

Die Dateneingabe

In jedem Programm muß festgelegt werden, wie und an welcher Stelle Daten eingegeben werden. Dies kann entweder im Run-Modus vor der Ausführung des Programms geschehen oder während einer Unterbrechung des Programms.

1. **Vorabeingabe**. Wenn ein variabler Wert in der ersten Zeile des Programms verwendet werden soll, geben Sie diesen vor dem Start des Programms in das X-Register ein. Wenn dieser Wert später verwendet wird, können Sie ihn in einem Speicherregister abspeichern (durch [STO]) und später im Programm mit der Anweisung [RCL] zurückrufen.

Diese Methode haben wir oben verwendet, wo ein Wert für *h* vor der Ausführung des Programms in das X-Register eingegeben wurde. Eine ENTER Anweisung ist nicht nötig, da die Anweisung zum Starten des Programms (hier: f A) sowohl die Zifferneingabe abschließt, als auch einen Stack Lift ermöglicht. Das obige Programm multipliziert im nächsten Schritt den Inhalt des X-Registers mit zwei.

Der Stack macht es sogar möglich mehr als eine Variable vor dem Ausführen des Programms einzugeben. Wenn Sie berücksichtigen, wie sich der Stack bei aufeinanderfolgenden Berechnungen verändert und wie der Stack (z.B. durch $x \gtrless y$) umgeordnet werden kann, dann erkennen Sie leicht, daß es möglich ist, Programme zu schreiben, die in das X, Y, Z und T-Register eingegebene Variablen verwenden.

2. **Direkte Eingabe**. Geben Sie die Daten so ein, wie sie im Programmverlauf verwendet werden. Setzen Sie eine Anweisung R/S (*run/stop*) in Ihr Programm an die Stelle, wo die Daten benötigt werden. Geben Sie hier Ihre Daten ein und drücken Sie R/S, um die Programmausfürung fortzusetzen.

Geben Sie keine variablen Daten direkt in den Programmtext ein. Jeder sich ändernde Wert eines Programms sollte mit jeder Programmausführung neu eingegeben werden.

Programmspeicher

Im Einschaltzustand (nach einem Löschen des Permanentspeichers) bietet Ihnen der HP-15C 322 Bytes für die Programmspeicherung und 21 Speicherregister. Die *meisten* Programmschritte (Anweisungen) verwenden ein Byte, einige benötigen zwei. Die Aufteilung der Speicherkapazität kann verändert werden; dies wird in Anhang C beschrieben. Der maximal verfügbare Programmspeicher besteht aus 448 Bytes (wobei die permanenten Speicherregister – R_I, R_0 und R_1 – erhalten bleiben); die Maximalzahl der Datenspeicherregister ist 67 (wobei kein Programmspeicher mehr vorhanden ist).

Beispiel

Die Dosenfabrik «Mutter's Küche» möchte ein Spaghetti Fertiggericht zusammenstellen, das in einer Verpackung drei verschiedene zylindrische Dosen enthält: eine Dose mit Spaghettisoße, eine mit geriebenem Käse, und eine mit Fleischstückchen. Die Firma muß nun die Grundfläche, die Gesamtoberfläche und das Volumen der drei Dosen berechnen. Außerdem möchte sie die Grundfläche der Gesamtverpackung und ihr Volumen berechnen.

Das Programm zur Berechnung dieser Informationen verwendet die folgenden Formeln und Daten:

Grundfläche = πr^2.
Volumen = Grundfläche × Höhe = $\pi r^2 h$.
Oberfläche = 2 × Grundfläche + Seitenfläche = $2\pi r^2 + 2\pi r h$.

Radius r	Höhe h	Grundfläche	Volumen	Oberfläche
2.5 cm	8.0 cm	?	?	?
4.0	10.5	?	?	?
4.5	4.0	?	?	?
	GESAMT	?	?	?

Vorgehensweise:

1. Geben Sie einen r-Wert in den Rechner ein und sichern Sie ihn für andere Berechnungen. Berechnen Sie die Grundfläche (πr^2), speichern Sie diese für spätere Berechnungen und addieren Sie diese Grundfläche zu dem Inhalt eines Registers, das am Ende aller Berechnungen die Summe aller Grundflächen enthalten wird.
2. Geben Sie den h-Wert ein, berechnen Sie das Volumen und addieren Sie diese Zahl zu dem Inhalt des Registers, das am Ende aller Berechnungen die Summe aller Volumen enthält.
3. Rufen Sie den r-Wert zurück. Dividieren Sie das Volumen durch r und multiplizieren Sie es mit zwei. Dies liefert den Wert der Seitenfläche. Den Wert der Oberfläche erhalten Sie, indem Sie den Grundflächenwert zurückrufen, ihn mit zwei multiplizieren und den Wert der Seitenfläche addieren. Summieren Sie die Oberflächenwerte in einem Register auf.

Geben Sie *nicht* die aktuellen Werte in das Programm ein, sondern bereiten Sie die Eingabe innerhalb Ihres Programmes vor. Da diese Werte variieren, werden sie vor und/oder während der Ausführung des Programmes eingegeben.

Geben Sie das folgende Programm ein, um das obige Problem zu lösen. Das Display zeigt die Zeilennummer und den Tastencode (die Reihe und Spalte der Taste auf dem Tastenfeld) an, der später im Unterabschnitt «zusätzliche Informationen» erklärt wird.

Tastenfolge	Anzeige	
g P/R	**000–**	Schaltet den Rechner in den Programm-Modus (**PRGM** erscheint in der Anzeige).
f CLEAR PRGM	**000–**	Löscht den Programmspeicher, Anzeige der Zeile 000.

Tastenfolge	Anzeige	
f LBL A	001– 42,21,11	Weist diesem Programm das Label «A» zu.
STO 0	002– 44 0	Speichert den Inhalt des X-Registers in das Register R_0. r muß sich im X-Register befinden, bevor das Programm ausgeführt werden kann.
g x^2	003– 43 11	Quadriert den Inhalt des X-Registers (der r sein wird).
g π	004– 43 26	
×	005– 20	πr^2, die GRUNDFLÄCHE einer Dose.
STO 4	006– 44 4	Speichert die GRUNDFLÄCHE in R_4.
STO + 1	007– 44,40, 1	Speichert die Summe aller GRUNDFLÄCHEN in R_1.
R/S	008– 31	Stoppt die Ausführung des Programms, um den Wert der GRUNDFLÄCHE anzuzeigen und um die Eingabe eines h-Wertes zu ermöglichen.
×	009– 20	Das Multiplizieren von h mit der GRUNDFLÄCHE ergibt den Wert des VOLUMENS.
f PSE	010– 42 31	Kurze Unterbrechung, um das Volumen anzuzeigen.
STO + 2	011– 44,40, 2	Speichert die Summe der Volumen aller Dosen in R_2.
RCL	012– 45 0	Ruft den Wert von r zurück.
÷	013– 10	Dividiert das Volumen durch r.
2	014– 2	
×	015– 20	$2\pi rh$, die SEITENFLÄCHE einer Dose.
RCL 4	016– 45 4	Zurückrufen der GRUNDFLÄCHE der Dose.
2	017– 2	Mutipliziert die GRUNDFLÄCHE mit zwei (Boden und Deckel).
×	018– 20	

Tastenfolge	Anzeige	
[+]	019– 40	SEITENFLÄCHE + GRUNDFLÄCHE = OBERFLÄCHE.
[STO] [+] 3	020– 44,40, 3	Speichert die Summe aller Oberflächen in R_3.
[g] [RTN]	021– 43 32	Beendet das Programm, der Rechner springt zurück zur Zeile 000.

Lassen Sie uns das Programm nun ausführen:

Tastenfolge	Anzeige	
[g] [P/R]		Schaltet den Rechner in den Run-Modus (**PRGM** erlischt in der Anzeige).
[f] CLEAR [REG]		Löschen *aller* Speicherregister. Die Anzeige wird nicht verändert.
2.5	2.5	Eingabe des *r*-Wertes der ersten Dose.
[f] [A] (or: [GSB] [A])	19.6350	Startet das Programms A. In der Anzeige erscheint der Wert der GRUNDFLÄCHE der ersten Dose. (**running** blinkt in der Anzeige während der Ausführung des Programmes auf.)
8	8	Eingabe des *h*-Wertes der ersten Dose; danach Fortsetzung der Programmausführung.
[R/S]	157.0796	VOLUMEN der ersten Dose.
	164.9336	OBERFLÄCHE der ersten Dose.
4	4	Eingabe des *r*-Wertes der zweiten Dose.
[R/S]	50.2655	GRUNDFLÄCHE der zweiten Dose.
10.5	10.5	Eingabe des *h*-Wertes der zweiten Dose.
[R/S]	527.7876	VOLUMEN der zweiten Dose.
	364.4247	OBERFLÄCHE der zweiten Dose.
4.5	4.5	Eingabe des *r*-Wertes der dritten Dose.
[R/S]	63.6173	GRUNDFLÄCHE der dritten Dose.

Tastenfolge	Anzeige	
4	4	Eingabe des *h*-Wertes der dritten Dose.
R/S	254.4690	VOLUMEN der dritten Dose.
	240.3318	OBERFLÄCHE der dritten Dose.
RCL 1	133.5177	Summe aller GRUND-FLÄCHEN.
RCL 2	939.3362	Summe aller VOLUMEN.
RCL 3	769.6902	Summe aller OBER-FLÄCHEN.

Dieses Programm illustriert die wesentlichen Techniken bei der Programmierung. Es zeigt ebenfalls wie Daten im Programm- oder Run-Modus durch die Eingabe, das Speichern oder den Rückruf von Werten (input und output) manipuliert werden können. Hierfür werden die Tasten ENTER, STO und RCL, die Speicherarithmetik und die vorprogrammierten Unterbrechungen der Programmausführung verwendet.

Zusätzliche Informationen

Programmanweisungen

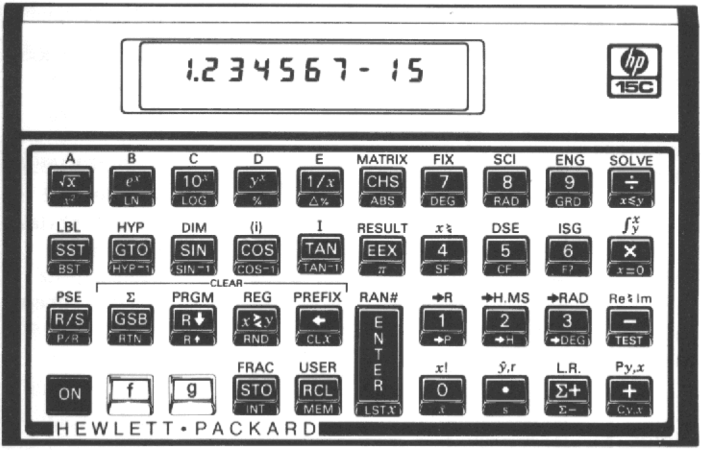
Jede Ziffer, jeder Dezimalpunkt und jede Funktionstaste wird als *Anweisung* betrachtet und wird in eine *Zeile* des Programmspeichers abgelegt. Eine Anweisung kann Vorwahltasten enthalten (wie z.B. f, GTO, STO und LBL), belegt aber trotzdem nur eine Zeile. Die meisten Anweisungen benötigen ein Byte des Programmspeichers; einige brauchen jedoch zwei. Eine vollständige Auflistung der Anweisungen, die zwei Bytes benötigen, finden Sie im Anhang C.

Codierung der Anweisungen

Jede Taste des HP-15C – mit Ausnahme der Tasten für die Ziffern 0 bis 9 – werden im Programm-Modus durch einen zweistelligen «Tastencode» identifiziert, der sich auf die Position der Taste auf dem Tastenfeld des Rechners bezieht.

Anweisung	Code	
STO + 1	006– 44,40, 1	Sechste Programmzeile.
f DSE I	XXX–42, 5,25	DSE hat den Code «5».

Die erste Ziffer des Tastencodes bezieht sich auf die Reihe, in der sich die Taste befindet (1 bis 4, von oben nach unten), und die zweite Ziffer auf die Spalte (1, 2, 3, ..., 9, 0). Ausnahme: der Tastencode einer Ziffer besteht aus der Ziffer selbst.



Tastencode 25: zweite Reihe, fünfte Taste.

Speicherkonfiguration

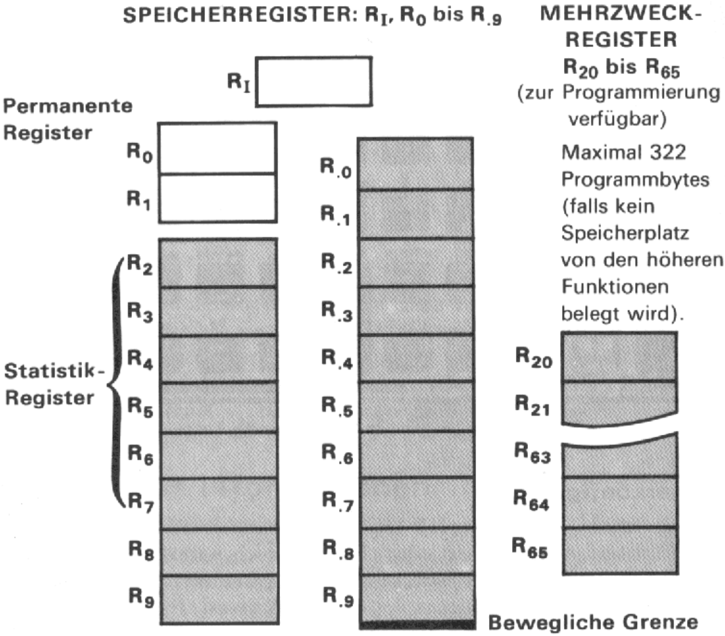
Für den üblichen Gebrauch des HP-15C ist ein Verständnis der Speicherkonfiguration nicht erforderlich. Es ist jedoch unentbehrlich, wenn Sie die Leistungsfähigkeit des HP-15C bei der Speicherung und Programmierung voll ausnutzen wollen. Je mehr Sie programmieren, desto hilfreicher wird dieses Wissen für Sie sein. Die Konfiguration und Aufteilung des Speichers wird im Anhang C («Speicheraufteilung») ausführlich beschrieben.

Sobald in der Anzeige die Meldung **Error 10** erscheint, haben Sie die Speichergrenzen Ihres HP-15C erreicht. Eine Neuaufteilung des Speichers gibt Ihnen wesentlich mehr Flexibilität bei der Speicherung von Informationen.

Der Speicher des HP-15C besteht aus 67 Registern (R_0 bis R_{65} und dem Indexregister), die zwischen Daten- und Programmspeicher (bzw. den höheren Funktionen des HP-15C) aufgeteilt sind.

- 46 Register stehen sowohl für die Programmierung, als auch für die höheren Funktionen zur Verfügung ([SOLVE], [∫], der komplexe Stack und die [MATRIX] Funktionen). Da jedes Register 7 Bytes Speicherkapazität hat, entsprechen die 46 Register also 322 Programmbytes, wenn kein Speicher für die höheren Funktionen verwendet wird.
- 21 Register stehen für die Datenspeicherung zur Verfügung (R_0 bis R_9, $R_{.0}$ bis $R_{.9}$, und das Indexregister).

Standard-Speicherkonfiguration



Aufteilbare Register (schattiert)

Der Speicher kann neu aufgeteilt werden, indem man dem Rechner angibt, welches Register das Datenspeicherregister mit der höchsten Nummer sein soll; dadurch sind alle anderen Register für die Programmierung und die höheren Funktionen verfügbar.

Tastenfolge	Anzeige	
60 [f] [DIM] [(i)] *	60.0000	Das Register R_{60} und alle Speicher mit niedrigeren Nummern werden zu Datenspeicherregistern; fünf Register (R_{61} bis R_{65}) bleiben als Programmspeicher verfügbar.

* Das optionale Auslassen der Taste [f] *nach* einer anderen Vorwahltaste wird auf Seite 78, «Abgekürzte Tastenfolgen», erklärt.

Tastenfolge	Anzeige	
1 [f] [DIM] [(i)]	1.0000	R_1 und R_0 sind verfügbar für die Datenspeicherung; R_2 bis R_{65} für Programme und die höheren Funktionen.
19 [f] [DIM] [(i)]	19.0000	Standardaufteilung: R_{19} ($R_{.9}$) und die «tieferen» Register sind Datenspeicherregister; R_{20} und R_{65} sind verfügbar für Programme und die höheren Funktionen.
[RCL] [DIM] [(i)]	19.0000	Zeigt das augenblicklich höchste Datenspeicherregister an.

Die Funktionen [DIM] und [MEM] (*memory status*) werden in Anhang C ausführlich beschrieben.

Beachten Sie, daß eine Fehlermeldung in der Anzeige erscheint (*ausgehend von der obigen Speicheraufteilung*), wenn

1. Sie versuchen ein höheres Register als R_{19} ($R_{.9}$), das anfänglich das höchste für die Dateneingabe verfügbare Register ist, aufzurufen (**Error 3**), oder wenn
2. Sie 322 Bytes mit einem Programm belegt haben und versuchen noch mehr Programmzeilen abzuspeichern (**Error 4**), oder wenn
3. Sie versuchen, eine höhere Funktion auszuführen, ohne genügend Speicherkapazität zur Verfügung zu haben (**Error 10**).

Begrenzung von Programmen

Ende eines Programms. Nicht jedes Programm muß mit einer [RTN] Anweisung oder [R/S] enden. Am Ende des Programmspeichers befindet sich *automatisch* eine [RTN] Anweisung, so daß Sie diese nicht an Ihr Programm anfügen müssen. Dadurch sparen Sie eine Zeile des Speichers. Außerdem kann ein Programm damit «enden», daß durch die Anweisung [GTO] (siehe Abschnitt 7) die Ausführung mit einer anderen Routine fortgesetzt wird.

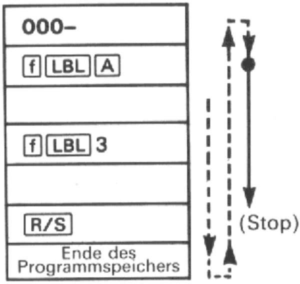
Labels. Labels innerhalb eines Programms oder eines Unterprogramms sind Markierungen, die dem Rechner anzeigen, wo die Ausführung beginnen soll. Nach einer [f] *Label* oder [GSB] *Label* Anweisung sucht der Rechner in den folgenden Zeilen (also nach «unten») des Programmspeichers nach dem entsprechenden Label.

* Bei der Speicheraufteilung und indirekten Adressierung, werden die Register $R_{.0}$ bis $R_{.9}$ als R_{10} bis R_{19} bezeichnet.

Falls nötig wird diese Suche nach dem Erreichen des Endes des Programmspeichers bei der Zeile 000 fortgesetzt. Nach dem Auffinden des gesuchten Labels, wird die Suche abgebrochen, und die Ausführung des Programms beginnt.
Tritt ein Label innerhalb eines laufenden Programmes auf, hat dies keine Auswirkungen auf das Ausführen des Programmes, d.h. die Ausführung wird einfach mit der nächsten Programmanweisung fortgesetzt. Dadurch ist es möglich ein Unterprogramm innerhalb eines Programms mit einem Label zu versehen. (Mehr über Unterprogramme finden Sie in Abschnitt 9.)
Da der Rechner nur in eine Richtung, ausgehend von der augenblicklichen Position im Programmspeicher, seine Suche beginnt, ist es möglich (wenngleich auch nicht ratsam), identische Labels für verschiedene Programme zu verwenden. Die Ausführung beginnt in der ersten Zeile, die das gewünschte Label enthält.

Wenn eine [f] [A] Eingabe die Suche nach «A» hier beginnt,

setzt sie sich nach unten im Speicher fort, springt vom Ende des Speichers in die Zeile 000 und endet am Label «A». Hier startet die Ausführung und setzt sich fort, bis eine Haltanweisung erfolgt (andere Labels werden ignoriert).



Unerwartete Programmunterbrechungen

Auslösen einer beliebigen Taste. Durch Drücken einer beliebigen Taste wird die Programmausführung angehalten. Die momentan sich in der Abarbeitung befindliche Anweisung wird jedoch noch vollständig ausgeführt, bevor das Programm stoppt.
Unterbrechung durch einen Fehler. Die Programmausführung wird unmittelbar angehalten, wenn der Rechner versucht eine unerlaubte Operation auszuführen, die zu einer **Error** Anzeige führt.
Um die Zeilennummer und den Tastencode der einen Fehler verursachenden Anweisung (die Zeile in der das Programm angehalten wurde), anzeigen zu lassen, drücken Sie irgendeine Taste um die **Error** Meldung zu löschen und schalten den Rechner in den Programm-Modus.
Blinkt die Anzeige wenn das Programm stoppt, liegt eine Overflowsituation vor (siehe Seite 61). Das Drücken von [←], [ON] oder [g] [CF] 9 beendet das Blinken.

Verkürzte Tastenfolgen

In manchen Fällen wird die Vorwahltaste [f], die in den Tastenfolgen für einige Programmanweisungen normalerweise enthalten sein sollte, nicht benötigt.

Die folgende Regel gilt für die Verwendung von *verkürzten Tastenfolgen*: die Vorwahltaste [f] ist unnötig nach einer anderen Vorwahltaste. (Seite 19 enthält eine Liste der Vorwahltasten.)

Beispielsweise wird [f] [LBL] [f] [A] zu [f] [LBL] [A], [f] [DIM] [f] [(i)] zu [f] [DIM] [(i)] und [STO] [f] [RAN#] zu [STO] [RAN#]. Das Auslassen der Taste [f] ist in diesen Fällen nicht zweideutig, da die Funktionen mit der Vorwahltaste [f] in diesen Fällen die einzig logischen Funktionen sind. Der Tastencode solcher Anweisungen enthält die unnötige Taste [f] nicht, selbst wenn Sie die Taste eingeben.

User-Modus

Der User-Modus ist eine Servicefunktion, durch die Sie sich beim Abrufen von Programmen das Bedienen bestimmter Tasten ersparen können. Durch Drücken von [f] [USER] werden die Primärfunktionen und die Funktionen mit der Vorwahltaste [f] der Tasten [A] bis [E] vertauscht. Dies verdeutlicht die folgende Illustration (der Rechner befindet sich im User-Modus, die Statusanzeige **USER** erscheint in der Anzeige):

[f] Vorwahl	A	B	C	D	E
Primärfunktion	[$\sqrt{x}$]	[e^x]	[10^x]	[y^x]	[$1/x$]
[g] Vorwahl	x^2	LN	LOG	%	Δ%

Durch nochmaliges Drücken von [f] [USER] wird der User-Modus ausgeschaltet.

Auswertung von Polynomen mit dem Horner-Schema

Einige Ausdrücke, wie z.B. Polynome, enthalten dieselbe Variable mehrmals. So enthält beispielsweise der Ausdruck

$$f(x) = Ax^4 + Bx^3 + Cx^2 + Dx + E$$

die Variable x viermal. Ein Programm zur Lösung dieser Gleichung könnte wiederholt eine gespeicherte Kopie von x aus einem Speicherregister abrufen. Eine kürzere Lösung dieses Problems besteht jedoch darin den Stack mit der Konstanten zu laden (siehe Seite 41).

Das Horner-Schema ist eine nützliche Methode zur Umordnung von Polynomen, bei der die Zahl von Rechenschritten und die Rechenzeit verkürzt wird. Sie ist besonders nützlich bei Funktionen [SOLVE] und [$\int_y^x$], deren Ausführung ziemlich lange dauert, da sie Unterprogramme verwenden.

Das Horner-Schema besteht in einer Umordnung des Polynoms in eine ineinandergeschachtelte Form, die die Exponenten größer als eins eliminiert:

$$Ax^4 + Bx^3 + Cx^2 + Dx + E$$

$$(Ax^3 + Bx^2 + Cx + D)x + E$$

$$((Ax^2 + Bx + C)x + D)x + E$$

$$(((Ax + B)x + C)x + D)x + E$$

Beispiel: Schreiben Sie ein Programm für die Berechnung von $5x^4 + 2x^3$ in der Form $(((5x + 2)x)x)x$, und berechnen Sie den Ausdruck für den Wert $x = 7$:

Tastenfolge	Anzeige	
g P/R	000–	Unterstellt, daß der Rechner auf Zeile 000 positioniert ist (gegebenenfalls ist der Programmspeicher zu löschen).
f LBL B	001–42,21,12	
5	002– 5	
×	003– 20	$5x$.
2	004– 2	
+	005– 40	$5x + 2$.
×	006– 20	$(5x + 2)x$.
×	007– 20	$(5x + 2)x^2$.
×	008– 20	$(5x + 2)x^3$.
g RTN	009– 43 32	
g P/R		Run-Modus. Anzeige des letzten Ergebnisses.
7 ENTER ENTER ENTER	7.0000	Lädt der Stack (X-, Y-, Z- und T-Register) mit dem Wert 7.
f B	12,691.0000	

Nicht programmierbare Funktionen

Im Programm-Modus kann beinahe jede Funktion des Tastenfeldes als Anweisung in den Programmspeicher eingegeben werden. Die folgenden Funktionen sind Ausnahmen und können *nicht* als Programmanweisungen gespeichert werden:

f CLEAR PREFIX	g BST	SST
f CLEAR PRGM	g MEM	←
f (i)	g P/R	ON / ·
f USER	GTO CHS *nnn*	ON / −

Übungsaufgaben

1. In dem Dorf Sonance wurde eine 12 Uhr-Sirene auf dem Turm des Feuerwehrhauses angebracht. Die Lautstärke an der Eingangstür des Feuerwehrhauses, 3,2 m von der Sirene entfernt, beträgt 138 Dezibel. Schreiben Sie ein Programm, das ermöglicht die Lautstärke in verschiedenen Entfernungen von der Sirene berechnen zu können.

   Verwenden Sie die Gleichung $L = L_0 - 20 \log (r/r_0)$, wobei
   L_0 die bekannte Lautstärke (138 db) eines Punktes nahe an der Sirene ist,
   r_0 den Abstand dieses Punktes von der Sirene (3,2 m) bezeichnet
   L die unbekannte Lautstärke an einem zweiten Punkt ist und
   r den Abstand in Meter dieses zweiten Punktes von der Schallquelle darstellt.
   Wie groß ist die Lautstärke 3 km von der Sirene entfernt ($r = 3$ km)?

   Eine mögliche Tastenfolge zur Lösung des Problems ist:
   [g] [P/R] [f] [LBL] [C] 3,2 [÷] [g] [LOG] 20 [×] [CHS] 138 [+] [g] [RTN] [g] [P/R]. Diese Anweisungen benötigen 15 Programmzeilen und 15 Bytes des Speichers. Allgemeiner kann dieses Problem gelöst werden, indem man die Werte von r_0 und L_0 aus den Speicherregistern zurückruft oder indem man L_0, r und r_0 vor der Ausführung in den Stack lädt: L_0 [ENTER] r [ENTER] r_0.

   (Ergebnis: für $r = 3$ km, $L = 78.5606$ db.)

2. Eine «typische» große Tomate wiegt etwa 200 Gramm. Davon sind 188 g (94%) Wasser. Ein Bauer versucht eine Tomate mit einem niedrigeren Wasseranteil zu züchten. Schreiben Sie ein Programm, das die prozentuale Änderung des Wasseranteils einer gegebenen Tomate, im Vergleich zu einer typischen Tomate, berechnet. Verwenden Sie eine Programmunterbrechung, um das Wassergewicht der neuen Tomate einzugeben.

   Wie groß ist die prozentuale Änderung des Wasserinhaltes einer 230 g Tomate, die 205 g Wasser enthält?

   Eine mögliche Tastenfolge zur Lösung dieser Aufgabe ist:
   [f] [LBL] [D] .94 [ENTER] [R/S] (Eingabe des Wassergewichtes der neuen Tomate) [ENTER] [R/S] (Eingabe des Gewichtes der neuen Tomate) [÷] [g] [Δ%] [g] [RTN]. Dieses Programm benötigt 11 Programmzeilen bzw. 11 Bytes des Speichers.

   (Ergebnis: für die 230 g Tomate beträgt die prozentuale Änderung des Wassergewichtes –5.1804%.)

Abschnitt 7

Programmkorrektur

Sie können aus den unterschiedlichsten Gründen ein bereits gespeichertes Programm ändern wollen; sei es, daß Sie eine Anweisung hinzufügen oder löschen wollen (z.B. [STO], [PSE], oder [R/S]), oder daß Sie Fehler in ihrem Programm gefunden haben. Der HP-15C ist darauf ausgelegt, den Prozess der Programmkorrektur möglichst einfach und komfortabel zu gestalten.

Das Handwerkszeug

Die Korrektur eines Programms besteht aus zwei Schritten: dem Auffinden der gewünschten Zeile, in der die Korrektur stattfinden soll, und dem Ausführen der Korrektur (Löschen oder Einfügen von Programmanweisungen).

Positionieren des Rechners auf eine Zeile des Programmspeichers

Die GO TO ([GTO]) Anweisung. Wenn Sie im Run- oder Programm-Modus [GTO] [CHS] *nnn* drücken, wird der Rechner auf die Zeile *nnn* des Programmspeichers positioniert. Diese Tastenfolge ist nicht programmierbar. Sie dient nur zum *manuellen* Auffinden einer speziellen Position im Programmspeicher. Die Zeilennummer muß eine dreistellige Zahl mit $000 \leq nnn \leq 448$ sein.

Die Einzelschrittanweisung [SST]. Mit Hilfe der Anweisung [SST] (*single step*) können Sie den Programmspeicher zeilenweise durchlaufen. Diese Anweisung ist nicht programmierbar.

Im Programm-Modus: [SST] bewegt den Rechner um eine Programmzeile weiter und zeigt diese Anweisung an. Halten Sie die Taste niedergedrückt, durchläuft der Rechner fortlaufend Zeile für Zeile des Programmspeichers.

Im Run-Modus: Durch Drücken von [SST] wird die momentane Programmzeile angezeigt, solange die Taste niedergehalten wird. Nach dem Loslassen der Taste wird die momentane Anweisung ausgeführt und der Rechner springt zur nächsten auszuführenden Programmzeile.

Die Rückschrittanweisung (BST). Durch Drücken von g BST (*back step*) im Programm- oder Run-Modus wird der Rechner im Programmspeicher um eine Zeile zurückpositioniert. Wenn Sie BST gedrückt halten, läuft der Rechner kontinuierlich rückwärts durch den Programmspeicher. Die Programmanweisungen werden dabei *nicht* ausgeführt. BST ist nicht programmierbar.

Löschen von Programmzeilen

Das Löschen von Programmanweisungen wird durch Drücken der Taste ← (*back arrow*) *im Programm-Modus* erreicht. Springen Sie zuerst in die Zeile, die Sie löschen wollen, und drücken Sie dann ←. Alle verbleibenden Zeilen werden neu durchnumeriert, um eine durchgehende Numerierung der Programmzeilen zu erhalten.

Das Drücken von ← in Run-Modus hat keine Auswirkungen auf den Programmspeicher, sondern wird zum Löschen der Anzeige verwendet (siehe Seite 21).

Einfügen von Programmzeilen

Wenn Sie zusätzliche Anweisungen in ein Programm einfügen wollen, müssen Sie den Rechner zunächst auf die der Einfügung vorangehenden Programmzeile positionieren. Jede anschließend eingegebene Anweisung wird *nach* der sich augenblicklich in der Anzeige befindlichen Zeile eingefügt. Um eine Anweisung zu ändern, löschen Sie diese zuerst, und fügen dann die neue Version ein.

Beispiele

Lassen Sie uns das Dosenprogramm aus Abschnitt 6, Seite 71, nochmals aufgreifen und einige Anweisungen annehmen. (Von dem im folgenden aufgelisteten Dosenprogramm wird angenommen, daß es sich bereits im Programmspeicher befindet und mit der Zeile 001 beginnt.)

Löschen von Programmanweisungen: Wenn die Werte der Gesamtgrundfläche, des Volumens und der Gesamtoberfläche nicht benötigt werden, können wir die Speicherregisteradditionen (Zeilen 007, 011 und 020) löschen.

Änderungen: Durch die obigen Löschungen werden die Register R_2, R_3 und R_4 nicht mehr benötigt und können anderweitig verwendet werden. So kann die Höhe *h* vor der Ausführung des Programms in R_1 abgespeichert werden, wodurch die Notwendigkeit der R/S Anweisung in Zeile 001 entfällt. Diese Anweisung ist durch eine RCL 1 Anweisung zu ersetzen. Um das Programm vollständig zu «säubern», soll auch die Zeile 006, STO 4, in STO 2 und die Zeile 016, RCL 4 in RCL 2 abgeändert werden (da auch die Register R_2 und R_3 nicht mehr benötigt werden).

Diese Korrekturen sind in dem Diagramm der nächsten Seite dargestellt.

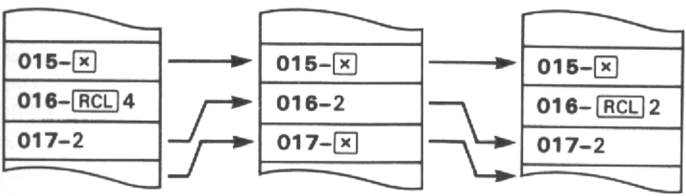
Ursprüngliche Version		Korrigierte Version	
006– STO 4	zu ändern	006– STO 2	neu
007– STO + 1	zu löschen	007– RCL 1	neu
008– R/S	zu ändern	008– ×	
009– ×		009– f PSE	
010– f PSE		010– RCL 0	
011– STO + 2	zu löschen	011– ÷	
012– RCL 0		012– 2	
013– ÷		013– ×	
014– 2		014– RCL 2	neu
015– ×		015– 2	
016– RCL 4	zu ändern	016– ×	
017– 2		017– +	
018– ×		018– g RTN	
019– +			
020– STO + 3	zu löschen		
021– g RTN			

Lassen sie uns mit der Korrektur am Ende des Programms beginnen. Auf diese Weise wird durch das Löschen von Programmanweisungen die Numerierung der vorangehenden Zeilen nicht geändert.

Tastenfolge	Anzeige	
g P/R	000–	Programm-Modus (angenommen der Rechner ist auf Zeile 000 positioniert).
GTO CHS 020 (oder verwenden Sie SST)	020–44,40, 3	Positioniert den Rechner auf Zeile 020.

Tastenfolge	**Anzeige**		
[←]	**019–**	**40**	Löschen von Zeile 020.
[g] [BST] (niederhalten)	**016–**	**45 4**	Als nächstes wird Zeile 016 ([RCL]) korrigiert.
[←]	**015–**	**20**	Löschen der Zeile 016.
[RCL] 2	**016–**	**45 2**	Abändern der Zeile 016 in [RCL] 2.
[GTO] [CHS] 11 (oder halten Sie [BST] gedrückt)	**011–**	**44,40, 2**	Positioniert den Rechner auf Zeile 011 ([STO] [+] 2).
[←]	**010–**	**42 31**	Löschen der Zeile 011.
[g] [BST]	**008–**	**31**	Halt! Eine Zeile zurück zu Zeile 008: [R/S].
[←]	**007–**	**44,40, 1**	Löschen von [R/S].
[RCL] 1	**008–**	**45 1**	Abändern der Zeile 008 in [RCL] 1.
[g] [BST]	**007–**	**44,40, 1**	Ein Schritt zurück zu Zeile 007.
[←]	**006–**	**44 4**	Löschen der Zeile 007 ([STO] [+] 1).
[+]	**005–**	**20**	Löschen der Zeile 006 ([STO] 4).
[STO] 2	**006–**	**44 2**	Abändern der Zeile 006 in [STO] 2.

Das Ersetzen einer Zeile vollzieht sich wie folgt:



Zusätzliche Informationen

Einzelschrittoperationen

Zeilenweise Ausführung eines Programms. Wollen Sie den Inhalt eines Programmes oder die Zeilennummer (Position) einer bestimmten Anweisung überprüfen, können Sie im *Programm-Modus* das Programm schrittweise durchlaufen.

Wenn Ihr Programm bei der Ausführung eine Fehlermeldung erzeugt, oder Sie entdecken auf andere Weise, daß Ihr Pogramm einen Fehler enthält, können Sie das Programm überprüfen, indem Sie es zeilenweise *ausführen* lassen. Dies geschieht durch das Drücken der Taste SST im *Run-Modus*.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
f CLEAR REG		Löschen der Speicherregister.
GTO A		Positioniert den Rechner auf die erste Zeile des Programms A.
8 STO 1	**8.0000**	Speichern einer Dosenhöhe.
2.5	**2.5**	Speichern eines Dosenradius.
SST (niedergedrückt)	**001– 42,21,11**	Tastencode für Zeile 001 (Label).
(loslassen)	**2.5000**	Ergebnis der Ausführung der Zeile 001.
SST	**002– 44 0** **2.5000**	STO 0. Ergebnis.
SST	**003– 43 11** **6.2500**	g x^2. Ergebnis.
SST	**004– 43 26** **3.1416**	g π. Ergebnis.
SST	**005– 20** **19.6350**	×. Ergebnis: die Grundfläche der Dose.

Die Funktion SST bewegt den Rechner nicht in unbelegte Bereiche des Programmspeichers. Stattdessen springt der Rechner zurück zu Zeile 000. (Im Run-Modus führt die Funktion SST jede Anweisung am Ende des Programmspeichers aus, wie z.B. RTN, GTO und GSB.)

Zeilenposition

Erinnern Sie sich, daß die Position des Rechners im Programmspeicher sich nicht verändert, wenn er ausgeschaltet wird, oder wenn zwischem dem Run-und Programm-Modus hin und her geschaltet wird. Nach dem Zurückkehren in den Programm-Modus befindet sich der Rechner in der gleichen Zeile, wie vor dem Verlassen des Programm-Modus. (Führen Sie ein Programm aus, das mit RTN endet, kehrt der Rechner zu Zeile 000 zurück.) Hat der Rechner sich selbst ausgeschaltet (nach einiger Zeit ohne Eingabe), müssen Sie ihn nur wieder einschalten, in den Programm-Modus zurückkehren (der Rechner «erwacht» immer im Run-Modus) und Sie sind in der Zeile, auf die der Rechner vor dem Ausschalten positioniert war.

Einfügen und Löschen

Nach dem Einfügen einer Zeile in ihr Programm wird im Display die Anweisung angezeigt, die Sie gerade eingegeben haben. Nach einem Löschen zeigt das Display die Zeile an, die der soeben gelöschten (jetzt nicht mehr existierenden) Zeile vorausgeht.

Wenn der gesamte verfügbare Speicherbereich belegt ist, akzeptiert der Rechner keine weiteren Programmanweisungen **Error 4**.

Initialisieren des Rechners

Die Inhalte der Speicherregister und die Einstellungen (Modi, Flags) des Rechners können ein Progamm beeinflussen, wenn das Programm diese Speicherregister verwendet oder irgendwie vom Status des Rechners abhängt. Ist der momentane Status für die erfolgreiche Ausführung des Programms unkorrekt, erhält man ein falsches Ergebnis. Es ist deshalb ratsam die Register zu löschen und den Rechner unmittelbar vor Ausführung des Programms oder innerhalb des Programms in die relevanten Modi zu schalten. Ein selbstinitialisierendes Programm ist sicherer, belegt aber mehr Programmzeilen.

Die folgenden Funktionen initialisieren den Rechner: [f] CLEAR [Σ], [f] CLEAR [PRGM], [f] CLEAR [REG], [g] [DEG], [g] [RAD], [g] [GRD], [g] [SF] und [g] [CF].

Übungsaufgaben

Es ist sinnvoll, die mehrfache Verwendung desselben Labels zu vermeiden. (Das sollte nicht sehr schwierig sein, da der HP-15C 25 verschiedene Labels zur Verfügung stellt.) Um sicher zu gehen kein Label nochmals verwendet zu haben, können Sie den Programmspeicher vorab löschen.

1. Das folgende Programm wird von dem Manager einer Bank verwendet, der damit die zukünftigen Erträge aus den Sparguthaben errechnet. Dazu wird die Formel $FV = PV\,(1 + i)^n$ benutzt. FV ist der zukünftige, PV der gegenwärtige Wert der Sparguthaben; i der Zinssatz pro Periode und n die Anzahl der Perioden. Geben Sie zuerst den Wert von PV in das Y-Register und danach in das X-Register ein. Zur Ausführung des Programms wird ein jährlicher Zinssatz von 7.5% (d.h. $i = 0{,}075$) angenommen.

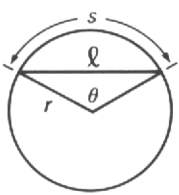
Tastenfolge	Anzeige	
f LBL · 1	001–42,21, .1	
f FIX 2	002–42, 7, 2	
1	003– 1	Verzinsung
·	004– 48	
0	005– 0	
7	006– 7	
5	007– 5	
x⪌y	008– 34	
y^x	009– 14	$(1+i)^n$.
×	010– 20	$PV(1+i)^n$.
g RTN	011– 43 32	

Laden Sie das Programm in den Rechner und berechnen Sie den Wert eines Sparguthabens von 1000 DM nach fünf Jahren. Berechnen Sie das gleiche für 2300 DM in vier Jahren. Beachten Sie, daß die Funktion GSB verwendet werden muß, um ein Programm zu starten, das ein numerisches Label hat. (Ergebnisse: 1435.63 DM; 3071.58 DM.)

Ändern Sie das Programm, so daß der jährliche Zinssatz 8% beträgt. Verwenden Sie das korrigierte Programm, um den Wert eines Sparguthabens von 500 DM nach vier Jahren bzw. von 1000 DM nach zehn Jahren zu berechnen. (Ergebnisse: 680.24 DM; 4317.85 DM.)

2. Schreiben Sie ein Programm, das die Länge einer Sekante über einem Winkel θ (in Grad) durch einen Kreis mit Radius r berechnet. Verwenden Sie dazu die folgende Formel:

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



Finden Sie ℓ für $\theta = 30°$ und $r = 25$.

(Ergebnis: 12,9410. Ein mögliches Programm ist: f LBL A, g DEG, f FIX 4, 2, ×, x⪌y, 2, ÷, SIN, ×, g RTN. Es wird angenommen, daß vor der Ausführung des Programms das X-Register mit r und das Y-Register mit θ geladen ist.)

Verändern Sie Ihr Programm so, daß auch die Länge s des von dem Winkel θ (*in Radiant*) ausgeschnittenen Kreisbogens berechnet und angezeigt wird. Verwenden Sie dazu die Formel

$$s = r\theta$$

Vervollständigen Sie die folgende Tabelle:

θ	r	ℓ	s
45°	50	?	?
90°	100	?	?
270°	100	?	?

(Ergebnis: 38.2683 und 39.2699; 141.4214 und 157.0796; 141.4214 und 471.2389.)

Eine mögliche Tastenfolge ist:

[f] [LBL] [A] [g] [DEG] [f] [FIX] 4 [STO] 0 2 [×] [x⇄y] [STO] 1
2 [÷] [SIN] [×] [f] [PSE] [f] [PSE] [RCL] 0 [RCL] 1 [f] [→RAD]
[×] [g] [RTN]).

Abschnitt 8

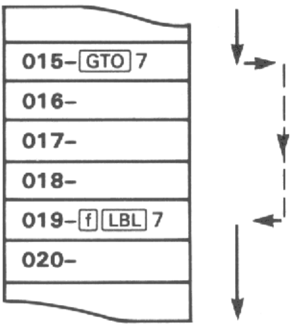
Programmverzweigungen

Die Anweisungen eines Programms werden im Normalfall der Reihe nach ausgeführt. Es ist jedoch oftmals wünschenswert, die weitere Ausführung eines Programms mit einer *anderen* Zeile, als der unmittelbar nächsten, fortzusetzen. Der HP-15C ermöglicht *einfache* oder vom Auftreten einer bestimmten *Bedingung* abhängige Verzweigungen. Durch einen Sprung in eine zurückliegende Zeile (die Zeilennummer ist kleiner als die der momentanen Zeile), läßt sich ein Programmteil mehr als einmal ausführen. Dieser Vorgang wird als *Programmschleife* bezeichnet.

Das Handwerkszeug

Programmverzweigungen

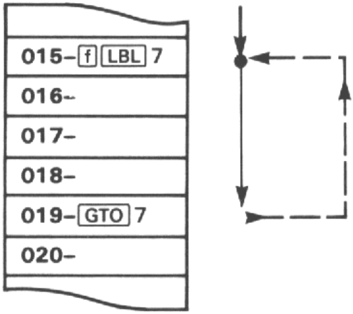
Die Go To (GTO) Anweisung. Einfache Verzweigungen – d.h. unbedingte Sprünge – werden durch die Anweisung GTO *Label* realisiert. In einem ablaufenden Programm bewirkt die Anweisung GTO die Übergabe der Programmausführung an den durch das Label spezifizierten Programmteil (und *nicht* an eine Zeilennummer).



Der Rechner sucht in Richtung der größeren Zeilennummern nach dem gewünschten Label. Falls nötig springt er am Programmspeicherende zurück zu Zeile 000, und setzt dort die Suche fort.

Programmschleifen. Wird durch eine Anweisung GTO ein Label in einer Zeile mit einer niedrigeren Nummer (d.h. in einer zurückliegenden Zeile) adressiert, werden die Anweisungen zwischen dem Label und der Anweisung GTO wiederholt ausgeführt – theoretisch unendlich oft.

Die Anzahl der Schleifendurchläufe kann durch einen bedingten Sprung oder durch eine [R/S] Anweisung innerhalb der Schleife bestimmt werden. Zusätzlich können Sie die Ausführung der Schleife durch Drücken einer beliebigen Taste, durch die die Programmausführung angehalten wird, abbrechen.



Vergleichsoperationen

Eine andere Möglichkeit, die Reihenfolge der Ausführung der Programmschritte zu verändern, ist eine *Vergleichsoperation*. Diese ja/nein (true/false) Abfrage vergleicht den Inhalt des X-Registers entweder mit Null oder mit dem Inhalt des Y-Registers. Der HP-15C enthält 12 verschiedene Vergleichsoperationen, zwei davon sind explizit auf dem Tastenfeld vorhanden, und 10 andere sind durch die Tastenfolge [g] [TEST] *n* zugänglich*.

1. Direkt verfügbar: [g] [$x \leq y$] und [g] [$x=0$].
2. Indirekt verfügbar: [g] [TEST] *n*.

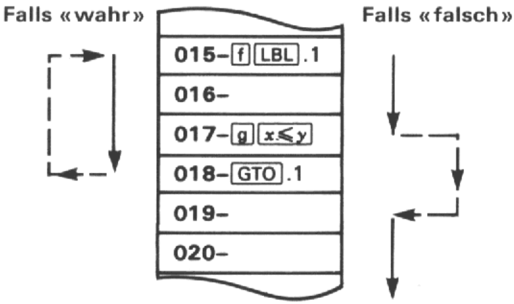
n	Abfrage	*n*	Abfrage
0	$x \neq 0$	5	$x = y$
1	$x > 0$	6	$x \neq y$
2	$x < 0$	7	$x > y$
3	$x \geq 0$	8	$x < y$
4	$x \leq 0$	9	$x \geq y$

Wenn in einem laufenden Programm das Ergebnis einer Vergleichsoperation «wahr» (true) lautet, wird die Programmausführung mit der auf die Vergleichs-

* Vier der Vergleichsoperationen können auch für komplexe Zahlen verwendet werden, siehe Abschnitt 11 auf Seite 132.

operation folgenden Anweisung fortgesetzt. Ist das Ergebnis einer Vergleichsoperation «falsch» (false), wird die auf die Abfrage *folgende Anweisung übersprungen* und die Ausführung mit der zweiten auf die Abfrage folgenden Zeile fortgesetzt. Oftmals wird unmittelbar nach einer Vergleichsoperation eine Anweisung [GTO] stehen, wodurch eine *bedingte Verzweigung* aufgebaut wird: d.h. die Anweisung [GTO] wird nur dann ausgeführt, wenn die Abfrage zu dem Ergebnis «true» geführt hat.

Programmausführung nach Abfrage



Flags

Eine zweite Gruppe von Abfrageoperationen besteht in der Abfrage des Zustands von Flags. Ein Flag ist ein Statusanzeiger, der entweder *gesetzt* (= true) oder «*nicht gesetzt*» (= false) ist. Einen nicht gesetzten Flag bezeichnet man auch als *gelöscht*. Im Verlauf einer Programmausführung kann ein Flag abgefragt werden, und eine Entscheidung davon abhängig gemacht werden, ob dieser Flag gesetzt ist oder nicht. Die Fortsetzung der Programmausführung nach Flagabfragen ist identisch mit der bei Vergleichsabfragen.

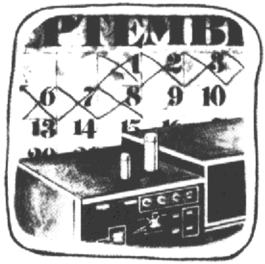
Der HP-15C verfügt über acht *Benutzer*-Flags (mit den Nummern 0 bis 7) und zwei *System*-Flags mit den Nummern 8 (Komplex-Modus) und 9 (Overflowbedingung). Die System-Flags werden später in diesem Abschnitt diskutiert. Alle Flags können gesetzt, gelöscht oder abgefragt werden, indem die entsprechende der folgenden Anweisungen verwendet wird:

- [g] [SF] *n*: der Flag mit der Nummer *n* (0 bis 9) wird *gesetzt*.
- [g] [CF] *n*: der Flag mit der Nummer *n* wird *gelöscht*.
- [g] [F?] *n*: der Flag mit der Nummer *n* wird *abgefragt*, ob er gesetzt oder gelöscht ist.

Ein gesetzter Flag *n* bleibt im gesetzten Zustand, bis er entweder mit der Funktion [CF] *n* gelöscht oder durch Löschen des Permanentspeichers in den ungesetzten Zustand gebracht wird.

Beispiele

Beispiel zu Programmsprüngen und Schleifen



In einem radiobiologischen Labor soll die Abnahme der Radioaktivität einer Probe von ^{131}I (Jod), einem radioaktiven Isotop, bestimmt werden. Schreiben Sie ein Programm, das die Radioaktivität der Probe in Zeitabstände von drei Tagen bestimmt, bis eine bestimmte Radioaktivität erreicht ist. Die Formel für N_t, der nach t Tagen verbliebenen Radioaktivität, ist

$$N_t = N_0 (2^{-t/k}),$$

wobei $k = 8$ Tage (die Halbwertszeit von ^{131}I) ist und N_0 den anfänglichen Wert der Radioaktivität darstellt.

Das folgende Programm verwendet eine Schleife, um die nach einer dreitägigen Zerfallszeit noch vorhandene Radioaktivität (gemessen in Millicurie mci) zu berechnen. Das Programm enthält eine Vergleichsoperation, die das Ergebnis überprüft und das Programm stoppt, sobald das Resultat einen vorgegebenen Wert unterschreitet.

Das Programm setzt voraus, daß t_1 – der erste Tag der Messung – in dem Register R_0 belassen, N_0 – die Anfangsmenge des Isotopes – in R_1 gespeichert wird und daß der Grenzwert der Radioaktivität sich im Register R_2 befindet.

Tastenfolge	Anzeige	
[g] [P/R]	000–	Programm-Modus.
[f] CLEAR [PRGM]	000–	(Nicht unbedingt nötig.)
[f] [LBL] [A]	001– 42,21,11	Jede Schleife kehrt zu dieser Zeile zurück.
[RCL] 0	002– 4 45 0	Ruft den augenblicklichen t-Wert, der sich in jeder Schleife ändert, zurück.
[f] [PSE]	003– 42 31	Stoppt die Ausführung, um t anzuzeigen.
8	004– 8	k.
[÷]	005– 10	
[CHS]	006– 16	$-t/k$.
2	007– 2	
[$x \gtrless y$]	008– 34	
[y^x]	009– 14	$2^{-t/k}$.

Tastenfolge	Anzeige	
RCL × 1	010– 45,20, 1	Multipliziert das Ergebnis der letzten Anweisung mit dem Inhalt des Registers von R_1 (N_0) und ergibt damit N_t, die Anzahl von mci des ^{131}I Isotopes nach t Tagen.
f PSE	011– 42 31	Stoppt die Programmausführung, um N_t anzuzeigen.
RCL 2	012– 45 2	Ruft den Grenzwert in das X-Register.
g TEST 9	013– 43,30, 9	$x \geqslant = y$? Vergleich des Grenzwertes (in X) und des Wertes von N_t (in Y).
g RTN	014– 43 32	Falls $x \geqslant = y$ stoppt das Programm.
3	015– 3	Falls $x \leqslant y$ wird die Programmausführung fortgesetzt.
STO + 0	016– 44,40, 0	Addiert 3 Tage zu t in R_0.
GTO A	017– 22 11	Springt zu «A» und wiederholt die Ausführung der Schleife zur Berechnung eines neuen N_t.

Beachten Sie, daß ohne die Anweisungen der Zeile 012 und 014, die Schleife unendlich oft durchlaufen werden würde (bis Sie die Ausführung manuell über das Tastenfeld abbrechen).

Führen Sie nun das Programm mit folgenden Werten aus: $t_1 = 2$ Tage, $N_0 = 100$ mci und einem Grenzwert von der Hälfte des N_0-Wertes.

Tastenfolge	Anzeige	
g P/R		Run-Modus
2 STO 0	2.0000	t_1.
100 STO 1	100.0000	N_0.
50 STO 2	50.0000	Grenzwert für N_t.
f A	2.0000	t_1.
	84.0896	N_1.
	5.0000	t_2.
	64.8420	N_2.
	8.0000	t_3.
	50.0000	N_3.
	50.0000	N_t Grenzwert erreicht; Ausführung wird beendet.

Beispiel zur Verwendung von Flags

Bei finanzmathematischen Berechnungen unterscheidet man zwei Zahlungsweisen: vorschüssig (d.h. zu Beginn eines festgelegten Zeitraums) und nachschüssig (d.h. am Ende eines festgelegten Zeitraums). Sie können innerhalb eines Programms zur Berechnung des gegenwärtigen Werts eines Darlehens oder einer Investition (bei gegebenem Zinssatz und Ein/Auszahlungsbetrag pro Periode) einen Flag als Statusanzeiger verwenden, der dem Programm mitteilt, ob die Zahlungen vor- oder nachschüssig erfolgen sollen.

Nehmen wir an, Sie wollen die Kosten für das Studium Ihrer Tochter planen. Sie rechnen mit einem Betrag von etwa 6000 DM im Jahr, also 500 DM pro Monat. Wenn Sie die monatlichen Zahlungen von einem Konto abbuchen lassen wollen, das jährlich mit 6% verzinst wird (die Zinsen werden jedoch monatlich abgerechnet, d.h. der jährliche Zins von 6% entspricht einem monatlichen Zinssatz von 0.5%), wieviel müssen Sie am Anfang der Universitätszeit einzahlen, um die monatlichen Zahlungen zu sichern?

Bei vorschüssiger Zahlungsweise
ist die folgende Formel anzuwenden:

$$V = -P \left[\frac{1-(1+i)^{-n}}{i} \right] (1+i)$$

Bei nachschüssiger Zahlungsweise gilt:

$$V = -P \left[\frac{1-(1+i)^{-n}}{i} \right]$$

Hierbei haben die einzelnen Variablen die folgende Bedeutung:

V ist der Gesamtbetrag, den Sie auf das Konto einzahlen müssen.

P ist die Höhe der *periodischen* (hier monatlichen) Zahlungen, die Sie von Ihrem Konto abbuchen wollen;

i ist der Zinssatz pro *Periode* (hier: die Periode, d.h. der Abrechnungszeitraum ist ein Monat, da der Zins monatlich gutgeschrieben wird); und

n ist die Anzahl von Perioden (Monate).

Das folgende Programm ist für beide Zahlungsweisen verwendbar. Es setzt voraus, daß vor der Ausführung des Programms der Wert für P in das Z-Register, der für n in das Y-Register und der Wert für i in das X-Register eingegeben wird.

Tastenfolge	Anzeige	
[g] [P/R]	000–	Programm-Modus.
[f] [LBL] [B]	001–42,21,12	Beginn bei «B», wenn die Zahlungen vorschüssig erfolgen.
[g] [CF] 0	002–43, 5, 0	Löscht Flag 0, um vorschüssige Zahlungsweise anzudeuten.
[GTO] 1	003– 22 1	Sprung zur Hauptroutine.
[f] [LBL] [E]	004–42,21,15	Beginn bei «E», wenn die Zahlungen nachschüssig erfolgen.
[g] [SF] 0	005–43, 4, 0	Setzt Flag 0, um nachschüssige Zahlungsweise anzudeuten.
[f] [LBL] 1	006–42,21, 1	Routine 1 (Hauptroutine).
[STO] 1	007– 44 1	Speichert i (aus dem X-Register).
1	008– 1	
[+]	009– 40	$(1+i)$.
[x⇄y]	010– 34	n in X; $(1+i)$ in Y.
[CHS]	011– 16	$-n$.
[y^x]	012– 14	$(1+i)^{-n}$.
[CHS]	013– 16	$-(1+i)^{-n}$.
1	014– 1	
[+]	015– 40	$1-(1+i)^{-n}$.
[RCL] [÷] 1	016–45,10, 1	Rückrufdivision mit R_1 (i), um $[1-(1+i)^{-n}]/i$ zu berechnen.
[×]	017– 20	Multiplikation mit P.
[g] [F?] 0	018–43, 6, 0	Flag 0 gesetzt?
[g] [RTN]	019– 43 32	Ende der Berechnung, wenn Flag 0 gesetzt (nachschüssige Zahlungsweise).
[RCL] 1	020– 45 1	Rückruf von i.
1	021– 1	
[+]	022– 40	$(1+i)$.
[×]	023– 20	Multiplikation mit zusätzlichem Faktor.
[g] [RTN]	024– 43 32	Ende der Berechnung, wenn Flag 0 gelöscht.

Berechnen Sie mit diesem Programm den Gesamtbetrag, den Sie benötigen, um 48 Monate lang 500 DM monatlich abheben zu können. Geben Sie den Zinssatz pro Periode in Form einer Dezimalzahl ein, d.h. 0.005 pro Monat. Berechnen Sie zuerst die Summe für den Fall, daß die Zahlungen am Anfang des Monats geleistet werden müssen (vorschüssige Zahlungsweise). Anschließend ist die benötigte Summe bei nachschüssiger Zahlungsweise zu berechnen.

Tastenfolge	Anzeige	
g P/R		Schaltet den Rechner in den Run-Modus.
500 ENTER	**500.0000**	Monatliche Zahlung.
48 ENTER	**48.0000**	Zahlungsperiode (4 Jahre × 12 Monate).
.005	**0.005**	Monatlicher Zinssatz in Form einer Dezimalzahl.
f B	**21,396.6097**	Einzahlungsbetrag bei vorschüssiger Zahlungsweise.

(Wiederholen Sie die Eingabe der Werte in den Stack.)

f E	**21,290.1589**	Einzahlungsbetrag bei nachschüssiger Zahlungsweise. (Der Unterschied zwischen diesem Betrag und dem erwarteten Gesamtbetrag der Studienkosten [24 000 DM] für die vier Jahre sind die auf den Einzahlungsbetrag aufgelaufenen Zinsen.)

Zusätzliche Informationen

Die Anweisung Go To

Im Gegensatz zu der nicht programmierbaren Tastenfolge GTO CHS *nnn*, kann die programmierbare Tastenfolge GTO *Label* nicht zu einer Verzweigung zu einer speziellen *Zeilennummer* verwendet werden, sondern nur zu einem Sprung zu einem *Programmlabel* (d.h. zu einer Zeile, die die Anweisung f LBL *Label* enthält).* Die Ausführung wird mit dieser Zeile fortgesetzt und kehrt nicht zur ursprünglichen Routine zurück, sofern dies nicht durch eine andere GTO Anweisung programmiert wird.

* Es ist jedoch möglich, mit Hilfe der indirekten Adressierung zu einer bestimmten *Zeilennummer* zu verzweigen; siehe Abschnitt 10.

Im Run-Modus (d.h. über das Tastenfeld) kann die Anweisung GTO *Label* Anweisung ebenfalls dazu verwendet werden, zu einer mit einem Label versehenen Zeile im Programmspeicher zu springen. Die adressierte Zeile wird jedoch nicht ausgeführt.

Programmschleifen

Programmschleifen sind eine Anwendung von Verzweigungsoperationen, bei denen eine GTO Anweisung zur wiederholten Ausführung einer Reihe von Programmbefehlen dient. Eine Programmschleife kann unendlich oft durchlaufen oder durch eine Vergleichsoperation beendet werden. Häufig wird eine Schleife dazu verwendet, die gleiche Berechnung mit verschiedenen Variablen wiederholt auszuführen. Zusätzlich kann ein Zähler in die Schleife eingebaut werden, der mit jeder Ausführung der Schleife erhöht wird. Dies ermöglicht die Anzahl der Schleifendurchläufe zu steuern. Der Wert dieses Zählers kann dann mit einer Vergleichsoperation überprüft werden, um zu entscheiden, ob die Schleife ein weiteres Mal ausgeführt werden soll oder nicht. (Diese Technik wird im Beispiel auf Seite 112 illustriert.)

Bedingte Verzweigungen

Die Anwendungen von bedingten Verzweigungen lassen sich in zwei Gruppen einteilen. Die erste Gruppe besteht in der Steuerung von Programmschleifen, wo entweder ein in der Schleife berechneter Wert oder ein Schleifenzähler abgefragt und in Abhängigkeit des Ergebnisses die Schleife verlassen oder fortgesetzt wird.

Die andere Hauptanwendung besteht darin, aus einer Anzahl von Möglichkeiten (Optionen) eine bestimmte auszuwählen. Betrachten Sie beispielsweise das Problem, das ein Kaufmann in Abhängigkeit des Umsatzes eine variable Bestellmenge aufgeben will. Sie könnten in diesem Fall ein Programm schreiben, das den Umsatz mit einem Testwert vergleicht und in Abhängigkeit des Ergebnisses die Höhe der Bestellmenge berechnet.

Vergleiche. Diese Abfragen vergleichen den Inhalt des X-Registers («x») entweder mit Null (z.B. $x=0$) oder mit «y», d.h. den Inhalt des Y-Registers (z.B. $x \leq y$). Bei einem x/y Vergleich muß daher der x- und y-Wert zuvor in das X-bzw.Y-Register eingegeben worden sein. Dazu ist es manchmal nötig, einen Vergleichswert abzuspeichern und ihn später zurückzurufen (d.h. in das X-Register zu bringen). Befindet sich ein Vergleichswert im Stack, so kann er gegebenenfalls mit den Anweisungen $x \gtrless y$, R↓ oder R↑ in das X-Register geschoben werden.

Vergleiche mit komplexen Zahlen und Matrix-Deskriptoren. In vier der bedingten Vergleiche können auch komplexe Zahlen und Matrizen Deskriptoren verwendet werden: [x=0], [TEST] 0 ($x \neq 0$), [TEST] 5 ($x = y$) und [TEST] 6 ($x \neq y$). (Weitere Erläuterungen dazu finden Sie in den Abschnitten 11 und 12.)

Flags

Mit Hilfe einer Vergleichsabfrage kann eine Option unter mehreren durch den Vergleich zweier Zahlen innerhalb eines Programms ausgewählt werden. Durch Flags können Sie die Auswahl einer Option von außen steuern.

Üblicherweise wird ein Flag in einem Programm zuerst gesetzt oder gelöscht, indem man einen anderen Startpunkt für das Programm wählt (durch unterschiedliche Label gekennzeichnet), je nachdem, welche Bedingung oder welchen Modus Sie verwenden wollen. (Beachten Sie hierzu das Beispiel auf Seite 95.)

Auf diese Weise kann ein Programm Eingaben in zwei verschiedene Modi wie z.B. Altgrad und Radiant berücksichtigen und das korrekte Ergebnis für den gewählten Modus berechnen. Sie können beispielsweise einen Flag setzen, wenn eine Konvertierung erfolgen soll, und den Flag löschen, wenn dies nicht notwendig ist.

Nehmen wir an, Sie wollen eine Gleichung lösen, die eine Temperaturangabe in Kelvin erfordert. Manchmal sind Ihre Werte jedoch in Grad Celsius angegeben. Sie können in diesem Fall ein Programm mit einem Flag verwenden, um Temperatureingaben in Kelvin und in Grad Celsius zu ermöglichen. Ein solches Programm könnte wie folgt aussehen:

[f] [LBL] [C]	Beginn des Programms bei «C» für eine Eingabe in Grad Celsius.
[g] [CF] 7	Löscht den Flag 7 (= false).
[GTO] 1	
[f] [LBL] [D]	Beginn des Programms bei «D» für eine Eingabe in Grad Kelvin.
[g] [SF] 7	Setzt den Flag 7 (= true).
[f] [LBL] 1	(Es wird angenommen, daß sich der Temperaturwert im X-Register befindet).
[g] [F?] 7	Abfrage von Flag 7 (überprüfen auf eine Celsius oder Kelvin Eingabe).
[GTO] 2	Ist Flag 7 gesetzt (Kelvineingabe), erfolgt ein Sprung zu der eigentlichen Berechnungsroutine, und die nächsten Schritte werden ausgelassen.

2 7 3 [+]	Ist Flag 7 nicht gesetzt (Celsiuseingabe) werden 273 zum Inhalt des X-Registers addiert, da °K = °C + 273 gilt.
[f] [LBL] 2	Die Berechnung setzt sich für beide Modi hier fort.
⋮	

Die System-Flags: Flag 8 und 9

Der Flag 8. Durch Setzen von Flag 8 wird der Komplex-Modus aktiviert (siehe Abschnitt 11). Die Statusanzeige **C** erscheint in der Anzeige. Wenn Sie den Rechner auf andere Weise in den Komplex-Modus schalten, wird Flag 8 automatisch gesetzt. Sie können den Komplex-Modus nur dadurch verlassen, indem Sie Flag 8 löschen; Flag 8 wird auf die gleiche Art und Weise gelöscht wie die anderen Flags auch.

Flag 9. Flag 9 wird bei Auftreten einer Overflowbedingung (siehe Seite 61) automatisch gesetzt und bedingt ein Blinken der Anzeige. Wenn Flag 9 während eines Programmablaufs gesetzt wird, beginnt die Anzeige erst nach dem Ende der Programmausführung zu blinken.

Flag 9 kann auf drei Arten gelöscht werden:

- Durch Drücken von [g] [CF] 9 (die übliche Methode einen Flag zu löschen).
- Durch Drücken von [←]. Dies löscht nur Flag 9 und beendet das Blinken, löscht aber nicht die Anzeige.
- Durch Ausschalten des Rechners. (Flag 9 wird nicht gelöscht, wenn sich der Rechner selbst ausschaltet.)

Wenn Sie Flag 9 manuell (durch [SF] 9) setzen, bewirkt dies ein Aufblinken der Anzeige, ungeachtet des Overflowstatus des Rechners. Wie oben beschrieben wird ein Programm jedoch erst vollständig abgearbeitet, bevor das Blinken der Anzeige beginnt. Flag 9 kann daher so in ein Programm eingebaut werden, daß er als ein visuelles Signal für das Auftreten einer ausgewählten Bedingung dient.

Unterprogramme

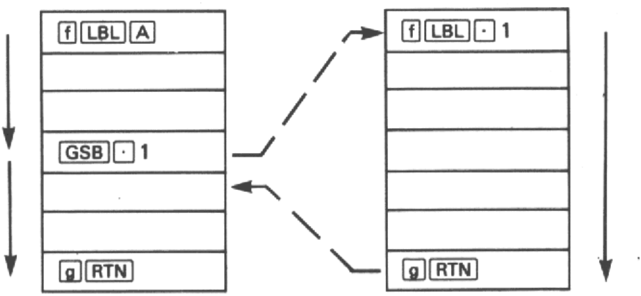
Wenn eine Folge von Anweisungen mehrmals innerhalb eines Programms verwendet werden soll, kann durch die Speicherung dieser Sequenz als Unterprogramm Platz gespart werden.

Das Handwerkszeug

Aufrufen eines Unterprogrammes und Rücksprung

Die Anweisung GSB (*go to subroutine*) wird in der gleichen Weise wie die Programmanweisung GTO ausgeführt, mit einem Unterschied: die Anweisung GSB setzt eine anstehende Rücksprungbedingung. Die Anweisung GSB *Label* bewirkt – genau wie GTO *Label** – einen Sprung zu der Zeile mit dem entsprechenden Label (A bis E; 0 bis 9 oder .0 bis .9). Von dort wird die weitere Ausführung fortgesetzt, *bis die erste Anweisung* RTN *auftritt. An diesem Punkt springt die Ausführung in die unmittelbar auf das letzte* GSB *folgende Zeile zurück*, und setzt sich von dort aus nach unten fort.

Ausführung eines Unterprogramms



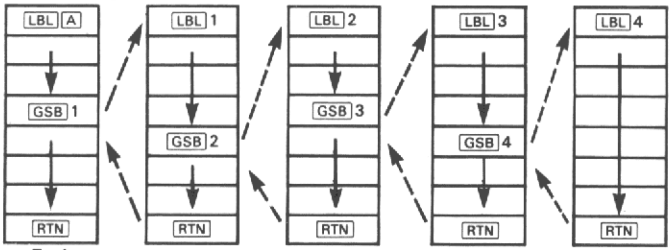
END
Übergabe der Kontrolle an Zeile 000 und Anhalten der Ausführung

RETURN
Übergabe der Kontrolle an die auf GSB · 1 folgende Zeile im Hauptprogramm

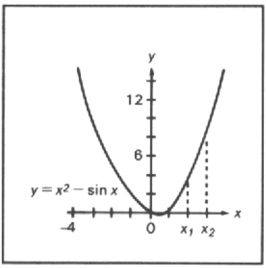
* Eine Anweisung ABS und GTO, gefolgt von einer Buchstabentaste, stellt eine abgekürzte Tastenfolge dar (siehe Seite 78).

Einschränkungen bei der Verwendung von Unterprogrammen

Ein Unterprogramm kann ein zweites Unterprogramm aufrufen und dieses wiederum ein drittes. Diese «Unterprogrammverschachtelung» – der Aufruf von Unterprogrammen innerhalb eines Unterprogrammes – ist auf sieben Ebenen beschränkt, wobei die Hauptprogrammebene nicht mitgezählt wird. Das folgende Diagramm verdeutlicht die Abarbeitung von verschachtelten Unterprogrammen:



Beispiele



Beispiel: Schreiben Sie ein Programm, das die Steigung einer Sekante durch die Punkte (x_1, y_1) und (x_2, y_2), die auf der nebenstehenden Kurve liegen, berechnet. Die Gleichung der Kurve sei $y = x^2 - \sin x$ (x in Radiant).

Die Steigung berechnet sich wie folgt:

$$\frac{y_2 - y_1}{x_2 - x_1} \text{ oder } \frac{(x_2{}^2 - \sin x_2) - (x_1{}^2 - \sin x_1)}{x_2 - x_1}$$

Für vorgegebene x_1 und y_2 Werte erfordert die Lösung, daß die Funktion y zweimal ausgewertet wird; einmal zur Berechnung von y_1 und einmal zur Berechnung von y_2. Da also dieselbe Rechnung zweimal ausgeführt werden muß, spart es Platz im Programm, y mit Hilfe eines Unterprogramms zu berechnen.

Das folgende Programm nimmt an, daß der Wert x_1 in das Y-Register und der Wert y_2 in das X-Register eingegeben wurde.

Hauptprogramm

g P/R	
f CLEAR PRGM	(Nicht programmierbar.)
000–	
001– f LBL 9	Beginn Hauptprogramm.
002– g RAD	Radiant-Modus.
003– STO 0	Speichert x_2 in R_0.
004– $x \gtrless y$	Bringt x_1 nach X, x_2 nach Y.
005– STO − 0	Speichert (x_2-x_1) in R_0.
006– GSB .3	Aufruf von Unterprogramm «.3» mit x_1.
	Rücksprung von Unterprogramm «.3».
007– CHS	$-y_1$.
008– $x \gtrless y$	Bringt x_2 ins X-Register.
009– GSB .3	Aufruf von Unterprogramm «.3» mit x_2.
	Rücksprung von Unterprogramm «.3».
010– +	y_2-y_1.
011– RCL ÷ 0	Rückruf von (x_2-x_1) aus R_0 und Berechnung von $(y_2-y_1)/(x_2-x_1)$.
012– g RTN	Programmende (Rücksprung zu Zeile 000).
Unterprogramm	
013– f LBL .3	Beginn des Unterprogramms «.3».
014– g x^2	x^2.
015– g LSTx	Rückruf von x.
016– SIN	Sinus von x.
017– −	$x^2-\sin x$; der Funktionswert.
018– g RTN	Rücksprung ins Hauptprogramm.

Berechnen Sie die Steigung für die folgenden x_1 und x_2 Werte: 0.52, 1.25; –1, 1; 0.81, 0.98. Denken Sie daran, GSB 9 zu verwenden (anstelle von f 9), da Sie eine Routine mit einer Ziffer als Label adressieren.

Ergebnisse: 1.1507; –0.8415; 1.1652.

Beispiel: Verschachtelungen: Das folgende Unterprogramm mit dem Label «.4» berechnet den Wert des Ausdrucks $\sqrt{x^2+y^2+z^2+t^2}$ als einen Teil einer Berechnung in einem größeren Programm.

Dieses Unterprogramm ruft ein weiteres Unterprogramm mit dem Label «.5» auf, um das wiederholte Quadrieren auszuführen.

Vor der Ausführung des Programms sind die Variablen t, z, y und x in die T-, Z-, Y- und X-Register zu laden.

Tastenfolge		
f LBL .4		Beginn des «Haupt»-Unterprogramms. x^2.
g x^2		
GSB .5		Berechnet y^2 und $x^2 + y^2$.
GSB .5	1	Berechnet z^2 und $x^2 + y^2 + z^2$.
GSB .5	2	Berechnet t^2 und $x^2 + y^2 + z^2 + t^2$.
$\sqrt{x}$	3	$\sqrt{x^2 + y^2 + z^2 + t^2}$
g RTN		Ende des «Haupt»-Unterprogramms; Rücksprung zum Hauptprogramm.
f LBL .5		Beginn des verschachtelten Unterprogramms.
$x \gtrless y$		
g x^2		Berechnet ein Quadrat und addiert den Wert zur momentanen Quadratsumme.
+		
g RTN		Ende des verschachtelten Unterprogramms; Rücksprung zum «Haupt»-Unterprogramm.

Wenn Sie dieses Unterprogramm allein (mit dem eingeschachtelten Unterprogramm) mit den Werten $x = 4.3$, $y = 7.9$, $z = 1.3$ und $t = 8.0$ ausführen, ergibt sich nach dem Drücken von GSB .4 das Resultat 12.1074.

Zusätzliche Informationen

Rücksprung aus einem Unterprogramm

Eine *anstehende Rücksprung* Bedingung bedeutet, daß das erste [RTN] nach einer [GSB] Anweisung einen Rücksprung in die auf die [GSB] Anweisung unmittelbar folgende Zeile bewirkt. Der Rücksprung findet also *nicht* in die Zeile 000 statt. Dadurch kann ein Unterprogramm in verschiedenen Teilen des Programms wiederverwendet werden: die Ausführung kehrt immer zu dem Punkt zurück, an dem die Programmverzweigung stattgefunden hat, auch wenn diese Zeile eine andere als bei der letzten Ausführung des Unterprogramms ist. Der einzige Unterschied zwischen den Anweisungen [GSB] und [GTO] ist somit die Stelle im Programm, an der die Ausführung *nach* der Anweisung [RTN] fortgesetzt wird.

Verschachtelte Unterprogramme

Bei dem Versuch, ein Unterprogramm aufzurufen, das tiefer als sieben Ebenen verschachtelt ist, unterbricht der Rechner die Ausführung und meldet **Error 5**, sobald er die [GSB] Anweisung auf der achten Ebene findet. Beachten Sie, daß die Anzahl der Unterprogramme nur durch die Größe des verfügbaren Programmspeichers beschränkt ist.

Abschnitt 10

Indexregister und Schleifensteuerung

Das Indexregister (R_I) ist ein sehr effektives Hilfsmittel für die Programmierung Ihres HP-15C. Zusätzlich zum einfachen Speichern und Rückrufen von Daten, kann der Inhalt des Indexregisters auch als Kontrollwert dienen, um

- Programmschleifen zu steuern,
- Speicherregister indirekt zu adressieren, einschließlich die Speicher oberhalb von $R_{.9}$ (R_{19}),
- indirekt sowohl zu Programmzeilen*nummern* als auch zu Labels zu verzweigen,
- indirekt das Anzeigeformat zu steuern und
- indirekte Flagkontrolloperationen auszuführen.

Die Tasten [I] und [(i)]

Direkte und indirekte Datenspeicherung mit dem Indexregister

Das Indexregister ist ein Datenspeicherregister, das direkt mit der Taste [I], oder indirekt mit der Taste [(i)]*, verwendet werden kann. Es ist sehr wichtig, daß Sie sich den Unterschied in der Verwendung dieser beiden Tasten deutlich klarmachen.

[I]	[(i)]
Die Funktion [I] vewendet den *Inhalt* des Indexregisters *selbst*.	Die Funktion [(i)] verwendet den Betrag des ganzzahligen Anteils des Wertes im Indexregister, um ein anderes Datenspeicherregister zu adressieren. Dies bezeichnet man als *indirekte Adressierung*.

* Beachten Sie, daß die Matrixfunktionen und die komplexen Funktionen die Tasten [I] und [(i)] auch verwenden, allerdings zu einem anderen Zweck (siehe Abschnitte 11 und 12).

Indirekte Programmkontrolle mit dem Indexregister

Die Taste [I] wird für alle Arten der indirekten Programmkontrolle verwendet *außer* zur indirekten Adressierung eines Registers. Daher wird die Taste [I] (und nicht [(i)]) für indirekte Programmverzweigungen, indirekte Kontrolle des Anzeigeformats und indirekte Flagkontrolle verwendet.

Steuerung einer Programmschleife

Das Zählen von Durchläufen durch eine Programmschleife und ihre Steuerung kann mit *jedem Speicherregister* ausgeführt werden: R_0 bis R_9, $R_{.0}$ bis $R_{.9}$ und dem Indexregister ([I]). Eine Schleifensteuerung kann auch *indirekt* mit der [(i)] Funktion ausgeführt werden.

Das Handwerkszeug

Die beiden Tasten [I] und [(i)] können in einer verkürzten Tastenfolge verwendet werden, in der die Vorwahltaste [f] wegfallen kann (siehe Seite 78).

Speicherung und Rückruf mit dem Indexregister

Direkte Operation. [STO] [I] und [RCL] [I]. Die Speicherung und der Rückruf von Werten zwischen dem X-Register und dem Indexregister geschieht in der gleichen Weise wie bei den übrigen Datenspeicherregistern (siehe Seite 42).

Indirekte Operation. Die Anweisung [STO] (oder [RCL]) [(i)] speichert einen Wert in dasjenige Datenspeicherregister (oder ruft einen Wert von dort zurück), das durch den ganzzahligen Anteil des Wertes (0 bis 65) im Indexregister adressiert wird. Dies wird in den folgenden Tabellen veranschaulicht.

Indirekte Adressierung

Inhalt von R_I:	Durch [(i)] adressiertes Register:	Übergabe der Kontrolle durch [GTO] [I] oder [GSB] an:*
± 0	R_0	[f] [LBL] 0
⋮	⋮	⋮
9	R_9	[f] [LBL] 9
10	$R_{.0}$	" " .0
11	$R_{.1}$	" " .1
⋮	⋮	⋮
19	$R_{.9}$	[f] [LBL] .9
20	R_{20}	" " [A]

*Nur für $R_I \geqslant 0$.

(Fortsetzung nächste Seite)

Indirekte Adressierung

Inhalt von R_I:	Durch [(i)] adressiertes Register:	Übergabe der Kontrolle durch [GTO] [I] oder [GSB] [I] an:*
21	R_{21}	[f] [LBL] [B]
22	R_{22}	" " [C]
23	R_{23}	" " [D]
24	R_{24}	" " [E]
⋮	⋮	—
65	R_{65}	—

*Nur für $R_I \geq 0$.

Indexregister-Arithmetik

Direkte Operation. [STO] oder [RCL] {[+], [−], [×], [÷]} [I]. Die Speicher- oder Rückruf-Arithmetik unter Verwendung des Indexregisters erfolgt in der gleichen Weise, wie bei den anderen Datenspeicherregistern (siehe Seite 43).

Indirekte Operation. Die Anweisung [STO] oder [RCL] {[+], [−], [×], [÷]} [(i)] führt eine Speicher- oder Rückruf-Arithmetik mit dem Inhalt des Datenregisters aus, das durch den ganzzahligen Anteil des Wertes (0 bis 65) im Indexregister adressiert wird (siehe obige Tabelle).

Vertauschen des Inhalts des X-Registers.

Direkte Operation. Die Anweisung [f] [x≷] [I] vertauscht den Inhalt des X-Registers und des Indexregisters. (Dies geschieht in der gleichen Weise, wie durch Drücken von [x≷] *n* die Inhalte des X- Registers und einem der Register R_0 bis $R_{.9}$ [je nach *n*] vertauscht würden.)

Indirekte Operation. Die Anweisung [f] [x≷] [(i)] vertauscht den Inhalt des X-Registers und desjenigen Datenspeicherregisters, das durch den Wert (0 bis 65) im Indexregister adressiert wird (siehe obige Tabelle).

Indirekte Verzweigungen zu einem Label oder einer Zeilennummer

Im Gegensatz zu [(i)] kann die Taste [I] zu *indirekten* Programmverzweigungen ([GTO] [I]) und zum *indirekten* Aufruf von Unterprogrammen ([GSB] [I]) verwendet werden. (Diese Operationen verwenden nur den ganzzahligen Anteil des Wertes im Indexregister R_I.) Die Taste [(i)] kann nur zur indirekten *Adressierung* eines Datenspeicherregisters verwendet werden.

Verzweigung zu einem Label. Ist der Inhalt von R_I eine *positive* Zahl, bewirken die Anweisungen GTO I und GSB I einen Sprung der Programmausführung zu dem *Label*, das dem Wert im Indexregister zugeordnet ist (siehe Tabelle auf Seite 107).

Nehmen wir beispielsweise an, daß das Indexregister den Wert 20.00500 enthält. Die Anweisung GTO I bewirkt in diesem Fall, daß die Programmausführung mit der Zeile f LBL A fortgesetzt wird (siehe Tabelle auf Seite 107).

Verzweigung zu einer Zeilennummer. Wenn R_I eine *negative* Zahl enthält, bewirken die Anweisungen GTO I und GSB I einen Sprung zu derjenigen *Zeilennummer*, die durch den Betrag des ganzzahligen Anteils des Wertes in R_I spezifiziert wird.

Wenn sich beispielsweise im Indexregister der Wert –20.00500 befindet, wird nach GTO I die Programmausführung mit Zeile 020 fortgesetzt.

Indirekte Flagkontrolle mittels I

Die Anweisungen SF I, CF I oder F? I setzen, löschen oder prüfen den durch ganzzahligen Anteil des Werts im Indexregister adressierten Flag.

Indirekte Kontrolle des Anzeigeformats mittels I

Die Anweisungen f FIX I, f SCI I und f ENG I setzen das Anzeigeformat in der üblichen Weise (siehe Seiten 58–59). Die Anzahl der Stellen wird dabei durch den ganzzahligen Anteil der Zahl im X-Register bestimmt, der zwischen 0 und 9* liegen muß.

Schleifensteuerung mit ISG und DSE

Die Funktion ISG (*Inkrement und Sprung, wenn größer*) und DSE (*Dekrement und Sprung, wenn kleiner oder gleich*) steuern die Ausführung einer Schleife, indem in einem gegebenen Register eine *Schleifenkontrollzahl* gespeichert und verändert wird. Ob die Programmausführung eine Zeile überspringt oder nicht, hängt von diesem Wert ab.

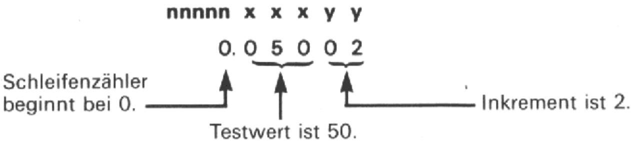
Die Tastenfolge für die Zuordnung einer Schleifenkontrollzahl zu einem Register ist: f {ISG, DSE} *Registeradresse*. Die Registeradresse kann 0 bis 9, .0 bis .9, I oder (i) sein.

Die Schleifenkontrollzahl. Die Schleifenkontrollzahl ist nach folgendem Schema aufgebaut:

	± **nnnnn**	Momentaner Wert des Zählers.
nnnnn.xxxyy wobei	**xxx**	Testwert.
	yy	Inkrement oder Dekrement.

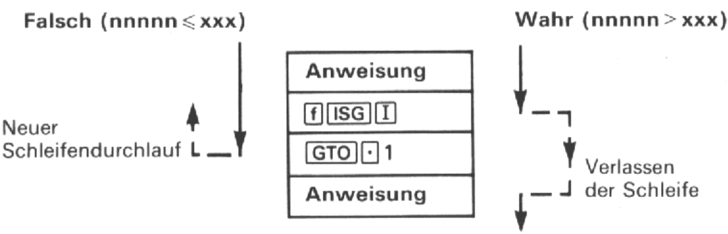
* Außer bei der Verwendung von ∫ (siehe Abschnitt 14).

Die Zahl 0.05002 wird beispielsweise als Schleifenkontrollzahl wie folgt interpretiert:

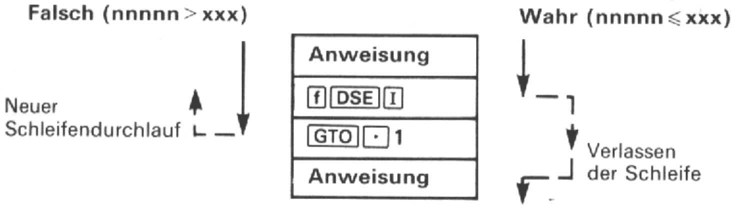


Ablauf der Operationen [ISG] und [DSE]. Bei jeder Ausführung von [ISG] oder [DSE] wird **nnnnn** (der ganzzahlige Anteil der Schleifenkontrollzahl) erhöht oder erniedrigt. Dies bewirkt eine Zählung der ausgeführten Schleifendurchläufe. Als nächstes wird **nnnnn** mit dem zuvor festgelegten Testwert **xxx** verglichen. Der Rechner verläßt die Schleife und überspringt die nächste Zeile, *wenn* **nnnnn** entweder größer ([ISG]) oder kleiner gleich ([DSE]) dem Testwert (**xxx**) ist. Der Betrag, um den **nnnnn** bei jedem Schleifendurchgang erhöht oder erniedrigt wird, ist durch **yy** gegeben.

Bei diesen Funktionen lautet die allgemeine Regel (im Gegensatz zu den anderen Abfrageoperationen): «überspringen, falls wahr».



[ISG]. Gegeben sei **nnnnn.xxxyy**. Bei jeder Ausführung der Schleife geschieht folgendes: der **nnnnn**-Wert wird zum Wert **nnnnn + yy**; dieser neue **nnnnn**-Wert wird mit **xxx** verglichen, und die nächste Zeile dann übersprungen, falls **nnnnn** größer als **xxx** ist. Die Schleife wird somit verlassen, sobald der **nnnnn**-Wert größer als der **xxx**-Wert wird.



DSE. Gegeben sei **nnnnn.xxxyy**. Bei jeder Ausführung der Schleife geschieht folgendes: der **nnnnn**-Wert wird zu **nnnnn-yy**, d.h. von **nnnnn** wird **yy** subtrahiert, und ergibt den neuen **nnnnn**-Wert. Dieser wird mit **xxx** verglichen und die nächste Zeile dann übersprungen, falls der neue **nnnnn**-Wert die Bedingung **nnnnn** ⩽ **xxx** erfüllt. Dies ermöglicht das Verlassen der Schleife, sobald die Anzahl der Schleifendurchläufe kleiner oder gleich einem Testwert ist.

Die folgende Tabelle zeigt als beispielhaft die Veränderung der Schleifenkontrollzahl nach jedem Durchlauf einer Schleife:

Iterationen

Operation	0	1	2	3	4
ISG	0.00602	2.00602	4.00602	6.00602	8.00602 (Überspringen der nächsten Zeile)
DSE	6.00002	4.00002	2.00002	0.00002 (Überspringen der nächsten Zeile)	

Beispiele

Beispiele zu Registeroperationen

Speicherung und Rückruf

Tastenfolge	Anzeige	
f CLEAR REG		Löscht alle Speicherregister.
12.3456	**12.3456**	
STO I	**12.3456**	Speichert R_I.
7 $\sqrt{x}$	**2.6458**	
STO (i)	**2.6458**	Speichert diesen Wert in $R_{.2}$ durch indirekte Adressierung ($R_I = 12.3456$).
RCL I	**12.3456**	Zurückrufen des Inhalts von R_I.

Tastenfolge	Anzeige	
RCL (i)	2.6458	Ruft den Inhalt von $R_{.2}$ indirekt zurück.
f x≷ .2	2.6458	Probe: gleicher Inhalt wird durch direkte Adressierung von $R_{.2}$ zurückgerufen.

Vertauschen des Inhalts des X-Registers

Tastenfolge	Anzeige	
f x≷ I	12.3456	Vertauscht den Inhalt des Registers R_I und des X-Registers.
RCL I	2.6458	Augenblicklicher Inhalt des Indexregisters.
f x≷ (i)	0.0000	Vertauscht den Inhalt des Registers R_2 (der Null ist) und X-Registers.
RCL (i)	2.6458	
f x≷ 2	2.6458	Zur Probe: direkte Adressierung des Registers R_2.

Speicherregister-Arithmetik

Tastenfolge	Anzeige	
10 STO + I	10.0000	Addiert 10 zum Inhalt von R_I.
RCL I	12.6458	Neuer Inhalt von R_I (= alter Inhalt + 10).
g π STO ÷ (i)	3.1416	Dividiert den Inhalt von $R_{.2}$ durch die Kreiskonstante π.
RCL (i)	0.8422	Neuer Inhalt von $R_{.2}$.
f x≷ .2	0.8422	Zur Probe: direktes Adressieren des Registers $R_{.2}$.

Beispiel: Schleifensteuerung mit DSE

Erinnern Sie sich noch an das Beispiel aus Abschnitt 8, Seite 93, das zur Berechnung eines radioaktiven Zerfalls eine Programmschleife verwendete? Dieses Programm benutzte eine Vergleichsoperation ($x \geq y$?), um die Schleife zu verlassen, sobald das Ergebnis einen vorgegebenen Grenzwert (50) überschritten hat. Wie wir nun in diesem Abschnitt gesehen haben, können Sie die Ausführung der Schleife auch über einen gespeicherten Schleifenzähler steuern, dessen Wert ständig mit den Funktionen ISG oder DSE überwacht wird.

Lassen Sie uns das Programm zur Berechnung des radioaktiven Zerfalls nochmals ausführen, wobei wir dieses Mal die Ausführung der Schleife auf drei Durchgänge beschränken, anstelle von der Vorgabe eines Grenzwertes in der ursprünglichen Version. Wir benutzen die Funktion DSE und speichern die Schleifenkontrollzahl in R_2: $\quad 3.\underbrace{000}\underbrace{01}.$

Anfangswert des Schleifenzählers — Testwert (Endwert) — Inkrementwert

Ändern Sie das Programm wie im folgenden gezeigt. (Wir nehmen dabei an, daß sich das Programm noch im Programmspeicher befindet.) Der Schleifenzähler wird in Register R_2 und eine Zeilennummer im Indexregister gespeichert.

Tastenfolge	Anzeige	
g P/R	**000–**	Programm-Modus.
GTO CHS 013	**013– 43,30, 9**	Zweite Abfrage innerhalb der Schleife.
← ←	**011– 42 31**	Löscht die Zeilen 013 und 012.
f DSE 2	**012–42, 5, 2**	Einfügen der DSE Anweisung (der Zähler ist in R_2 gespeichert).
GTO I	**013– 22 25**	Sprung zu der spezifizierten Zeile (015).

Wenn der Schleifenzähler (gespeichert in R_2) den Wert Null erreicht hat, wird die Zeile 013 übersprungen und zu Zeile 014 – eine RTN Anweisung – verzweigt, was ein Anhalten der Programmausführung bedingt. Solange der Schleifenzähler noch nicht den Wert Null erreicht hat, wird die Ausführung des Programms mit Zeile 015 fortgesetzt. Diese bewirkt einen Sprung in Zeile 015 und einen erneuten Schleifendurchlauf.

Um die neue Version des Programms auszuführen, geben Sie t_1 (erster Tag) in das Register R_0 , N_0 (anfängliche Isotopenmenge) in das Register R_1, den Schleifenzähler in das Register R_2 und die Nummer der Zeile, in die verzweigt werden soll, in das Indexregister ein.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
2 STO 0	**2.0000**	t_1.
100 STO 1	**100.0000**	N_0.
3.00001 STO 2	**3.0000**	Schleifenzähler. (Diese Anweisung könnte auch Teil des Programms sein.)

Tastenfolge	Anzeige	
15 [CHS] [STO] [I]	–15.0000	Nummer der Zeile, in die verzweigt werden soll.
[f] [A]	2.0000	Startet das Programm; Schleifenzähler hat den Wert 3.
	84.0896	
	5.0000	Schleifenzähler gleich 2.
	64.8420	
	8.0000	Schleifenzähler gleich 1.
	50.0000	
	50.0000	Schleifenzähler gleich 0; Ende des Programms.

Beispiel: Steuerung des Anzeigeformats

Das folgende Programm zeigt mittels einer [DSE] Schleife eine Zahl im Festkommaformat mit abnehmender Anzahl von Dezimalstellen an.

Tastenfolge	
[g] [P/R]	
[f] CLEAR [PRGM]	
[f] [LBL] [B]	
9	**nnnnn** = 9. Daher ist **xxx** = 0 und gemäß Voreinstellung gilt **yy** = 1 (**yy** kann nicht Null sein).
[STO] [I]	
[f] [LBL] 0	
[f] [FIX] [I]	
[RCL] [I]	
[f] [PSE]	Zeigt den momentanen Wert von **nnnnn** an.
[f] [DSE] [I]	**Der Wert in R_I** wird verkleinert und abgefragt. Die nächste Zeile wird übersprungen, falls **nnnnn** ≤ Testwert gilt.
[GTO] 0	Setzt die Ausführung der Schleife fort, falls **nnnnn** > Testwert (0) gilt.
[g] [TEST] 1 [GTO] 0	Prüft ob der **angezeigte Wert** größer als Null ist; die Schleife wird fortgesetzt, wenn **nnnnn** den Wert 0 erreicht hat. In der Anzeige erscheint jedoch nur noch der Wert 1.0.
[g] [RTN]	

Starten Sie das Programm, um die mögliche Festkommaformate des HP-15C anzeigen zu lassen:

Tastenfolge	**Anzeige**	
g P/R		Run-Modus.
f B	**9.000000000**	
	8.00000000	
	7.0000000	
	6.000000	
	5.00000	
	4.0000	
	3.000	
	2.00	
	1.0	
	0.	Anzeige während der f PSE Anweisung.
	0.	Anzeige bei Programmende.

Zusätzliche Informationen

Verwendung des Werts im Indexregister

Jede im Indexregister gespeicherte Zahl kann auf drei Arten benutzt werden:

- Verwendung von I als normales Datenspeicherregister. Der Inhalt von R_I kann direkt benutzt werden, d.h. gespeichert, zurückgerufen, vertauscht, zu einem anderen Wert addiert werden usw.
- Verwendung von I als Kontrollwert. Der Betrag des ganzzahligen Anteils des Inhalts von R_I spielt im Vergleich zum dezimalen Anteil eine besondere Rolle. Für indirekte Verzweigungen, die Flagkontrolle und die Steuerung des Anzeigeformats durch die Funktion I wird nur der ganzzahlige Anteil verwendet. Der dezimale Anteil dient nur zur Schleifensteuerung*.
- Verwendung von (i) als Verweis auf den Inhalt eines anderen Speicherregisters. Die Taste (i) benutzt das System der indirekten Adressierung, wie in der Tabelle auf Seite 107 und 108 gezeigt. (Der Inhalt des dadurch spezifizierten Registers kann dann als Schleifenkontrollzahl in der oben beschriebenen Weise verwendet werden.)

* Dies gilt auch für den Wert *irgendeines* Speicherregisters, das zur indirekten Schleifenkontrolle verwendet wird.

[ISG] und [DSE]

Der ganzzahlige Anteil (d.h. der Wert des Schleifenzählers) einer gespeicherten Kontrollzahl kann aus bis zu fünf Ziffern bestehen (**nnnnn.xxxyy**). Bei fehlender Angabe ist der Zähler (**nnnnn**) auf Null voreingestellt.

Der **xxx**-Wert, die ersten drei Dezimalstellen der Kontrollzahl, muß dreistellig spezifiziert werden. («5» muß beispielsweise in der Form «005» eingegeben werden.) Voreinstellung für **xxx** ist Null. Bei jeder Ausführung einer [ISG] oder [DSE] Anweisung wird **nnnnn** intern mit **xxx**, dem Endwert von **nnnnn** nach der Inkrementierung bzw. Dekrementierung, verglichen.

Der **yy** Wert muß zweistellig und ungleich Null sein. *Bei fehlender Angabe gilt als Ersatzwert* **01**. Bei jeder Ausführung von [ISG] oder [DSE] wird **nnnnn** um den Betrag von **yy** erhöht oder erniedrigt. **yy** und **xxx** sind Referenzwerte, die sich bei der Ausführung der Schleife nicht verändern.

Indirekte Anzeigesteuerung

Obwohl Sie das Indexregister auch zur manuellen Steuerung (d.h. über das Tastenfeld) der Anzeige verwenden können, wird diese Funktion am häufigsten bei der Programmierung benutzt. Diese Eigenschaft des Rechners ist besonders wertvoll bei der Anwendung von [$\int_y^x$], wo die Genauigkeit durch die Spezifikation der anzuzeigenden Ziffern festgesetzt werden kann (siehe Abschnitt 14).

Es gibt jedoch gewisse Begrenzungen bei der Anzeige, die zu beachten sind. Erinnern Sie sich, daß jede Anzeigeformatfunktion nur die Anzahl von Dezimalstellen, auf die das Ergebnis gerundet ist, verändert. Im Speicher des Rechners wird jede Zahl jedoch in der wissenschaftlichen Notation mit einem zehnstelligen Exponenten abgelegt.

Der ganzzahlige Anteil der Zahl im Indexregister spezifiziert die Anzahl von Dezimalstellen, auf die die Anzeige gerundet wird. Eine Zahl kleiner als Null wird durch Null ersetzt ([FIX] Format ohne Dezimalstellen). Eine Zahl größer als 9 wird durch 9 ersetzt ([FIX] Format mit 9 Dezimalstellen)*.

---

* Beachten Sie, daß in dem [SCI] und [ENG] Modus die maximale Mantisse in die Anzeige eine sechsstellige Zahl und der maximale Exponent eine zweistellige Zahl ist. Eine Formatzahl größer als sechs (und kleiner oder gleich 9) *verändert* jedoch die Dezimalstelle, auf der die Rundung stattfindet. (Siehe Seiten 58–59).

Eine Ausnahme bildet die Funktion [∫ₓʸ], wo die Anzeigeformatzahl in R_I von –6 bis +9 reichen kann (siehe Anhang E, Seite 247). Eine Zahl kleiner als Null hat keinen Einfluß auf das Anzeigeformat, verändert aber die Genauigkeit bei der Berechnung der Funktion [∫ₓʸ].

Teil III
Höhere Funktionen des HP-15C

Abschnitt 11

Berechnungen mit komplexen Zahlen

Der HP-15C ermöglicht Ihnen Rechnungen mit komplexen Zahlen, d.h. mit Zahlen der Form

$$a + ib,$$

wobei a der Realteil der komplexen Zahl,

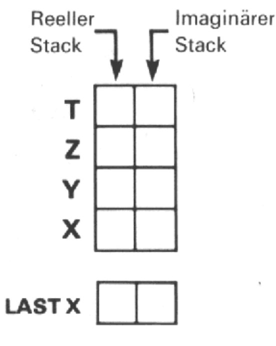
b der Imaginärteil der komplexen Zahl und

$i = \sqrt{-1}$ ist,

auszuführen. Wie Sie bald erkennen werden, besteht die Einfachheit einer komplexen Rechnung mit dem HP-15C darin, daß die meisten Operationen in der selben Weise wie bei einer Berechnung mit reellen Zahlen ausgeführt werden, wenn die komplexen Zahlen erst einmal eingegeben sind.

Komplex-Modus und komplexer Stack

Berechnungen mit komplexen Zahlen werden mit Hilfe eines komplexen Stacks ausgeführt. Dieser besteht aus *zwei* parallelen 4-Registerstacks (und zwei LAST X Registern). Einer dieser parallelen Stacks, als *reeller Stack* bezeichnet, enthält die Realteile, der in der Berechnung verwendeten komplexen Zahlen. (Dies ist der gleiche Stack, der bisher in den Berechnungen mit reellen Zahlen verwendet wurde.) Der andere Stack, als *imaginärer Stack* bezeichnet, enthält die Imaginärteile, der in der Berechnung verwendeten komplexen Zahlen.



Aufbauen des komplexen Stacks

Der imaginäre Stack wird automatisch aufgebaut (durch die Konvertierung von fünf Speicherregistern, siehe Anhang C), sobald Sie den Rechner in den Komplex-Modus umschalten; er existiert nicht, wenn der Komplex-Modus verlassen wird.

Das «Aktivieren» des Komplex-Modus geschieht wie folgt:

1) automatisch, wenn die Funktionen [f] [I] oder [f] [Re≷Im] ausgeführt werden, oder
2) durch das Setzen von Flag 8 (dem Komplex-Modus Flag).

Wenn der Rechner sich im Komplex-Modus befindet, erscheint die Statusanzeige **C** in der Anzeige. Dies deutet an, daß der Flag 8 gesetzt ist und ein komplexer Stack existiert. Im oder außerhalb des Komplex-Modus ist *die Zahl in der Anzeige immer der Inhalt des* **reellen** *X-Registers.*

> **Hinweis**: Im Komplex-Modus (durch die Statusanzeige **C** angezeigt) berechnet der HP-15C *alle* trigonometrischen Funktionen unter Verwendung der Einheit *Radiant. Die trigonometrischen Modi betreffen nur zwei Funktionen:* [→R] *und* [→P] (dies wird später in diesem Abschnitt näher erläutert).

Desaktivieren des Komplex-Modus

Da der Komplex-Modus fünf Speicherregister belegt, stehen Ihnen mehr Speicher für Programme und die höheren Funktionen zur Verfügung, wenn Sie den Komplex-Modus desaktivieren und nur mit reellen Zahlen arbeiten.

Zum Desaktivieren des Komplex-Modus müssen Sie Flag 8 löschen (Tastenfolge [g] [CF] 8). Die Statusanzeige **C** verschwindet aus der Anzeige.

Durch das Löschen des Permanentspeichers (siehe Seite 63) wird der Komplex-Modus ebenfalls desaktiviert. Unabhängig von der zur Desaktivierung des Komplex-Modus verwendeten Operation wird der imaginäre Stack immer gelöscht und die dort gespeicherten Imaginärteile gehen verloren.

Komplexe Zahlen und der Stack

Eingabe einer komplexen Zahl

Eine komplexe Zahl ist wie folgt einzugeben:

1. Tasten Sie den Realteil der Zahl in die Anzeige.
2. Drücken Sie die Taste [ENTER].
3. Tasten Sie den Imaginärteil der Zahl in die Anzeige.
4. Drücken Sie [f] [I]. (Wenn Sie nicht schon im Komplex-Modus sind, wird dadurch der komplexe Stack aufgebaut und die Statusanzeige **C** erscheint im Display.)

Beispiel: Addieren Sie $2 + 3i$ zu $4 + 5i$. (Die Operationen sind in dem auf die Tastenfolge folgenden Diagramm veranschaulicht.)

Tastenfolge	Anzeige	
[f] [FIX] 4		
2 [ENTER]	2.0000	Eingabe des Realteiles der ersten Zahl in das (reelle) Y-Register.
3	3	Eingabe des Imaginärteils der ersten Zahl in das (reelle) X-Register.
[f] [I]	2.0000	Aufbau des imaginären Stacks; Verschieben der 3 in das imaginäre X-Register und Herabfallen der 2 in das reelle X-Register.
4 [ENTER]	4.0000	Eingabe des Realteils der zweiten Zahl in das (reelle) Y-Register.
5	5	Eingabe des Imaginärteils der zweiten Zahl in das (reelle) X-Register.
[f] [I]	4.0000	Kopieren der 5 vom reellen X-Register in das imaginäre X-Register; Kopieren der 4 vom reellen Y-Register in das reelle X-Register und Stack Drop.
[+]	6.0000	Realteil der Summe.
[f] [(i)] (niedergedrückt)	8.0000	Anzeige des Imaginärteils der Summe. (Dies beendet zusätzlich die Zifferneingabe.)
(losgelassen)	6.0000	

Die Operationen des reellen und des imaginären Stacks während dieses Prozesses sind auf der nächsten Seite veranschaulicht. (Wir nehmen an, daß die Stackregister mit den eingesetzten Zahlen als Ergebnisse früherer Rechnungen schon geladen sind.) Beachten Sie, daß der imaginäre Stack, der rechts vom reellen Stack angedeutet ist, erst gebildet wird, nachdem Sie [f] [I] gedrückt haben. (Die schattierten Felder des Stacks deuten an, daß diese Inhalte überschrieben werden, sobald die nächste Zahl eingegeben oder zurückgerufen wird.)

	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
T	9		8		7		7		7	0
Z	8		7		6		6		7	0
Y	7		6		2		2		6	0
X	6		2		2		3		2	3

Tasten: 2 ENTER 3 f I

Die Ausführung von f I bewirkt einen Stack Drop des gesamten Stacks, eine Reproduzierung des T-Wertes und eine Übertragung des reellen X-Registers in das imaginäre X-Register.

Wenn die zweite Zahl eingegeben wird, verändert sich der Stack wie unten gezeigt. Beachten Sie, daß die Taste ENTER *beide* Stacks anhebt.

	Re	Im	Re	Im	Re	Im	Re	Im
T	7	0	7	0	6	0	6	0
Z	7	0	6	0	2	3	2	3
Y	6	0	2	3	4	0	4	0
X	2	3	4	0	4	0	5	0

Tasten: 4 ENTER 5

	Re	Im	Re	Im	Re	Im
T	6	0	6	0	6	0
Z	2	3	6	0	6	0
Y	4	0	2	3	6	0
X	5	0	4	5	6	8

Tasten: f I +

Eine zweite Methode zur Eingabe komplexer Zahlen besteht darin, den imaginären Teil zuerst einzugeben, und dann die Tasten Re≷Im und ← zu benutzen. Diese Methode ist auf Seite 127 unter «Eingabe einer komplexen Zahl mit Hilfe von ←» beschrieben.

Stack Lift im Komplex-Modus

Der Stacklift verändert den imaginären Stack in der selben Weise, wie er den reellen Stack verändert (der reelle Stack verhält sich inner- und außerhalb des Komplex-Modus immer gleich). *Dieselben Funktionen, die einen Stack Lift des reellen Stacks freigeben, sperren oder sich neutral verhalten, sind es auch, die einen Stack Lift des imaginären Stacks freigeben, sperren oder sich neutral verhalten.* (Dieser Prozess wird im Detail im Abschnitt 3 und Anhang B erklärt.)

Zusätzlich *bewirkt jede nichtneutrale Funktion, mit Ausnahme von* [←] *und* [CLx], *ein Löschen des imaginären X-Registers, wenn die nächste Zahl eingegeben wird.* Diese Eigenschaft erlaubt es ihnen Rechenoperationen im Komplex-Modus auszuführen und dafür dieselben Tastenfolgen zu verwenden wie außerhalb des Komplex-Modus*.

Umordnung des imaginären und des reellen Stacks

[Re≷Im] ***(real exchange imaginary)***. Durch das Drücken der Taste [Re≷Im] werden die Inhalte des reellen und des imaginären X-Registers ausgetauscht. Hierbei wird der Imaginärteil der Zahl in den Realteil umgewandelt und umgekehrt. Die Y-, Z- und T-Register werden dabei *nicht* verändert. Drücken Sie [f] [Re≷Im] *zweimal*, um die Zahlen in der ursprünglichen Form abzuspeichern.

Die Taste [Re≷Im] bewirkt zusätzlich ein Umschalten in den Komplex-Modus, falls sich der Rechner noch nicht in diesem Modus befindet.

Kurzzeitige Anzeige des imaginären X-Registers. Durch Drücken von [f] [(i)] wird *kurzzeitig* der Imaginärteil der Zahl im X-Register angezeigt, *ohne dabei den Realteil und den Imaginärteil dieser Zahl zu vertauschen.* Der Imaginärteil wird solange angezeigt, wie Sie die Tasten gedrückt halten.

Vorzeichenwechsel

Im Komplex-Modus beeinflußt die Taste [CHS] nur den Inhalt des reellen X-Registers, das imaginäre X-Register bleibt unverändert. Das erlaubt ihnen das Vorzeichen des Realteils oder des Imaginärteils unabhängig voneinander zu verändern. Um einen negativen Realteil oder Imaginärteil einzugeben, ändern Sie das Vorzeichen des entsprechenden Teiles im nachhinein.

Wenn Sie die additive Inverse einer komplexen Zahl, die *sich schon im X-Register befindet,* bilden wollen, können Sie jedoch nicht einfach die Taste [CHS] drücken, wie das außerhalb des Komplex-Modus möglich ist. Stattdessen müssen Sie eine der folgenden Operationen verwenden:

---

* Mit Ausnahme der Funktionen [→P] und [→R], siehe Seite 133.

- Multiplizieren der Zahl mit –1.
- Wenn Sie den Rest des Stacks nicht verändern wollen, drücken Sie CHS f Re≷Im CHS f Re≷Im.

Wenn Sie Vorzeichen von nur einem Teil der komplexen Zahl im X-Register ändern wollen:

- Drücken Sie CHS, um nur das Vorzeichen des *Realteils* zu ändern.
- Drücken Sie f Re≷Im CHS f Re≷Im, um nur das Vorzeichen des *Imaginärteils* zu ändern, d.h. Sie bilden die konjugiert komplexe Zahl.

Löschen einer komplexen Zahl

Zwangsläufig werden Sie einmal eine komplexe Zahl löschen müssen. Sie können nur einen Teil auf einmal löschen, aber anschließend beide Teile überschreiben (da ← und CLx den Stack sperren).

Löschen des reellen X-Registers. Durch Drücken von ← (oder g CLx) im Komplex-Modus wird nur der Inhalt des reellen X-Registers gelöscht; die Zahl im imaginären X-Register wird *nicht* gelöscht.

Beispiel: Ändern Sie $6 + 8i$ in $7 + 8i$ und subtrahieren Sie diese Zahl von der vorigen Eingabe. (Verwenden Sie f Re≷Im oder f (i), um den Imaginärteil des X-Registers anzuzeigen.) Nehmen Sie an, daß *a, b, c* und *d* Teile von komplexen Zahlen darstellen.

	Re	Im	Re	Im	Re	Im	Re	Im
T	*a*	*b*	*a*	*b*	*a*	*b*	*a*	*b*
Z	*c*	*d*	*c*	*d*	*c*	*d*	*a*	*b*
Y	6	0	6	0	6	0	*c*	*d*
X	6	8	0	8	7	8	–1	–8
Tasten:			←		7		– (oder jede andere Operation)	

Da Löschoperationen den Stack Lift sperren (wie oben beschrieben), nimmt der als nächstes eingegebene Wert die Stelle des gerade gelöschten Wertes ein. Wenn Sie den Realteil durch Null ersetzen wollen, drücken Sie nach dem Löschen die Taste ENTER oder irgendeine andere Taste, um die Eingabe zu beenden (andernfalls überschreibt die nächste Eingabe die Null); der Imaginärteil bleibt unverändert. Sie können danach mit irgendeiner Rechnerfunktion fortfahren.

Löschen des imaginären X-Registers. Um den Inhalt im imaginären X-Register zu löschen, drücken Sie f Re≷Im und dann ←. Drücken Sie f Re≷Im nochmals, um die Null, oder jede andere eingegebene Zahl, in das imaginäre X-Register zurückzuladen.

Beispiel: Ersetzen Sie $-1-8i$ durch $-1+5i$.

	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
T	*a*	*b*	*a*	*b*	*a*	*b*	*a*	*b*	*a*	*b*
Z	*c*	*d*	*c*	*d*	*c*	*d*	*c*	*d*	*c*	*d*
Y	*e*	*f*	*e*	*f*	*e*	*f*	*e*	*f*	*e*	*f*
X	−1	−8	−8	−1	0	−1	5	−1	−1	5
Tasten:			Re≷Im		←		5		Re≷Im (Fortsetzung mit beliebiger Operation.)	

Löschen des imaginären und des reellen X-Registers. Wenn Sie *sowohl* den Real- *als auch* den Imaginärteil einer komplexen Zahl löschen oder ersetzen wollen, drücken Sie einfach ←, um den Stack zu sperren, und geben danach die neue Zahl ein. Wenn die neue Zahl rein reell (einschließlich $0+0i$) ist, können Sie die alte, komplexe Zahl schnell löschen oder ersetzen, indem Sie ← gefolgt von Null oder der neuen, reellen Zahl drücken.

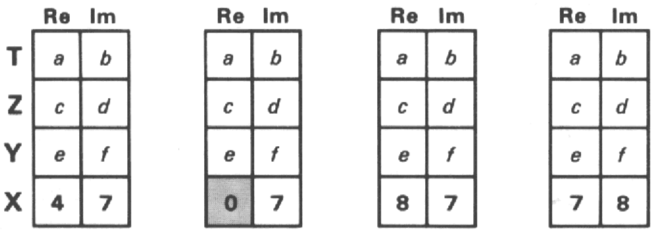
Beispiel: Ersetzen Sie $-1+5i$ durch $4+7i$.

	Re	Im	Re	Im	Re	Im	Re	Im	Re	Im
T	*a*	*b*	*a*	*b*	*c*	*d*	*c*	*d*	*c*	*d*
Z	*c*	*d*	*c*	*d*	*e*	*f*	*e*	*f*	*c*	*d*
Y	*e*	*f*	*e*	*f*	4	5	4	5	*e*	*f*
X	−1	5	0	5	4	5	7	0	4	7
Tasten:			←		4 ENTER		7		f I (Fortsetzung mit beliebiger Operation.)	

Eingabe einer komplexen Zahl mit Hilfe der Taste [←]. Die Löschfunktionen [←] und [CLx] können auch zusammen mit der Taste [Re≷Im] als alternative Methode zur Eingabe (oder zum Löschen) einer komplexen Zahl verwendet werden. Mit dieser Methode können Sie eine komplexe Zahl eingeben und dabei nur das X-Register verwenden. Der übrige Stack bleibt unverändert. (Dies ist möglich, da [←] und [CLx] den Stack Lift sperren.) Bei der Ausführung der Funktion [Re≷Im] wird zusätzlich ein imaginärer Stack aufgebaut, falls dies noch nicht geschehen ist.

Beispiel: Geben Sie 9 + 8i ein, ohne dabei den Stack anzuheben, und berechnen Sie das Quadrat dieser Zahl.

Tastenfolge	**Anzeige**	
([←])	(0.0000)	Verhindert ein Anheben des Stacks bei der Eingabe der nächsten Ziffer. Lassen Sie diesen Schritt aus, wenn Sie den X-Wert retten und dafür den T-Wert verlieren wollen.
8	8	Geben Sie den Imaginärteil zuerst ein.
[f] [Re≷Im]	7.0000	Anzeige des Realteils; Komplex-Modus wird aktiviert.
[←]	0.0000	Sperrt den Stack. (Andernfalls würde die Taste [Re≷Im] einen Stack Lift bewirken.)
9	9	Eingabe des Realteils. Die Ziffereingabe ist noch nicht abgeschlossen.
[g] [x^2]	20.0000	Realteil.
[f] [(i)] (gedrückt)	144.0000	Imaginärteil.
(loslassen)	17.0000	



Tasten: [←] 8 [f] [Re≷Im]

	Re	Im		Re	Im		Re	Im		Re	Im
T	*a*	*b*		*a*	*b*		*a*	*b*		*a*	*b*
Z	*c*	*d*		*c*	*d*		*c*	*d*		*c*	*d*
Y	*e*	*f*		*e*	*f*		*e*	*f*		*e*	*f*
X	**7**	**8**		**0**	**8**		**9**	**8**		**17**	**144**
Tasten:			←			9			g x^2		

Eingabe einer reellen Zahl

Wir haben nun schon zwei Arten der Eingabe einer komplexen Zahl kennengelernt. Aber es gibt einen kürzeren Weg eine rein reelle Zahl einzugeben: tasten (oder rufen) Sie einfach die Zahl in die Anzeige (zurück), genauso wie Sie es außerhalb des Komplex-Modus tun würden. Dabei wird automatisch eine Null in das imaginäre X-Register gesetzt (solange die vorhergehende Operation nicht ← oder CLx war, siehe Seite 124).

Die Veränderungen des reellen und des imaginären X-Registers während dieser Operation sind unten verdeutlicht. (Wir nehmen an, daß die zuletzt gedrückte Taste nicht ← war und die Stackinhalte aus früheren Berechnungen resultieren.)

	Re	Im		Re	Im		Re	Im	
T	*a*	*b*		*c*	*d*		*e*	*f*	
Z	*c*	*d*		*e*	*f*		**17**	**144**	
Y	*e*	*f*		**17**	**144**		**4**	**0**	
X	**17**	**144**		**4**	**0**		**4**	**0**	
Tasten:			4			ENTER			(Gefolgt von einer anderen Zahl.)

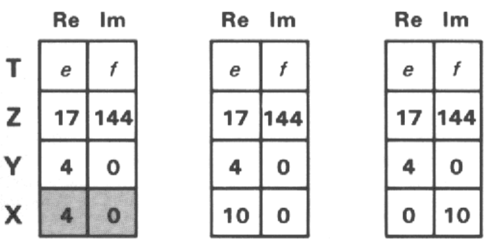
Eingabe einer rein imaginären Zahl

Wenn sich der Rechner im Komplex-Modus befindet, können Sie die Eingabe einer rein imaginären Zahl verkürzen: tasten Sie einfach die (imaginäre) Zahl ein und drücken Sie [Re≷Im].

Beispiel: Geben Sie $0 + 10i$ ein. (Wir nehmen an, daß die zuletzt ausgeführte Funktion nicht [←] oder [CLx] war.)

Tastenfolge	Anzeige	
10	**10**	Eingabe von 10 in das angezeigte, reelle X-Register und Null in das imaginäre X-Register.
[f] [Re≷Im]	**0.0000**	Austausch des reellen und des imaginären X-Registers. Das Display zeigt, daß sich im reellen X-Register die Null befindet, wie es bei einer rein imaginären Zahl der Fall sein sollte.

Die Veränderungen des reellen und des imaginären Stacks, während der soeben beschriebenen Operation, werden im folgenden Diagramm veranschaulicht. (Wir nehmen an, daß die Stackregister die Ergebnisse des vorigen Beispiels enthalten.)



Tasten: 10 [f] [Re≷Im] (Fortsetzung mit beliebiger Operation.)

Beachten Sie, daß [Re≷Im] nur die Inhalte des reellen und imaginären *X-Registers* austauscht und *nicht* auch die der übrigen Register.

Speichern und Rückrufen von komplexen Zahlen

Die Funktionen [STO] und [RCL] wirken *nur auf das reelle X-Register*; daher muß der Imaginärteil einer komplexen Zahl getrennt gespeichert oder zurückgerufen werden. Die Tastenfolgen für diese Operation können als Teil eines Programms eingegeben und dann automatisch ausgeführt werden*.

Um $a + ib$ vom reellen und komplexen X-Register in die Register R_1 und R_2 zu speichern, können Sie die folgende Tastenfolge verwenden:

[STO] 1 [f] [Re≷Im] [STO] 2

WennSie danach [f] [Re≷Im] drücken, ist der Stack in seinen Ausgangszustand gebracht. Um $a + ib$ aus den Registern R_1 und R_2 zurückzurufen, können Sie die folgende Tastenfolge verwenden:

[RCL] 1 [RCL] 2 [f] [I]

Wollen Sie vermeiden, daß der Rest des Stacks verändert wird, können Sie die folgende Tastenfolge verwenden:

[RCL] 2 [f] [Re≷Im] [←] [RCL] 1.

(Verwenden Sie im Programm-Modus [g] [CLx] anstelle von [←].)

Operationen mit komplexen Zahlen

Fast alle Funktionen mit *reellen Zahlen* werden dasselbe Ergebnis liefern, gleich ob sie im oder außerhalb des Komplex-Modus ausgeführt werden**, sofern das Resultat ebenfalls reell ist. Mit anderen Worten, der Komplex-Modus beschneidet die Möglichkeiten, mit reellen Zahlen zu rechnen, *nicht*.

Alle Funktionen, die nicht innerhalb dieses Abschnittes erwähnt wurden, ignorieren den imaginären Stack.

---

* Sie können die Matrixfunktionen des HP-15C dazu verwenden (siehe Abschnitt 12), das Speichern und Rückrufen komplexer Zahlen noch einfacher zu gestalten. Durch das Dimensionieren einer $n \times 2$ Matrix, können n komplexe Zahlen als Zeilen der Matrix gespeichert werden. (Diese Technik wird in dem Handbuch *HP-15C – Fortgeschrittene Funktionen*, in Abschnitt 3, beschrieben.)

** Ausnahmen sind die Funktionen [→P] und [→R], die im Komplex-Modus verschieden operieren, um die Konvertierung einer komplexen Zahl in Polarkoordinaten zu erleichtern (siehe Seite 133).

Funktionen einer Variablen

Die folgenden Funktionen wirken auf den Real- und Imaginärteil der Zahl im X-Register und geben den Real- und Imaginärteil des Ergebnisses in diese Register zurück.

[√x] [x²] [LN] [LOG] [1/x] [10ˣ] [eˣ] [ABS] [→P] [→R]

Alle trigonometrischen und hyperbolischen Funktionen und ihre Umkehrfunktionen gehören ebenfalls zu dieser Gruppe*.

Die Funktion [ABS] berechnet den Betrag der Zahl im X-Register (die Quadratwurzel aus der Summe der Quadrate des Realteils und des Imaginärteils); als Imaginärteil des Betrages wird der Wert Null zurückgegeben.

Die Funktion [→P] transformiert in Polarkoordinaten und [→R] transformiert in Rechteckskoordinaten (siehe Seite 133).

Bei der Verwendung trigonometrischer Funktionen betrachtet der Rechner die Zahlen im reellen und imaginären X-Register als in *Radiant* angegeben, ungeachtet des augenblicklichen trigonometrischen Modus. Um trigonometrische Funktionen mit Argumenten, die in Altgrad angegeben sind, zu berechnen, müssen diese vor der Ausführung der trigonometrischen Funktion durch [RAD] in Radiant-Werte umgewandelt werden.

Funktionen zweier Variablen

Die folgenden Funktionen wirken sowohl auf den Realteil als auch auf den Imaginärteil der Zahlen in den X- und Y-Registern und geben den Realteil und den Imaginärteil des Ergebnisses in die X-Register. Die Ausführung dieser Funktionen beinhalten einen Stack Drop beider Stacks.

[+] [−] [×] [÷] [yˣ]

Funktionen zur Stackmanipulation

Im Komplex-Modus verändern die folgenden Funktionen gleichzeitig den reellen und den imaginären Stack. Dies geschieht in derselben Art und Weise, wie diese Funktionen den gewöhnlichen Stack, d.h. den Stack, wenn der Rechner sich nicht im Komplex-Modus befindet, verändern.

---

* Weitere Informationen zu komplexen trigonometrischen Funktionen und den Berechnungen im Komplex-Modus liefert Ihnen das Handbuch *«HP-15C – Fortgeschrittene Funktionen»*.

Die Funktion [x⪌y], z.B., vertauscht den Realteil *und* den Imaginärteil der Zahlen im X- und Y-Register.

[x⪌y] [R↓] [R↑] [ENTER] [LSTx]

Vergleichsoperationen

Zur Programmierung stehen die vier unten aufgeführten Vergleichsoperationen im Zusammenhang mit komplexen Zahlen zur Verfügung: [x=0] und [TEST] 0 vergleichen die *komplexe* Zahl im (reellen und imaginären) X-Register mit $0 + 0i$, während [TEST] 5 und [TEST] 6 die *komplexen* Zahlen in den (reellen und imaginären) X- und Y-Registern vergleichen. Alle nicht aufgeführten Vergleichsoperationen ignorieren den imaginären Stack.

[x=0] [TEST] 0 ($x \neq 0$) [TEST] 5 ($x = y$) [TEST] 6 ($x \neq y$)

Beispiel: Komplexe Arithmetik. Die charakteristische Impedanz eines Stufenverteilernetzes kann mit der folgenden Formel berechnet werden:

$$Z_0 = \sqrt{\frac{A}{B}},$$

wobei A und B komplexe Zahlen sind. Finden Sie Z_0 für die hypothetischen Werte $A = 1.2 + 4.7i$ und $B = 2.7 + 3.2i$.

Tastenfolge	**Anzeige**	
1.2 [ENTER] 4.7 [f] [I]	**1.2000**	Eingabe von A in das reelle und imaginäre X-Register.
2.7 [ENTER] 3.2 [f] [I]	**2.7000**	Eingabe von B in das reelle und imaginäre X-Register, und Verschieben von A in das reelle und imaginäre Y-Register.
[÷]	**1.0428**	Berechnung von A/B.
[√x]	**1.0491**	Berechnung von Z_0 und Anzeige des Realteils.
[f] [(i)] (niedergedrückt)	**0.2406**	Anzeige des Imaginärteils von Z_0 solange [(i)] gedrückt wird.
(losgelassen)	**1.0491**	Wieder Anzeige des Realteils von Z_0.

Komplexe Ergebnisse bei Berechnungen mit reellen Zahlen

Im vorigen Beispiel wurde durch die Eingabe von komplexen Zahlen der Komplex-Modus (automatisch) aktiviert. Es wird jedoch manchmal nötig sein, daß sich der Rechner auch bei Berechnungen mit *reellen* Zahlen im Komplex-Modus befindet, wie z.B. zur Berechnung von $\sqrt{-5}$. (Ohne daß der Rechner sich im Komplex-Modus befindet, würde diese Operation die Meldung **Error 0** – unerlaubte mathematische Operation – erzeugen.) Um den Rechner zu irgendeinem Zeitpunkt in den Komplex-Modus zu schalten *und dabei den Stack nicht zu verändern*, setzen Sie den Flag 8, bevor Sie die fragliche Funktion* ausführen.

Beispiel: Die Berechnung des Arcussinus von 2.404 würde normalerweise eine **Error 0** Meldung liefern. Nehmen wir an, daß sich 2.404 im X-Register befindet; der komplexe Wert Arcussinus von 2.404 kann dann wie folgt berechnet werden:

Tastenfolge	Anzeige	
[g] [SF] 8		Aktiviert den Komplex-Modus.
[g] [SIN⁻¹]	**1.5708**	Realteil von Arcussinus 2.404.
[f] [(i)] (gedrückt)	**–1.5239**	Imaginärteil des Arcussinus 2.404.
(losgelassen)	**1.5708**	Nach dem Loslassen der Taste [(i)] wird der Realteil wieder angezeigt.

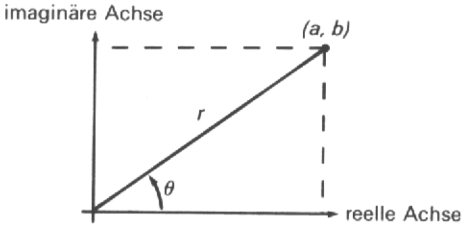
Transformation von Polar- und Rechteckskoordinaten

In vielen Anwendungen sind komplexe Zahlen in Polarkoordinaten angegeben, manchmal wird dabei die Phasenschreibweise verwendet. Der HP-15C nimmt jedoch an, daß jede komplexe Zahl in *Rechteckskoordinaten* ausgedrückt ist. Deshalb müssen alle in Polarkoordinaten- oder Phasendarstellung gegebenen Zahlen vor der Anwendung einer Funktion im Komplex-Modus in rechtwinklige Form gebracht werden.

---

* *Zweimaliges* Drücken von [f] [Re≷Im] hat die gleiche Wirkung. Diese Tastenfolge wird deshalb nicht verwendet, weil sie die Zahlen im reellen X- und Y-Register in einer einzigen komplexen Zahl zusammenfassen würde.

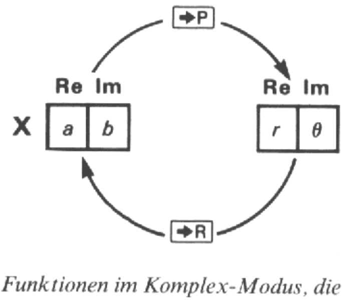
$$a + ib = \begin{cases} r(\cos\theta + i\sin\theta) = re^{i\theta} & \text{(Polarform)} \\ r\angle\theta & \text{(Phasenschreibweise)} \end{cases}$$



[→R] und [→P] können zur Umwandlung der Rechtecksdarstellung einer komplexen Zahl in die Polardarstellung bzw. umgekehrt verwendet werden. Im *Komplex-Modus* arbeiten diese Funktionen wie folgt:

[f] [→R] konvertiert die Polarkoordinatendarstellung einer komplexen Zahl in Rechteckskoordinaten, indem der Betrag der Zahl im reellen X-Register durch den a-Wert ersetzt wird, und die Phase (der Winkel θ) im imaginären X-Register durch den b-Wert.

[g] [→P] konvertiert die Rechtecksdarstellung einer komplexen Zahl in ihre Polarkoordinatendarstellung, indem der Realteil a im reellen X-Register durch den Betrag, und der Imaginärteil im imaginären X-Register durch den Winkel (die Phase) θ ersetzt wird.



Dies sind die einzigen Funktionen im Komplex-Modus, die von dem augenblicklichen trigonometrischen Status des Rechners beeinflußt werden. In diesen Fällen muß der Wert für θ in den durch den momentanen trigonometrischen Modus bestimmten Einheiten interpretiert bzw. eingegeben werden.

Beispiel: Berechnen Sie die Summe $2(\cos 65° + i \sin 65°) + 3(\cos 40° + i \sin 40°)$ und drücken Sie das Ergebnis in Polarkoordinaten aus. (Oder in der Phasennotation: berechnen Sie $2 \angle 65° + 3 \angle 40°$.)

Tastenfolge	Anzeige	
g DEG		Altgrad-Modus gesetzt für alle Umwandlungen von Polar- in Rechteckskoordinaten.
2 ENTER	**2.0000**	
65 f I	**2.0000**	Statusanzeige **C** erscheint in der Anzeige; der Komplex-Modus ist aktiviert.
f →R	**0.8452**	Transformation von Polarkoordinaten in rechtwinklige Koordinaten; der Realteil (*a*) wird angezeigt
3 ENTER	**3.0000**	
40 f I	**3.0000**	
f →R	**2.2981**	Transformation von Polarkoordinaten in rechtwinklige Koordinaten; der Realteil (*a*) wird angezeigt.
+	**3.1434**	
g →P	**4.8863**	Transformation von Rechteckskoordinaten in Polarform; *r* wird angezeigt.
f (i) (gedrückt)	**49.9612**	θ (in Altgrad)
(losgelassen)	**4.8863**	

Übungsaufgaben

Wenn Sie die folgenden Problemstellungen durchrechnen, werden Sie erkennen, daß Berechnungen mit komplexen Zahlen mit Ihrem HP-15C genauso einfach und rasch auszuführen sind, wie die Berechnungen mit reellen Zahlen. Die meisten mathematischen Operationen erfordern sogar genau die gleichen Tastenfolgen, wenn die Zahlen erst einmal eingegeben sind.

1. Berechnen Sie $$\frac{2i(-8+6i)^3}{(4-2\sqrt{5}\,i)(2-4\sqrt{5}\,i)}$$

Tastenfolge	Anzeige	
2 [f] [Re⇄Im]	0.0000	$2i$. Anzeige des Realteils.
8 [CHS] [ENTER]	-8.0000	
6 [f] [I]	-8.0000	$-8+6i$.
3 [y^x]	352.0000	$(-8+6i)^3$.
[×]	-1,872.0000	$2i(-8+6i)^3$.
4 [ENTER]	4.0000	
5 [$\sqrt{x}$]	2.2361	
2 [CHS] [×]	-4.4721	$-2\sqrt{5}$.
[f] [I]	4.0000	$4-2\sqrt{5}i$.
[÷]	-295.4551	$\dfrac{2i(-8+6i)^3}{4-2\sqrt{5}i}$.
2 [ENTER] 5 [$\sqrt{x}$]	2.2361	
4 [CHS] [×]	-8.9443	
[f] [I]	2.0000	$2-4\sqrt{5}i$.
[÷]	9.3982	Realteil des Ergebnisses.
[f] [(i)]	-35.1344 9.3982	Komplexe Lösung: 9.3982–35.1344i.

2. Schreiben Sie ein Programm, das die Funktion $\omega = \dfrac{2z+1}{5z+3}$ für verschiedene Werte von z berechnet. (ω ist eine konforme Abbildung.) Berechnen Sie die Funktion für $z = 1 + 2i$.
(Ergebnis: $0.3902 + 0.0122i$. Eine mögliche Tastenfolge ist:

[f] [LBL] [A] [ENTER] [ENTER] 2 [×] 1 [+] [x⇄y] 5 [×] 3 [+] [÷] [R/S] [f] [Re⇄Im] [g] [RTN].)

3. Versuchen Sie sich an einem komplexen Polynom und ändern Sie dazu das Beispiel von Seite 80 ab. Sie können dasselbe Programm verwenden, um das Polynom $P(z) = 5z^4 + 2z^3$ zu berechnen, wobei z eine komplexe Zahl ist.

Laden Sie den Stack mit $z = 7 + 0i$ und versuchen Sie, ob Sie dasselbe Ergebnis wie auf Seite 80 erhalten.

(Ergebnis: $12{,}691.0000 + 0.0000i$.)

Führen Sie jetzt das Programm mit $z = 1 + i$ aus.

(Ergebnis: $-24.0000 + 4.0000i$.)

Zusätzliche Informationen

Das Handbuch *HP-15C Fortgeschrittene Funktionen* enthält detailliertere und technisch tiefergehende Aspekte der Verwendung komplexer Zahlen in den verschiedenen Funktionen des HP-15C. Zusätzlich finden Sie zahlreiche Anwendungen. Dieses Handbuch deckt die folgenden Bereiche ab:

- Genauigkeitsbetrachtungen.
- Hauptzweige mehrdeutiger Funktionen.
- Komplexe Kontourintegrale.
- Komplexe Potentiale.
- Speichern und Zurückrufen komplexer Zahlen unter Verwendung einer Matrix.
- Berechnung der *n*-ten Wurzel einer komplexen Zahl.
- Komplexe Lösungen einer Gleichung.
- Verwendung von [SOLVE] und [$\int_y^x$] im Komplex-Modus.

Abschnitt 12

Matrizenrechnung

Mit Hilfe der Matrizenoperationen des HP-15C sind Sie in der Lage, schwierige algebraische Probleme elegant zu lösen. Der Rechner kann mit bis zu fünf Matrizen gleichzeitig arbeiten; diese werden nach den Tasten [A] bis [E], mit denen auf sie zugegriffen wird, **A** bis **E** genannt. Sie können die Größe jeder Matrix einzeln festlegen, einzelne Matrixelemente eingeben und abfragen, und Matrixoperationen durchführen, sowohl für reelle als auch komplexe Matrizen. (Am Ende dieses Abschnitts finden Sie eine Zusammenstellung der Matrixfunktionen.)

Eine häufige Anwendung der Matrizenrechnung besteht in der Lösung von linearen Gleichungssystemen. Gesucht sind zum Beispiel die Werte x_1 und x_2, die folgendes Gleichungssystem lösen:

$$3.8x_1 + 7.2x_2 = 16.5$$

$$1.3x_1 - 0.9x_2 = -22.1$$

Diese Gleichungen können in Matrizenform als $\mathbf{AX} = \mathbf{B}$ dargestellt werden, wobei

$$\mathbf{A} = \begin{bmatrix} 3.8 & 7.2 \\ 1.3 & -0.9 \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad \text{und } \mathbf{B} = \begin{bmatrix} 16.5 \\ -22.1 \end{bmatrix}.$$

Mit den nachfolgenden Tastenfolgen können Sie dieses Matrixproblem sehr einfach lösen. (Die dabei benutzten Matrizenoperationen werden später in diesem Abschnitt erklärt.)

Zuerst müssen die zwei gegebenen Matrizen dimensioniert und ihre Elemente eingegeben werden, von links nach rechts in jeder Zeile, von der ersten bis zur letzten Zeile. Definieren Sie die Matrix **C** als die Matrix, die das Ergebnis Ihrer Matrizenrechnung aufnehmen soll ($\mathbf{C} = \mathbf{X}$).

Tastenfolge	Anzeige	
g CF 8		Desaktiviert Komplex-Modus.
2 ENTER f DIM A	2.0000	Dimensioniert **A** als 2×2-Matrix.
f MATRIX 1	2.0000	Vorbereitung zur automatischen Eingabe der Matrixelemente im User-Modus.
USER	2.0000	Statusanzeige **USER** erscheint.
3.8 STO A	A 1.1	Bezeichnet Matrix **A**, Zeile 1, Spalte 1. (Eine solche Anzeige erscheint bei der Eingabe jedes Elements und bleibt erhalten, solange Sie die Buchstabentaste gedrückt halten.)
	3.8000	Speichert a_{11}.
7.2 STO A	7.2000	Speichert a_{12}.
1.3 STO A	1.3000	Speichert a_{21}.
.9 CHS STO A	–0.9000	Speichert a_{22}.
2 ENTER 1 f DIM B	1.0000	Dimensioniert **B** als 2×1-Matrix.
16.5 STO B	16.5000	Speichert b_{11}.
22.1 CHS STO B	–22.1000	Speichert b_{21}.
f RESULT C	–22.1000	Definiert Matrix **C** als Ergebnismatrix.

In Matrix-Notation hat die Matrizengleichung $\mathbf{AX} = \mathbf{B}$ die Lösung

$$\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$$

wobei $\mathbf{A}^{-1}$ die inverse Matrix zu **A** ist. Sie können diese Operation ausführen, indem Sie die «Deskriptoren» der Matrizen **A** und **B** in die Register X und Y eingeben und dann ÷ drücken.

(Ein Deskriptor besteht aus dem Namen und den Dimensionen der Matrix.) Beachten Sie, daß einfache Zahlen auch als 1 × 1-Matrizen behandelt werden können.

Tastenfolge	Anzeige		
RCL MATRIX B	b	2 1	Eingabe des Deskriptors der 2 × 1-Konstantenmatrix **B**.
RCL MATRIX B	A	2 2	Eingabe des Deskriptors der 2 × 2-Koeffizientenmatrix **A** ins X-Register, der Deskriptor von **B** wandert ins Y-Register.
÷	running		Anzeige während der Operationsausführung.
	C	2 1	Deskriptor der 2 × 1-Ergebnismatrix **C**.

Rufen Sie nun die Elemente der Matrix **C** ab, die Lösung der Matrizengleichung. (Verlassen Sie dann den User-Modus und löschen Sie alle Matrizen.)

Tastenfolge	Anzeige	
RCL C	C 1,1	Bezeichnet Matrix **C**, Zeile 1, Spalte 1.
	–11.2887	Wert von c_{11} (x_1).
RCL C	8.2496	Wert von c_{21} (x_2).
f USER	8.2496	Verlassen des User-Modus.
f MATRIX 0	8.2496	Löscht alle Matrizen.

Die Lösung des Gleichungssystems ist $x_1 = -11.2887$ und $x_2 = 8.2496$.

Bemerkung: Die Beschreibung der Matrizenrechnungen in diesem Abschnitt setzt Vorkenntnisse in Matrixalgebra voraus.

Matrixdimensionen

Der Speicher kann bis zu 64 Matrixelemente aufnehmen. Sie können diesen Speicherraum für eine einzige Matrix verwenden oder auf bis zu fünf Matrizen aufteilen.

Eine Matrixinvertierung kann beispielsweise mit einer reellen 8 × 8-Matrix (oder, wie später erläutert, mit einer komplexen 4 × 4-Matrix)* durchgeführt werden.

Um Speicherplatz einzusparen, sind alle Matrizen zu Anfang auf 0 × 0 dimensioniert. Bei der Dimensionierung einer Matrix wird ihr automatisch die entsprechende Anzahl Datenspeicherregister zugeordnet. Möglicherweise müssen Sie die dem Matrizenspeicher zugeteilte Anzahl Register erhöhen, bevor Sie eine Matrix dimensionieren oder eine bestimmte Matrixoperation ausführen können. Anhang C beschreibt die Speicherorganisation des HP-15C, und wie Sie die Anzahl der dem Matrizenspeicher derzeit zugeteilten Register feststellen und ändern können.

Dimensionieren einer Matrix

Um eine Matrix auf *y* Zeilen und *x* Spalten zu dimensionieren, geben Sie diese Zahlen in die entsprechenden Register X und Y ein, und führen Sie dann [f] [DIM], gefolgt von der Taste mit dem Namen der Matrix, aus.

1. Tasten Sie die Anzahl der Zeilen (*y*) ein und drücken Sie [ENTER], um diese ins Y-Register zu bringen.
2. Tasten Sie die Anzahl der Spalten (*x*) ins X-Register.
3. Drücken Sie [f] [DIM], gefolgt vom Namen der Matrix (wahlweise [A] bis [E]**).

Y	**Anzahl der Zeilen**
X	**Anzahl der Spalten**

* Die Matrizenfunktionen in diesem Abschnitt arbeiten nur mit reellen Matrizen. (Im Komplex-Modus wird der imaginäre Stack während Matrixoperationen ignoriert.) Der HP-15C hat jedoch vier Matrizenfunktionen, mit denen Sie komplexe Matrizen in *reeller Darstellung* bearbeiten können. Siehe dazu die Seiten 160–173.

** Sie brauchen vor diesen Tasten nicht [f] zu drücken (siehe «Verkürzte Tastenfolgen», Seite 78).

Beispiel: Dimensionieren Sie **A** als 2 × 3-Matrix.

Tastenfolge	**Anzeige**	
2 [ENTER]	**2.0000**	Eingabe der Anzahl der Zeilen ins X-Register.
3	**3**	Eingabe der Anzahl der Spalten ins X-Register.
[f] [DIM] [A]	**3.0000**	Dimensioniert **A** als 2 × 3-Matrix.

Anzeige von Matrixdimensionen

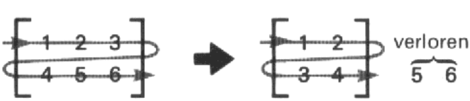
Sie können die Dimensionen einer Matrix auf zwei Weisen zur Anzeige bringen:

- Drücken Sie [RCL] [MATRIX], gefolgt von der Namenstaste der Matrix. Der Name der Matrix steht links, das Zahlenpaar Anzahl Zeilen/Anzahl Spalten rechts in der Anzeige.
- Drücken Sie [RCL] [DIM], gefolgt von der Namenstaste der Matrix. Die Anzahl der Zeilen steht dann im Y-Register, die Anzahl der Spalten im X-Register.

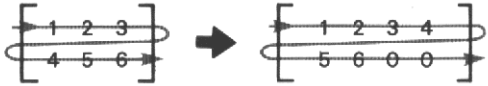
Tastenfolge	**Anzeige**	
[RCL] [MATRIX] [B]	**b 0 0**	Matrix **B** hat 0 Zeilen und 0 Spalten, da sie nicht explizit dimensioniert wurde.
[RCL] [DIM] [A]	**3.0000**	Anzahl der Spalten von **A**.
[$x \gtrless y$]	**2.0000**	Anzahl der Zeilen von **A**.

Ändern der Dimensionen einer Matrix

Matrixelemente werden im Speicher in folgender Reihenfolge abgelegt: Zeile für Zeile, beginnend mit der ersten Zeile, von links nach rechts in jeder Zeile. Beim Verkleinern der Dimensionen einer Matrix werden diese Werte nach dem gleichen Schema der Matrix wieder zugeordnet, übrigbleibende Werte gehen verloren. Beispiel: Neudimensionierung einer 2 × 3-Matrix in eine 2 × 2-Matrix:



Beim Vergrößern der Dimensionen einer Matrix wird den nicht bestimmten zusätzlichen Elementen der Wert 0 zugeteilt. Wenn beispielsweise die folgende 2 × 3-Matrix in eine 2 × 2-Matrix umdimensioniert wird:



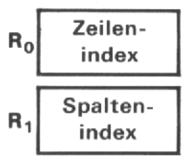
Nach Beendigung Ihrer Matrixoperationen wollen Sie normalerweise alle fünf Matrizen wieder auf die Größe 0 × 0 redimensionieren, damit der Speicherplatz wieder für Programmzeilen oder andere Daten zur Verfügung steht. Mit [f] [MATRIX] 0 werden alle fünf Matrizen gleichzeitig auf 0 × 0 dimensioniert. (Eine einzelne Matrix kann mit der Anweisung 0 [f] [DIM], gefolgt von der Namenstaste, auf 0 × 0 dimensioniert werden.)

Speichern und Abrufen von Matrixelementen

Sie können Matrixelemente auf zwei Arten speichern und abrufen. Mit der ersten Methode haben Sie Zugriff auf alle Elemente in ihrer Reihenfolge. Mit der zweiten Methode können Sie Matrixelemente einzeln bearbeiten.

Speichern und Abrufen aller Elemente in ihrer Reihenfolge

Der HP-15C benutzt im Normalfall die Speicherregister R_0 und R_1, um den Zeilen- und Spaltenindex eines Matrixelements zu bestimmen. Im User-Modus werden die Zeilen- und Spaltenindices beim Abspeichern oder Aufrufen von Matrixelementen *automatisch* inkrementiert, von links nach rechts in jeder Zeile, von der ersten bis zur letzten Zeile.



Mit [f] [MATRIX] 1 setzen Sie die Zeilen- und Spaltenindices in R_0 und R_1 auf 1.

Um Matrizenelemente sequentiell zu speichern oder abzurufen:

1. Stellen Sie sicher, daß die Matrix richtig dimensioniert ist.
2. Drücken Sie [f] [MATRIX] 1. Dies speichert den Wert 1 in R_0 und R_1, so daß der Zugriff bei Zeile 1 und Spalte 1 beginnt.
3. Setzen Sie den User-Modus. In diesem Modus werden die Register R_0 bzw. R_1 nach Bearbeitung jedes Elements automatisch inkrementiert, so daß sie das nächste Matrixelement adressieren. Dazu untenstehendes Beispiel.
4. Falls Sie Elemente abspeichern wollen, geben Sie den Wert ein, der in Zeile 1/Spalte 1 abzulegen ist.
5. Drücken Sie [STO] bzw. [RCL], gefolgt von der Namenstaste der Matrix.
6. Wiederholen Sie die Schritte 4 und 5 für alle Matrixelemente. Die Zeilen- und Spalteninkrementierung orientiert sich an den Dimensionen der Matrix.

Solange die Taste mit dem Buchstaben, der für den Namen der Matrix steht, bei einer [STO] oder [RCL] Operation gedrückt wird, steht in der Anzeige der Name der Matrix und die Zeile und Spalte des zu bearbeitenden Elements. Wird die Namenstaste für länger als etwa 3 Sekunden gedrückt, zeigt der Rechner **null** an; es wird dann kein Wert gespeichert oder abgerufen, Zeilen- und Spaltenindex bleiben unverändert. (Auch der Stack wird nicht verändert.) Nach der Bearbeitung des letzten Matrixelements werden Zeilen- und Spaltenindex auf 1 gesetzt.

Beispiel: Geben Sie die folgenden Werte für die Elemente der zuvor dimensionierten Matrix **A** ein. (Die Matrix **A** muß auf 2 × 3 dimensioniert sein.)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

Tastenfolge	Anzeige	
[f] [MATRIX] 1		Setzt Zeilen- und Spaltenindex in R_0 und R_1 auf 1. (Anzeige enthält das letzte Ergebnis.)
[f] [USER]		Aktiviert User-Modus.
1 [STO] [A]	**A 1,1**	Zeile 1/ Spalte 1 von **A**. Nur angezeigt, solange [A] gedrückt wird.
	1.0000	Wert von a_{11}.
2 [STO] [A]	**2.0000**	Wert von a_{12}.
3 [STO] [A]	**3.0000**	Wert von a_{13}.
4 [STO] [A]	**4.0000**	Wert von a_{21}.
5 [STO] [A]	**5.0000**	Wert von a_{22}.
6 [STO] [A]	**6.0000**	Wert von a_{23}.
[RCL] [A]	**A 1,1**	Ruft Element a_{11} ab. (R_0 und R_1 wurden zuvor zurückgesetzt.)
	1.0000	Wert von a_{11}.
[RCL] [A]	**2.0000**	Wert von a_{12}.
[RCL] [A]	**3.0000**	Wert von a_{13}.
[RCL] [A]	**4.0000**	Wert von a_{21}.
[RCL] [A]	**5.0000**	Wert von a_{22}.
[RCL] [A]	**6.0000**	Wert von a_{23}.
[f] [USER]	**6.0000**	Verlassen des User-Modus.

Überprüfen und Ändern einzelner Matrixelemente

Sie können den Wert eines Matrixelementes auf zwei Arten überprüfen (abrufen) und ändern (speichern). Bei der ersten Methode werden die Datenregister R_0 und R_1 wie oben beschrieben benutzt – außer, daß ihre Werte bei desaktiviertem User-Modus nicht automatisch inkrementiert werden. Bei der zweiten Methode wird der Stack zur Definition der Zeilen/Spaltenindices benutzt.

Verwendung von R_0 und R_1. Um auf ein bestimmtes Matrixelement zuzugreifen, speichern Sie den Zeilenindex in R_0 und den Spaltenindex in R_1 ab. Diese Zahlen werden nicht automatisch inkrementiert (solange sich der Rechner nicht im User-Modus befindet).

- Um den Wert des Elements abzurufen (nachdem Sie dessen Zeilen- und Spaltenindex eingegeben haben), drücken Sie [RCL], gefolgt von der die Matrix bezeichnenden Buchstabentaste.
- Um einen Wert für das Element abzuspeichern (nachdem Sie dessen Zeilen- und Spaltenindex eingegeben haben), laden Sie diesen Wert in das X-Register und drücken Sie [STO], gefolgt von der die Matrix bezeichnenden Buchstabentaste.

Beispiel: Speichern Sie den Wert 9 als Element a_{23} in Zeile 2, Spalte 3 der Matrix **A** vom vorhergehenden Beispiel ab.

Tastenfolge	Anzeige	
2 [STO] 0	**2.0000**	Speichert den Zeilenindex in R_0.
3 [STO] 1	**3.0000**	Speichert den Spaltenindex in R_1.
9	**9**	Eingabe des Wertes des neuen Elements ins X-Register.
[STO] [A]	**A 2,3**	Zeile 2, Spalte 3 von **A**.
	9.0000	Wert von a_{23}.

Verwendung des Stacks. Sie können ein bestimmtes Matrixelement auch mit Hilfe der Stackregister adressieren. Sie brauchen dann nicht die Inhalte von R_0 und R_1 zu ändern.

- Um den Wert eines Elements abzurufen, geben Sie zuerst den Zeilenindex und dann den Spaltenindex in den Stack ein. Drücken Sie danach [RCL] [g], gefolgt von der die Matrix bezeichnenden Buchstabentaste. Der Wert des Elements wird dann ins X-Register geladen (Zeilen- und Spaltenindex gehen dabei verloren).
- Um den Wert eines Elementes zu speichern, geben Sie zuerst den Wert, dann den Zeilenindex und zuletzt den Spaltenindex des Elements in den Stack ein. Drücken Sie dann [STO] [g], gefolgt von der die Matrix bezeichnenden Buchstabentaste. (Zeilen- und Spaltenindex gehen dabei verloren; der Wert des Elements steht anschließend im X-Register.)

Beachten Sie, daß dies die einzigen Operationen sind, bei denen die blaue Vorwahltaste [g] vor einer goldfarbenen Buchstabentaste gedrückt werden muß.

Beispiel: Rufen Sie unter Verwendung des Stack das Element a_{21} der Matrix **A** des vorhergehenden Beispiels ab.

Tastenfolge	**Anzeige**	
2 [ENTER] 1	1	Eingabe des Zeilenindex ins Y-Register und des Spaltenindex ins X-Register.
[RCL] [g] [A]	4.0000	Wert von a_{21}.

Besetzen einer Matrix mit einer Konstanten

Um eine Zahl in allen Matrixelementen abzuspeichern, geben Sie diese Zahl in die Anzeige und drücken Sie [STO] [MATRIX], gefolgt von der die Matrix bezeichnenden Buchstabentaste.

Matrizenoperationen

Matrizenoperationen haben sehr viel mit einfacher Zahlenarithmetik gemeinsam. Bei der Zahlenarithmetik müssen Sie die Zahlen bestimmen, mit denen operiert werden soll; und oft bestimmen Sie auch ein Register, wo das Ergebnis abgelegt werden soll. Bei der Rechnung mit Matrizen müssen Sie ebenfalls die Matrizen bestimmen (mittels eines *Deskriptors*), mit denen operiert werden soll. Für viele Berechnungen müssen Sie weiterhin eine Matrix bestimmen, die das Ergebnis aufnehmen soll. Diese wird im folgenden als *Ergebnismatrix* bezeichnet.

Matrizenoperationen bestehen in der Regel aus einer Vielzahl von Rechenschritten; deshalb wird während der meisten Matrizenoperationen die Anzeige **running** erscheinen.

Matrix-Deskriptoren

Sie haben schon früher in diesem Abschnitt gesehen, daß die Funktion [RCL] [MATRIX], gefolgt von einer die Matrix bezeichnenden Buchstabentaste, den Namen und die Dimensionen einer Matrix zur Anzeige bringt. Der Name der Matrix wird hier als *Deskriptor* der Matrix bezeichnet. Matrix-Deskriptoren können wie Zahlen zwischen Stack- und Datenregistern hin- und hergeschoben werden; benutzen Sie dazu die Funktionen [STO], [RCL], [ENTER] usw. Ein Matrix-Deskriptor erscheint immer zusammen mit der Angabe der derzeitigen Dimensionen der Matrix in der Anzeige.

Mit den Deskriptoren legen Sie fest, welche Matrizen in einer Matrixoperation verwendet werden.

Die im folgenden noch besprochenen Matrizenoperationen wirken auf die Matrizen im X-Register und (bei einigen Operationen) im Y-Register.

Zwei Matrizenoperationen – Berechnung der Determinante und Lösung der Matrixgleichung **AX** = **B** – erfordern die Berechnung einer *LR*-Zerlegung der im X-Register bestimmten Matrix*. Eine Matrix in *LR*-Form wird durch zwei Gedankenstriche gekennzeichnet, die in der Anzeige ihres Deskriptors dem Matrixnamen folgen. (Die Verwendung von Matrizen in *LR*-Form wird auf Seite 160 beschrieben.)

Die Ergebnismatrix

Bei vielen der in diesem Abschnitt besprochenen Operationen muß eine Matrix definiert werden, in der das Ergebnis der Operation abgelegt werden soll. Diese Matrix wird *Ergebnismatrix* genannt.

Andere Matrizenoperationen benötigen *keine* Ergebnismatrix. (Dies wird dann in der Beschreibung dieser Operationen vermerkt.) Diese Operationen legen das Ergebnis der Operation in der Ausgangsmatrix (wenn das Ergebnis eine Matrix ist, z.B. eine Transponierte) oder im X-Register (wenn das Ergebnis ein Skalar, z.B. ein Zeilenindex ist) ab.

Bevor Sie eine Operation durchführen, die eine Ergebnismatrix benutzt, müssen Sie die Ergebnismatrix bestimmen. Drücken Sie dazu [f] [RESULT] und dann die die Matrix bezeichnende Buchstabentaste. (Falls der Deskriptor der vorgesehenen Ergebnismatrix bereits im X-Register steht, drücken Sie stattdessen [STO] [RESULT].) Die spezifizierte Matrix bleibt solange Ergebnismatrix, bis eine andere dazu bestimmt wird**. Um den Deskriptor einer Ergebnismatrix anzuzeigen, drücken Sie [RCL] [RESULT].

Wenn Sie eine Operation durchführen, die die Ergebnismatrix benutzt, wird die Matrix automatisch auf die richtige Größe redimensioniert. Falls diese Redimensionierung mehr zusätzliche Elemente erfordert, als im Matrixspeicher noch vorhanden sind (*maximal* 64 für alle fünf Matrizen), kann die Operation nicht ausgeführt werden.

---

* Die *LR*-Zerlegung einer Matrix ist eine weitere Matrix, die eine untere Dreiecksmatrix **L** und eine obere Dreiecksmatrix **R** enthält, deren Produkt **LR** der Matrix **A** entspricht (möglicherweise mit vertauschten Zeilen). In dem Handbuch «*HP-15C – Fortgeschrittene Funktionen*» wird die LR-Zerlegung ausführlich diskutiert.

** Nach dem Löschen des Permanentspeichers ist Matrix **A** als Ergebnismatrix voreingestellt.

Diese Beschränkung kann oft dadurch umgangen werden, daß Sie eine der Matrizen, die als Operand auftreten, als Ergebnismatrix bestimmen. (Es gibt jedoch Operationen, in denen die Ergebnismatrix nicht gleichzeitig eine der Operandenmatrizen sein darf – dies wird in der Beschreibung der entsprechenden Operationen vermerkt.)

Solange Sie die Taste einer beliebigen Matrixoperation, deren Ergebnis in der Ergebnismatrix gespeichert wird, gedrückt halten, wird der Deskriptor der Ergebnismatrix angezeigt. Wird die Taste innerhalb von drei Sekunden losgelassen, wird die Operation ausgeführt und der Deskriptor der Ergebnismatrix ins X-Register gespeichert. Wird die Taste länger gehalten, wird die Operation nicht ausgeführt, und in der Anzeige erscheint die Meldung **null**.

Kopieren einer Matrix

Mit Hilfe der Tastenfolge [STO] [MATRIX] werden die Elemente einer Matrix in die entsprechenden Elemente einer anderen Matrix kopiert:

1. Drücken Sie [RCL] [MATRIX], gefolgt von der Buchstabentaste mit dem Namen der zu kopierenden Matrix. Der Deskriptor dieser Matrix erscheint in der Anzeige.
2. Drücken Sie [STO] [MATRIX], gefolgt von der Buchstabentaste mit dem Namen der Duplikat-Matrix.

Wenn die Dimensionen der beiden Matrizen nicht übereinstimmen, wird die Duplikat-Matrix auf die Größe des Originals redimensioniert. Die Duplikat-Matrix braucht nicht dimensioniert zu sein, um diese Operation auszuführen.

Beispiel: Kopieren Sie die Matrix **A** des vorhergehenden Beispiels in die Matrix **B**.

Tastenfolge	Anzeige	
[RCL] [MATRIX] [A]	A 2 3	Zeigt den Deskriptor der zu kopierenden Matrix an.
[STO] [MATRIX] [B]	A 2 3	Redimensioniert Matrix **B** und kopiert **A** nach **B**.
[RCL] [MATRIX] [B]	b 2 3	Zeigt den Deskriptor der neuen Matrix an.

Operationen auf einer einzelnen Matrix

Die folgende Tabelle enthält Funktionen, die nur auf der im X-Register spezifizierten Matrix operieren. Operationen auf einer Matrix und einer Zahl in einem anderen Stackregister werden unter Skalare Operationen (Seite 151) beschrieben.

Tastenfolge	Ergebnis im X-Register	Auswirkung auf die im X-Register spezifizierte Matrix	Auswirkung auf die Ergebnismatrix
CHS	Keine Änderung.	Wechselt Vorzeichen aller Elemente.	Keine.***
1/x	Deskriptor der Ergebnis-Matrix.	Keine.***	Inverse der spezifizierten Matrix.****
f MATRIX 4	Deskriptor der Transponierten.	Wird durch die Transponierte ersetzt.	Keine.***
f MATRIX 7	Zeilennorm der spezifizierten Matrix.*	Keine.	Keine.
f MATRIX 8	Euklidische Norm der spezifizierten Matrix.**	Keine.	Keine.
f MATRIX 9	Determinante der spezifizierten Matrix.	Keine.***	*LR*-Zerlegung der spezifizierten Matrix.****

* Die Zeilennorm ist die größte Summe der Absolutwerte der Elemente einer Zeile der spezifizierten Matrix.

** Die Euklidische oder Frobeniusnorm ist die Wurzel über der Summe der Quadrate aller Elemente der spezifizierten Matrix.

*** Falls die Ergebnismatrix nicht die im X-Register spezifizierte Matrix ist.

**** Ist die spezifizierte Matrix *singulär* (d.h. nicht invertierbar), dann verändert der HP-15C die *LR*-Form um einen Betrag, der gewöhnlich klein im Vergleich zum Rundungsfehler ist. Bei 1/x ist die berechnete Inverse die inverse Matrix zu einer vom singulären Original kaum verschiedenen Matrix. (Detaillierte Information dazu finden Sie im Handbuch «*HP-15C – Fortgeschrittene Funktionen*».)

Beispiel: Berechnen Sie die Transponierte von Matrix **B**. Matrix **B** wurde in den vorhergehenden Beispielen wie folgt besetzt:

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix}.$$

Tastenfolge	**Anzeige**	
RCL MATRIX B	b 2 3	Deskriptor der 2 × 3-Matrix **B** in der Anzeige.
f MATRIX 4	b 3 2	Deskriptor der 3 × 2-Transponierten.

Matrix **B** (die im User-Modus mit RCL B abgerufen werden kann) hat nun folgenden Inhalt:

$$\mathbf{B} = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 9 \end{bmatrix}.$$

Skalare Operationen

Skalare Operationen führen arithmetische Operationen mit einem Skalar (einer Zahl) und jedem Element einer Matrix durch. Der Skalar und der Deskriptor der Matrix müssen in den Registern X und Y plaziert werden. (Beachten Sie, daß das Ergebnis der Funktionen − und ÷ von der Position des Skalars und der Matrix im Stack abhängt.) Die Ergebniswerte werden in den entsprechenden Elementen der Ergebnismatrix abgespeichert.

Die folgende Tabelle zeigt die möglichen Operationen.

Operation	Elemente der Ergebnismatrix*: Matrix im Y-Register, Skalar im X-Register	Elemente der Ergebnismatrix*: Skalar im Y-Register, Matrix im X-Register
[+]	Addiert einen skalaren Wert zu jedem Matrixelement.	Addiert einen skalaren Wert zu jedem Matrixelement.
[×]	Multipliziert jedes Matrixelement mit dem Skalar.	Multipliziert jedes Matrixelement mit dem Skalar.
[−]	Subtrahiert den Skalar von jedem Matrixelement.	Subtrahiert jedes Matrixelement vom Skalarwert.
[÷]	Teilt jedes Matrixelement durch den Skalarwert.	Berechnet die Inverse der Matrix und multipliziert jedes Element mit dem Skalar.

* Die spezifizierte Matrix kann als Ergebnismatrix verwendet werden.

Beispiel: Berechnen Sie die Matrix **B** = 2**A**, subtrahieren Sie danach 1 von jedem Element in **B**. Benutzen Sie die Matrix **A** aus dem letzten Beispiel:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix}.$$

Tastenfolge	Anzeige	
[f] [RESULT] [B]		Bestimmt Matrix **B** als Ergebnismatrix.
[RCL] [MATRIX] [A]	A 2 3	Zeigt den Deskriptor der Matrix **A** an.
2 [×]	b 2 3	Redimensioniert **B** auf die Größe von **A**, multipliziert die Elemente von **A** mit 2, speichert diese Werte in den entsprechenden Elementen von **B** und zeigt den Deskriptor der Ergebnismatrix an.

Tastenfolge	Anzeige		
1 [-]	b	2 3	Subtrahiert 1 von den Elementen der Matrix **B** und speichert diese Werte in denselben Elementen von **B**.

Das Ergebnis kann im User-Modus mit [RCL] [B] abgerufen werden und lautet:

$$\mathbf{B} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 17 \end{bmatrix}.$$

Arithmetische Operationen

Wenn Sie die Deskriptoren zweier Matrizen in die Register X und Y eingegeben haben, können Sie mit [+] oder [-] die Summe oder Differenz dieser Matrizen berechnen.

Tasten	Berechnung*
[+]	$\mathbf{Y} + \mathbf{X}$
[-]	$\mathbf{Y} - \mathbf{X}$

*Resultat wird in der Ergebnismatrix abgelegt, die **X** oder **Y** sein kann.

Beispiel: Berechnen Sie **C** = **B**–**A**, wo **A** und **B** die Matrizen des vorhergehenden Beispiels sind.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix} \text{ und } \mathbf{B} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 17 \end{bmatrix}.$$

Tastenfolge	Anzeige		
[f] [RESULT] [C]			Bestimmt **C** als Ergebnismatrix.
[RCL] [MATRIX] [B]	b	2 3	Ruft den Deskriptor der Matrix **B** ab. (Dieser Schritt ist nicht notwendig, wenn der Deskriptor schon im X-Register steht.)
[RCL] [MATRIX] [A]	A	2 3	Ruft den Deskriptor der Matrix **A** ins X-Register, der Deskriptor von **B** wird ins Y-Register geschoben

Tastenfolge	**Anzeige**	
[−]	**C 2 3**	Berechnet **B–A** und speichert die Werte in der redimensionierten Matrix **C**.

Das Ergebnis ist $\mathbf{C} = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 8 \end{bmatrix}$

Multiplikation von Matrizen

Sie können mit Matrix-Deskriptoren in den Registern X und Y drei verschiedene Matrizenprodukte bilden. Die folgende Tabelle zeigt die Ergebnisse der auf einer Matrix **X** im X-Register und einer Matrix **Y** im Y-Register ausgeführten Operationen. $\mathbf{X}^{-1}$ ist die inverse Matrix zu **X**, und $\mathbf{Y}^T$ ist die Transponierte zu **Y**.

Tasten	Berechnung*
[×]	$\mathbf{Y X}$
[f] [MATRIX] 5	$\mathbf{Y}^T\mathbf{X}$
[÷]	$\mathbf{X}^{-1}\mathbf{Y}$

*Das Ergebnis wird in der Ergebnismatrix abgelegt. Bei [÷] kann **Y** (aber nicht **X**) als Ergebnismatrix spezifiziert werden; bei allen Operationen kann weder **X** noch **Y** die Ergebnismatrix sein.

Hinweis: Wenn Sie mittels der Funktion [÷] den Ausdruck $\mathbf{A}^{-1}\mathbf{B}$ berechnen wollen, müssen Sie die Deskriptoren der Matrizen in der umgekehrten Reihenfolge **B**, **A** eingeben.*

Die Werte der Elemente der Ergebnismatrix werden nach den üblichen Regeln der Matrizenmultiplikation berechnet.

In der Operation [MATRIX] 5 wird die im Y-Register spezifizierte Matrix nicht verändert, obwohl ihre Transponierte berechnet wird. Das Ergebnis stimmt mit dem der Nacheinanderausführung von [MATRIX] 4 (Transposition) und [×] überein.

* Diese Reihenfolge würden Sie auch bei der Eingabe von b und a zur Berechnung von $a^{-1}b = b/a$ benutzen.

Bei der Ausführung von [÷] wird die im X-Register spezifizierte Matrix durch ihre *LR*-Zerlegung ersetzt. Die Funktion [÷] berechnet $\mathbf{X}^{-1}\mathbf{Y}$ mittels einer direkteren Methode als [1/x] und [×] und liefert in kürzerer Zeit ein genaueres Ergebnis als diese.

Beispiel: Benutzen Sie die Matrizen **A** und **B** des vorhergehenden Beispiels, um $\mathbf{C} = \mathbf{A}^T\mathbf{B}$ zu berechnen.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 9 \end{bmatrix} \quad \text{und} \quad \mathbf{B} = \begin{bmatrix} 1 & 3 & 5 \\ 7 & 9 & 17 \end{bmatrix}$$

1 4   29 37
2 4   37
3 9

Tastenfolge	Anzeige		
[RCL] [MATRIX] [A]	A	2 3	Aufruf des Deskriptors von **A**.
[RCL] [MATRIX] [B]	b	2 3	Aufruf des Deskriptors der Matrix **B** ins X-Register; verschiebt den Deskriptor von **A** ins Y-Register.
[f] [RESULT] [C]	b	2 3	Bestimmt Matrix **C** als Ergebnismatrix.
[f] [MATRIX] 5	C	3 3	Berechnet $\mathbf{A}^{-1}\mathbf{B}$ und speichert das Ergebnis in der Matrix **C**, die auf 3 × 3 redimensioniert wurde.

Die Ergebnismatrix **C** ist

$$\mathbf{C} = \begin{bmatrix} 29 & 39 & 73 \\ 37 & 51 & 95 \\ 66 & 90 & 168 \end{bmatrix}.$$

Lösen der Gleichung AX = B

Matrizengleichungen der Form **AX** = **B**, wobei **A** die Koeffizientenmatrix, **B** die Konstantenmatrix und **X** die Lösungsmatrix ist, können mittels der Funktion $\boxed{\div}$ gelöst werden. Der Deskriptor der Konstantenmatrix **B** muß im Y-Register und der Deskriptor der Koeffizientenmatrix **A** muß im X-Register stehen. Dann berechnet die Funktion $\boxed{\div}$ die Lösung $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$.*

Y	Konstanten-matrix
X	Koeffizienten-matrix

Denken Sie daran, daß die Funktion $\boxed{\div}$ die Koeffizientenmatrix durch ihre *LR*-Zerlegung ersetzt, und daß diese Matrix nicht als Ergebnismatrix bestimmt werden darf. Weiterhin erhalten Sie mit $\boxed{\div}$ eine schnellere und genauere Lösung als durch die Hintereinanderausführung von $\boxed{1/x}$ und $\boxed{\times}$.

Am Anfang dieses Abschnitts haben Sie die Lösung eines linearen Gleichungssystems berechnet, dessen Konstantenmatrix und Lösungsmatrix aus nur einer Spalte bestand. Im folgenden Beispiel werden Sie sehen, daß Sie mit dem HP-15C auch Lösungen für mehrere Konstantenvektoren, d.h. für eine Konstantenmatrix und eine Lösungsmatrix mit mehr als einer Spalte, berechnen können.

Beispiel: Silas Farmer schreibt folgende Zusammenstellung seiner drei letzten Lieferungen von Kohl und Broccoli.

* Falls **A** singulär ist, verändert der HP-15C die *LR*-Form von **A** um einen im Vergleich zum Rundungsfehler kleinen Betrag. Die berechnete Lösung basiert dann auf einer regulären Matrix, die sich von der singulären Originalmatrix nur geringfügig unterscheidet.

	Woche		
	1	2	3
Gesamtgew. (kg)	274	233	331
Gesamtwert	$120.32	$112.96	$151.36

Silas hat $0.24 pro Kilogramm für seinen Kohl und $0.68 pro Kilogramm für seine Broccoli erhalten. Berechnen Sie mittels Matrizenoperationen die jede Woche ausgelieferte Menge Kohl und Broccoli.

Lösung: Die Lieferung jeder Woche wird von zwei linearen Gleichungen (eine für das Gewicht und eine für den Wert) mit zwei Unbekannten (das Gewicht des Kohls und das des Broccoli) dargestellt. Mit folgender Matrizengleichung können alle drei Wochen gleichzeitig bilanziert werden

$$\begin{bmatrix} 1 & 1 \\ 0.24 & 0.86 \end{bmatrix} \begin{bmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \end{bmatrix} = \begin{bmatrix} 274 & 233 & 331 \\ 120.32 & 112.96 & 151.36 \end{bmatrix}$$

oder

$$\mathbf{AD} = \mathbf{B}$$

wobei die erste Zeile der Matrix **D** das Gewicht des Kohls in den drei Wochen und die zweite Zeile das Gewicht des Broccoli darstellt.

Tastenfolge	**Anzeige**	
2 [ENTER] [f] [DIM] [A]	**2.0000**	Dimensioniert **A** als 2 × 2-Matrix.
[f] [MATRIX] 1	**2.0000**	Setzt Zeilen- und Spaltenindex in R_0 und R_1 auf 1.
[f] [USER]	**2.0000**	Aktiviert User-Modus.
1 [STO] [A]	**1.0000**	Speichert a_{11}.
[STO] [A]	**1.0000**	Speichert a_{12}.
.24 [STO] [A]	**0.2400**	Speichert a_{21}.
.86 [STO] [A]	**0.8600**	Speichert a_{22}.
2 [ENTER] 3 [f] [DIM] [B]	**3.0000**	Dimensioniert **B** als 2 × 3-Matrix.

Tastenfolge	**Anzeige**		
274 [STO] [B]	**274.0000**		Speichert b_{11}.*
233 [STO] [B]	**233.0000**		Speichert b_{12}.
331 [STO] [B]	**331.0000**		Speichert b_{13}.
120.32 [STO] [B]	**120.3200**		Speichert b_{21}.
112.96 [STO] [B]	**112.9600**		Speichert b_{22}.
151.36 [STO] [B]	**151.3600**		Speichert b_{23}.
[f] [RESULT] [D]	**151.3600**		Bestimmt **D** als Ergebnismatrix.
[RCL] [MATRIX] [B]	**b**	**2 3**	Ruft den Deskriptor der Konstantenmatrix ab.
[RCL] [MATRIX] [A]	**A**	**2 2**	Ruft den Deskriptor der Koeffizientenmatrix **A** ins X-Register und schiebt den Deskriptor der Konstantenmatrix **B** ins Y-Register.
[÷]	**d**	**2 3**	Berechnet $\mathbf{A}^{-1}\mathbf{B}$ und speichert das Ergebnis in **D**.
[RCL] [D]	**186.0000**		Ruft d_{11} ab (Gewicht des Kohls der ersten Woche).
[RCL] [D]	**141.0000**		Ruft d_{12} ab (Gewicht des Kohls der zweiten Woche).
[RCL] [D]	**215.0000**		Ruft d_{13} ab.
[CL] [D]	**88.0000**		Ruft d_{21} ab.
[RCL] [D]	**92.0000**		Ruft d_{22} ab.
[RCL] [D]	**116.0000**		Ruft d_{23} ab.
[f] [USER]	**116.0000**		Verlassen des User-Modus.

* Beachten Sie, daß Sie vor dem Abspeichern der Elemente von Matrix **B** nicht [f] [MATRIX] 1 drücken mußten, da nach dem Abspeichern des letzten Elements von **A** der Zeilen- und der Spaltenindex in R_0 und R_1 automatisch auf 1 zurückgesetzt wurden.

Silas Lieferungen waren:

	Woche		
	1	2	3
Kohl (kg)	186	141	215
Broccoli (kg)	88	92	116

Berechnung des Residuums

Mit dem HP-15C können Sie ein Residuum berechnen – d.h. die Matrix

$$\text{Residuum} = \mathbf{R} - \mathbf{YX},$$

wobei **R** die Ergebnismatrix und **X** und **Y** die in den Registern X und Y spezifizierten Matrizen sind.

Diese Operation kann zum Beispiel für iterative Verbesserungen bei der Lösung eines Gleichungssystems oder für lineare Regressionsprobleme eingesetzt werden. Ist zum Beispiel **C** eine mögliche Lösung von **AX** = **B**, dann sagt Ihnen der Ausdruck **B**–**AC**, wie gut diese Lösung die Gleichung erfüllt. (Weitere Informationen zur iterativen Verbesserung von Lösungen und zur linearen Regression finden Sie im Handbuch «*HP-15C – Fortgeschrittene Funktionen*».)

Die Residuumsfunktion ([MATRIX] 6) berechnet das oben definierte Residuum aus dem momentanen Inhalt der Ergebnismatrix und aus den in den Registern X und Y bestimmten Matrizen. Das Residuum wird in der Ergebnismatrix abgespeichert, deren alter Inhalt verlorengeht. Eine in den Registern X oder Y gegebene Matrix kann nicht gleichzeitig Ergebnismatrix sein.

Die Verwendung von [MATRIX] 6 anstelle von [×] und [−] ergibt ein genaueres Ergebnis, vor allem dann, wenn das Residuum klein im Vergleich zu den zu subtrahierenden Matrizen ist.

Berechnung des Residuums:

1. Geben Sie den Deskriptor der Matrix **Y** ins Y-Register ein.
2. Geben Sie den Deskriptor der Matrix **X** ins X-Register ein.
3. Bestimmen Sie die Matrix **R** als Ergebnismatrix.
4. Drücken Sie [f][MATRIX] 6. Das Residuum wird in der ursprünglichen Ergebnismatrix (**R**) abgelegt; der Deskriptor der Ergebnismatrix erscheint im X-Register.

Verwendung von Matrizen in *LR*-Form

Wie schon früher bemerkt, erzeugen zwei Matrizenoperationen (Berechnung der Determinanten und Lösen der Matrizengleichung **AX = B**) eine *LR*-Zerlegung der im X-Register spezifizierten Matrix. Der Deskriptor einer solchen Matrix wird durch zwei Gedankenstriche gekennzeichnet. Die Elemente einer Matrix in *LR*-Form sind von denen der Originalmatrix verschieden.

Der Deskriptor einer Matrix in *LR*-Form kann jedoch anstelle des Originaldeskriptors benutzt werden, um Operationen mit der Inversen oder der Determinante einer Matrix durchzuführen. Das heißt, für folgende Operationen kann entweder die Originalmatrix oder ihre *LR*-Zerlegung benutzt werden:

1/x
÷ für die Matrix im X-Register
MATRIX 9

Diese drei Funktionen liefern das gleiche Ergebnis, wenn anstelle der Originalmatrix ihre *LR*-Zerlegung invertiert wird.

Wenn Sie zum Beispiel die Matrizengleichung **AX = B** lösen, wird Matrix **A** in ihre *LR*-Form zerlegt. Wollen Sie nun dieselbe Gleichung für eine andere Matrix **B** noch einmal lösen, können Sie dies tun, *ohne* die Matrix **A** erneut eingeben zu müssen; Sie erhalten das richtige Ergebnis auch mit der *LR*-Matrix.

Bei allen anderen Matrizenoperationen wird eine *LR*-zerlegte Matrix *nicht* als Darstellung der Originalmatrix erkannt. Stattdessen werden die Elemente der *LR*-Matrix so verwendet, wie sie im Matrizenspeicher stehen, und das Ergebnis ist nicht identisch mit dem Ergebnis, das Sie bei Verwendung der Originalmatrix erhalten würden.

Berechnungen mit komplexen Matrizen

Mit Hilfe des HP-15C können Sie Matrizenmultiplikationen und Matrixinversionen auch mit komplexen Matrizen (Matrizen, deren Elemente komplexe Zahlen sind) durchführen und komplexe Gleichungssysteme (Systeme, deren Koeffizienten und rechten Seiten komplex sind) lösen.

Der HP-15C speichert und verarbeitet allerdings nur reelle Matrizen. Die Berechnungen mit komplexen Matrizen sind völlig unabhängig von den im letzten Abschnitt beschriebenen Rechenoperationen mit komplexen Zahlen.

Sie brauchen nicht in den Komplex-Modus zu schalten, wenn Sie mit komplexen Matrizen rechnen wollen.

Stattdessen werden Operationen mit komplexen Matrizen mit von den ursprünglichen komplexen Matrizen abgeleiteten reellen Matrizen unter Anwendung einiger zusätzlicher Umformungen ausgeführt. Diese Umformungen werden mit Hilfe von vier Rechnerfunktionen durchgeführt. Dieser Abschnitt beschreibt die Durchführung von Operationen auf komplexen Matrizen. (Weitere Beispiele für Berechnungen mit komplexen Matrizen finden Sie im Handbuch «*HP-15C – Fortgeschrittene Funktionen*»).

Abspeichern der Elemente einer komplexen Matrix

Gegeben sei eine komplexe $m \times n$-Matrix $\mathbf{Z} = \mathbf{X} + i\mathbf{Y}$, wobei $\mathbf{X}$ und $\mathbf{Y}$ reelle $m \times n$-Matrizen sind. Diese Matrix kann im Rechner als «geteilte» $2m \times n$-Matrix dargestellt werden:

$$\mathbf{Z}^P = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \begin{matrix} \} \text{ Realteil} \\ \} \text{ Imaginärteil} \end{matrix}$$

Der Index P deutet an, daß die komplexe Matrix durch eine geteilte Matrix dargestellt wird.

Alle Elemente von $\mathbf{Z}^P$ sind reelle Zahlen; die Elemente in der oberen Hälfte repräsentieren die Elemente des Realteils (Matrix $\mathbf{X}$), die der unteren Hälfte repräsentieren die Elemente des Imaginärteils (Matrix $\mathbf{Y}$). Die Elemente von $\mathbf{Z}^P$ werden in der bekannten Weise in einer der fünf Matrizen (zum Beispiel $\mathbf{A}$) abgespeichert, wie zu Beginn dieses Abschnitts beschrieben wurde.

Ist beispielsweise $\mathbf{Z} = \mathbf{X} + i\mathbf{Y}$ mit

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} \quad \text{und } \mathbf{Y} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix},$$

dann kann **Z** im Rechner auf folgende Weise dargestellt werden

$$\mathbf{A} = \mathbf{Z}^P = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \hdashline y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix}$$

Nehmen wir an, Sie wollen eine Berechnung mit einer Matrix durchführen, die nicht als Summe einer reellen und einer imaginären Matrix (so wie obenstehende Matrix **Z**) dargestellt ist, sondern deren Elemente aus komplexen Zahlen bestehen

$$\mathbf{Z} = \begin{bmatrix} x_{11} + iy_{11} & x_{12} + iy_{12} \\ x_{21} + iy_{21} & x_{22} + iy_{22} \end{bmatrix}.$$

Diese Matrix kann im Rechner durch eine sehr ähnlich aussehende Matrix dargestellt werden – ignorieren Sie einfach das *i* und das Plus-Zeichen. Die obenstehende 2×2-Matrix kann im Rechner durch folgende 2×4-Matrix in «Komplexform» dargestellt werden.

$$\mathbf{A} = \mathbf{Z}^C = \begin{bmatrix} x_{11} & y_{11} & x_{12} & y_{12} \\ x_{21} & y_{21} & x_{22} & y_{22} \end{bmatrix}.$$

Der Index *C* bedeutet, daß die komplexe Matrix in einer komplexen «Ersatzform» dargestellt ist.

Obwohl eine komplexe Matrix *zu Beginn* durch eine Matrix derselben Form wie $\mathbf{Z}^C$ dargestellt werden kann, setzen die Operationen zur Multiplikation und Inversion komplexer Matrizen eine Darstellung der Form $\mathbf{Z}^P$ voraus. Der HP-15C verfügt über zwei Transformationen, die Ihnen den Übergang von der $\mathbf{Z}^C$-in die $\mathbf{Z}^P$-Darstellung komplexer Matrizen und umgekehrt ermöglichen.

Tasten	Transform.	in
f Py,x	$\mathbf{Z}^C$	$\mathbf{Z}^P$
g Cy,x	$\mathbf{Z}^P$	$\mathbf{Z}^C$

Um eine dieser Transformationen durchzuführen, rufen Sie den Deskriptor von $\mathbf{Z}^C$ oder $\mathbf{Z}^P$ in die Anzeige und drücken Sie die oben angegebenen Tasten.

Die Transformation wird mit der angegebenen Matrix durchgeführt; die Ergebnismatrix bleibt unverändert.

Beispiel: Speichern Sie die komplexe Matrix

$$\mathbf{Z} = \begin{bmatrix} 4+3i & 7-2i \\ 1+5i & 3+8i \end{bmatrix}$$

die in der Form $\mathbf{Z}^C$ gegeben ist, in dieser Form ab; und transformieren Sie sie dann in die Form $\mathbf{Z}^P$.

Dazu können Sie die Elemente der Matrix $\mathbf{Z}^C$ in die Matrix **A** abspeichern und dann die Funktion [Py,x] anwenden, wobei

$$\mathbf{A} = \mathbf{Z}^C = \begin{bmatrix} 4 & 3 & 7 & -2 \\ 1 & 5 & 3 & 8 \end{bmatrix}.$$

Tastenfolge	**Anzeige**		
[f] [MATRIX] 0			Löscht alle Matrizen.
2 [ENTER] 4 [f] [DIM] [A]	**4.0000**		Dimensioniert **A** auf 2×4.
[f] [MATRIX] 1	**4.0000**		Initialisiert R_0 und R_1 auf 1.
[f] [USER]	**4.0000**		Aktiviert den User-Modus.
4 [STO] [A]	**4.0000**		Speichert a_{11}.
3 [STO] [A]	**3.0000**		Speichert a_{12}.
7 [STO] [A]	**7.0000**		Speichert a_{13}.
2 [CHS] [STO] [A]	**–2.0000**		Speichert a_{14}.
1 [STO] [A]	**1.0000**		Speichert a_{21}.
5 [STO] [A]	**5.0000**		Speichert a_{22}.
3 [STO] [A]	**3.0000**		Speichert a_{23}.
8 [STO] [A]	**8.0000**		Speichert a_{24}.
[f] [USER]	**8.0000**		Verlassen des User-Modus.
[RCL] [MATRIX] [A]	**A**	**2 4**	Zeigt den Deskriptor der Matrix **A** an.
[f] [Py,x]	**A**	**4 2**	Transformiert $\mathbf{Z}^C$ in $\mathbf{Z}^P$ und redimensioniert Matrix **A**.

Matrix **A** stellt nun die komplexe Matrix **Z** in $\mathbf{Z}^P$-Form dar.

$$\mathbf{A} = \mathbf{Z}^P = \left[\begin{array}{cc} 4 & 7 \\ 1 & 3 \\ \hdashline 3 & -2 \\ 5 & 8 \end{array}\right] . \quad \begin{array}{l} \left.\vphantom{\begin{array}{c}4\\1\end{array}}\right\} \text{Realteil} \\ \left.\vphantom{\begin{array}{c}3\\5\end{array}}\right\} \text{Imaginärteil} \end{array}$$

Komplexe Transformationen

Um das Produkt zweier komplexer Matrizen zu berechnen, müssen Sie eine weitere Transformation durchführen (das gleiche gilt für die Invertierung einer komplexen Matrix). Diese Transformationen bewirken die Umwandlungen zwischen der $\mathbf{Z}^P$-Darstellung einer komplexen $m \times n$-Matrix und einer $2m \times 2n$-Matrix der folgenden Form

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \mathbf{X} & -\mathbf{Y} \\ \mathbf{Y} & \mathbf{X} \end{bmatrix} .$$

Die Matrix $\tilde{\mathbf{Z}}$, die durch die Transformation MATRIX 2 erzeugt wird, besteht aus doppelt sovielen Elementen wie $\mathbf{Z}^P$.

Folgende Beispielmatrizen zeigen den Zusammenhang zwischen $\tilde{\mathbf{Z}}$ und $\mathbf{Z}^P$.

$$\mathbf{Z}^P = \left[\begin{array}{cc} 1 & -6 \\ \hdashline -4 & 5 \end{array}\right] \quad \longleftrightarrow \quad \tilde{\mathbf{Z}} = \left[\begin{array}{cc;{2pt/2pt}cc} 1 & -6 & 4 & -5 \\ \hdashline -4 & 5 & 1 & -6 \end{array}\right]$$

Die folgende Tabelle führt die Transformationen auf, die die $\mathbf{Z}^P$-Darstellung einer komplexen Matrix in die $\tilde{\mathbf{Z}}$-Form und umgekehrt umwandeln.

Tasten	Transform.	in
f MATRIX 2	$\mathbf{Z}^P$	$\tilde{\mathbf{Z}}$
f MATRIX 3	$\tilde{\mathbf{Z}}$	$\mathbf{Z}^P$

Um eine dieser Transformationen auszuführen, rufen Sie den Deskriptor von $\mathbf{Z}^P$ oder $\tilde{\mathbf{Z}}$ in die Anzeige, und drücken Sie dann die oben angegebenen Tasten. Die Transformation wird mit der spezifizierten Matrix ausgeführt; die Ergebnismatrix wird nicht verändert.

Invertierung einer komplexen Matrix

Die Beziehung $(\tilde{\mathbf{Z}})^{-1} = (\widetilde{\mathbf{Z}^{-1}})$ kann zur Berechnung der Inversen einer komplexen Matrix benutzt werden.

Um die Inverse, $\mathbf{Z}^{-1}$, einer komplexen Matrix $\mathbf{Z}$ zu berechnen:

1. Speichern Sie die Elemente von $\mathbf{Z}$ entweder in der Form $\mathbf{Z}^P$ oder $\mathbf{Z}^C$ ab.
2. Rufen Sie den Deskriptor der $\mathbf{Z}$ darstellenden Matrix in die Anzeige.
3. Falls $\mathbf{Z}$ in der Darstellung $\mathbf{Z}^C$ eingegeben wurde, drücken Sie [f] [Py,x], um $\mathbf{Z}^C$ in $\mathbf{Z}^P$ zu transformieren.
4. Drücken Sie [f] [MATRIX] 2, um $\mathbf{Z}^P$ in $\tilde{\mathbf{Z}}$ zu transformieren.
5. Bestimmen Sie eine Ergebnismatrix. Diese kann dieselbe Matrix sein, in welcher auch $\tilde{\mathbf{Z}}$ gespeichert ist.
6. Drücken Sie [1/x]. Damit berechnen Sie die Matrix $(\tilde{\mathbf{Z}})^{-1}$, die identisch ist mit $(\widetilde{\mathbf{Z}^{-1}})$. Die Werte dieser Matrixelemente werden in der Ergebnismatrix gespeichert, und der Deskriptor der Ergebnismatrix erscheint im X-Register.
7. Drücken Sie [f] [MATRIX] 3, um $(\widetilde{\mathbf{Z}^{-1}})$ in $(\mathbf{Z}^{-1})^P$ zu transformieren.
8. Wenn Sie die Inverse in der Form $(\mathbf{Z}^{-1})^C$ dargestellt haben wollen, drücken Sie [g] [Cy,x].

Sie können die komplexen Elemente von $\mathbf{Z}^{-1}$ rekonstruieren, indem Sie die Elemente von $\mathbf{Z}^P$ oder $\mathbf{Z}^C$ abrufen und wie zuvor beschrieben kombinieren.

Beispiel: Berechnen Sie die Inverse der komplexen Matrix $\mathbf{Z}$ aus dem vorhergehenden Beispiel,

$$\mathbf{A} = \mathbf{Z}^P = \begin{bmatrix} 4 & 7 \\ 1 & 3 \\ \hdashline 3 & -2 \\ 5 & 8 \end{bmatrix}.$$

Tastenfolge	**Anzeige**		
[RCL] [MATRIX] [A]	**A**	**4 2**	Ruft den Deskriptor der Matrix **A** ab.
[f] [MATRIX] 2	**A**	**4 4**	Transformiert $\mathbf{Z}^P$ in $\tilde{\mathbf{Z}}$ und redimensioniert die Matrix **A**.

Tastenfolge	Anzeige		
f RESULT B	A	4 4	Bestimmt **B** als Ergebnismatrix.
1/x	b	4 4	Berechnet $(\tilde{\mathbf{Z}})^{-1} = (\widetilde{\mathbf{Z}^{-1}})$ und speichert das Ergebnis in Matrix **B**.
f MATRIX 3	b	4 2	Transformiert $(\mathbf{Z}^{-1})$ in $(\mathbf{Z}^{-1})^P$.

Die geteilte Darstellung von $\mathbf{Z}^{-1}$ ist in Matrix **B** abgelegt.

$$\mathbf{B} = \left[\begin{array}{rr} -0.0254 & 0.2420 \\ -0.0122 & -0.1017 \\ \hdashline -0.2829 & -0.0022 \\ 0.1691 & -0.1315 \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{r}0\\0\end{array}}\right\} \text{Realteil} \\ \left.\vphantom{\begin{array}{r}0\\0\end{array}}\right\} \text{Imaginärteil} \end{array}$$

Multiplikation komplexer Matrizen

Das Produkt zweier komplexer Matrizen kann über die Beziehung $(\mathbf{YX})^P = \tilde{\mathbf{Y}}\mathbf{X}^P$ berechnet werden. Zur Berechnung von **YX**, wobei **Y** und **X** komplexe Matrizen sind:

1. Speichern Sie die Elemente von **Y** und **X** in einer der Darstellungen $\mathbf{Z}^P$ oder $\mathbf{Z}^C$.
2. Rufen Sie den Deskriptor der **Y** repräsentierenden Matrix in die Anzeige.
3. Falls die Elemente von **Y** in der Form $\mathbf{Y}^C$ eingegeben wurden, drücken Sie f Py,x, um $\mathbf{Y}^C$ in $\mathbf{Y}^P$ zu transformieren.
4. Drücken Sie f MATRIX 2, um $\mathbf{Y}^P$ in $\tilde{\mathbf{Y}}$ umzuformen.
5. Rufen Sie den Deskriptor der **X** darstellenden Matrix in die Anzeige.
6. Falls die Elemente von **X** in der Form $\mathbf{X}^C$ eingegeben wurden, drücken Sie f Py,x, um $\mathbf{X}^C$ in $\mathbf{X}^P$ zu transformieren.
7. Bestimmen Sie die Ergebnismatrix; diese darf nicht eine der beiden Operandenmatrizen sein.

8. Drücken Sie [×], um $\tilde{\mathbf{Y}}\mathbf{X}^P = (\mathbf{YX})^P$ zu berechnen. Die Werte dieser Matrixelemente werden in der Ergebnismatrix abgelegt, und der Deskriptor der Ergebnismatrix wird ins X-Register gebracht.
9. Wenn Sie das Produkt in der Darstellung $(\mathbf{YX})^C$ wünschen, drücken Sie [g] [Cy,x].

Beachten Sie, daß $\mathbf{X}^P$ nicht in $\tilde{\mathbf{X}}$ umgewandelt wird.

Sie können die komplexen Elemente der Produktmatrix **YX** finden, indem Sie die Elemente von $(\mathbf{YX})^P$ oder $(\mathbf{YX})^C$ abrufen und nach den zuvor beschriebenen Regeln kombinieren.

Beispiel: Berechnen Sie das Produkt $\mathbf{ZZ}^{-1}$, wo **Z** die komplexe Matrix des vorhergehenden Beispiels ist.

Da die Elemente beider Matrizen schon gespeichert sind, überspringen Sie die Schritte 1, 3, 4 und 6.

Tastenfolge	**Anzeige**		
[RCL] [MATRIX] [A]	A	4 4	Zeigt den Deskriptor der Matrix **A** an.
[RCL] [MATRIX] [B]	b	4 2	Zeigt den Deskriptor der Matrix **B** an.
[f] [RESULT] [C]	b	4 2	Bestimmt **C** als Ergebnismatrix.
[×]	C	4 2	Berechnet $\tilde{\mathbf{Z}}(\mathbf{Z}^{-1})^P = (\mathbf{ZZ}^{-1})^P$.
[f] [USER]	C	4 2	Schaltet in den User-Modus.
[RCL] [C]	C 1,1		Matrix **C**, Zeile 1, Spalte 1. (Kurzzeitig angezeigt, solange die letzte Taste gedrückt gehalten wird.)
	1.0000		Wert von c_{11}.
[RCL] [C]	–2.8500	–10	Wert von c_{12}.
[RCL] [C]	–4.0000	–11	Wert von c_{21}.
[RCL] [C]	1.0000		Wert von c_{22}.
[RCL] [C]	1.0000	–11	Wert von c_{31}.
[RCL] [C]	3.8000	–10	Wert von c_{32}.
[RCL] [C]	1.0000	–11	Wert von c_{41}.
[RCL] [C]	–1.0500	–10	Wert von c_{42}.
[f] [USER]	–1.0500	–10	Verlassen des User-Modus.

Die Elemente der Matrix **C** sind

$$\mathbf{C} = \left[\begin{array}{cc} 1.0000 & -2.8500 \times 10^{-10} \\ -4.0000 \times 10^{-11} & 1.0000 \\ \hdashline 1.0000 \times 10^{-11} & 3.8000 \times 10^{-10} \\ 1.0000 \times 10^{-11} & -1.0500 \times 10^{-10} \end{array}\right] = (\mathbf{ZZ}^{-1})^P,$$

wobei die obere Hälfte der Matrix **C** dem reellen Teil von $\mathbf{ZZ}^{-1}$ und die untere Hälfte dem imaginären Teil entspricht. Zerlegung der Matrix **C** ergibt

$$\mathbf{ZZ}^{-1} = \begin{bmatrix} 1.0000 & -2.8500 \times 10^{-10} \\ -4.0000 \times 10^{-11} & 1.0000 \end{bmatrix}$$

$$+ i \begin{bmatrix} 1.0000 \times 10^{-11} & 3.8000 \times 10^{-11} \\ 1.0000 \times 10^{-11} & -1.0500 \times 10^{-10} \end{bmatrix}$$

Wie erwartet,

$$\mathbf{ZZ}^{-1} \cong \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + i \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Lösung der komplexen Gleichung $\mathbf{AX} = \mathbf{B}$

Sie können die komplexe Matrixgleichung $\mathbf{AX} = \mathbf{B}$ durch Berechnung von $\mathbf{X} = \mathbf{A}^{-1}\mathbf{B}$ lösen. Berechnen Sie dazu $\mathbf{X}^P = (\tilde{\mathbf{A}})^{-1}\mathbf{B}^P$.

Um die Gleichung $\mathbf{AX} = \mathbf{B}$ zu lösen, wobei die Matrizen **A**, **B** und **X** komplexe Matrizen sind:

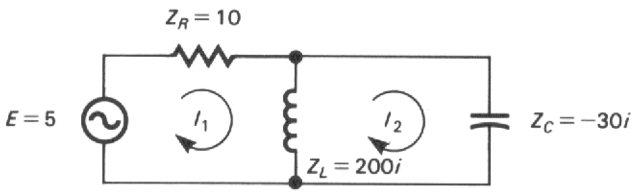
1. Speichern Sie die Elemente von **A** und **B** entweder in der Form $\mathbf{Z}^P$ oder $\mathbf{Z}^C$ ab.
2. Rufen Sie den Deskriptor der **B** darstellenden Matrix in die Anzeige.
3. Falls die Elemente von **B** in der Form $\mathbf{B}^C$ eingegeben wurden, drücken Sie $\boxed{f}$ $\boxed{Py,x}$, um $\mathbf{B}^C$ in $\mathbf{B}^P$ umzuformen.

4. Rufen Sie den Deskriptor der **A** repräsentierenden Matrix in die Anzeige.
5. Falls die Elemente von **A** in der Form $\mathbf{A}^C$ eingegeben wurden, drücken Sie [f] [Py,x] um $\mathbf{A}^C$ in $\mathbf{A}^P$ umzuformen.
6. Drücken Sie [f] [MATRIX] 2, um $\mathbf{A}^P$ in $\tilde{\mathbf{A}}$ umzuformen.
7. Spezifizieren Sie die Ergebnismatrix; diese darf nicht mit der **A** repräsentierenden Matrix übereinstimmen.
8. Drücken Sie [÷]; dies bewirkt Berechnung von $\mathbf{X}^P$. Die Werte dieser Matrix werden in der Ergebnismatrix abgespeichert, und der Deskriptor der Ergebnismatrix wird im X-Register abgelegt.
9. Wenn Sie die Lösung in der Form $\mathbf{X}^C$ wünschen, drücken Sie [g] [Cy,x].

Beachten Sie, daß $\mathbf{B}^P$ nicht in $\tilde{\mathbf{B}}$ umgeformt wird.

Sie können die komplexen Elemente der Lösung gewinnen, indem Sie die Elemente von $\mathbf{X}^P$ oder $\mathbf{X}^C$ abrufen und den zuvor beschriebenen Regeln entsprechend kombinieren.

Beispiel: Der Student der Elektrotechnik A.C. Dimmer möchte den folgenden elektrischen Schaltkreis berechnen. Die Impedanzen der Bauteile sind in komplexer Form gegeben. Bestimmen Sie die Ströme I_1 und I_2 in komplexer Darstellung.



Dieser Schaltkreis kann durch die komplexe Matrizengleichung

$$\begin{bmatrix} 10+200i & -200i \\ -200i & (200-30)i \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 0 \end{bmatrix}$$

oder

$$\mathbf{AX} = \mathbf{B}$$

dargestellt werden.

In geteilter Form gilt:

$$\mathbf{A} = \left[\begin{array}{cc} 10 & 0 \\ 0 & 0 \\ \hdashline 200 & -200 \\ -200 & 170 \end{array}\right] \text{ und } \mathbf{B} = \left[\begin{array}{c} 5 \\ 0 \\ \hdashline 0 \\ 0 \end{array}\right],$$

wobei die Elemente mit dem Wert Null Real- und Imaginärteilen mit Wert Null entsprechen.

Tastenfolge	Anzeige	
4 [ENTER] 2 [f] [DIM] [A]	2.0000	Dimensioniert die Matrix **A** auf 4 × 2.
[f] [MATRIX] 1	2.0000	Setzt die Anfangszeile und -Spalte in R_0 und R_1 auf 1.
[f] [USER]	2.0000	Schaltet in den User-Modus.
10 [STO] [A]	10.0000	Speichert a_{11}.
0 [STO] [A]	0.0000	Speichert a_{12}.
[STO] [A]	0.0000	Speichert a_{21}.
[STO] [A]	0.0000	Speichert a_{22}.
200 [STO] [A]	200.0000	Speichert a_{31}.
[CHS] [STO] [A]	-200.0000	Speichert a_{32}.
[STO] [A]	-200.0000	Speichert a_{41}.
170 [STO] [A]	170.0000	Speichert a_{42}.
4 [ENTER] 1 [f] [DIM] [B]	1.0000	Dimensioniert die Matrix **B** als 4 × 1-Matrix.
0 [STO] [MATRIX] [B]	0.0000	Speichert den Wert 0 in allen Elementen von **B**.
5 [ENTER] 1 [ENTER]	1.0000	Spezifiziert den Wert 5 für das Element in Zeile 1, Spalte 1.
[STO] [g] [B]	5.0000	Speichert den Wert 5 in b_{11}.
[RCL] [MATRIX] [B]	b 4 1	Ruft den Deskriptor der Matrix **B** ab.
[RCL] [MATRIX] [A]	A 4 2	Lädt den Deskriptor der Matrix **A** ins X-Register und schiebt den Deskriptor der Matrix **B** ins Y-Register.

Tastenfolge	Anzeige		
f MATRIX 2	A	4 4	Transformiert $\mathbf{A}^P$ in $\tilde{\mathbf{A}}$.
f RESULT C	A	4 4	Spezifiziert Matrix **C** als Ergebnismatrix.
÷	C	4 1	Berechnet $\mathbf{X}^P$ und speichert deren Werte in **C**.
g Cy,x	C	2 2	Transformiert $\mathbf{X}^P$ in $\mathbf{X}^C$.
RCL C	0.0372		Ruft c_{11} ab.
RCL C	0.1311		Ruft c_{12} ab.
RCL C	0.0437		Ruft c_{21} ab.
RCL C	0.1543		Ruft c_{22} ab.
f USER	0.1543		Verlassen des User-Modus.
f MATRIX 0	0.1543		Alle Matrizen werden auf 0×0 redimensioniert.

Die von der komplexen Matrix **X** dargestellten Ströme können aus **C** abgelesen werden:

$$\mathbf{X} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} = \begin{bmatrix} 0.0372 + 0.1311i \\ 0.0437 + 0.1543i \end{bmatrix}.$$

Sie haben zur Lösung des vorhergehenden Beispiels 24 Register des Matrizenspeichers belegt – 16 für die 4×4 Matrix **A** (die ursprünglich als eine 4×2 Matrix eingegeben wurde und eine komplexe 2×2 Matrix repräsentiert) und je vier für die Matrizen **B** und **C** (die jeweils eine komplexe 2×1 Matrix darstellen). (Wenn Sie Matrix **B** als Ergebnismatrix spezifizieren, benötigen Sie vier Register weniger.) Beachten Sie, daß **X** und **B** nicht auf Vektoren (d.h., einspaltige Matrizen) beschränkt sind und deshalb unter Umständen auch mehr Speicherplatz belegen können.

Der HP-15C verfügt über genügend Speicherraum, um mit der oben beschriebenen Methode die komplexe Matrizengleichung **AX** = **B** für Matrizen **X** und **B** mit bis zu sechs Spalten zu lösen, falls **A** eine 2×2 Matrix ist. Ist **A** eine 3×3 Matrix, so können **X** und **B** maximal zwei Spalten besitzen*. (Die zulässige Anzahl von Spalten verdoppelt sich, wenn **B** als Ergebnismatrix bestimmt wird.) Falls **X** und **B** mehr Spalten aufweisen, oder falls **A** eine 4×4 Matrix ist, können Sie die Gleichung mit der folgenden Alternativmethode lösen.

*Sofern der gesamte verfügbare Speicherplatz dem Common Pool zugeordnet ist (MEM: **1 64 0-0**). Siehe Anhang C, Speicheraufteilung.

Bei dieser Methode werden im Unterschied zur vorhergehenden Invertierung und Multiplikation getrennt und daher weniger Register benötigt.

1. Speichern Sie alle Elemente von **A** in der Form $\mathbf{A}^P$ oder $\mathbf{A}^C$ ab.
2. Rufen Sie den Deskriptor der **A** darstellenden Matrix in die Anzeige.
3. Falls die Elemente von **A** in der Form $\mathbf{A}^C$ eingegeben wurden, drücken Sie f Py,x um $\mathbf{A}^C$ in $\mathbf{A}^P$ zu transformieren.
4. Drücken Sie f MATRIX 2, um $\mathbf{A}^P$ in $\tilde{\mathbf{A}}$ zu transformieren.
5. Drücken Sie STO RESULT, um die **A** repräsentierende Matrix als Ergebnismatrix zu bestimmen.
6. Drücken Sie 1/x um $(\tilde{\mathbf{A}})^{-1}$ zu berechnen.
7. Redimensionieren Sie **A** auf die Hälfte der Zeilen, die im Deskriptor der Matrix angezeigt werden.
8. Speichern Sie die Elemente von **B** in der Form $\mathbf{B}^P$ oder $\mathbf{B}^C$ ab.
9. Rufen Sie den Deskriptor der **A** repräsentierenden Matrix in die Anzeige.
10. Rufen Sie den Deskriptor der **B** repräsentierenden Matrix in die Anzeige.
11. Falls die Elemente von **B** in der Darstellung $\mathbf{B}^C$ eingegeben wurden, transformieren Sie sie mit f Py,x in die Form $\mathbf{B}^P$.
12. Drücken Sie f MATRIX 2 um $\mathbf{B}^P$ in $\tilde{\mathbf{B}}$ zu transformieren.
13. Spezifizieren Sie die Ergebnismatrix; sie darf keine der Operandenmatrizen sein.
14. Drücken Sie ×.
15. Drücken Sie f MATRIX 4, um die Ergebnismatrix zu transponieren.
16. Drücken Sie f MATRIX 2.
17. Redimensionieren Sie die Ergebnismatrix auf die Hälfte der Zeilen, die im Deskriptor der Matrix nach dem letzten Schritt angezeigt wurden.
18. Drücken Sie RCL RESULT, um den Deskriptor der Ergebnismatrix anzuzeigen.

19. Drücken Sie [f] [MATRIX] 4 um $\mathbf{X}^P$ zu berechnen.
20. Wenn Sie die Lösung in der Darstellung $\mathbf{X}^C$ haben wollen, drücken Sie [g] [Cy,x].

Das Handbuch «*HP-15C – Fortgeschrittene Funktionen*» enthält eine Anwendung, bei deren Lösung diese Methode angewandt wird; siehe «Lösung eines großen komplexen Gleichungssystems» in dem genannten Handbuch.

Verschiedene Operationen mit Matrizen

Anwendung von Registeroperationen auf Matrixelemente

Wenn Sie nach einer der folgenden Funktionstasten eine eine Matrix bezeichnende Buchstabentaste drücken, wird die Operation auf dem durch die Register R_0 und R_1 indizierten Matrixelement ausgeführt, als ob es ein Datenregister wäre.

[STO]*	[RCL]*
[STO] {[+], [−], [×], [÷]}	[RCL] {[+], [−], [×], [÷]}
[DSE]	[ISG]
[x⋛]	

Matrix-Deskriptoren im Indexregister

Bei bestimmten Applikationen kann es sinnvoll sein, eine vorprogrammierte Sequenz von Matrixoperationen auf den Matrizen **A** bis **E** auszuführen. In diesen Fällen können Sie durch Spezifikation eines Matrix-Deskriptors im Indexregister, die jeweils als Operandenmatrizen zu verwendenden Matrizen definieren.

Mit einem Matrix-Deskriptor im Indexregister können Sie folgende Operationen ausführen:

- Durch Drücken von [(i)] nach einer der oben aufgezählten Funktionen wird diese Operation auf dem in R_0 und R_1 indizierten Element der in R_I gegebenen Matrix ausgeführt.
- Durch Drücken von [(i)] nach [STO] [g] oder nach [RCL] [g] wird diese Operation auf dem Element der in R_I gegebenen Matrix ausgeführt, dessen Zeilen- und Spaltenindex in den Registern Y und X enthalten sind.

---

* Im User-Modus werden die Zeilen- und Spaltenindices in R_0 und R_1 den Matrixdimensionen entsprechend inkrementiert.

- Durch Drücken von [f] [DIM] [I] wird die in R_I gegebene Matrix auf die in den Registern X und Y vorhandenen Werte dimensioniert.
- Durch Drücken von [RCL] [DIM] [I] werden die Dimensionen der in R_I gegebenen Matrix aus den Registern X und Y abgerufen.
- Das Drücken von [GSB] [I] oder [GTO] [I] bewirkt die gleiche Operation wie [GSB] oder [GTO] gefolgt von dem Namen der in R_I gegebenen Matrix. (Dies ist keine Matrixoperation im eigentlichen Sinne; nur der Name der Matrix wird benutzt.)

Vergleichsoperationen mit Matrix-Deskriptoren

Mit Matrix-Deskriptoren in den Registern X und Y können Sie vier Vergleiche durchführen – [x=0], [TEST] 0 ($x \neq 0$), [TEST] 5 ($x \neq y$) und [TEST] 6 ($x \neq y$). Vergleiche können zur Steuerung des Programmablaufs benutzt werden (siehe Abschnitt 8).

Befindet sich ein Deskriptor einer Matrix im X-Register, so ist das Ergebnis von [x=0] immer «falsch» (= false) und das Ergebnis von [TEST] 0 immer «wahr» (= true) (unabhängig von den Werten der Matrixelemente).

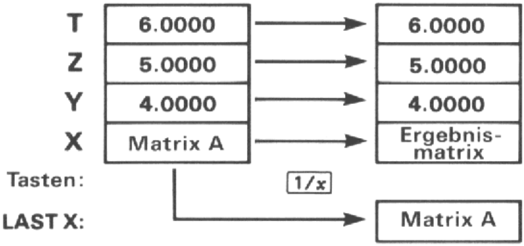
Sind bei der Ausführung der Vergleiche [TEST]5 und [TEST]6 Matrix-Deskriptoren in den Registern X und Y, so sind x und y genau dann gleich, wenn die beiden Deskriptoren identisch sind. Der Vergleich wird *zwischen den Deskriptoren, nicht zwischen den Elementen* der gegebenen Matrizen ausgeführt.

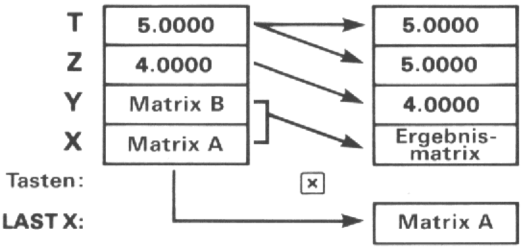
Andere Vergleiche sind mit Matrix-Deskriptoren nicht möglich.

Stackbewegungen bei Matrizenoperationen

Das Verhalten des Stack bei Matrizenoperationen ist dem Verhalten bei skalaren Berechnungen sehr ähnlich.

Bei einigen Matrizenoperationen wird das Ergebnis in der Ergebnismatrix abgespeichert. Die Argumente, ein oder zwei Deskriptoren oder Zahlen im X-Register oder in den Registern X und Y, werden in der Operation kombiniert, und der Deskriptor der Ergebnismatrix wird im X-Register abgelegt. (Das Argument im X-Register bleibt im LAST X-Register erhalten.)



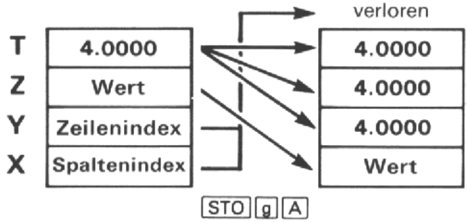


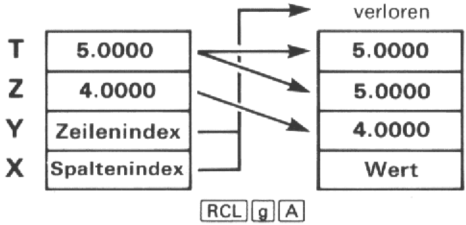
Verschiedene Matrizenfunktionen operieren nur auf der im X-Register gegebenen Matrix und speichern das Ergebnis in der gleichen Matrix ab. In diesen Operationen wird der Stack (einschließlich LAST X-Register) nicht bewegt, obwohl sich gegebenenfalls die Anzeige ändert, um die neuen Dimensionen anzuzeigen.

Die Funktionen MATRIX 7, MATRIX 8 und MATRIX 9 retten den im X-Register enthaltenen Deskriptor ins LAST X-Register und laden die Norm oder (bei MATRIX 9) die Determinante ins X-Register. Die Register Y, Z und T bleiben unverändert.

Wenn Sie (bei freigegebenem Stack) Deskriptoren oder Matrixelemente ins X-Register laden, werden Deskriptoren und Zahlen, die sich bereits im Stack befinden, angehoben, und der Inhalt des T-Registers geht verloren. (Das LAST X Register wird nicht beeinflußt.) Beim Abspeichern von Deskriptoren oder Matrixelementen bleiben Stack und LAST X-Register unverändert.

Im Gegensatz zu den oben beschriebenen Operationen beeinflussen die Funktionen STO g und RCL g das LAST X-Register nicht; ihre Funktionsweise wird im folgenden beschrieben.

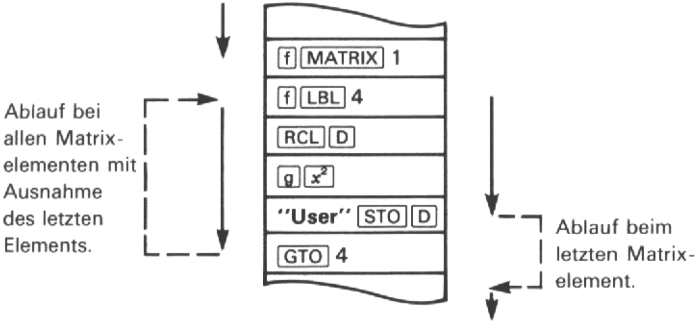




Matrizenoperationen in Programmen

Befindet sich der Rechner während der Programmeingabe im User-Modus, wenn Sie die Anweisung STO oder RCL {A bis E, (i)} eintasten, um ein Matrixelement zu speichern oder abzurufen, dann wird der Bindestrich nach der Zeilennummer durch ein **u** ersetzt. Bei der Programmausführung wird die Anweisung dieser Zeile dann so interpretiert, als ob sich der Rechner im User-Modus befände; d.h., die Zeilen- und Spaltennummern in R_0 und R_1 werden automatisch entsprechend den Dimensionen der gegebenen Matrix inkrementiert. Damit ist es möglich, sequentiell auf Matrixelemente zuzugreifen. (Die Statusanzeige **USER** hat keine Auswirkungen auf die Programmausführung.)

Nachdem das letzte Element mit der «User»-Anweisung STO oder RCL bearbeitet ist und R_0 und R_1 auf 1 zurückgesetzt sind, wird in der Programmausführung die nächste Zeile übersprungen. Dies ermöglicht die Programmierung einer Programmschleife, die alle Matrixelemente speichert oder abruft und danach das Programm fortsetzt. Das folgende Beispiel-Programm quadriert alle Elemente der Matrix **D**:



Die Funktionen MATRIX 7 (Zeilennorm) und MATRIX 8 (Frobenius-Norm) können in Programmen auch als bedingte Verzweigungen benutzt werden. Wenn das X-Register einen Matrix-Deskriptor enthält, berechnen diese Funktionen wie gewöhnlich die Norm, und setzen die Programmausführung bei der folgenden Zeile fort. Enthält das X-Register jedoch eine Zahl, wird bei der Programmausführung die nächste Zeile übersprungen. In beiden Fällen bleibt der Inhalt des X-Registers im LAST X-Register erhalten. Damit können Sie während der Programmausführung testen, ob ein Matrix-Deskriptor im X-Register steht.

Zusammenfassung der Matrizenfunktionen

Tastenfolge	Anzeige
g Cy,x	Transformiert $\mathbf{Z}^P$ in $\mathbf{Z}^C$.
CHS	Ändert das Vorzeichen aller Elemente der im X-Register gegebenen Matrix.
f DIM {A bis E, I}	Dimensioniert die gegebene Matrix.
f MATRIX 0	Dimensioniert alle Matrizen auf 0 × 0.
f MATRIX 1	Setzt Zeilen- und Spaltenindex in R_0 und R_1 auf 1.
f MATRIX 2	Transformiert $\mathbf{Z}^P$ in $\tilde{\mathbf{Z}}$.
f MATRIX 3	Transformiert $\tilde{\mathbf{Z}}$ in $\mathbf{Z}^P$.
f MATRIX 4	Berechnet die Transponierte der im X-Register gegebenen Matrix.
f MATRIX 5	Multipliziert die Transponierte der im Y-Register gegebenen Matrix mit der im X-Register gegebenen Matrix und speichert das Ergebnis in der Ergebnismatrix.

Tastenfolge	Anzeige
f MATRIX 6	Berechnet das Residuum und speichert es in die Ergebnismatrix.
f MATRIX 7	Berechnet die Zeilensummennorm der Matrix im X-Register.
f MATRIX 8	Berechnet die Frobenius-Norm oder Euklidische Norm der Matrix im X-Register.
f MATRIX 9	Berechnet die Determinante der Matrix im X-Register und legt die *LR*-Zerlegung dieser Matrix in der Ergebnismatrix ab.
f Py,x	Transformiert $\mathbf{Z}^C$ in $\mathbf{Z}^P$.
RCL {A bis E, I}	Ruft das in den Registern X und Y indizierte Element der gegebenen Matrix ab.
RCL DIM {A bis E I}	Ruft die Dimensionen der gegebenen Matrix in die Register Y und X ab.
RCL MATRIX {A bis E}	Zeigt den Deskriptor der gegebenen Matrix an.
RCL RESULT	Zeigt den Deskriptor der Ergebnismatrix an.
f RESULT {A bis E, I}	Spezifiziert die gegebene Matrix als Ergebnismatrix.
STO {A bis E, (i)}	Speichert den angezeigten Wert in dasjenige Element der gegebenen Matrix, dessen Zeilen- und Spaltenindex sich in R_0 und R_1 befinden.
STO g {A bis E, (i)}	Speichert den Wert des Z-Registers in das in den Registern X und Y indizierte Element der gegebenen Matrix.
STO MATRIX {A bis E}	Ist ein Deskriptor in der Anzeige, so werden alle Elemente dieser Matrix in die entsprechenden Elemente der angegebenen Matrix kopiert. Ist eine Zahl in der Anzeige, wird deren Wert in allen Elementen der angegebenen Matrix gespeichert.

Tastenfolge	Anzeige
STO RESULT	Spezifiziert die im X-Register gegebene Matrix als Ergebnismatrix.
f USER	Nach jedem STO oder RCL {A bis E, (i)} werden Zeilen- bzw. Spaltenindex in R_0 und R_1 automatisch erhöht.
1/x	Invertiert die im X-Register gegebene Matrix und speichert die Inverse in der Ergebnismatrix ab.
+, −	Sind in den Registern X und Y Deskriptoren gegeben, werden die entsprechenden Elemente beider gegebenen Matrizen addiert oder subtrahiert. Steht nur in einem der beiden Register ein Deskriptor, wird der Skalar im anderen Register zu jedem Element der Matrix addiert oder subtrahiert. Das Ergebnis wird in der Ergebnismatrix abgelegt.
×	Enthalten beide Register X und Y Matrix-Deskriptoren, so wird das Produkt der beiden Matrizen (**YX**) berechnet. Enthält nur ein Register einen Deskriptor, so wird jedes Element der entsprechenden Matrix mit dem Skalar im anderen Register multipliziert. Das Ergebnis steht in der Ergebnismatrix.
÷	Befinden sich Deskriptoren in X und Y, so wird die Inverse der Matrix im X-Register mit der im Y-Register gegebenen Matrix multipliziert. Wenn nur im Y-Register ein Deskriptor steht, werden alle Elemente dieser Matrix durch den Skalar im X-Register dividiert. Steht nur im X-Register ein Deskriptor, so wird jedes Element der Inversen dieser Matrix mit dem Skalar im Y-Register multipliziert. Das Resultat wird in der Ergebnismatrix abgelegt.

Zusätzliche Information

Mehr Details und technische Hinweise sowie Anwendungen zu den Matrixfunktionen des HP-15C finden Sie im Handbuch «*HP-15C – Fortgeschrittene Funktionen*». Dort behandelte Themen sind: Methode der kleinsten Quadrate, Lösung nichtlinearer Gleichungen, schlecht konditionierte und singuläre Matrizen, Genauigkeitsbetrachtungen, iterative Verbesserungen und Erzeugen der Einheitsmatrix.

Abschnitt 13

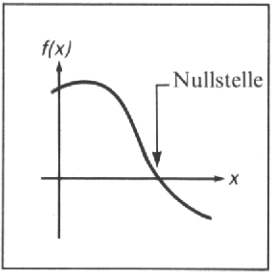
Nullstellenbestimmung

In vielen Anwendungen werden Sie eine Gleichung der Form

$$f(x) = 0.*$$

zu lösen haben.

Man sucht also Werte von x, die diese Gleichung erfüllen. Ein solcher Wert wird als *Lösung* der Gleichung $f(x) = 0$ und als *Nullstelle* der Funktion $f(x)$ bezeichnet. Wenn diese Lösungen (oder Nullstellen) reelle Zahlen sind, spricht man von *reellen Lösungen* (oder Nullstellen). In vielen Fällen können die Nullstellen einer Gleichung mittels bekannter mathematischer Formeln algebraisch bestimmt werden. Dies ist jedoch nicht immer möglich. Wenn analytische Methoden nicht ausreichen, werden numerische Verfahren zur Bestimmung der Nullstellen einer Funktion verwendet. Die Taste SOLVE beinhaltet einen sehr effizienten numerischen Algorithmus, mit dem Sie schnell und einfach die *reellen Lösungen* einer großen Anzahl unterschiedlicher Gleichungen berechnen können.**



Verwendung von SOLVE

Bei der Berechnung der Lösung einer Gleichung ruft SOLVE ein Unterprogramm, das *Sie* für die Auswertung der Funktion $f(x)$ zur Verfügung stellen müssen, wiederholt auf und führt es aus.

---

* Prinzipiell kann *jede* Gleichung mit einer Veränderlichen in dieser Form ausgedrückt werden. Sie können beispielsweise die Gleichung $f(x) = a$ und $f(x) = g(x)$ als $f(x)-a = 0$ und $f(x) = g(x)$ darstellen.

** Die Funktion SOLVE verwendet nicht den komplexen Stack. Informationen über die Berechnung von komplexen Lösungen einer Gleichung finden Sie im Handbuch «*HP-15C: Fortgeschrittene Funktionen*».

Folgende Regeln gelten für die Verwendung von [SOLVE]:

1. Geben Sie im Programm-Modus ein Unterprogramm ein, das die Funktion $f(x)$ auswertet. Dieses Unterprogramm muß mit einer Label-Anweisung ([f] [LBL] *Label*) anfangen. Nach Ausführung des Unterprogramms muß der Wert von $f(x)$ im X-Register stehen.

   Im Run-Modus:

2. Geben Sie zwei Anfangsnäherungen (durch [ENTER] getrennt) der gesuchten Lösung in die X- und Y-Register ein. Diese Anfangsnäherungen teilen dem 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 [f] [SOLVE] gefolgt von dem Label mit dem Ihr Unterprogramm beginnt. Der Rechner sucht darauf nach der gesuchten Nullstelle der Funktion und zeigt das Ergebnis an. Wenn die Funktion, die Sie untersuchen, mehr als eine Nullstelle besitzt, wird die Routine beendet, sobald die erste Nullstelle gefunden ist. Um weitere Lösungen zu bestimmen, müssen Sie neue Anfangsnäherungen eingeben und danach [SOLVE] nochmals drücken.

Unmittelbar bevor [SOLVE] Ihr Unterprogramm aufruft, wird der Wert von x in die X-, Y-, T- und Z-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 fortlaufend Ihrem Unterprogramm zur Verfügung. (Diese Technik wird auf Seite 41 ausführlich beschrieben.)

Beispiel: Verwenden Sie [SOLVE], um die Werte von x zu bestimmen, für die

$$f(x) = x^2 - 3x - 10 = 0.$$

gilt.

Mit Hilfe des Horner-Schemas (siehe Seite 79) können Sie die Funktion wie folgt darstellen:

$$f(x) = (x - 3)x - 10.$$

Tasten Sie im Programm-Modus das folgende Unterprogramm zur Berechnung von $f(x)$ ein.

Tastenfolge	**Anzeige**	
[g] [P/R]	000–	Programm-Modus.
[f] CLEAR [PRGM]	000–	Löscht den Programmspeicher.

Tastenfolge	Anzeige	
f LBL 0	001– 42,21, 0	LBL Anweisung definiert den Anfang des Unterprogramms. Das Unterprogramm unterstellt, daß der Stack mit x geladen ist.
3	002– 3	
−	003– 30	Berechnet $x-3$.
×	004– 20	Berechnet $(x-3)x$.
1	005– 1	
0	006– 0	
−	007– 30	Berechnet $(x-3)x-10$.
g RTN	008– 43 32	

Geben Sie im Run-Modus zwei Anfangsnäherungen in die X- und Y-Register ein. Versuchen Sie es mit den Werten 0 und 10, um eine eventuelle positive Lösung zu suchen.

Tastenfolge	Anzeige*	
g P/R		Run-Modus.
0 ENTER	0.0000	Anfangsnäherungen.
10	10	

Jetzt können Sie die gewünschte Lösung berechnen, indem Sie f SOLVE 0 drücken. Das Ergebnis erscheint jedoch nicht sofort in der Anzeige. Der HP-15C verwendet einen iterativen Algorithmus** zur Berechnung der Nullstellen. Der Algorithmus wertet nun wiederholt die gegebene Funktion aus, wobei jedesmal das von Ihnen zur Verfügung gestellte Unterprogramm aufgerufen wird, und vergleicht die ermittelten Funktionswerte. Die Berechnung einer Nullstelle dauert im allgemeinen 30 Sekunden bis 2 Minuten; diese Zeit kann im Einzelfall aber auch noch überschritten werden.

Drücken Sie f SOLVE und lassen Sie sich eine der leistungsfähigsten Operationen Ihres Rechners demonstrieren. Die Meldung **running** blinkt in der Anzeige, solange SOLVE ausgeführt wird.

---

* Drücken Sie f FIX 4 damit die Anzeige mit der in diesem Beispiel verwendeten Anzeige übereinstimmt. Das Anzeigeformat beeinflußt in keiner Weise die Funktion SOLVE.

** Ein *Algorithmus* ist ein Verfahren zur Lösung mathematischer Probleme, das aus einer Anzahl vordefinierter Einzeloperationen besteht. Ein iterativer Algorithmus enthält einen Teil, der zur Lösung des Problems wiederholt ausgeführt wird.

Tastenfolge	Anzeige	
f SOLVE 0	5.0000	Die gesuchte Lösung.

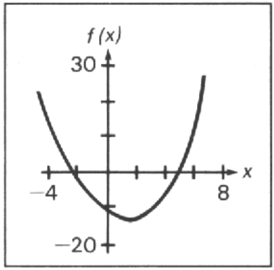
Nachdem die Nullstelle ermittelt und das Ergebnis angezeigt ist, können Sie im Stack überprüfen, ob der angezeigte Wert tatsächlich eine Nullstelle von $f(x)=0$ ist. Wie Sie wissen, steht die gesuchte Lösung im X-Register. Das Y-Register enthält die vorletzte Näherung der Nullstelle, die sehr nahe an der angezeigten Lösung im X-Register liegen sollte. Das Z-Register enthält den für die berechnete Lösung resultierenden Funktionswert.

Tastenfolge	Anzeige	
R↓	5.0000	Vorletzte Näherung der Nullstelle.
R↓	0.0000	Wert der Funktion, wenn für x die Nullstelle eingesetzt wird. Hier gilt $f(x)=0$.

Quadratische Gleichungen, wie in unserem Beispiel, besitzen zwei Nullstellen. Mit zwei neuen Anfangsnäherungen können Sie die Lage der zweiten Nullstelle ermitteln. Versuchen Sie es mit 0 und −10, um eine negative Nullstelle zu finden.

Tastenfolge	Anzeige	
0 ENTER	0.0000	Anfangsnäherungen.
10 CHS	−10	
f SOLVE 0	−2.0000	Die zweite Nullstelle.
R↓	−2.0000	Vorletzte Näherung für die zweite Nullstelle.
R↓	0.0000	Der Wert der Funktion, wenn für x die zweite Nullstelle eingesetzt wird.

Sie haben jetzt die zwei Lösungen der Gleichung $f(x) = 0$ gefunden. Beachten Sie, daß Sie diese Gleichung auch algebraisch hätten lösen können – die Ergebnisse wären die gleichen gewesen, die Sie mit [SOLVE] bekommen haben.



Graph von $f(x)$

Die Anwendbarkeit und Leistungsfähigkeit der Funktion [SOLVE] zeigt sich erst richtig, wenn Sie die Nullstellen einer Gleichung bestimmen wollen, die nicht algebraisch berechnet werden können.

Beispiel: Lokalmatador Chuck Fahr 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.

Benutzen Sie das folgende Unterprogramm, um die Höhe zu berechnen.

Tastenfolge	**Anzeige**	
[g] [P/R]	**000–**	
[f] [LBL] [A]	**001– 42,21,11**	Das Unterprogramm beginnt mit dem Label «A».
2	**002– 2**	Das Unterprogramm setzt voraus, daß t in das X- und Y-Register geladen ist.
0	**003– 0**	
[÷]	**004– 10**	

Tastenfolge	Anzeige		
CHS	005–	16	$-t/20$.
e^x	006–	12	
CHS	007–	16	$-e^{-t/20}$.
1	008–	1	
+	009–	40	$1-e^{-t/20}$.
5	010–	5	
0	011–	0	
0	012–	0	
0	013–	0	
×	014–	20	$5000\,(1-e^{-t/20})$.
x⇄y	015–	34	Lädt einen neuen t-Wert in das X-Register.
2	016–	2	
0	017–	0	
0	018–	0	
×	019–	20	$200t$.
−	020–	30	$5000\,(1-e^{-t/20})-200t$.
g RTN	021–	43 32	

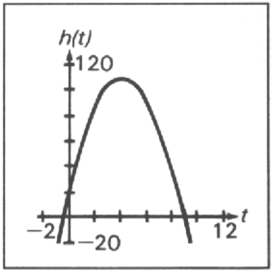
Schalten Sie den Rechner in den Run-Modus und tasten Sie danach die zwei Anfangsnäherungen für die Zeit (z.B. 5 und 6 Sekunden) ein. Führen Sie anschließend SOLVE aus.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
5 ENTER	5.0000	Anfangsnäherungen.
6	6	
f SOLVE A	9.2843	Die gesuchte Nullstelle.

Überprüfen Sie die Lösung, indem Sie den Inhalt der Y- und X-Register untersuchen.

Tastenfolge	Anzeige	
R↓	9.2843	Vorletzte Näherung der Lösung
R↓	0.000	Der Wert der Funktion, wenn für t die gefundene Lösung eingesetzt wird. Hier gilt $h=0$.

Fahr's Hammer fällt nach 9.2843 Sekunden wieder zu Boden – ein gewaltiger Wurf.



Graph von *h*(*t*)

Wenn keine Lösung gefunden wird

Sie haben gesehen, wie die Funktion SOLVE 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 statt dessen **Error 8**.

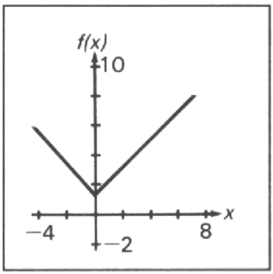
Beispiel: Für die Gleichung

$$|x| = -1$$

gibt es keine reelle Lösung, da der Absolutwert der Funktion niemals negativ sein kann. Wir schreiben die Gleichung erst in die gewünschte Form

$$|x| + 1 = 0$$

und versuchen mit Hilfe von SOLVE eine Lösung zu finden.



Graph von $f(x) = |x| + 1$

Tastenfolge	Anzeige	
g P/R	000–	Programm-Modus.
f LBL 1	001–42,21, 1	
g ABS	002– 43 16	
1	003– 1	
+	004– 40	
g RTN	005– 43 32	

Da die Betragsfunktion ihr Minimum für $x = 0$ annimmt, wollen wir Anfangsnäherungen rechts und links von Null wählen, beispielsweise 1 und –1. Versuchen Sie nun eine Nullstelle zu berechnen.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
1 ENTER	**1.0000**	Anfangsnäherungen.
1 CHS	–1	
f SOLVE 1	**Error 8**	Diese Anzeige bedeutet, daß keine Lösung gefunden wurde.
←	**0.0000**	Löscht die **Error**-Anzeige.

Wie Sie sehen, hat der HP-15C die Suche nach einer Lösung der Gleichung $f(x) = 0$ abgebrochen, sobald er feststellte, daß zumindest in der Nähe des anfänglichen angegebenen Bereichs keine Nullstelle existiert. Die Meldung **Error 8** bedeutet nicht, daß eine unzulässige Operation ausgeführt wurde; vielmehr teilt der Rechner Ihnen dadurch mit, daß SOLVE keine Nullstelle in dem von Ihnen angegebenen Bereich finden konnte.

Wenn der HP-15C die Suche nach einer Nullstelle abbricht und eine Fehlermeldung ausgibt, liegt einer der drei folgenden Fälle vor:

- Wenn aufeinanderfolgende Iterationen einen konstanten Funktionswert ungleich Null ergeben, bricht die Funktionsausführung mit der Fehlermeldung **Error 8** ab.
- Wenn die Funktionswerte andeuten, daß das *betragsmäßige Minimum* der Funktion in dem untersuchten Bereich nicht gleich Null ist, wird die Ausführung mit der Anzeige **Error 8** unterbrochen.
- Wenn innerhalb des Unterprogramms ein unerlaubtes Argument in einer mathematischen Operation verwendet wird, wird die Ausführung mit der Anzeige **Error 0** unterbrochen.

Im Falle eines konstanten Funktionswertes gibt es keine Anzeige dafür, daß die Funktion gegen Null geht. Das kann eintreten, wenn die 10 signifikanten Ziffern eines Funktionswertes konstant sind (wenn der Graph der 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.

Im Fall, wenn die Funktion ein betragsmäßiges Minimum ungleich Null annimmt, hat die Routine eine Folge von betragsmäßig abnehmenden Funktionswerten gefunden. Es wurde jedoch kein Wert von x gefunden, für den der Graph der Funktion die x-Achse berührt oder geschnitten hat.

Der letzte Fall deutet eher auf mögliche Unzulänglichkeiten im Unterprogramm als auf bei der Anwendung der Lösungsroutine zu beachtende Einschränkungen hin. Unerlaubte Operationen können durch die Angabe von Anfangsnäherungen vermieden werden, die den Suchvorgang in einem Bereich konzentrieren, in dem ein Ergebnis dieser Art nicht vorkommen kann. Da die [SOLVE] Routine auf möglichst rasche Konvergenz ausgelegt ist, wird die Funktion u.U. innerhalb eines großen Bereichs untersucht. Es ist daher sinnvoll, 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, Variablen zu skalieren, um große Zahlen zu vermeiden.

Das erfolgreiche Auffinden einer Nullstelle mit Hilfe der [SOLVE] Routine hängt maßgeblich von der zu untersuchenden Funktion und den Anfangsnäherungen ab. Das bloße Vorhandensein einer Nullstelle ist keine Garantie, daß sie mit [SOLVE] 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 Nullstelle überhaupt existiert.

Wahl der Anfangsnäherungen

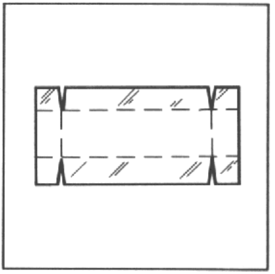
Wenn Sie mit Hilfe von [SOLVE] die Lösung einer Gleichung finden wollen, geben Sie mit den beiden Anfangsnäherungen diejenigen Werte von x vor, mit denen die Routine den Suchvorgang beginnt. Im allgemeinen wird damit die Wahrscheinlichkeit, daß Sie eine gesuchte Nullstelle auch finden, durch Ihre Kenntnis der zur Analyse vorliegenden Funktion bestimmt. Die Vorgabe realistischer und wohlüberlegter Anfangsnäherungen erleichtert wesentlich die Berechnung einer Nullstelle.

Die Anfangsnäherungen können auf verschiedene Art und Weise gewählt werden:

Wenn die Variable x auf einen Bereich beschränkt ist, in dem eine Lösung denkbar ist, sollten die Anfangsnäherungen 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 betrachteten Funktionen das physikalische Problem nur innerhalb bestimmter Bereiche beschreiben. 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 ein betragsmäßiges Minimum ungleich Null annimmt.

Beispiel: Aus einem Eisenblech, 8 dm lang und 4 dm breit, soll ein offener Behälter mit einem Volumen von 7.5 dm^3 geformt werden. Wie muß das Blech abgekantet werden? (Ein hoher Behälter ist einem niedrigeren vorzuziehen.)



Lösung: Gesucht wird die Höhe des Behälters (d.h. die Länge des Blechs, das entlang der vier Seiten *umgebogen* werden muß), der das gewünschte Volumen ergibt. Wenn x die Höhe ist (die Länge des zu biegenden Blechs) dann ist die Länge des Behälters $(8-2x)$ und die Breite $(4-2x)$.

Das Volumen wird durch die Gleichung

$$V = (8-2x)\ (4-2x)x$$

gegeben.

Zerlegt man diesen Ausdruck und wendet dann das Horner-Schema (Seite 79) an, ergibt sich

$$V = 4((x-6x)x + 8)x$$

Um ein Volumen von $V = 7.5$ zu erhalten, berechnen Sie die Nullstellen der folgenden Funktion

$$f(x) = 4((x-6)x + 8)x-7.5 = 0$$

Das folgende Unterprogramm berechnet $f(x)$:

Tastenfolge	**Anzeige**	
g P/R	**000–**	Programm-Modus.
f LBL 3	**001– 42,21, 3**	Label.
6	**002– 6**	Setzt voraus, daß der Stack mit x geladen ist.

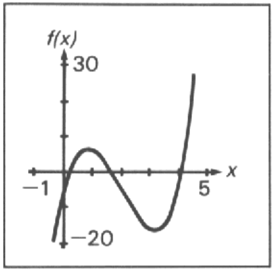
Tastenfolge	Anzeige		
[-]	003–	30	
[×]	004–	20	$(x-6)x$.
8	005–	8	
[+]	006–	40	
[×]	007–	20	$((x-6)x+8)x$.
4	008–	4	
[×]	009–	20	$4((x-6)x+8)x$.
7	010–	7	
[·]	011–	48	
5	012–	5	
[-]	013–	30	
[g] [RTN]	014–	43 32	

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 Anfangsnäherungen 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. Anfangsnäherungen von 1 und 2 dm sind daher angemessen.

Ermitteln Sie die gewünschte Höhe:

Tastenfolge	Anzeige	
[g] [P/R]		Run-Modus.
1 [ENTER]	1.0000	Anfängsnäherungen.
2	2	
[f] [SOLVE] 3	1.5000	Die gewünschte Höhe.
[R↓]	1.5000	Vorletzte Näherung.
[R↓]	0.0000	Funktionswert der Lösung.

Mit der Höhe von 1.5 dm ergibt sich ein Behälter 5 dm lang, 1 dm breit und 1.5 dm hoch.



Graph von $f(x)$

Wenn Sie die obere Beschränkung für die Höhe ignorieren und Anfangsnäherungen 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 die Anfangsnäherungen 0 und 1 dm verwenden, beträgt die Höhe 0.2974 dm – ein Behälter mit einer unerwünschten flachen Form.

Wenn Sie das Verhalten einer Funktion untersuchen wollen, können Sie einfach die als Unterprogramm enthaltene Funktion für einige Werte von x berechnen. Dazu laden Sie den Wert von x in den Stack. Führen Sie das Unterprogramm aus, um den Funktionswert zu berechnen (drücken Sie f *Buchstabenlabel* oder GSB *Label*).

Die Ergebniswerte können als Kurve graphisch dargestellt werden. Dieser Vorgang ist dann besonders nützlich, wenn Sie den Verlauf der Funktion nicht kennen. Eine einfach aussehende Funktion kann einen stark variierenden Graph aufweisen, den Sie gar nicht erwartet haben. Eine Nullstelle in der Umgebung einer solchen lokalen Variation ist vielleicht schwer zu finden, wenn die Anfangsnäherungen nicht in der Nähe der Nullstelle liegen.

Wenn Sie sich keine Vorstellung hinsichtlich der Art der Funktion oder der Lage der von Ihnen gewünschten Nullstellen machen können, bleibt Ihnen der Versuch, durch Probieren zu einem Ergebnis zu gelangen. Die erfolgreiche Lösung hängt in diesem Fall zum Teil von der Funktion selbst ab. Die folgende Vorgehensweise ist häufig – wenn auch nicht immer – erfolgreich.

- Wenn Sie zwei verhältnismäßig große negative und positive Anfangsnäherungen 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, indem Sie von den bekannten Nullstellen weit entfernte Anfangsnäherungen verwenden.

- Viele Funktionen zeigen ein spezielles Verhalten, wenn das Argument gegen Null geht. Bei verschachtelten Funktionsausdrücken ist es daher sinnvoll, zu überprüfen, wann die als Argument dienende Funktion gegen Null geht, und dementsprechend die Anfangsnäherungen zu wählen.

Obgleich Sie die Funktion SOLVE üblicherweise mit zwei verschiedenen Anfangsnäherungen versorgen, können Sie SOLVE durchaus auch mit ein und demselben Wert im X- und Y-Register verwenden. Bei identischen Anfangsnäherungen wird intern eine zweite Anfangsnäherung erzeugt. Wenn die einzelne Anfangsnäherung ungleich Null ist, unterscheidet sich die zweite Anfangsnäherung in der siebten signifikanten Ziffer. Wenn Ihre Anfangsnäherung gleich Null ist, wird 1×10^{-7} als zweite Näherung verwendet. Die Suche nach einer Nullstelle wird dann ganz normal wie mit zwei Anfangsnäherungen durchgeführt.

Verwendung von SOLVE in Programmen

Die SOLVE Routine kann als Teil eines Programms verwendet werden. Dabei ist darauf zu achten, daß X- und Y-Register vor dem Aufruf der SOLVE Routine die Anfangsnäherungen enthalten. Nach Ausführung von SOLVE steht der x-Wert im X-Register und der entsprechende Funktionswert im Z-Register. Ist der x-Wert eine Nullstelle, führt das Programm die nächste Zeile aus. Ist der x-Wert keine Nullstelle, wird die nächste Zeile übersprungen. (Weitere Informationen über die Interpretation von Nullstellen finden Sie auf Seite 226 unter der Überschrift «Auswertung der Ergebnisse».) 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, indem es neue Anfangsnäherungen wählt oder einen Parameter der Funktion ändert.

Bei der Verwendung von SOLVE innerhalb eines Programms wird eine der sieben möglichen Rücksprungadressen belegt. Da SOLVE selbst ein Unterprogramm aufruft, dürfen höchstens fünf andere Rücksprünge anstehen. Wird SOLVE dagegen über das Tastenfeld verwendet, wird keine Rücksprungadresse benötigt, so daß den von SOLVE aufgerufenen Unterprogrammen sechs Unterprogrammebenen zur Verfügung stehen. Beachten Sie, daß die Fehlermeldung **Error 5** erscheint, wenn sieben Rücksprünge anstehen und ein weiteres Unterprogramm aufgerufen wird. (Siehe Seite 105.)

Einschränkungen bei der Verwendung von [SOLVE]

Bei der Verwendung von [SOLVE] ist zu beachten, daß diese Funktion nicht rekursiv verwendet werden kann. Das heißt, daß [SOLVE] nicht in einem Unterprogramm vorkommen kann, das bei der Ausführung von [SOLVE] aufgerufen wird. In diesem Fall wird das Programm mit der Fehlermeldung **Error 7** abgebrochen. Es *ist* jedoch möglich, [SOLVE] zusammen mit der Funktion [∫] zu verwenden.

Speicheranforderungen

[SOLVE] verwendet bei der Ausführung fünf Speicherregister. (In Anhang C wird die automatische Zuteilung dieser Register erläutert.) Wenn keine fünf unbelegte Register mehr vorhanden sind, kann [SOLVE] nicht ausgeführt werden, und die Meldung **Error 10** erscheint in der Anzeige.

Eine Routine, die die Funktionen [SOLVE] und [∫] kombiniert, benötigt 23 Register an Speicherbereich.

Zusätzliche Informationen

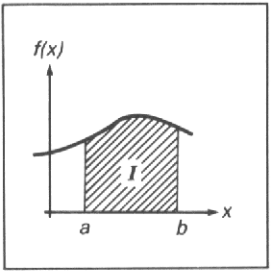
In Anhang D, «[SOLVE] im Detail», werden weitere Techniken und Anwendungen in Verbindung mit [SOLVE] beschrieben. Dazu gehören:

- Arbeitsweise von [SOLVE].
- Genauigkeit der Lösung.
- Auswertung der Ergebnisse.
- Bestimmung mehrerer Nullstellen.
- Verkürzung der Rechenzeit.

Abschnitt 14

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 beschrieben 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 Grenzen $x=a$ und $x=b$* begrenzt wird.

Wenn sich ein Integral nur schwierig oder gar nicht analytisch bestimmen läßt, kann es nur mit numerischen Methoden berechnet werden. Bis jetzt waren Sie dazu auf mehr oder weniger aufwendige Computerprogramme angewiesen. Mit dem HP-15C können Sie numerische Integrationen einfach mittels der Funktion [$\int_y^x$]** durchführen.

Verwendung von [$\int_y^x$]

[$\int_y^x$] ist wie folgt anzuwenden:

1. Tasten Sie im Programm-Modus ein Unterprogramm ein, das die zu integrierende Funktion $f(x)$ auswertet. Dieses Unterprogramm muß mit einer Label-Anweisung beginnen ([f] [LBL] *Label*) und den Wert von $f(x)$ im X-Register ablegen.

* Es wird vorausgesetzt, daß $f(x)$ im Integrationsintervall nicht negativ ist.

** Die Funktion [$\int_y^x$] verwendet nicht den imaginären Stack. Weitere Informationen über die Verwendung von [$\int_y^x$] im Komplex-Modus finden Sie im Handbuch «*HP-15C: Fortgeschrittene Funktionen*».

Im Run-Modus:

2. Geben Sie die untere Integrationsgrenze (a) in das X-Register ein und drücken Sie dann [ENTER], damit sie in das Y-Register angehoben wird.
3. Geben Sie die obere Integrationsgrenze (b) in das X-Register ein.
4. Drücken Sie [f] [$\int_y^x$] gefolgt von dem für das Unterprogramm verwendeten Label.

Beispiel: In gewissen Zweigen der Physik und der Ingenieurwissenschaften wird die Berechnung der *Besselschen Funktion* benötigt. Die Besselsche Funktion erster Art 0-ter Ordnung hat folgende Form:

$$J_0(x) = \frac{1}{\pi}\int_0^{\pi} \cos(x \sin\theta)\, d\theta.$$

Berechnen Sie

$$J_0(1) = \frac{1}{\pi}\int_0^{\pi} \cos(\sin\theta)\, d\theta.$$

Tasten Sie im Programm-Modus das folgende Unterprogramm ein, um die Funktion $f(\theta) = \cos(\sin\theta)$ zu berechnen.

Tastenfolge	**Anzeige**	
[g] [P/R]	000–	Programm-Modus.
[f] CLEAR [PRGM]	000–	Löscht den Programmspeicher.
[f] [LBL] 0	001– 42,41, 0	Beginn des Unterprogramms mit einer [LBL] Anweisung. Das Unterprogramm setzt den Wert θ im X-Register voraus.
[SIN]	002– 23	Berechnet sin (θ).
[COS]	003– 24	Berechnet cos (sin θ).
[g] [RTN]	004– 43 32	

Geben Sie nun im Run-Modus die untere Integrationsgrenze in das Y-Register und die obere Integrationsgrenze in das X-Register ein. Außerdem muß zur Berechnung dieses speziellen Beispiels der trigonometrische Modus Radiant gewählt werden.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
0 ENTER	0.0000	Geben Sie die untere Grenze, 0, in das Y-Register ein.
g π	3.1416	Geben Sie die obere Grenze, π, in das X-Register.
g RAD	3.1416	Wählt den trigonometrischen Modus Radiant.

Jetzt können Sie f ∫ 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-15C einen ausgeklügelten iterativen Algorithmus. Dieser Algorithmus berechnet die zu integrierende Funktion $f(x)$ für viele Werte von x innerhalb des Integrationsintervalls. Dabei wird für jeden Wert von x das für diesen Zweck zur Verfügung gestellte Unterprogramm durchlaufen, um den Funktionswert zu ermitteln. Wenn der Rechner ein Unterprogramm häufig durchlaufen muß – wie es der Fall ist, wenn Sie ∫ drücken – dann können Sie das Ergebnis nicht sofort erwarten. Die Rechenzeit für ein Integral dürfte im allgemeinen zwischen 30 Sekunden und 2 Minuten liegen, und kann in einzelnen Fällen noch länger sein. Später wird erläutert, wie Sie die Ausführungszeit verkürzen können. Drücken Sie jedoch nun f ∫ 0 und legen dann eine Pause ein (oder lesen schon weiter); der HP-15C übernimmt den strapaziösen Teil der Arbeit für Sie.

Tastenfolge	Anzeige	
f ∫ 0	2.4040	$= \int_0^\pi \cos(\sin\theta)\,d\theta.$

Im allgemeinen muß das Ergebnis noch mit eventuellen Konstanten, die außerhalb des Integrationszeichens stehen, multipliziert werden. In diesem Fall muß das Integral mit $1/\pi$ multipliziert werden, um $J_0(1)$ zu erhalten.

Tastenfolge	Anzeige	
g π	3.1416	
÷	0.7652	$J_0(1)$.

Bevor das Unterprogramm zur Berechnung von $f(x)$ aufgerufen wird, speichert die [∫]-Routine – genau wie [SOLVE] – den Wert x in das X-, 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. (Eine Technik zur Auswertung von Polynomen, die voraussetzt, daß der Stack mit dem x-Wert geladen ist, wird auf Seite 79 beschrieben.)

> **Hinweis**: Da der Rechner den x-Wert in jedes Stack-Register speichert, wird der vorherige Inhalt des Stacks mit dem x-Wert überschrieben. 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.
>
> Es kann vorkommen, daß Sie ein Unterprogramm, das Sie für die [∫] Routine entwickelt haben, auch zur Berechnung 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 [ENTER] [ENTER] [ENTER] laden.

Beispiel: Die Besselsche Funktion erster Art 1-ter Ordnung hat die Form:

$$J_1(x) = \frac{1}{\pi} \int_0^{\pi} \cos(\theta - x \sin \theta)\, d\theta.$$

Berechnen Sie $$J_1(1) = \frac{1}{\pi} \int_0^{\pi} \cos(\theta - \sin \theta)\, d\theta.$$

Tasten Sie das folgende Unterprogramm zur Berechnung der Funktion $f(\theta) = \cos(\theta - \sin\theta)$ ein.

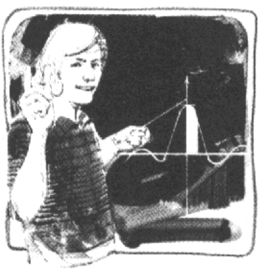
Tastenfolge	Anzeige	
[g] [P/R]	**000–**	Programm-Modus.
[f] [LBL] 1	**001– 42,21, 1**	Das Unterprogramm beginnt mit einem Label.

Tastenfolge	Anzeige		
SIN	002–	23	Berechnet sin θ.
−	003–	30	Da der ∫ Algorithmus vor der Ausführung des Unterprogramms den θ-Wert in das Y-Register speichert, wird hier mit der − Operation (θ–sin θ) berechnet.
COS	004–	24	Berechnet cos (θ–sin θ).
g RTN	005–	43 32	

Geben Sie nun im Run-Modus die untere und obere Integrationsgrenze in das Y- und X-Register ein. Achten Sie darauf, daß der trigonometrische Modus Radiant eingeschaltet ist und drücken Sie dann f ∫ 1, um das Integral zu berechnen. Dieses Integral müssen Sie schließlich noch mit $1/\pi$ multiplizieren, um $J_1(1)$ zu erhalten.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
0 ENTER	0.0000	Eingabe der Untergrenze ins Y-Register.
g π	3.1416	Eingabe der Obergrenze ins X-Register.
g RAD	3.1416	(Falls nicht bereits im Radiant-Modus.)
f ∫ 1	1.3825	$= \int_0^\pi \cos(\theta - \sin\theta)\, d\theta$.
g π ÷	0.4401	$J_1(1)$.

Beispiel: In der Nachrichtentechnik wird für manche Zwecke (z.B. die Stromübertragung in idealisierten Netzwerken) ein Integral der folgenden Form (auch *Integralsinus* genannt) benötigt:



$$Si(t) = \int_0^t \frac{\sin x}{x}\, dx.$$

Berechnen Sie *Si*(2).

Tasten Sie im Programm-Modus das Unterprogramm zur Berechnung von $f(x) = (\sin x)/x$* ein.

Tastenfolge	Anzeige	
g P/R	000–	Programm-Modus.
f LBL .2	001– 42,21,.2	Beginn des Unterprogramms mit einer Labelanweisung.
SIN	002– 23	Berechnet sin x.
$x \gtrless y$	003– 34	Da der ∫ Algorithmus den x-Wert vor der Ausführung des Unterprogramms in das Y-Register speichert, bewirkt hier die $x \gtrless y$ Operation, daß der x-Wert in das X-Register und sin x in das Y-Register gespeichert wird.
÷	004– 71	Dividiert sin x durch x.
g RTN	005– 43 32	

Geben Sie nun die Integrationsgrenzen in das X- und Y-Register ein. Drücken Sie im trigonometrischen Modus Radiant die Tasten f ∫ .2, um das Integral zu berechnen:

Tastenfolge	Anzeige	
0 ENTER	0.0000	Lädt die untere Integrationsgrenze in das Y-Register.
2	2	Lädt die obere Integrationsgrenze in das X-Register.
g RAD	2.0000	(Falls der Rechner nicht schon im Radiant-Modus ist.)
f ∫ .2	1.6054	*Si*(2).

* Der Versuch die Funktion $f(x) = (\sin x)/x$ für $x = 0$ als untere Integrationsgrenze zu berechnen, würde die Fehlermeldung **Error 0** zur Folge haben, weil eine Division durch Null zum Programmabbruch führt. Das Integral würde nicht berechnet werden. Der ∫ Algorithmus wertet jedoch die Funktion normalerweise *nicht* an den unteren Integrationsgrenzen aus, so daß der Rechner auch das Integral einer Funktion, die in diesen Punkten nicht definiert ist, berechnen kann. Nur wenn das Integrationsintervall sehr klein ist, oder die Anzahl der Stützstellen sehr groß ist, wird die Funktion an den Integrationsgrenzen ausgewertet.

Genauigkeit von [∫ʸₓ]

Die Genauigkeit des Integrals einer Funktion hängt von der Genauigkeit der Funktion selbst ab. Daher wird auch die Genauigkeit eines mit [∫ʸₓ] berechneten Integrals durch die Genauigkeit der mit Ihrem Unterprogramm berechneten Funktion begrenzt*. Zur Spezifikation der Genauigkeit einer Funktion wählen Sie ein Anzeigeformat, das *nur* die Stellen anzeigt, auf die Sie die Funktionswerte als genau erwarten**. 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 hier zeigen, wie Sie die Genauigkeit eines berechneten Integrals bestimmen können, nachdem wir einige Bemerkungen über das Anzeigeformat gemacht haben.

Sie werden sich erinnern, daß der HP-15C über drei Arten von Anzeigeformaten verfügt: [FIX], [SCI] und [ENG]. Im allgemeinen spielt es keine Rolle, welches Anzeigeformat verwendet wird, da für viele Integrale das Ergebnis in jedem der Formate identisch ist (vorausgesetzt, daß in Abhängigkeit der Wertigkeit der Funktion die Anzahl der Stellen in der Anzeige richtig gewählt wurde). Bei der Bestimmung von Integralen wird im allgemeinen das [SCI] Anzeigeformat verwendet; wir benutzen im weiteren Verlauf dieses Abschnitts ebenfalls dieses Format.

> **Hinweis**: Erinnern Sie sich, daß Sie die Anzahl der angezeigten Ziffern des einmal mit [SCI], [ENG] oder [FIX] gewählten Anzeigeformats ändern können, indem Sie eine Zahl in das Indexregister speichern und dann [f] [FIX] [I], [f] [SCI] [I] oder [f] [ENG] [I] drücken (siehe Abschnitt 10). Diese Eigenschaft ist häufig besonders dann nützlich, wenn [∫ʸₓ] als Teil eines Programms ausgeführt wird.

---

* Es ist möglich, daß Integrale von Funktionen mit gewissen Charakteristika (wie scharfe Zacken oder sehr schnelle Oszillationen) falsch berechnet werden. *Dies ist jedoch sehr unwahrscheinlich.* In Anhang E werden die Eigenschaften von problematischen Funktionen 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 die Rundungsfehler bei der Bestimmung der Funktionswerte. Erläuterung dem Handbuch «*HP-15C: Fortgeschrittene Funktionen*».

*** Der Grund hierfür wird in Anhang E beschrieben.

Da die Genauigkeit jedes Integrals durch die Genauigkeit der Funktion begrenzt wird (die wiederum durch das Anzeigeformat begrenzt ist), kann der Rechner nicht den genauen, sondern nur einen *angenäherten* Wert (*Approximation*) des Integrals bestimmen. Der HP-15C speichert eine Fehlerabschätzung* der Approximation des Integrals in das Y-Register und schreibt die Approximation in das X-Register. Um die Genauigkeit einer Approximation zu überprüfen, holen Sie sich die Fehlerabschätzung einfach mit [x≷y] in die Anzeige.

Beispiel: Wählen Sie das Anzeigeformat [SCI] 2 und berechnen Sie das Integral $J_1(1)$ des Beispiels auf Seite 197:

Tastenfolge	**Anzeige**		
0 [ENTER]	**0.0000**		Gibt die untere Integrationsgrenze in das Y-Register ein.
[g] [π]	**3.1416**		Gibt die obere Integrationsgrenze in das X-Register ein.
[g] [RAD]	**3.1416**		(Falls der Rechner nicht schon im Radiant-Modus ist.)
[f] [SCI] 2	**3.14**	**00**	Wählt das Anzeigeformat [SCI] 2.
[f] [∫] 1	**1.38**	**00**	[SCI] 2 Approximation des Integrals.
[x≷y]	**1.88**	**–03**	Fehlerabschätzung der [SCI] 2 Approximation.

Das Integral hat den Wert 1.38 ± 0.00188. Da die Fehlerabschätzung das Ergebnis erst in der dritten Dezimalstelle beeinflußt, können die angezeigten Ziffern der Approximation als genau betrachtet werden. Es ist im allgemeinen jedoch sehr schwierig vorauszusagen, wieviele Ziffern einer Approximation durch die Fehlerabschätzung nicht beeinflußt werden. Dies hängt im allgemeinen von der zu integrierenden Funktion, den Integrationsgrenzen und dem Anzeigeformat ab.

---

*Kein Algorithmus, der in der numerischen Integration verwendet wird, kann den genauen Unterschied zwischen der Approximation und dem tatsächliche Integral feststellen. Der vom HP-15C verwendete Algorithmus bestimmt aber eine obere Grenze für diese Differenz, die als Fehlerabschätzung der Approximation bezeichnet wird. Wenn beispielsweise das Integral von *Si*(2) gleich 1.6054 ± 0.0001 ist, so ist die Approximation des Integrals 1.6054 und die Fehlerabschätzung 0.0001. Damit kennen Sie zwar nicht den exakten Unterschied zwischen dem tatsächlichen Integral und der Approximation, aber Sie wissen, daß diese Differenz nicht größer als 0.0001 ist.

Wenn die Fehlerabschätzung größer als gewünscht ist, können Sie die Genauigkeit der Approximation verbessern, indem Sie eine größere Anzahl von Ziffern im Anzeigeformat spezifizieren und die Berechnung wiederholen*.

Wenn Sie die Berechnung einer Approximation wiederholen wollen, erspart Ihnen der HP-15C die Mühe, die Integrationsgrenzen nochmals in das X- und Y-Register zu laden. Nach der Berechnung eines Integrals stehen nicht nur das Ergebnis und die Fehlerabschätzung im X- bzw. Y-Register, sondern auch die untere und obere Integrationsgrenze im T- und Z- Register. Um diese Grenzen für eine erneute Berechnung des Integrals wieder in das X-Register und Y-Register zu laden, drücken Sie einfach [R↓] [R↓].

Beispiel: Das Integral $J_1(1)$ soll jetzt in der vierten und nicht in der zweiten Stelle nach dem Dezimalpunkt genau sein.

Tastenfolge	Anzeige		
[f] [SCI] 4	**1.8826**	**–03**	Wählt das Anzeigeformat [SCI] 4.
[R↓] [R↓]	**3.1416**	**00**	Vertauscht die Stackinhalte zyklisch solange, bis die obere Integrationsgrenze im X-Register erscheint.
[f] [$\int_y^x$] 1	**1.3825**	**00**	[SCI] 4 Approximation des Integrals.
[$x \gtrless y$]	**1.7091**	**–05**	Fehlerabschätzung der [SCI] 4 Approximation.

Die Fehlerabschätzung läßt erkennen, daß die Approximation wenigstens bis auf vier Stellen nach dem Dezimalpunkt genau ist. Beachten Sie, daß die Fehlerabschätzung bei [SCI] 4 etwa ein Prozent der Fehlerabschätzung bei [SCI] 2 beträgt. Im allgemeinen verkleinert Fehlerabschätzung einer [$\int_y^x$] Approximation für jede zusätzliche Ziffer im Anzeigeformat um den Faktor 10.

* Vorausgesetzt, daß $f(x)$ immer noch auf die angezeigten Ziffern genau berechnet wird.

In dem vorherigen Beispiel war aus der Fehlerabschätzung zu schließen, daß die Approximation nur auf vier Stellen nach dem Dezimalpunkt genau ist. Wenn Sie aber kurzzeitig alle zehn Stellen des Ergebnisses anzeigen und diese Zahl mit dem tatsächlichen Wert des Integrals (eigentlich eine Approximation, die auf eine ausreichende Anzahl Stellen nach dem Dezimalpunkt genau ist) vergleichen, werden Sie finden, daß unser Ergebnis in der Tat genauer ist, als es die Fehlerabschätzung vermuten läßt.

Tastenfolge	**Anzeige**	
[x≷y]	**1.3825 00**	Ruft die Approximation in die Anzeige zurück.
[f] CLEAR [PREFIX]	**1382459676**	Alle zehn Ziffern der Approximation.

Das Integral beträgt auf acht Stellen nach dem Dezimalpunkt genau 1.38245969. Das Ergebnis ist also auf *sieben* nicht nur vier Stellen nach dem Dezimalpunkt genau. Da die Fehlerabschätzung durchweg sehr vorsichtig berechnet wird, *werden die durch den Rechner bestimmten Approximationen im allgemeinen eine höhere Genauigkeit aufweisen, als die Fehlerabschätzung andeutet*. Im Normalfall läßt sich jedoch nicht feststellen, *wie genau* eine Näherung ist.

Mit der Genauigkeit und Fehlerabschätzung von [$\int_y^x$] Approximationen werden wir uns im Anhang E noch eingehend befassen.

Verwendung von [$\int_y^x$] in Programmen

Bei der Verwendung von [$\int_y^x$] ist die Einschränkung zu beachten, daß diese Funktion nicht rekursiv verwendet werden kann; d.h., daß [$\int_y^x$] nicht in einem Unterprogramm stehen kann, das bei der Ausführung von [$\int_y^x$] aufgerufen wird. Eine Berechnung von Mehrfachintegralen ist demzufolge nicht möglich. In diesem Fall wird das Programm mit der Fehlermeldung **Error 7** abgebrochen. Es ist jedoch möglich, [$\int_y^x$] in Verbindung mit der Funktion [SOLVE] zu verwenden.

Sobald Sie [$\int_y^x$] innerhalb eines Programms verwenden, wird eine der sechs möglichen Rücksprungadressen belegt. Da [$\int_y^x$] selbst ein Unterprogramm aufruft, dürfen höchstens fünf andere Rücksprünge anstehen. Wenn Sie [$\int_y^x$] hingegen über das Tastenfeld ausführen, wird keine Rücksprungadresse benötigt, so daß den von [$\int_y^x$] aufgerufenen Unterprogrammen sechs zusätzliche Unterprogrammebenen zur Verfügung stehen.

Beachten Sie, daß die Fehlermeldung **Error 5** erscheint, wenn sieben Rücksprünge anstehen und ein weiteres Unterprogramm aufgerufen wird. (Siehe Seite 105.)

Speicheranforderungen

$\int_y^x$ verwendet bei der Ausführung 23 Speicherregister. (Im Anhang C wird erklärt wie diese automatisch vom Speicher abgeteilt werden.) Sind keine 23 unbelegte Speicherregister vorhanden, kann $\int_y^x$ nicht ausgeführt werden, und die Meldung **Error 10** erscheint in der Anzeige.

Ein Routine, die die Funktionen $\int_y^x$ und SOLVE gemeinsam verwendet, benötigt ebenfalls 23 Register.

Zusätzliche Informationen

Der in diesem Abschnitt behandelte Stoff läßt Sie $\int_y^x$ in vielen Anwendungsbereichen erfolgreich einsetzen. In Anhang E werden weitere Techniken und Anwendungen in Verbindung mit $\int_y^x$ beschrieben. Dazu gehören:

- Arbeitsweise von $\int_y^x$.
- Genauigkeit, Fehlerabschätzung und Rechenzeit.
- Fehlerabschätzung und Anzeigeformat.
- Faktoren, die zu falschen Ergebnissen führen können.
- Faktoren, die die Rechenzeit verkürzen.
- Anzeige der momentanen Approximation.

Anhang A

Fehlerbedingungen

Wenn in einer Berechnung eine unzulässige Operation – z.B. eine Division durch Null – enthalten ist, erscheint Meldung **Error** und eine Zahl. Jede Fehlermeldung kann durch Auslösen einer beliebigen Taste gelöscht werden; die vorhergehende Anzeige erscheint dann wieder.

Im folgenden sind die möglichen Fehlermeldungen und die zugehörigen Fehlerbedingungen gelistet. (Die Beschreibung von **Error 2** umfaßt eine Liste der benutzten statistischen Formeln.)

Error 0: Unzulässige mathematische Operation

Unzulässiges Argument in einer der folgenden mathematischen Routinen:

[÷], wo $x = 0$.

[y^x], wo

- im reellen Modus $y < 0$ und x nicht ganzzahlig.
- im reellen Modus $y = 0$ und $x \leqslant 0$.
- im Komplex-Modus $y = 0$ und $\mathrm{Re}(x) \leqslant 0$.

[$\sqrt{x}$], wo im reellen Modus $x < 0$.

[1/x], wo $x = 0$.

[LOG], wo

- im reellen Modus $x \leqslant 0$; oder
- im Komplex-Modus $x = 0$.

[LN], wo

- im reellen Modus $x \leqslant 0$; oder
- im Komplex-Modus $x = 0$.

[SIN⁻¹], wo im reellen Modus $|x| > 1$.

[COS⁻¹], wo im reellen Modus $|x| > 1$.

[STO] [÷], wo $x = 0$.

[RCL] [÷], wo der Inhalt des adressierten Registers 0 ist.

[Δ%], wo der Wert im Y-Register 0 ist.

[HYP⁻¹] [COS], wo im reellen Modus $x < 1$.

[HYP⁻¹] [TAN], wo im reellen Modus $|x| > 1$.

[Cy,x] oder [Py,x], wo

- x oder y nicht ganzzahlig ist;

- $x < 0$ oder $y < 0$;
- $x > y$;
- x oder $y \geq 10^{10}$.

Error 1: Unzulässige Matrixoperation

Anwendung einer Funktion auf eine Matrix, die keine Matrixfunktion ist; d.h. versuchte Ausführung einer nicht für Matrizen zulässigen Operation auf einem Register (sei es das X- oder Y-Register oder ein Datenregister), das eine Matrix enthält.

Error 2: Unzulässige statistische Operation

$\boxed{\bar{x}}$	$n = 0$
$\boxed{s}$	$n \leq 1$
$\boxed{\hat{y},r}$	$n \leq 1$
$\boxed{\text{L.R.}}$	$n \leq 1$

Die Meldung **Error 2** erscheint auch, wenn bei der Berechnung einer der folgenden Formeln eine Division mit Null oder die Quadratwurzel einer negativen Zahl auftreten würde:

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

$$s_x = \sqrt{\frac{M}{n(n-1)}} \qquad s_y = \sqrt{\frac{N}{n(n-1)}} \qquad r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M} \qquad B = \frac{M \Sigma y - P \Sigma x}{n \cdot M}$$

$$\hat{y} = \frac{M \Sigma y + P(n \cdot x - \Sigma x)}{n \cdot M}$$

wo:

$$M = n \Sigma x^2 - (\Sigma x)^2$$

$$N = n \Sigma y^2 - (\Sigma y)^2$$

$$P = n \Sigma xy - \Sigma x \Sigma y$$

(A und B sind die bei der Operation $\boxed{\text{L.R.}}$ berechneten Werte, d.h. $y = Ax + B$.)

Error 3: Unzulässige Adressierung eines Registers oder eines Matrixelements

Das adressierte Speicherregister bzw. das indizierte Matrixelement ist nicht vorhanden.

Error 4: Unzulässige Zeilennummer oder unzulässiger Labelaufruf

Die durch die Zeilennummer adressierte Programmzeile ist momentan nicht belegt oder existiert nicht (Zeilennummer >448); Versuch, mehr als 448 Zeilen in den Programmspeicher zu laden; das aufgerufene Label existiert nicht.

Error 5: Unterprogrammverschachtelung zu tief

Eine Unterprogrammverschachtelung enthält mehr als sieben Verschachtelungsebenen.

Error 6: Unzulässiger Flag

Versuch, einen Flag größer 9 aufzurufen.

Error 7: Rekursiver Aufruf von [SOLVE] oder [$\int_y^x$]

Ein mit [SOLVE] aufgerufenes Unterprogramm enthält ebenfalls eine [SOLVE] Anweisung; ein mit [$\int_y^x$] aufgerufenes Unterprogramm enthält ebenfalls eine [$\int_y^x$] Anweisung.

Error 8: Keine Nullstelle

[SOLVE] findet mit der gegebenen Anfangsnäherung keine Nullstelle.

Error 9: Service

Bei der Funktionsprüfung wurde ein Schaltkreisfehler entdeckt, oder während des Tastentests wurde eine falsche Taste gedrückt. Siehe Anhang F.

Error 10: Nicht ausreichender Speicherplatz

Es ist nicht genügend Speicherplatz verfügbar, um eine gegebene Operation auszuführen.

Error 11: Unzulässiges Matrix-Argument

Inkonsistente oder unzulässige Matrixargumente bei einer gegebenen Matrizenoperation:

[+] oder [−] bei inkompatiblen Dimensionen.

[×], wo:
- nicht kompatible Dimensionen auftreten; oder
- die Ergebnismatrix eine der Argumentmatrizen ist.

[1/x] bei nicht quadratischer Matrix.

Skalar/Matrix [÷], wenn die Matrix nicht quadratisch ist.

[÷], wo:
- die Matrix im X-Register nicht quadratisch ist;
- die Dimensionen nicht kompatibel sind; oder
- die Ergebnismatrix die Matrix im X-Register ist.

[MATRIX] 2, wo der Operand ein Skalar oder die Anzahl der Zeilen ungerade ist.

[MATRIX] 3, wo der Operand ein Skalar oder die Anzahl der Zeilen ungerade ist.

[MATRIX] 4, wo der Operand ein Skalar ist.

[MATRIX] 5, wo:
- ein Operand ein Skalar ist;
- die Dimensionen nicht kompatibel sind; oder
- die Ergebnismatrix mit einem der Argumente identisch ist.

[MATRIX] 6, wo:
- ein Operand ein Skalar ist;
- die Dimensionen nicht kompatibel sind (einschließlich der Ergebnismatrix); oder
- die Ergebnismatrix mit einem der Argumente identisch ist.

[MATRIX] 9, wenn die Matrix nicht quadratisch ist.

[RCL] [DIM] [I], wenn der Inhalt von RI ein Skalar ist.

[DIM] [I], wenn der Inhalt von RI ein Skalar ist.

[STO] [RESULT], wenn der Operand ein Skalar ist.

[Py,x], wenn die Anzahl der Spalten ungerade ist.

[Cy,x], wenn die Anzahl der Zeilen ungerade ist.

Pr Error (*Power Error*)

Permanentspeicher wurde wegen Stromausfall gelöscht.

Anhang B

Stack Lift und LAST X

Bei der Konstruktion des HP-15C wurde auf eine möglichst «natürliche» Arbeitsweise Wert gelegt. Wie Sie beim Durcharbeiten dieses Handbuchs gemerkt haben werden, brauchen Sie nur selten über die Abläufe im automatischen Speicherstack nachzudenken – Sie lösen jedes Problem wie mit Papier und Bleistift, immer eine Operation nach der anderen.

Es kann jedoch insbesondere bei der Programmierung des HP-15C von Interesse sein, die Auswirkungen einer bestimmten Operation auf den Stack zu kennen. Die folgenden Erläuterungen sollen Ihnen dabei helfen.

Abschluß der Zifferneingabe

Die meisten Operationen des Rechners, sei es nun bei der Ausführung einer Anweisung innerhalb eines Programms oder über das Tastenfeld, beenden die Eingabe von Ziffern. Dies besagt, daß der Rechner jede nach Abschluß einer dieser Operationen eingegebene Ziffer als Teil einer neuen Zahl auffaßt.

[0] drücke [9]	[CHS]	[←]
[·]	[EEX]	

Stack Lift

Die Operationen des Rechners lassen sich je nach ihrer Auswirkung auf den Stack Lift in drei Klassen einteilen. Dies sind stacksperrende (stack-*disabling*) Operationen, stackfreigebende (stack-*enabling*) Operationen und *neutrale* Operationen.

Im Komplex-Modus wird jede Operation sowohl auf den reellen als auch auf den imaginären Stack-Registern ausgeführt. Die Auswirkungen bzgl. des Stack Lifts sind identisch. Weiterhin wird *nach jeder beliebigen Operation außer* [←] *oder* [CLx] bei Eingabe einer Zahl in die Anzeige (reelles X-Register) gleichzeitig eine Null ins imaginäre X-Register geladen.

Sperrende Operationen

Stack Lift. Der Rechner verfügt über vier stacksperrende Operationen*. Bei diesen Operationen wird der Stack Lift gesperrt, so daß eine nachfolgende Eingabe den momentanen Inhalt des angezeigten X-Registers überschreibt und der Stack nicht nach oben verschoben wird. Die folgenden Operationen sperren den Stack:

[ENTER] [CLx] [Σ+] [Σ−]

Imaginäres X-Register. Bei der Eingabe oder dem Abruf der nächsten Zahl nach einer der Operationen [ENTER], [Σ+] oder [Σ−] in die Anzeige (reelles X-Register) wird eine Null in das imaginäre X-Register geladen. Eingabe oder Abruf einer Zahl nach [←] oder [CLx] verändert den Inhalt des imaginären X-Registers *nicht*.

Freigebende Operationen

Stack Lift. Die meisten Operationen auf dem Tastenfeld des Rechners, einschließlich der mathematischen Funktionen einer und zweier Variablen wie [x^2] und [×], sind stackfreigebende Operationen. Diese Operationen geben den Stack frei, so daß bei Eingabe einer neuen Zahl der Stackinhalt nach oben verschoben wird. Dies gilt für den reellen und den imaginären Stack gleichermaßen. (Schattierung des X-Registers bedeutet, daß die nächste Zahleneingabe oder der nächste Abruf den derzeitigen Inhalt überschreibt.)

T	*t*	*z*	*y*	*y*
Z	*z*	*y*	*x*	*x*
Y	*y*	*x*	4.0000	4.0000
X	*x*	4	4.0000	3

Tasten: 4 [ENTER] 3

(Unterstellt freigegebenen Stack.)	Anheben des Stacks.	Stack gesperrt.	Kein Stack Lift.

* Siehe Fußnote auf Seite 36.

T	y	y	y	y
Z	x	x	x	x
Y	4.0000	53.1301	53.1301	53.1301
X	3	5.0000	0.0000	7
Tasten:		g →P	g CLx	7
		Stack freigegeben.	Stack gesperrt.	Kein Stack Lift.

Imaginäres X-Register. Nach allen den Stack freigebenden Funktionen wird *bei Eingabe oder Abruf der nächsten Zahl in die Anzeige* eine Null ins imaginäre X-Register geschrieben.

Neutrale Operationen

Stack Lift. Einige Operationen, wie z.B. FIX, sind neutral, d.h. der bestehende Status des Stack wird durch sie nicht verändert. Wenn beispielsweise der Stack Lift mit ENTER gesperrt worden ist, so bewirkt das Drücken von f FIX *n* und die Eingabe einer neuen Zahl ein Überschreiben des Inhalts des X-Registers und keine Verschiebung des Stacks. Im anderen Fall, bei einer Freigabe des Stacks z.B. durch √x, bewirkt die Ausführung einer FIX Anweisung, gefolgt von der Eingabe einer Ziffernsequenz, eine Verschiebung des Stacks nach oben*.

Die folgenden Operationen sind neutral:

FIX	GRD	USER	R/S
SCI	GTO CHS *nnn*	CLEAR PREFIX	P/R
ENG	BST	CLEAR REG	(i)**
DEG	SST	CLEAR Σ	
RAD	MEM	PSE	

Imaginäres X-Register. Die zuvor aufgeführten Operationen verhalten sich auch bezüglich Löschung des imaginären X-Registers neutral.

* Alle Zifferntasten sind *während* der Zifferneingabe ebenfalls neutral. Nach der Zifferneingabe geben CHS und EEX den Stack frei; ← sperrt den Stack.

** Gemeint ist die Tastenfolge f (i), die zur Anzeige des imaginären X-Registers dient.

LAST X Register

Bei den folgenden Operationen wird x ins LAST X-Register «gerettet»:

[−]	[x^2]	[HYP^{-1}] [COS]	[%]
[+]	[SIN]	[HYP^{-1}] [TAN]	[Δ%]
[×]	[COS]	[→H.MS]	[→P]
[÷]	[TAN]	[→H]	[→R]
[ABS]	[SIN^{-1}]	[→DEG]	[Py,x] *
[FRAC]	[COS^{-1}]	[→RAD]	[Cy,x] *
[INT]	[TAN^{-1}]	[LN]	[Σ+]
[RND]	[HYP] [SIN]	[e^x]	[Σ−]
[$1/x$]	[HYP] [COS]	[LOG]	[$\hat{y},r$]
[$x!$]	[HYP] [TAN]	[10^x]	[MATRIX] 5 bis 9
[$\sqrt{x}$]	[HYP^{-1}] [SIN]	[y^x]	[$\int_y^x$] **

* Außer bei der Verwendung als Matrix-Funktion.

** [$\int_y^x$] verwendet das LAST X Register auf besondere Weise; siehe Anhang E.

Anhang C

Speicheraufteilung

Aufbau des Speichers

Datenregister, Programmzeilen, und die höheren Funktionen* teilen sich den Speicherbereich des HP-15C. Die *Verfügbarkeit* von Speicherplatz für einen speziellen Zweck hängt von der momentanen *Aufteilung* des Speicherbereichs und natürlich von der gesamten Speicherkapazität des Rechners ab.

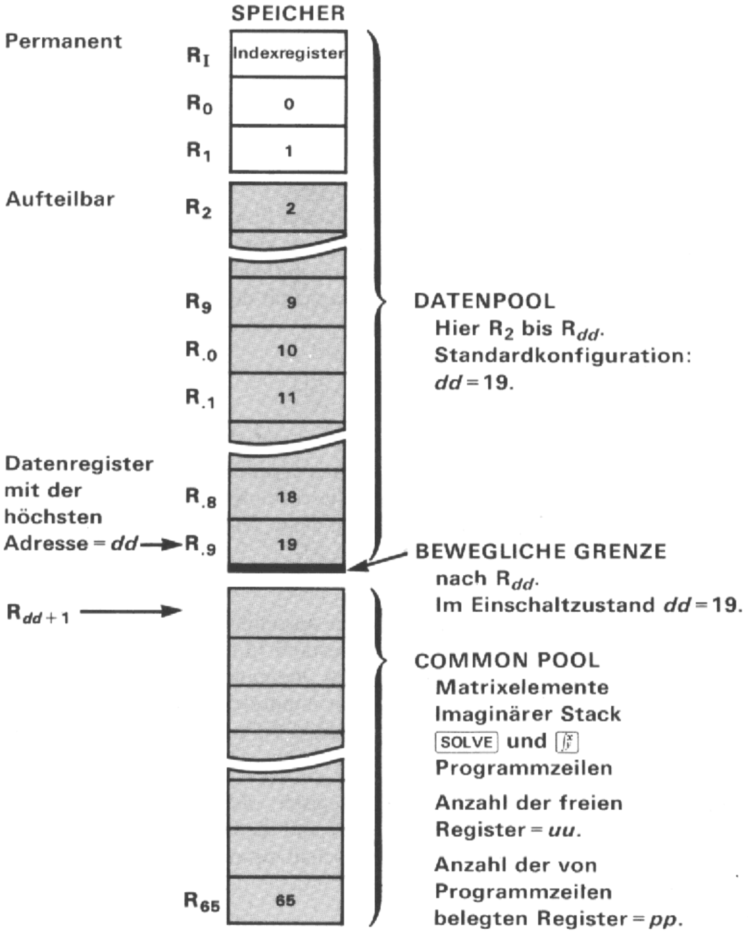
Register

Der Speicher des HP-15C ist auf *Register*basis organisiert. Er ist in zwei *Pools* aufgeteilt, und die Zugehörigkeit zu einem dieser Pools legt fest, ob ein Register der Daten- oder der Programmspeicherung dient. Die Gesamtanzahl der für diese beiden Pools verfügbaren Register ist 67.

- Der *Datenpool* enthält die Register, die ausschließlich für Datenspeicherung vorgesehen sind. Im Einschaltzustand (d.h. nach dem allerersten Einschalten oder einem Löschen des Permanentspeichers) sind dies 21 Register. Dieser Pool enthält immer mindestens drei Register: R_I, R_0 und R_1.

- Der *Common Pool* enthält freie Register, die für Programmzeilen, Matrixelemente, den imaginären Stack und die Operationen [SOLVE] und [∫] verwendet werden können. Im Einschaltzustand sind 46 freie Register im Common Pool verfügbar.

* Die Funktionen [SOLVE], [∫], Komplex-Modus sowie die Matrixfunktionen benötigen befristet zusätzlichen Speicherplatz; wir kommen darauf später in diesem Anhang zurück.



Verfügbarer Speicher: 64 Register, numeriert von R_2 bis R_{65}. $[(9dd-1) + uu + pp + \text{(Matrixelemente)} + \text{(imaginärer Stack)} + (\boxed{\text{SOLVE}} \text{ und } \boxed{\int_y^x})] = 64$. Bei der Speicheraufteilung und indirekten Adressierung werden die Datenregister $R_{.0}$ bis $R_{.9}$ mit R_{10} bis R_{19} bezeichnet.

Speicher-Status (MEM)

Drücken Sie g MEM, um die derzeitige Speicheraufteilung anzuzeigen. Die Anzeige bleibt erhalten, solange Sie MEM* drücken, und enthält vier Zahlen:

dd uu pp-b

wobei:

dd = Nummer des *höchsten* Registers im Datenspeicherpool (die Gesamtanzahl Datenregister ist also *dd* = 2, einschließlich R_0 und R_I);

uu = Anzahl der *freien* Register im Common Pool;

pp = Anzahl der *Programmzeilen* enthaltenden Register; und

b = die Zahl noch verfügbarer Bytes, bevor *uu* um ein Register erniedrigt (und *pp* erhöht) wird, um sieben weitere Programmzeilen zur Verfügung zu stellen.

Im Einschaltzustand gilt:

19 46 0-0

Die bewegliche Grenze zwischen den Datenregistern und dem Common Pool liegt immer zwischen R_{dd} und R_{dd+1}.

Neuaufteilung des Speicherbereichs

Der Speicher umfaßt 67 Register zu je 7 Bytes. 64 dieser Register (R_2 bis R_{65}) können wahlweise dem Datenspeicherpool oder dem Common Pool zugewiesen werden.

Die Funktion DIM (i)

Wenn Sie entweder mehr Speicherraum im Common Pool (z.B. für Programmspeicherung) oder im Datenspeicherpool (aber nicht gleichzeitig!) benötigen, können Sie mit DIM (i) auf die folgende Weise die notwendige Speicherraum-Umverteilung durchführen**:

* MEM ist nicht programmierbar.

** DIM (*dimension*) wird auch benutzt, um Matrizen zu dimensionieren. Hier jedoch dient es zur «Dimensionierung» der Größe des Datenspeicherpools.

1. Geben Sie *dd*, die Nummer des *höchsten benötigten Datenregisters, das zugeteilt werden soll,* in die Anzeige. $1 \leqslant dd \leqslant 65$. Im freien Pool (und daher gegebenenfalls für Programmierung verfügbar) befinden sich $65 - dd$ Register.
2. Drücken Sie [f] [DIM] [(i)].

Sie können die Neuaufteilung des Speichers auf zwei Weisen zur Anzeige bringen:

- Rufen Sie mit [RCL] [DIM] [(i)] die Nummer des höchsten Datenregisters *dd* in den Stack (programmierbar).
- Mit [g] [MEM] (wie zuvor erläutert) erhalten Sie eine vollständige Anzeige des Speicher-Status (*dd uu pp-b*).

Tastenfolge	**Anzeige**	
(Annahme: *Der Programmspeicher ist gelöscht*)*		
1 [f] [DIM] [(i)] [g] [MEM] (gehalten)	**1.0000** **1 64 0-0**	R_I, R_0 und R_1 sind der Datenspeicherung zugeteilt. 64 Register sind noch frei; noch kein Register enthält Programmanweisungen.
19 [f] [DIM] [(i)] [RCL] [DIM] [(i)]	**19.0000** **19.0000**	R_{19} ($R_{.9}$) ist das Datenregister mit der höchsten Nummer. Im Common Pool stehen noch 46 Register zur Verfügung.

Beschränkungen bei der Neuaufteilung

Der Permanentspeicher erhält die vorgegebene Speicherkonfiguration solange, bis er gelöscht wird oder [DIM] [(i)] erneut ausgeführt wird. Wenn Sie versuchen, weniger als ein Register dem Datenpool zuzuordnen, gilt $dd = 1$. Beim Versuch, mehr als 65 Register dem Datenpool zuzuordnen erfolgt die Anzeige **Error 10**.

* Bei nicht gelöschtem Programmspeicher stehen weniger freie Register (*uu*) zur Verfügung, da sie dem Programmspeicher (*pp*) zugeordnet sind. In diesem Fall wäre $pp > 0$ und *b* könnte variieren.

Beachten Sie beim Umwandeln von Registern:

- Sie können nur *freie* Register des Common Pools umwandeln. Wenn Sie z.B. versuchen, mit Programmanweisungen belegte Register umzuwandeln, meldet der Rechner **Error 10**.
- Sie können belegte Datenregister umwandeln, *die gespeicherten Daten sind dann verloren*. Beim Versuch, ein «verlorenes», d.h. nichtexistentes Datenregister zu adressieren, erhalten Sie die Meldung **Error 3**. Es ist daher ratsam, Daten zuerst in den unteren Registern abzuspeichern, da diese als letzte umgewandelt werden.

Programmspeicher

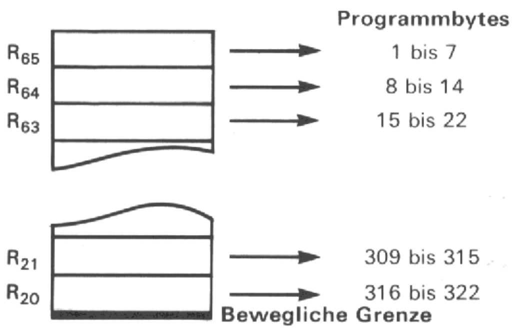
Wie schon erwähnt, besteht jedes Register aus sieben Bytes. Programmanweisungen belegen ein oder zwei Bytes im Speicher. Die meisten Programmzeilen benötigen nur ein Byte; die Anweisungen, die zwei Bytes belegen, sind auf Seite 218 aufgelistet.

Die *maximale* Programmkapazität des HP-15C beträgt 448 Programmbytes (64 umwandelbare Register mit je sieben Bytes). Im Einschaltzustand kann der Programmspeicher bis zu 322 Programmbytes aufnehmen (46 zugeordnete Register zu je sieben Bytes).

Automatische Speicherumwandlung

Innerhalb des Common Pools wird der Programmspeicher je nach Bedarf automatisch erweitert. Ein freies Register, das höchste verfügbare zuerst, wird in sieben Bytes Programmspeicher umgewandelt.

Umwandlung freier Register in Programmspeicher



Ihre allererste Programmanweisung verwandelt das freie Register R_{65} (alle sieben Bytes) in ein Programmspeicherregister. Die achte Programmanweisung wandelt R_{64} um usw., bis Sie an die Grenze des Common Pools stoßen. Register des Datenspeicherpools (beim Einschalten die Register unterhalb R_{20}) sind nicht ohne Neuverteilung mit DIM (i) für die Programmierung verfügbar.

2-Byte Programmanweisungen

Die folgenden Anweisungen belegen zwei Bytes im Programmspeicher. (Alle übrigen Anweisungen belegen nur ein Byte.)

f LBL . *Label*
f GTO . *Label*
g CF (*n* oder I)
g SF (*n* oder I)
g F? (*n* oder I)
f FIX (*n* oder I)
f SCI (*n* oder I)
f ENG (*n* oder I)
f SOLVE
f ∫
f MATRIX {0 bis 9}
f x≷ {2 bis 9, .0 bis .9}
f DSE {2 bis 9, .0 bis .9}
f ISG {2 bis 9, .0 bis .9}
STO {+, −, ×, ÷}
RCL {+, −, ×, ÷}
STO MATRIX {A bis E}
STO {A bis E, (i)} im User-Modus
RCL {A bis E, (i)} im User-Modus
STO g (i)
RCL g (i)

Speicheranforderungen der höheren Funktionen

Die vier höheren Funktionen benötigen vorübergehend Registerplatz aus dem Common Pool.

Funktion	Benötigte Register
SOLVE	5 } 23 bei gleichzeitiger Verwendung
∫	23 }
Komplexer Stack	5
Matrizen	1 pro Matrixelement

Bei [SOLVE] und [∫] wird der benötigte Speicherplatz automatisch zur Verfügung gestellt und zurückgegeben*. Der Speicherplatz wird also nur für die Dauer der Operation belegt.

Bei jeder Ausführung von [f] [I], [f] [Re≷Im] oder [g] [SF] 8 wird dem imaginären Stack Speicherplatz zugeteilt. Dieser Speicherplatz wird bei Eingabe von [g] [CF] 8 zurückverwandelt. Matrixelementen wird Speicherplatz erst dann zugeteilt, *wenn eine Matrix* (*mit* [DIM]) *dimensioniert wird.* [MATRIX] 0 dimensioniert *alle* Matrizen auf 0×0.

* Falls Sie [SOLVE] oder [∫] *in der Ausführung* durch Tastendruck unterbrechen, können Sie die zugewiesenen Register mit [g][RTN] oder mit [f] CLEAR [PRGM] im Run-Modus wieder freigeben.

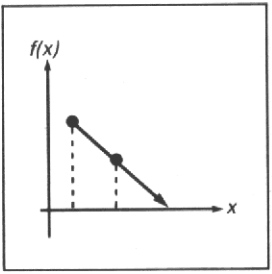
Anhang D

[SOLVE] im Detail

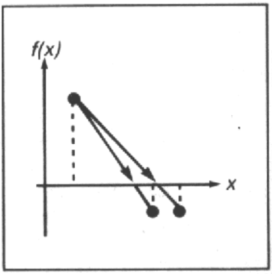
Die grundlegenden Informationen zur Anwendung des [SOLVE] Algorithmus finden Sie in Abschnitt 13, Nullstellenbestimmung. Dieser Anhang enthält tiefergehende Betrachtungen und zusätzliche Aspekte zur Funktion [SOLVE].

Arbeitsweise von [SOLVE]

Zur optimalen Anwendung von [SOLVE] ist ein Verständnis der Arbeitsweise des Algorithmus unerläßlich.



Bei der Suche nach Nullstellen der spezifizierten Funktion benutzt der Algorithmus die Funktionswerte von zwei oder drei vorhergehenden Näherungen, um den Verlauf des Graphen der Funktion zu approximieren. Mit Hilfe dieses Kurvenverlaufs wird eine verbesserte Näherung der Schnittstelle des Graphen mit der x-Achse berechnet. Dann wird mit Hilfe des Funktions-Unterprogrammes der Wert der Funktion an der neuen Näherungsstelle berechnet. Dieser Vorgang wird vom [SOLVE] Algorithmus solange wiederholt, bis eine Nullstelle gefunden oder zu einem Fehlerausgang verzweigt wird.



Falls zwei Näherungen Funktionswerte mit verschiedenem Vorzeichen ergeben, nimmt der Algorithmus an, daß der Graph der Funktion im Intervall zwischen diesen beiden Näherungen die x-Achse mindestens einmal schneidet. Dieses Intervall wird dann systematisch verkleinert, bis eine Nullstelle gefunden wird.

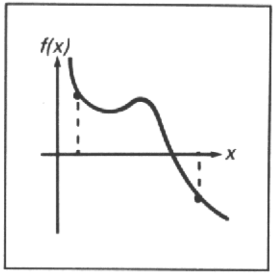
Eine Nullstelle gilt als gefunden, wenn entweder der berechnete Funktionswert gleich Null ist, oder wenn zwei sich auf der letzten Stelle nur um eine Einheit unterscheidende Näherungen Funktionswerte verschiedenen Vorzeichens ergeben. In diesem Fall wird die Ausführung beendet und die Näherung angezeigt.

Wie in Abschnitt 13, Seite 186 erläutert wurde, deuten bestimmte Ergebnisse während der Iteration an, daß keine Nullstelle existiert. Der Grund dafür liegt darin, daß der Rechner keine neue Näherung mit einem näher bei Null liegenden Funktionswert findet. In diesem Fall wird **Error 8** angezeigt.

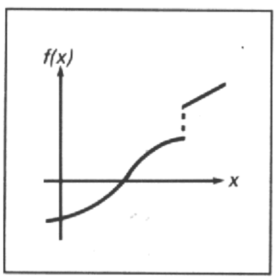
Beachten Sie, daß der «Vorhersage»prozeß mit den von Ihnen vorgegebenen Näherungen gestartet wird. Sorgfältig gewählte Anfangsnäherungen können das Auffinden der gesuchten Nullstelle sehr vereinfachen und verkürzen.

Erfüllt Ihre Funktion eine der folgenden vier Bedingungen, dann wird *immer* eine Nullstelle gefunden, vorausgesetzt sie existiert und liegt im Zahlenbereich des Rechners:

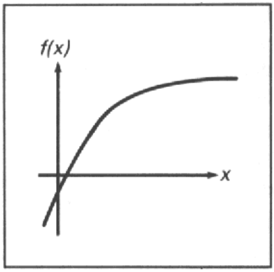
- Zwei beliebige Näherungen haben verschiedene Vorzeichen.



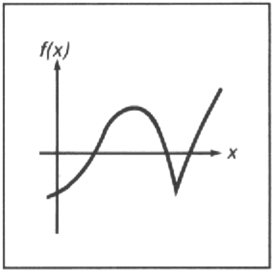
- Die Funktion ist auf dem betrachteten Intervall monoton wachsend oder fallend.



- Der Graph der Funktion ist auf dem betrachteten Intervall konvex oder konkav.



- Die Funktion enthält keine lokalen Minima oder Maxima zwischen benachbarten Nullstellen.



Weiterhin wird angenommen, daß der SOLVE Algorithmus nicht durch eine unerlaubte Operation unterbrochen wird.

Genauigkeit der Nullstelle

Mit dem SOLVE Algorithmus wird die Nullstelle einer Gleichung exakt bestimmt. Der Funktionswert der angezeigten Nullstelle ist entweder exakt gleich Null oder ein zehnstelliger Funktionswert eines direkt an die Schnittstelle mit der x-Achse angrenzenden Wertes. Jede solche Nullstelle ist auf ± 3 Einheiten in der 10. Stelle genau.

Normalerweise ist die berechnete Nullstelle eine genaue Näherung der theoretischen (auf unendlich viele Stellen bestimmten) Nullstelle. Unter gewissen Bedingungen kann jedoch ein von der exakten Nullstelle abweichendes Ergebnis auftreten.

Ist das Ergebnis einer Berechnung betragsmäßig kleiner als $1.000000000 \times 10^{-99}$, wird es gleich Null gesetzt. Dies wird als «Underflow» bezeichnet. Wenn Ihr Unterprogramm bei der Auswertung von x in einem bestimmten Intervall einen Underflow erzeugt, und dies den berechneten Funktionswert beeinflußt, dann ist die Nullstelle in diesem Bereich möglicherweise ungenau. Die Gleichung

$$x^4 = 0$$

hat zum Beispiel eine Nullstelle bei $x = 0$. Wegen des Underflow berechnet [SOLVE] jedoch den Wert **1.5060 –25** (bei Anfangsnäherungen 1 und 2). Ein weiteres Beispiel

$$1/x^2 = 0$$

besitzt eine Nullstelle im Unendlichen. Wegen des Underflows berechnet [SOLVE] die Nullstelle **3.1707 49** (bei Anfangsnäherungen 10 und 20). In beiden Beispielen hat der Algorithmus einen Wert x gefunden, dessen Funktionswert Null ist. Wenn Sie die Auswirkungen des Underflow verstehen, können Sie solche Ergebnisse leicht interpretieren.

Die Genauigkeit eines berechneten Wertes kann manchmal durch einen Rundungsfehler negativ beeinflusst werden, wenn eine Zahl mit unendlich vielen Dezimalstellen auf 10 gültige Stellen gerundet werden muß. Wenn die Berechnung der Funktionswerte für einen Bereich von x in Ihrem Unterprogramm erhöhte Genauigkeit erfordert, kann das mit [SOLVE] erhaltene Ergebnis ungenau sein. Die Gleichung

$$|x-5| = 0$$

hat z.B. eine Lösung bei $x = \sqrt{5}$. Da $\sqrt{5}$ nicht mit 10 Stellen *exakt* dargestellt werden kann, ergibt [SOLVE] die Anzeige **Error 8** (für beliebige Anfangswerte), da die Funktion nie gleich Null wird und nie das Vorzeichen wechselt. Andererseits besitzt die Gleichung

$$[(|x| + 1) + 10^{15}]^2 = 10^{30}$$

keine Nullstellen, da die linke Seite immer größer als die rechte Seite ist. Wegen Rundungsfehlern in der Berechnung von

$$f(x) = [(|x| + 1) = 10^{15}]^2 - 10^{30},$$

wird für die Anfangswerte 1 und 2 die Nullstelle **1.0000** gefunden. Wenn Sie Situationen, in denen ein Rundungsfehler die Ausführung von SOLVE beeinflussen kann, erkennen, können Sie das Ergebnis entsprechend beurteilen und vielleicht die Funktion umformuliern, um so den Rundungsfehler zu verringern.

In einer Vielzahl praktischer Anwendungen sind die Parameter einer Gleichung – oder vielleicht die Gleichung selbst – nur *Approximationen*. Physikalische Parameter besitzen eine inhärente Ungenauigkeit. Mathematische Darstellungen physikalischer Prozesse sind nur Modelle dieser Prozesse, nur genau in dem Maße, wie die zugrundeliegenden Annahmen zutreffen. Ein Verständnis dieser und anderer Ungenauigkeiten kann zu Ihrem Vorteil sein. Wenn Sie Ihr Funktionsunterprogramm so schreiben, daß ein für praktische Zwecke vernachlässigbarer Funktionswert Null gesetzt wird, können Sie gewöhnlich bei der Anwendung von SOLVE in sonst langdauernden Näherungsprozessen viel Zeit sparen.

Beispiel: Diskuswerfer wie Chuck Fahr können ihre Scheibe 105 und mehr Meter hoch werfen. Fahrs Würfe erreichen gewöhnlich eine Höhe von 107 Metern. Wie lange dauert es bei diesem außergewöhnlichen Wurf, bis die Scheibe 107 Meter Höhe erreicht?

Lösung: Die gesuchte Lösung ist der Wert t, für den $h = 107$ m gilt. Geben Sie das Unterprogramm von Seite 184 ein, das die Höhe der Scheibe berechnet. Dieses Unterprogramm kann in einem neuen Funktions-Unterprogramm benutzt werden, welches

$$f(t) = h(t) - 107$$

berechnet. Das folgende Unterprogramm berechnet $f(t)$:

Tastenfolge	**Anzeige**	
g P/R	**000–**	Programm-Modus.
f LBL B	**001–42,21,12**	Beginn mit neuem Label.
GSB A	**002– 32 11**	Berechnung von $h(t)$.

Tastenfolge	**Anzeige**	
1	**003– 1**	
0	**004– 0**	
7	**005– 7**	Berechnet $h(t)-107$.
−	**006– 30**	
g RTN	**007– 43 32**	

Verwenden Sie die Anfangswerte 0 und 1 und führen Sie SOLVE mit B aus, um den Zeitpunkt zu finden, an dem der Diskus zum ersten Mal die Höhe 107 m erreicht.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
0 ENTER	0.0000	Anfangsnäherungen.
1	1	
f SOLVE B	4.1718	Gesuchte Nullstelle.
R↓	4.1718	Vorletzte Näherung der Nullstelle.
R↓	0.0000	Funktionswert von $f(t)$ an der Nullstelle.

Die Scheibe braucht 4.1718 Sekunden, um eine Höhe von genau 107 m zu erreichen. (Das Auffinden dieser Lösung dauert ca. eine Minute.)

Nehmen Sie jedoch an, daß die Funktion $h(t)$ nur auf ganze Meter genau ist. Sie können dann Ihr Unterprogramm ändern, so daß es $f(t) = 0$ setzt, sobald der berechnete Wert für $f(t)$ kleiner als ein Meter ist. Ändern Sie Ihr Unterprogramm folgendermaßen:

Tastenfolge	Anzeige	
g P/R	000–	Programm-Modus.
GTO CHS 006	006– 30	Zeile vor der RTN-Anweisung.
g ABS	007– 43 16	Betrag von $f(t)$.
·	008– 48	Toleranz.
5	009– 5	
g TEST 7	010–43,30, 7	Abfrage auf $x > y$ und Rückgabe von Null falls $(0.5 > \lvert f(t)\rvert$.
g CLx	011– 43 35	
g TEST 0	012–43,30, 0	Abfrage auf $x \neq 0$; Rückruf von $f(t)$, wenn diese Bedingung erfüllt ist.
g LSTx	013– 43 36	

Führen Sie SOLVE erneut aus:

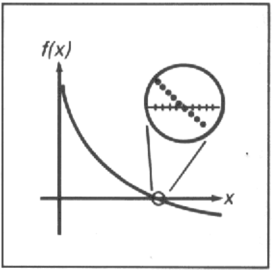
Tastenfolge	Anzeige	
g P/R		Run-Modus.
0 ENTER	0.0000	Anfangsnäherungen.
1	1	
f SOLVE B	4.0681	Gesuchte Nullstelle.
R↓	4.0681	Vorletzte Näherung der Nullstelle.
R↓	0.0000	Wert der modifizierten Funktion $f(t)$ an der Nullstelle.

Nach 4.0681 Sekunden hat die Scheibe eine Höhe von 107 ± 0.5 Metern erreicht. Diese Lösung ist zwar von der zuvor erhaltenen verschieden, aber dennoch korrekt, wenn die Ungenauigkeit der Höhengleichung in Betracht gezogen wird. (Und diese Lösung wird in weniger als der Hälfte der Zeit berechnet.)

Interpretation von Ergebnissen

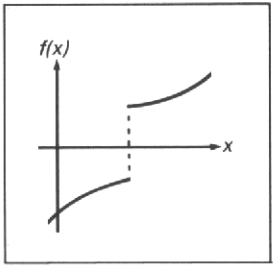
Die von SOLVE in den Registern X, Y und Z abgelegten Zahlen können Ihnen bei der Bewertung der Ergebnisse einer Nullstellensuche helfen*. Auch wenn keine Nullstelle gefunden wurde, sind die Ergebnisse trotzdem von Bedeutung.

Wenn SOLVE auf eine Nullstelle der spezifizierten Gleichung stößt, werden die Nullstelle und deren Funktionswert in den Registern X und Z abgelegt. Für den Funktionswert wird der Wert Null erwartet. Wenn Sie einen von Null verschiedenen Funktionswert erhalten, schneidet der Graph der Funktion die x-Achse nicht bei der angegebenen Nullstelle, sondern in deren unmittelbaren Umgebung. In den meisten Fällen wird der Funktionswert sehr nahe an Null liegen.

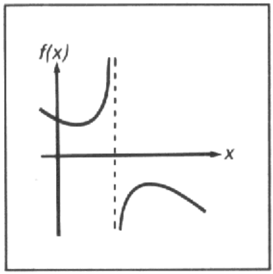


* Die Zahl im T-Register ist die bei der letzten Ausführung des Funktions-Unterprogrammes im Y-Register gebliebene Zahl. Im Normalfall ist dieser Wert nicht von Interesse.

Die folgende Situation, wo SOLVE eine Nullstelle mit von Null verschiedenem Funktionswert findet, verdient besondere Beachtung: besitzt der Graph dieser Funktion eine Unstetigkeitsstelle, die die x-Achse überspringt, wird von SOLVE ein an die Sprungstelle angrenzender x-Wert als Nullstelle angegeben. Dies ist sinnvoll, weil ein großer Unterschied im Funktionswert zweier nebeneinanderliegender x-Werte auch ein sehr starker stetiger Anstieg sein kann. Weil der Algorithmus dies nicht unterscheiden kann, wird die Nullstelle zu Ihrer Beurteilung angezeigt.



Eine Funktion kann einen *Pol*, eine Unendlichkeitsstelle, besitzen. Wenn der Funktionswert im Pol sein Vorzeichen wechselt, wird der zugehörige x-Wert als Nullstelle der entsprechenden Funktion aufgefaßt, genau wie bei jeder anderen Sprungstelle über die x-Achse. Bei solchen Funktionen wird jedoch der nach dem Auffinden der Nullstelle im Z-Register abgelegte Funktionswert der Nullstelle sehr groß sein. Ist der Pol ein x-Wert, der mit 10 Stellen *genau* dargestellt werden kann, benutzt das Unterprogramm möglicherweise diesen Wert und bricht die Berechnung mit einer Fehlermeldung vorzeitig ab. Dies kann natürlich durch den Einbau von Abfragen in Ihr Unterprogramm vermieden werden.



Beispiel: In seiner Untersuchung der in einem Strukturelement auftretenden Spannungen hat der Ingenieur K. Smart die Scherspannung

$$Q = \begin{cases} 3x^3 - 45x^2 + 350 & \text{für } 0 < x < 10 \\ 1000 & \text{für } 10 \leq x < 14 \end{cases}$$

gefunden, wobei Q die Scherspannung in Newton pro Quadratmeter und x den Abstand vom einen Ende in Metern darstellt.

Schreiben Sie ein Unterprogramm zur Berechnung der Scherspannung für beliebige Werte von x. Berechnen Sie mit SOLVE die Stelle, die scherspannungsfrei ist.

Lösung: Schreiben Sie die Gleichung für die Scherspannung mit Hilfe des Horner-Schemas um, um die Programmierung effizienter zu machen:

$$Q = (3x-45)x^2 + 350 \qquad \text{für } 0 < x < 10.$$

Tastenfolge	Anzeige	
g P/R	000–	Programm-Modus.
f LBL 2	001–42,21, 2	
1	002– 1	
0	003– 0	Abfrage auf Bereich von x.
g $x \leqslant y$	004– 43 10	
GTO 9	005– 22 9	Verzweigung wenn $x \geqslant 10$.
g CLx	006– 43 35	
3	007– 3	
×	008– 20	$3x$.
4	009– 4	
5	010– 5	
−	011– 30	$(3x-45)$.
×	012– 20	
×	013– 20	$(3x-45)x^2$.
3	014– 3	
5	015– 5	
0	016– 0	
+	017– 40	$(3x-45)x^2+350$.
g RTN	018– 43 32	Ende des Unterprogramms.
f LBL 9	019–42,21, 9	Unterprogramm für $x \geqslant 10$.
EEX	020– 26	
3	021– 3	$10^3 = 1000$.
g RTN	022– 43 32	Ende des Unterprogramms.

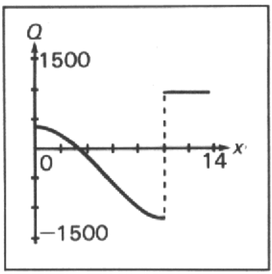
Führen Sie SOLVE mit den Anfangsnäherungen 7 und 14 aus, um am äußeren Ende des Balkens beginnend nach verschwindender Scherspannung zu suchen.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
7 ENTER	**7.0000**	Anfangsnäherungen.
14	**14**	
f SOLVE 2	**10.0000**	Mögliche Nullstelle.
R↓ R↓	**1,000.0000**	Spannung ungleich Null.

Der hohe Spannungswert an der Nullstelle deutet auf eine Unstetigkeit hin. Das ist eine Stelle des Balkens, wo die Spannung abrupt von positiven zu negativen Werten wechselt. Wenden Sie SOLVE noch einmal, aber diesmal vom anderen Ende (Anfangswerte 0 und 7), an.

Tastenfolge	Anzeige	
0 ENTER	**0.0000**	Anfangsnäherungen.
7	**7**	
f SOLVE 2	**3.1358**	Mögliche Nullstellen.
R↓ R↓	**2.0000 –07**	Vernachlässigbare Spannung.

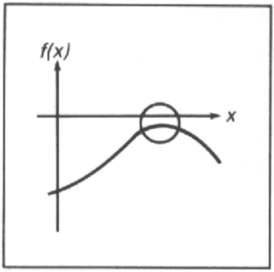
Smart's Balken ist ungefähr 3.1358 m von diesem Ende entfernt scherspannungslos und erfährt in 10.0000 m Abstand eine abrupte Spannungsänderung.



Graph von $Q(x)$

Falls keine Nullstelle gefunden und deshalb **Error 8** angezeigt wird, können Sie mit ← oder einer beliebigen anderen Taste die Anzeige löschen und den Wert erhalten, dessen Funktionswert Null am nächsten liegt. Auch die Zahlen in den Registern Y und Z können oft auf den Verlauf der Funktion nahe der geschätzten Nullstelle hinweisen und deshalb bei der Beurteilung der Nullstelle von Nutzen sein.

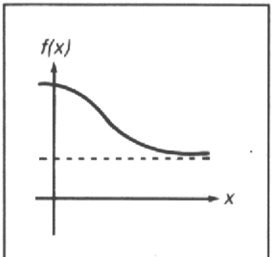
Beendet der Algorithmus seine Suche nach einer Nullstelle in der Nähe eines *betragsmäßigen* lokalen Minimums, löschen Sie die **Error 8** Anzeige und betrachten sie die Inhalte der X-, Y- und Z-Register, indem Sie den Stack nach unten schieben. Ist der Funktionswert, d.h. der Wert im Z-Register, beinahe Null, haben Sie möglicherweise eine Nullstelle gefunden – die in das X-Register zurückgegebene Zahl kann dem Wert der theoretischen Nullstelle sehr nahe kommen.

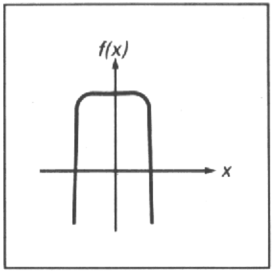


Sie können dieses mögliche Minimum weiter erforschen, indem Sie den Stack wieder nach oben verschieben, bis sich die beiden letzten Näherungen wieder in den X- und Y-Registern befinden. Danach führen Sie SOLVE nochmals aus und verwenden dieses Mal diese Werte als Anfangsnäherungen. Wenn ein wirkliches Minimum gefunden wurde, wird wiederum **Error 8** angezeigt, und der Wert im X-Register wird in etwa derselbe sein wie zuvor, jedoch möglicherweise etwas näher am Minimum liegen.

Selbstverständlich, können Sie SOLVE bewußt dazu verwenden, um die Lage eines lokalen Minimums einer Funktion zu suchen. In diesem Fall müssen Sie die Suche jedoch auf die unmittelbare Umgebung des Minimums beschränken. Denken Sie daran, daß SOLVE immer versucht, eine *Nullstelle* zu finden.

Stoppt der Algorithmus die Suche nach einer Nullstelle und zeigt **Error 8** an, weil er im Bereich einer waagrechten Asymptote sucht (d.h. die Funktionswerte sind über ein großes Intervall von x Werten im wesentlichen konstant), unterscheiden sich die Näherungen im X- und Y-Register gewöhnlich deutlich voneinander. Der Wert im Z-Register ist der Wert der möglichen Asymptote. Wenn Sie SOLVE mit den Werten in den Registern X und Y als Anfangsnäherungen nochmals ausführen, kann eine horizontale Asymptote wiederum **Error 8** verursachen, aber die Register X und Y werden danach von den vorhergehenden verschiedene Werte enthalten. Die Zahl im Z-Register wird im wesentlichen ihren Wert beibehalten.





Wird infolge einer in einem «flachen» Gebiet konzentrierten Suche **Error 8** angezeigt, sind die Näherungen in den Registern X und Y relativ eng zusammen oder extrem klein. Führen Sie SOLVE noch einmal aus und benutzen Sie als Anfangsnäherungen die Werte in den Registern X und Y (oder vielleicht etwas weiter auseinander liegend). Wenn die Funktion in diesem Bereich kein betragsmäßiges Minimum enthält, wird der Algorithmus seine Suche ausdehnen und gegebenenfalls ein signifikantes Ergebnis finden.

Beispiel: Untersuchen Sie das Verhalten der Funktion

$$f(x) = 3 + e^{-|x|/10} - 2e^{x^2 e^{-|x|}}$$

deren Werte mit dem folgenden Unterprogramm berechnet werden können.

Tastenfolge	Anzeige	
g P/R	000–	Programm-Modus.
f LBL .0	001–42,21, .0	
g ABS	002– 43 16	
CHS	003– 16	
e^x	004– 12	$e^{-\|x\|}$.
$x \gtrless y$	005– 34	Lädt den x-Wert ins X-Register.
g x^2	006– 43 11	
×	007– 20	$x^2e^{-\|x\|}$.
e^x	008– 12	
2	009– 2	
×	010– 20	
CHS	011– 16	$-2e^{x^2e^{-\|x\|}}$.
$x \gtrless y$	012– 34	Lädt den x-Wert ins X-Register.
g ABS	013– 43 16	
CHS	014– 16	
1	015– 1	
0	016– 0	
÷	017– 10	$-\|x\|/10$.
e^x	018– 12	

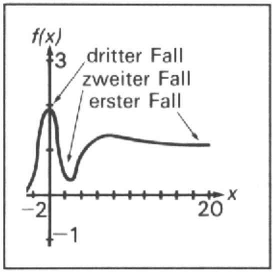
Tastenfolge	Anzeige	
[+]	019– 40	$e^{-\lvert x\rvert/10} - 2e^{x^2 e^{-\lvert x\rvert}}$.
3	020– 3	
[+]	021– 40	$3 + e^{-\lvert x\rvert/10} - 2e^{x^2 e^{-\lvert x\rvert}}$.
[g] [RTN]	022– 43 32	

Führen Sie [SOLVE] mit folgenden *einzelnen* Anfangsnäherungen durch: 10, 1 und 10^{-20}.

Tastenfolge	Anzeige	
[g] [P/R]		Run-Modus.
10 [ENTER]	10.0000	Einzelne Anfangsnäherung.
[f] [SOLVE] .0	Error 8	
[←]	455.4335	Beste Näherung.
[R↓]	48,026,721.85	Vorletzte Näherung.
[R↓]	1.0000	Funktionswert.
[g] [R↑] [g] [R↑]	455.4335	Restaurieren des Stack.
[f] [SOLVE] .0	Error 8	
[←]	48,026,721.85	Neue Näherung.
[R↓] [R↓]	1.0000	Gleicher Funktionswert (eine Asymptote).
1 [ENTER]	1.0000	Einzelne Anfangsnäherung.
[f] [SOLVE] .0	Error 8	
[←]	2.1213	Beste Näherung.
[R↓]	2.1471	Vorletzte Näherung.
[R↓]	0.3788	Funktionswert.
[g] [R↑] [g] [R↑]	2.1213	Restaurieren des Stack.
[f] [SOLVE] .0	Error 8	
[←]	2.1213	Gleiche Näherung.
[R↓] [R↓]	0.3788	Gleicher Funktionswert (ein Minimum).
[EEX] [CHS] 20 [ENTER]	1.0000 –20	Einzelne Anfangsnäherung.
[f] [SOLVE] .0	Error 8	
[←]	1.0000 –20	Beste Näherung.
[R↓]	1.1250 –20	Vorletzte Näherung.
[R↓]	2.0000	Funktionswert.

Tastenfolge	Anzeige		
g R↑ g R↑	1.0000	-20	Restaurieren des Stack.
f SOLVE .0	Error 8		
←	1.1250	-20	Neue Näherung.
R↓	1.5626	-16	Vorletzte Näherung.
R↓	2.0000		Gleicher Funktionswert.

In allen drei Fällen hat sich SOLVE bei der Nullstellensuche von dem Graphen der Funktion in der Umgebung der Ausgangsnäherung leiten lassen. Mit dem Wert 10 als Ausgangswert hat SOLVE die waagerechte Asymptote gefunden (Wert 1.0000). Mit 1 als Ausgangswert wurde das Minimum 0.3788 bei $x = 2.1213$ gefunden. Bei dem Ausgangswert 10^{-20} war die Funktion auf dem untersuchten kleinen Intervall im wesentlichen konstant.



Berechnung mehrerer Nullstellen

Viele Gleichungen, die Sie untersuchen, besitzen mehr als eine Nullstelle. Im folgenden werden einige Techniken zur Berechnung mehrerer Nullstellen einer Funktion vorgestellt.

Die einfachste Methode besteht darin, die Suche in die verschiedenen Bereiche von x zu lenken, die Nullstellen enthalten. Ihre Anfangsnäherungen geben den Bereich vor, der zuerst untersucht wird. Diese Methode wurde bei allen Beispielen in Abschnitt 13 angewandt. Sie können auf diese Weise oft alle Nullstellen einer Gleichung finden.

Eine andere Methode ist die *Reduktion* einer Gleichung. Durch Reduktion werden Nullstellen eliminiert. Dazu muß die Gleichung so verändert werden, daß die schon gefundenen Nullstellen keine Nullstellen mehr sind, aber alle anderen Nullstellen Nullstellen bleiben.

Hat die Funktion $f(x)$ den Wert Null bei $x = a$, dann bleibt die Funktion $f(x)/(x-a)$in diesem Bereich von Null verschieden (falls a eine einfache Nullstelle von $f(x)$ ist). Mit dieser Information können Sie eine bekannte Nullstelle eliminieren. Fügen Sie einfach entsprechende Programmzeilen an das Ende Ihres Funktions-Unterprogramms an. Diese Zeilen sollten die bekannte Nullstelle (auf 10 Stellen genau) vom x-Wert subtrahieren, und den Funktionswert durch diesen Ausdruck dividieren.

In vielen Fällen ist die benutzte Nullstelle eine einfache Nullstelle, und die neue Funktion führt SOLVE von der bekannten Nullstelle weg.

Andererseits kann die Nullstelle eine *mehrfache* Nullstelle sein; d.h., nicht nur die Funktion, sondern auch deren Steigung (und ggf. auch die nächsthöheren Ableitungen) nehmen an der betreffenden Stelle den Wert Null an. Ist die bekannte Nullstelle Ihrer Gleichung eine mehrfache Nullstelle, dann wird diese durch einfaches Dividieren mit dem oben gegebenen Divisor nicht eliminiert. Die Gleichung

$$f(x) = x(x-a)^3 = 0$$

zum Beispiel hat eine dreifache Nullstelle bei $x = a$. Diese Nullstelle wird durch einfaches Dividieren von $f(x)$ mit $(x-a)$ nicht eliminiert. Sie kann aber durch Division mit $(x-a)^3$ eliminiert werden.

Beispiel: Finden Sie die Nullstellen von

$$60x^4 - 944x^3 + 3003x^2 + 6171x - 2890 = 0.$$

durch Reduktion. Nach dem Horner-Schema kann diese Gleichung in die Form

$$(((60x - 944)x + 3003)x + 6171)x - 2890 = 0.$$

umgeschrieben werden. Schreiben Sie ein Unterprogramm, das dieses Polynom auswertet.

Tastenfolge	Anzeige	
g P/R	000-	Programm-Modus.
f CLEAR PRGM	000-	
f LBL 2	001-42,21, 2	
6	002- 6	
0	003- 0	
×	004- 20	
9	005- 9	
4	006- 4	
4	007- 4	
−	008- 30	
×	009- 20	
3	010- 3	
0	011- 0	

Tastenfolge	Anzeige	
0	012–	0
3	013–	3
+	014–	40
×	015–	20
6	016–	6
1	017–	1
7	018–	7
1	019–	1
+	020–	40
×	021–	20
2	022–	2
8	023–	8
9	024–	9
0	025–	0
–	026–	30
g RTN	027–	43 32

Geben Sie im Run-Modus zwei große negative Anfangsnäherungen (z.B. –10 und –20) ein und berechnen Sie mit SOLVE die größte negative Nullstelle.

Tastenfolge	Anzeige	
g P/R		Run-Modus.
10 CHS ENTER	–10.0000	Anfangsnäherungen.
20 CHS	–20	
f SOLVE 2	–1.6667	Erste Nullstelle.
STO 0	–1.6667	Speichert die Nullstelle zur Reduktion.
R↓ R↓	4.0000 –06	Funktionswert nahe an Null.

Schalten Sie in den Programm-Modus zurück und fügen Sie Anweisungen zur Elimination der soeben gefundenen Nullstelle in Ihr Unterprogramm ein.

Tastenfolge	Anzeige		
g P/R	000–		Programm-Modus.
g BST g BST	026–	30	Zeile vor RTN.
x⇄y	027–	34	Lädt x ins X-Register.
RCL 0	028–	45 0	Division durch (x–a), wo a bekannte Nullstelle.
–	029–	30	
÷	030–	10	

Berechnen Sie nun mit den gleichen Anfangsnäherungen die nächste Nullstelle.

Tastenfolge	Anzeige	
g P/R	4.0000 -06	Run-Modus.
10 CHS ENTER	-10.0000	Alte Anfangsnäherungen.
20 CHS	-20	
f SOLVE 2	0.4000	Zweite Nullstelle.
STO 1	0.4000	Speichert die Nullstelle zur Reduktion.
R↓ R↓	0.0000	Wert der reduzierten Funktion.

Modifizieren Sie jetzt Ihr Unterprogramm, um die zweite Nullstelle zu eliminieren.

Tastenfolge	Anzeige	
g P/R	000-	Programm-Modus.
g BST g BST	030- 10	Zeile vor RTN.
$x \gtrless y$	031- 34	Lädt x ins X-Register.
RCL 1	032- 45 1	Reduktion der zweiten Nullstelle.
−	033- 30	
÷	034- 10	

Berechnen Sie wieder mit den gleichen Anfangsnäherungen die nächste Nullstelle.

Tastenfolge	Anzeige	
g P/R	0.0000	Run-Modus.
10 CHS ENTER	-10.0000	Alte Anfangsnäherungen.
20 CHS	-20	
f SOLVE 2	8.4999	Dritte Nullstelle.
STO 2	8.4999	Speichert die Nullstelle zur Reduktion.
R↓ R↓	-1.0929 -07	Wert der reduzierten Funktion.

Erweitern Sie nun Ihr Unterprogramm, um die dritte Nullstelle zu eliminieren.

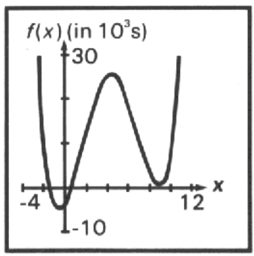
Tastenfolge	Anzeige	
g P/R	000-	Programm-Modus.
g BST g BST	034- 10	Zeile vor RTN.
$x \gtrless y$	035- 34	Lädt x ins X-Register.

Tastenfolge	Anzeige	
RCL 2	036 45 2	Reduktion der dritten Nullstelle.
−	037− 30	
÷	038− 10	

Berechnung der vierten Nullstelle.

Tastenfolge	Anzeige	
g P/R	−1.0929 −07	
10 CHS ENTER	−10.0000	Alte Anfangsnäherungen.
20 CHS	−20	
f SOLVE 2	8.5001	Vierte Nullstelle.
STO 3	8.5001	Speichert die Nullstelle zur Ref.
R↓ R↓	−0.0009	Wert der reduzierten Funktion.

Sie haben jetzt mit immer den gleichen Anfangswerten vier Nullstellen für dieses Polynom vierten Grades gefunden. Die beiden letzten Nullstellen liegen jedoch sehr nahe beisammen und sind in Wirklichkeit nur eine Nullstelle (zweiten Grades). Aus diesem Grund wurde diese Nullstelle nicht eliminiert, als Sie Reduktion an dieser Stelle anzuwenden versuchten. (Rundungsfehler sind für die kleinen positiven und negativen Funktionswerte für die x-Werte zwischen 8.4999 und 8,5001 verantwortlich; die Funktion hat für $x = 8.5$ genau den Wert Null.)



Graph von $f(x)$

Im allgemeinen kennen Sie die Ordnung der Nullstelle, die Sie eliminieren wollen, nicht im voraus. Falls nach dem Versuch, eine Nullstelle zu eliminieren, SOLVE die selbe Nullstelle nocheinmal berechnet, können Sie auf verschiedene Weise fortfahren:

- Benutzen Sie andere Anfangswerte für die reduzierte Funktion, und wenden Sie SOLVE wiederholt an.
- Reduzieren Sie das Polynom noch einmal, um die mehrfache Nullstelle zu eliminieren. Wenn Sie die Ordnung der Nullstelle nicht kennen, müssen Sie diesen Vorgang möglicherweise mehrmals wiederholen.

- Untersuchen Sie das Verhalten der reduzierten Funktion bei x-Werten nahe bei der bekannten Nullstelle. Wenn die berechneten Funktionswerte die x-Achse ohne großen Sprung schneiden, deutet das auf eine weitere Nullstelle oder auf eine Nullstelle höherer Ordnung hin.
- Analysieren Sie die Originalfunktion und ihre Ableitungen algebraisch. Sie können daraus möglicherweise deren Verhalten für x-Werte nahe der bekannten Nullstelle erkennen. (Eine Taylorentwicklung kann beispielsweise eine Nullstelle höherer Ordnung andeuten.)

Begrenzen der Laufzeit

Manchmal ist es sinnvoll, die Zeit zur Berechnung einer Nullstelle zu begrenzen. Sie können dies mit zwei Methoden erreichen – Zählen der Iterationen oder Vorgabe einer Genauigkeitsschranke.

Zählen von Iterationen

Bei der Suche nach einer Nullstelle berechnet SOLVE gewöhnlich weit mehr als zehn Werte der vorgegebenen Funktion. Gelegentlich wird das Funktions-Unterprogramm mehr als hundertmal aufgerufen. (SOLVE beendet seine Ausführung jedoch auf jeden Fall selbst.) Da Ihr Funktions-Unterprogramm für jede neue Näherung wieder durchlaufen wird, kann es die Anzahl durchgeführter Iterationen zählen und beschränken. Dies kann auf einfache Weise mit der Anweisung ISG erreicht werden, die die Iterationszahl im Indexregister (oder einem anderen Datenregister) festhält.

Wenn Sie in diesem Register vor dem Aufruf von SOLVE ein geeignetes Maximum abspeichern, kann Ihr Unterprogramm den SOLVE Algorithmus unterbrechen, wenn die erlaubte Iterationszahl überschritten wird.

Vorgabe einer Genauigkeitsschranke

Sie können die zur Nullstellenbestimmung benötigte Zeit verkürzen, wenn Sie eine maximal erlaubte Ungenauigkeit für Ihre Funktion vorgeben. Ihr Unterprogramm sollte mit dem Wert Null zurückkehren, falls der berechnete Funktionswert innerhalb der angegebenen Genauigkeitschranke liegt. Die vorgegebene Toleranz sollte für praktische Zwecke vernachlässigbar klein sein oder der Rechengenauigkeit angepasst sein. Mit dieser Technik sparen Sie die Zeit ein, die dazu verwendet wird, die Nullstelle genauer zu bestimmen, als durch die Aufgabe gerechtfertigt ist. (Das Beispiel auf Seite 224 arbeitet mit dieser Methode.)

Weitere Informationen

Im Handbuch *HP-15C – Fortgeschrittene Funktionen* finden Sie weitere, tiefergehende Techniken und Anwendungen zu SOLVE. Folgende Themen sind dort behandelt:

- Anwendung von SOLVE auf Polynome.
- Lösen von Gleichungssystemen.
- Berechnung lokaler Extrema einer Funktion.
- SOLVE bei finanzmathematischen Berechnungen.
- SOLVE im Komplex-Modus.
- Berechnung der komplexen Nullstellen einer Funktion.

Anhang E

$\boxed{\int_y^x}$ im Detail

Die grundlegenden Informationen zur Anwendung des $\boxed{\int_y^x}$ Algorithmus finden Sie in Abschnitt 14. Dieser Anhang enthält weitergehende Aspekte zu $\boxed{\int_y^x}$, die von Interesse sind, wenn Sie $\boxed{\int_y^x}$ häufig benutzen.

Arbeitsweise von $\boxed{\int_y^x}$

Der $\boxed{\int_y^x}$ Algorithmus berechnet das Integral einer Funktion $f(x)$, indem er einen gewichteten Mittelwert der Funktionswerte an ausreichend vielen Stützstellen von x innerhalb des Integrationsintervalls bildet. Die Genauigkeit des Ergebnisses eines derartigen Stützstellenalgorithmus hängt von der Anzahl der einbezogenen Stützstellen ab: ganz allgemein gilt, je mehr Stützstellen, desto größer die Genauigkeit. Wenn $f(x)$ an unendlich vielen Stellen berechnet werden könnte, würde der Algorithmus (unter Vernachlässigung der Ungenauigkeit der Berechnung der Funktionswerte) eine exakte Lösung liefern.

Die Auswertung der Funktionswerte an unendlich vielen Stellen würde unendlich lange dauern. Dies ist jedoch nicht erforderlich, da die Maximalgenauigkeit des Integrals ohnehin durch die Genauigkeit der berechneten Funktionswerte begrenzt ist. Mit einer begrenzten Anzahl von Stützstellen kann der Algorithmus ein Integral berechnen, dessen Wert so genau ist, wie es die $f(x)$ anhaftende Ungenauigkeit zuläßt.

Der $\boxed{\int_y^x}$ Algorithmus betrachtet zuerst nur einige wenige Stützstellen und liefert ein entsprechend ungenaues Ergebnis. Sind diese Approximationen noch nicht so exakt, wie die Genauigkeit von $f(x)$ zuläßt, wird der Algorithmus mit einer größeren Anzahl von Stützstellen wiederholt. Diese Iterationen werden fortgesetzt, jedesmal mit verdoppelter Stützstellenzahl, bis die resultierende Approximation die Maximalgenauigkeit erreicht hat, die die Unsicherheit von $f(x)$ zuläßt.

Die Ungenauigkeit der berechneten Approximation ist eine vom Anzeigeformat abgeleitete Zahl, die von der Ungenauigkeit der Funktionswerte bestimmt wird*. Am Ende jeder Iteration vergleicht der Algorithmus die erhaltene Approximation mit den zwei vorhergehenden Approximationen. Ist der Unterschied zwischen einer beliebigen dieser Approximationen und den andern beiden kleiner als die in der letzten Approximation tolerierte Ungenauigkeit, so wird der Algorithmus beendet; die letzte Approximation steht im X-Register und die zugehörige Fehlerabschätzung im Y-Register.

Es ist sehr unwahrscheinlich, daß die Fehler in drei aufeinanderfolgenden Approximationen – d.h. der Unterschied zwischen der Approximation und dem tatsächlichen Wert des Integrals – immer größer als der Unterschied zwischen den Approximationen selbst ist. Deshalb ist der Fehler der letzten Approximation kleiner als deren Fehlerabschätzung**. Obwohl der Fehler der letzten Approximation nicht bekannt ist, ist es sehr unwahrscheinlich, daß er die angezeigte Fehlerabschätzung der Approximation überschreitet. Mit anderen Worten, die Fehlerabschätzung im Y-Register ist ziemlich sicher die obere Grenze des Unterschiedes zwischen der Approximation und dem tatsächlichen Integral.

Genauigkeit, Fehlerabschätzung und Rechenzeit

Die Genauigkeit einer $\int_x^y$ Approximation ändert sich nicht immer, wenn Sie die Stellenzahl im Anzeigeformat um *nur eine* Stelle vergrößern, die Fehlerabschätzung wird jedoch verkleinert. Entsprechend kann sich die für eine Integration benötigte Rechenzeit mit einer Änderung des Anzeigeformats ebenfalls ändern, muß aber nicht.

Beispiel: Die Besselfunktion erster Art vierter Ordnung kann dargestellt werden als

$$J_4(x) = \frac{1}{\pi} \int_0^\pi \cos(4\theta - x \sin\theta)\, d\theta.$$

* Der Zusammenhang zwischen Anzeigeformat, Genauigkeit der Funktionswerte und Fehler der Integrationsnäherung wird später in diesem Abschnitt erläutert.

** Vorausgesetzt, daß $f(x)$ nicht zu rasch variiert; dies wird später in diesem Anhang genauer erläutert.

Berechnen Sie das Integral in dem Ausdruck für $J_4(1)$,

$$\int_0^{\pi} \cos(4\theta - \sin\theta)\, d\theta.$$

Schalten Sie zuerst in den Programm-Modus und geben Sie ein Unterprogramm zur Berechnung von $f(\theta) = \cos(4\theta - \sin\theta)$ ein.

Tastenfolge	Anzeige	
[g] [P/R]	000–	Programm-Modus.
[f] CLEAR [PRGM]	000–	
[f] [LBL] 0	001–42,21, 0	
4	002– 4	
[×]	003– 20	
[x⪌y]	004– 34	
[SIN]	005– 23	
[–]	006– 30	
[COS]	007– 24	
[g] [RTN]	008– 43 32	

Schalten Sie nun in den Run-Modus und geben Sie die Integrationsgrenzen in die Register X und Y ein. Setzen Sie den Trigonometrischen Modus auf Radiant und das Anzeigeformat auf [SCI] 2. Drücken Sie dann [f] [∫y x] 0, um das Integral zu berechnen.

Tastenfolge	Anzeige	
[g] [P/R]		Run-Modus.
0 [ENTER]	0.0000	Eingabe der unteren Integrationsgrenze ins Y-Register.
[g] [π]	3.1416	Eingabe der oberen Integrationsgrenze ins X-Register.
[g] [RAD]	3.1416	Umschalten auf Radiant.
[f] [SCI] 2	3.14 00	Anzeigeformat [SCI] 2.
[f] [∫y x] 0	7.79 –03	[SCI] 2 Approximation.
[x⪌y]	1.45 –03	Fehlerabschätzung der [SCI] 2 Approximation.

Die Fehlerabschätzung besagt, daß sämtliche angezeigten Stellen der Approximation als genau angesehen werden können. Tatsächlich ist diese Approximation sogar noch genauer, als die Fehlerabschätzung andeutet.

Tastenfolge	Anzeige		
[x≷y]	**7.79**	**–03**	Anzeige der Approximation.
[f] CLEAR [PREFIX] (gehalten)	**7785820888**		Alle 10 Stellen der [SCI] 2 Approximation.

Der tatsächliche Wert dieses Integrals, auf fünf Stellen genau, ist 7.7805×10^{-3}. Der Approximationsfehler beträgt daher $(7.7858–7.7805) \times 10^{-3} = 5.3 \times 10^{-6}$. Dieser Fehler ist beträchtlich kleiner als die Fehlerabschätzung von 1.45×10^{-3}. Die Fehlerabschätzung ist nur eine Obergrenze für den Approximationsfehler; der tatsächliche Fehler ist im allgemeinen kleiner.

Berechnen Sie nun das Integral mit [SCI] 3 und vergleichen Sie die damit erreichte Approximation mit dem [SCI] 2 Ergebnis.

Tastenfolge	Anzeige		
[f] [SCI] 3	**7.786**	**–03**	Ändert das Anzeigeformat in [SCI] 3.
[R↓] [R↓]	**3.142**	**00**	Verschiebt den Stack nach unten, bis die Obergrenze im X-Register erscheint.
[f] [∫] 0	**7.786**	**–03**	[SCI] 3 Approximation.
[x≷y]	**1.448**	**–04**	Fehlerabschätzung der [SCI] 3 Approximation.
[x≷y]	**7.786**	**–03**	Bringt die Approximation wieder zur Anzeige.
[f] CLEAR [PREFIX] (gehalten)	**7785820888**		Alle 10 Stellen der [SCI] 3 Approximation.

Die Approximationen in [SCI] 2 und [SCI] 3 sind auf allen 10 Stellen identisch: die Genauigkeit mit [SCI] 3 ist nicht besser als mit [SCI] 2, obwohl die Fehlerabschätzung in [SCI] 3 kleiner ist. Warum? Erinnern Sie sich, daß die Genauigkeit jeder Approximation vor allem von der Anzahl der Stützstellen, deren Funktionswerte benutzt werden, abhängt. Der [∫y] Algorithmus erhöht die Anzahl der Stützstellen, bis der Unterschied dreier aufeinanderfolgender Approximationen kleiner als die Ungenauigkeit aufgrund des Anzeigeformats ist. Nach einer bestimmten Iteration kann die Abweichung der Approximationen untereinander bereits so klein im Vergleich zur Ungenauigkeit der Funktionswerte sein, daß sie immer noch kleiner als diese Ungenauigkeit ist, selbst wenn diese um einen Faktor 10 vermindert wird. In einem solchen Fall muß der Algorithmus, nachdem Sie die Anzeigegenauigkeit um eine Stelle erhöht und damit die Ungenauigkeit des Funktionswertes verringert haben, keine weiteren Stützstellen mehr berechnen, und das Approximationsergebnis ist mit dem bei größerer Ungenauigkeit berechneten Ergebnis identisch.

Falls Sie die zwei vorhergehenden Berechnungen auf Ihrem Rechner nachvollzogen haben, werden Sie bemerkt haben, daß Sie zur Berechnung des Integrals mit [SCI] 3 nicht länger gebraucht haben als mit [SCI] 2. Die Rechenzeit hängt nämlich im wesentlichen nur von der Anzahl der Stützstellen ab, deren Funktionswerte berechnet werden müssen, um eine ausreichende Genauigkeit zu erzielen. Der Algorithmus hatte für die Integration mit [SCI] 3 nicht mehr Stützstellen auszuwerten als mit [SCI] 2, deshalb wurde dafür auch nicht mehr Zeit benötigt.

Oft wird jedoch bei erhöhter Anzeigegenauigkeit die Funktion an zusätzlichen Stützstellen berechnet werden müssen, und die Integration wird dann länger dauern. Berechnen Sie nun das gleiche Integral im [SCI] 4 Anzeigeformat.

Tastenfolge	Anzeige		
[f] [SCI] 4	**7.7858**	**–03**	Anzeigeformat [SCI] 4.
[R↓] [R↓]	**3.1416**	**00**	Verschiebt den Stack nach unten, bis die obere Integrationsgrenze im X-Register erscheint.
[f] [∫y] 0	**7.7807**	**–03**	[SCI] 4 Approximation.

Diese Approximation benötigte ungefähr doppelt soviel Zeit wie die Approximationen mit [SCI] 2 und [SCI] 3. In diesem Fall mußte der Algorithmus etwa doppelt soviele Stützstellen auswerten, um eine Approximation ausreichender Genauigkeit zu erhalten. Beachten Sie aber, daß Ihre Geduld belohnt wurde: die Genauigkeit dieses Ergebnisses ist um fast zwei Stellen besser als die Genauigkeit der Approximation mit halb sovielen Stützstellen.

Die vorhergehenden Beispiele zeigen, daß die Approximation in verschiedenen Anzeigeformaten manchmal ein genaueres Ergebnis liefert, manchmal aber auch nicht. Ob die Genauigkeit verbessert wird oder nicht, hängt von der betreffenden Funktion ab und kann im Normalfall nur durch Probieren herausgefunden werden.

Weiterhin müssen Sie ein genaueres Ergebnis mit doppelter Rechenzeit bezahlen. Diesen unvermeidbaren Zusammenhang zwischen Genauigkeit und Rechenzeit müssen Sie im Auge behalten, wenn Sie durch erhöhte Genauigkeit bei der Funktionsauswertung die Genauigkeit des Ergebnisses verbessern wollen.

Die Rechenzeit für das Integral einer gegebenen Funktion hängt nicht nur von der durch das Anzeigeformat gegebenen Anzahl gültiger Stellen, sondern auch in gewissem Maß von den Integrationsgrenzen ab. Wenn die Berechnung eines Integrals unvernünftig viel Zeit erfordert, ist möglicherweise die Integrationsbreite (die Differenz der Grenzen) zu groß in Bezug auf gewisse Eigenschaften der zu integrierenden Funktion. Bei den meisten Problemstellungen brauchen Sie sich aber um die Auswirkungen der Wahl der Integrationsgrenzen auf die Rechenzeit nicht zu kümmern. Wir kommen später in diesem Abschnitt auf dieses Problem und auf mögliche Lösungstechniken zurück.

Fehlerabschätzung und Anzeigeformat

Das Unterprogramm, das Sie zur Berechnung von $f(x)$ zur Verfügung stellen müssen, kann diesen Wert nicht genau bestimmen, sondern muß ihn auf die im Anzeigeformat gegebene Stellenzahl runden. Es berechnet also nicht $f(x)$, sondern

$$\hat{f}(x) = f(x) \pm \delta_1(x),$$

wobei $\delta_1(x)$ der Rundungsfehler ist.

Falls $f(x)$ ein physikalisches Problem beschreibt, ist die Funktion, die Sie integrieren wollen, nicht $f(x)$, sondern stattdessen

$$F(x)=f(x)\pm\delta_2(x),$$

wo $\delta_2(x)$ für die Ungenauigkeit steht, mit der $f(x)$ das tatsächliche physikalische Problem beschreibt.

Da $f(x)=\hat{f}(x)\pm\delta_1(x)$, ist die zu integrierende Funktion

$$F(x)=\hat{f}(x)\pm\delta_1(x)\pm\delta_2(x)$$

oder
$$F(x)=\hat{f}(x)\pm\delta(x),$$

wobei $\delta(x)$ den mit $\hat{f}(x)$ verknüpften Gesamtfehler beschreibt.

Das gesuchte Integral hat deshalb die Form

$$\int_a^b F(x)\,dx=\int_a^b [\hat{f}(x)\pm\delta(x)]\,dx$$

$$=\int_a^b \hat{f}(x)\,dx\pm\int_a^b \delta(x)\,dx$$

$$=I\pm\Delta$$

wobei I die Approximation des gesuchten Integrals und Δ den mit dieser Approximation verknüpften Fehler darstellt. Der $\int_y^x$ Algorithmus legt I im X-Register und eine Abschätzung für Δ im Y-Register ab.

Die Ungenauigkeit $\delta(x)$ von $\hat{f}(x)$, des von Ihrem Unterprogramm berechneten Funktionswertes, wird folgendermaßen bestimmt. Nehmen Sie an, die betrachtete Funktion sei auf drei Stellen genau. Setzen Sie deshalb das Anzeigeformat auf SCI 2. Die Anzeige enthält dann nur die genauen Stellen in der Mantisse des Funktionswertes: zum Beispiel: **1.23 -04**.

Da die Zahl im X-Register auf die durch das Anzeigeformat festgelegte Ziffernzahl gerundet wird, ist der Fehler der Funktionswerte in diesem Fall $\pm 0.005\times10^{-4}=\pm0.5\times10^{-2}\times10^{-4}=\pm0.5\times10^{-6}$.

Deshalb ist, wenn Sie das Anzeigeformat auf [SCI] n* oder [ENG] n (dabei muß n ganzzahlig sein) festsetzen, der Fehler des Funktionswertes

$$\delta(x) = 0.5 \times 10^{-n} \times 10^{m(x)}$$

$$= 0.5 \times 10^{-n+m(x)}$$

In dieser Formel ist n die im Anzeigeformat vorgegebene Stellenzahl, und $m(x)$ ist der Exponent des Funktionswertes im [SCI] Format.

Der Fehler ist proportional zum Faktor $10^{m(x)}$, der die Größe des Funktionswertes an der Stelle x darstellt. In den Formaten [SCI] und [ENG] ist der Fehler des Funktionswertes von der Größe des Funktionswertes abhängig und deshalb *relativ*.

Wird andererseits ein Funktionswert mit [FIX] n angezeigt, bedingt der Rundungsfehler der Anzeige einen Fehler des Funktionswertes von

$$\delta(x) = 0.5 \times 10^{-n}.$$

Da dieser Fehler nicht vom Funktionswert abhängt, erhalten Sie im [FIX] 4 Format einen *absoluten* Fehler.

Bei jeder Berechnung eines Funktionswertes an einer Stelle x bestimmt der [∫] Algorithmus auch den Wert $\delta(x)$, den Fehler des Funktionswertes von x. Dieser wird mit Hilfe der Stellenzahl des Anzeigeformats und (falls das Anzeigeformat ein [SCI] oder [ENG] Format ist) der Größe $m(x)$ an der Stelle x berechnet.

---

* Obwohl Sie mit [SCI] 8 oder 9 im allgemeinen die *gleiche Anzeige* erhalten wie mit [SCI] 7, ist der resultierende Fehler *dennoch kleiner*. (Das gleiche gilt für das Format [ENG].) Ein negativer Wert für n (der mit Hilfe des Indexregisters gesetzt werden kann) beeinflußt den Fehler einer Berechnung mit [∫]. Der kleinste Wert von n, der auf den Fehler noch einen Einfluß hat, ist −6. Eine Zahl kleiner als −6 in R_I wird als −6 interpretiert.

Die Zahl Δ, der Approximationsfehler, ist das Integral über $\delta(x)$:

$$\Delta = \int_a^b \delta(x)\,dx$$

$$= \int_a^b [0.5 \times 10^{-n+m(x)}]\,dx.$$

Zur Berechnung dieses Integrals wird $f(x)$ in der gleichen Weise an den Stützstellen ausgewertet, wie dies bei der Berechnung der Integralapproximation mittels $f(x)$ erfolgt.

Da Δ proportional zum Faktor 10^{-n} ist, ändert sich der Approximationsfehler um ungefähr den Faktor 10 für jede zusätzliche Stelle in der Anzeige. Dieser Faktor wird im [SCI] oder [ENG] Format im allgemeinen jedoch nicht genau 10 sein, da nach einer Änderung der vorgegebenen Stellenzahl andere Stützstellen ausgewertet werden, und $\delta(x) \sim 10^{m(x)}$ andere Werte annimmt.

Beachten Sie, daß bei der Auswertung eines Integrals im [FIX] Format $m(x) = 0$ gilt; der berechnete Fehler ist dann

$$\Delta = 0.5 \times 10^{-n}(b-a).$$

Normalerweise benötigen Sie keinen *genauen* Wert für den Fehler in der Funktion (das würde oft eine sehr komplizierte Untersuchung erfordern). Gewöhnlich werden Sie ein [SCI] oder [ENG] Anzeigeformat bevorzugen, wenn der Fehler leichter im Sinne eines *relativen* Fehlers zu schätzen ist. Kann dagegen der Fehler besser als *absoluter* Wert geschätzt werden, so ist ein [FIX] Format vorzuziehen. Besitzt die zu integrierende Funktion auf dem Integrationsbereich extrem kleine Funktionswerte *und* die Fehlertoleranz der Funktionswerte ist gleichfalls sehr klein, sollten Sie das [FIX] Format vermeiden, da Sie möglicherweise sehr absonderliche Ergebnisse erhalten können. Das gleiche trifft für [SCI] Anzeigeformate zu, wenn die Größe von Funktionswerten viel kleiner als die gegebene Fehlertoleranz wird. Erhalten Sie bei der Berechnung eines Integrals seltsam anmutende Ergebnisse, sollten Sie die Berechnung in einem anderen Anzeigeformat wiederholen.

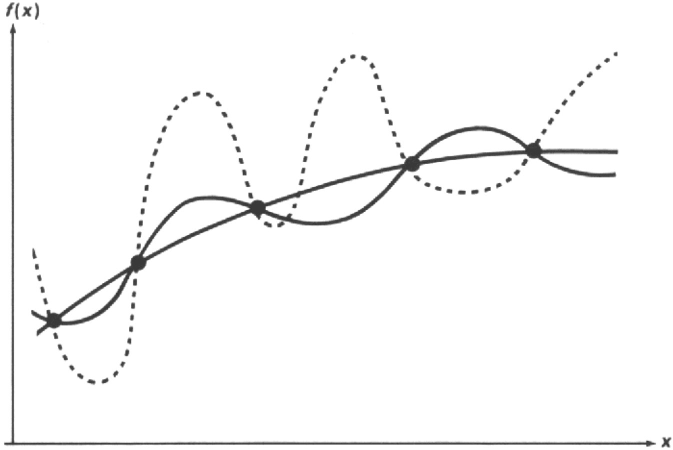
Mögliche Ursachen für fehlerhafte Ergebnisse

Obwohl der ∫ Algorithmus des HP-15C einer der besten verfügbaren Algorithmen ist, kann er Ihnen in bestimmten Situationen – wie fast alle Algorithmen für numerische Integration – ein falsches Ergebnis liefern. *Die Wahrscheinlichkeit dafür ist allerdings sehr gering.*

Dieser Algorithmus ist so ausgelegt, daß er für praktisch alle glatt verlaufenden Funktionen zuverlässige Ergebnisse liefert. Nur bei extrem sprunghaft verlaufenden Funktionen gehen Sie ein gewisses Risiko ein, ein ungenaues Ergebnis zu erhalten. Solche Funktionen kommen in physikalischen Problemstellungen jedoch kaum vor und können gegebenenfalls leicht erkannt und bearbeitet werden.

Wie auf Seite 240 erwähnt, berechnet der Algorithmus ∫ den Funktionswert $f(x)$ für verschiedene Stützstellen x innerhalb des Integrationsintervalls. Der Approximation des Integrals von $f(x)$ wird dann als gewichtetes Mittel der Funktionswerte der Stützstellen berechnet.

Da der Algorithmus außer den Funktionswerten in den Stützstellen keine weitere Information über $f(x)$ besitzt, kann er $f(x)$ nicht von anderen Funktionen unterscheiden, die mit $f(x)$ an allen Stützstellen übereinstimmen. Die Illustration auf der folgenden Seite verdeutlicht diese Situation anhand dreier der unendlich vielen möglichen Funktionen, deren Graphen auf einem Abschnitt des Integrationsintervalls an endlich vielen Stützstellen übereinstimmen.



Mit diesen Stützstellen berechnet der Algorithmus gleiche Approximationen für das Integral aller gezeichneten Funktionen. Die tatsächlichen Integrale der beiden mit durchgezogenen Linien gezeichneten Funktionen sind ungefähr gleich, die Approximation wird also ziemlich genau sein, wenn $f(x)$ eine dieser beiden Funktionen ist. Das Integral der punktiert gezeichneten Funktion besitzt jedoch einen von den andern beiden Integralen ziemlich verschiedenen Wert, und Sie werden für diese Funktion eine ungenaue Approximation des Integrals erhalten.

Der [∫] Algorithmus untersucht den Kurvenverlauf der Funktion durch Funktionsauswertungen an immer enger beisammen liegenden Stützstellen. Beschränken sich die Fluktuationen der Funktion nicht auf einen engen Bereich des Integrationsintervalls, werden diese Fluktuationen sehr wahrscheinlich in einem entsprechenden Iterationsdurchgang entdeckt. Danach wird die Anzahl der Stützstellen erhöht, bis nachfolgende Iterationen Approximationen liefern, die den Verlauf der stärksten *charakteristischen* Fluktuationen berücksichtigen.

Betrachten Sie zum Beispiel die Approximation für

$$\int_0^\infty xe^{-x}dx.$$

Da Sie dieses Integral numerisch auswerten, ist es naheliegend (aber dennoch irreführend, wie Sie sehen werden), die obere Integrationsgrenze als 10^{99} zu wählen – die größtmögliche Zahl, die Sie in den Rechner eingeben können. Versuchen Sie es und sehen Sie, was passiert.

Geben Sie ein Unterprogramm zur Auswertung der Funktionswerte $f(x) = xe^{-x}$ ein.

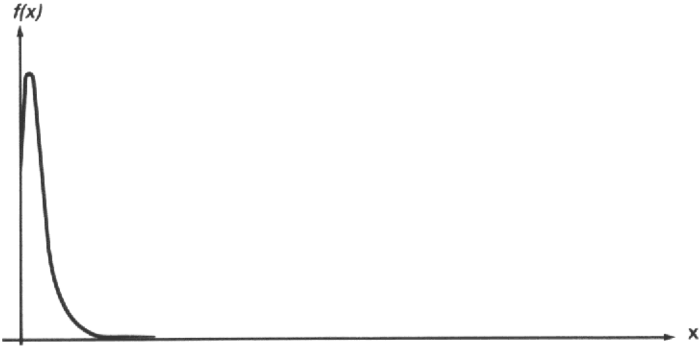
Tastenfolge	Anzeige	
[g] [P/R]	000–	Programm-Modus.
[f] [LBL] 1	001–42,21, 1	
[CHS]	002– 16	
[e^x]	003– 12	
[×]	004– 20	
[g] [RTN]	005– 43 32	

Schalten Sie den Rechner in den Run-Modus. Wählen Sie das Anzeigeformat [SCI] 3 und geben Sie die Integrationsgrenzen in die Register X und Y ein.

Tastenfolge	Anzeige	
[g] [P/R]		Run-Modus.
[f] [SCI] 3		Wählt Anzeigeformat [SCI] 3.
0 [ENTER]	0.000 00	Eingabe der unteren Integrationsgrenze ins Y-Register.
[EEX] 99	1 99	Eingabe der oberen Integrationsgrenze ins X-Register.
[f] [∫] 1	0.000 00	Approximation des Integrals.

Die vom Rechner gegebene Lösung ist mit Sicherheit unrichtig; das Integral von $f(x) = xe^{-x}$ von 0 bis ∞ hat exakt den Wert 1. Das Problem ist aber *nicht*, daß Sie ∞ durch 10^{99} dargestellt haben, da das Integral dieser Funktion von 0 bis 10^{99} beinahe den Wert 1 hat.

Der Grund für dieses falsche Ergebnis wird offensichtlich, wenn Sie den Graphen von $f(x)$ auf dem Integrationsintervall betrachten:

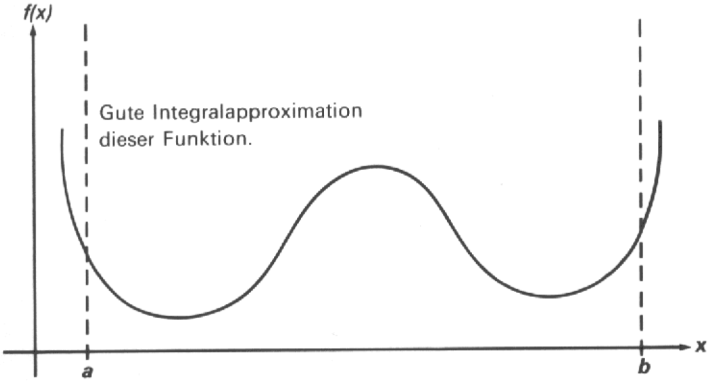


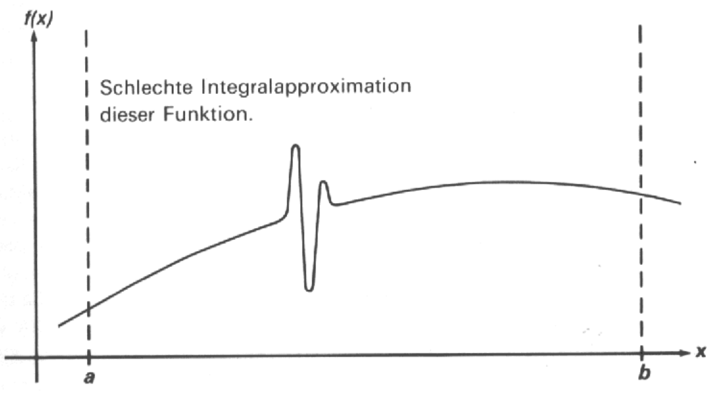
Der Graph hat einen Peak nahe des Ursprungs. (Um $f(x)$ darzustellen, wurde die Breite des Peaks beträchtlich vergrößert. In maßstabsgetreuer Darstellung auf dem Integrationsintervall wäre der Peak nicht von der vertikalen Achse des Graphen unterscheidbar.) Da keine Stützstelle den Peak des Graphen entdeckte, nahm der Algorithmus an, daß die Funktion auf dem ganzen Integrationsintervall identisch Null sei. Sogar bei erhöhter Anzahl von Stützstellen im Modus [SCI] 9 würde für diese spezielle Funktion und dieses spezielle Integrationsintervall keine der zusätzlichen Stützstellen den Peak entdecken. (Bessere Methoden zur Bewältigung derartiger Probleme werden am Ende des nächsten Unterabschnitts «Erhöhte Rechenzeiten» erwähnt.)

Sie haben gesehen, wie eine für das Gesamtverhalten einer Funktion uncharakteristische Fluktuation zu falschen Ergebnissen des [∫] Algorithmus führen kann. Glücklicherweise treten derart pathologische Funktionen selten genug auf, daß Sie wahrscheinlich nie in die Verlegenheit kommen werden, eine davon unwissentlich integrieren zu müssen.

Funktionen, die zu unrichtigen Ergebnissen führen könnten, werden durch die rasche Variation ihrer Werte und der Werte ihrer ersten Ableitungen charakterisiert. Grundsätzlich gilt, je stärker die Variation der Werte der Funktion oder ihrer Ableitungen, und je niederer die Ordnung der derartig variierenden Ableitung, desto länger braucht der Algorithmus zur Berechnung des Integrals, und desto ungenauer ist möglicherweise die resultierende Approximation.

Beachten Sie, daß die Stärke der Variation der Funktion (oder ihrer ersten Ableitungen) im Verhältnis zur Breite des Integrationsintervalls beurteilt werden muß. Bei einer gegebenen Anzahl von Stützstellen kann eine Funktion mit drei Fluktuationsstellen durch ihre Stützstellenwerte besser dargestellt werden, wenn diese über das ganze Intervall verteilt sind, als wenn sie auf einem kleinen Teil des Intervalls konzentriert sind. (Diese beiden Situationen werden in den folgenden zwei Illustrationen veranschaulicht.) Wenn Sie die Variationen oder Fluktuationen als Oszillationen der Funktion interpretieren, ist das interessierende Kriterium das Verhältnis der Schwingungsperiode zur Gesamtlänge des Integrationsintervalls: je größer dieses Verhältnis, desto schneller findet der Algorithmus ein Ergebnis, und desto zuverlässiger wird dieses sein.





In vielen Fällen werden Sie die zu integrierende Funktion gut genug kennen, um zu wissen, ob die Funktion irgendwo große (im Vergleich zum Integrationsintervall) Schwankungen aufweist. Wenn Sie die Funktion nicht kennen, und Sie Grund zur Annahme haben, daß sie Schwierigkeiten machen könnte, können Sie mit dem zur Berechnung der Funktionswerte geschriebenen Unterprogramm schnell ein paar Punkte berechnen und mit diesen eine Skizze des Funktionsverlaufs anfertigen.

Wenn Sie aus irgendeinem Grund die Gültigkeit eines erhaltenen Näherungswertes Ihrer Integration anzweifeln müssen, können Sie mit einer sehr einfachen Methode Ihren Verdacht überprüfen: Teilen Sie Ihr Integrationsintervall in zwei oder mehr Abschnitte, und addieren Sie die Näherungswerte der Integrale über diese Abschnitte. Die Funktion wird in diesem Fall an vollständig neuen Stützstellen berechnet, und findet deshalb zuvor verborgene Peaks mit größerer Wahrscheinlichkeit. Falls die anfängliche Approximation richtig ist, ist sie mit der Summe der Approximationen über den Teilabschnitten identisch.

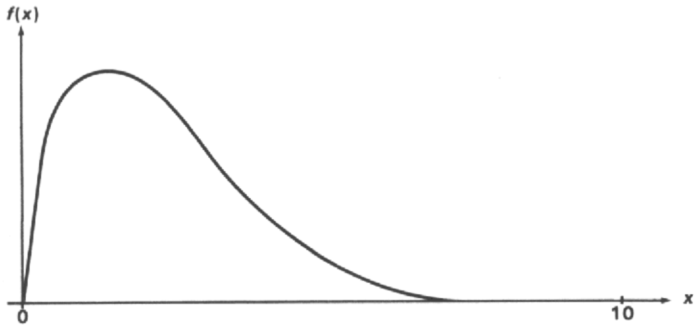
Erhöhte Rechenzeiten

Im vorhergehenden Beispiel (S. 251) lieferte der Algorithmus ein falsches Resultat, weil er den Peak der Funktion gar nicht entdeckte. Dies war deshalb der Fall, weil die Variation der Funktion im Vergleich zur Intervallbreite zu schnell war. Bei kleinerer Intervallbreite könnten Sie die richtige Lösung erhalten, aber das würde sehr lange dauern, falls die Intervallbreite noch immer zu groß ist.

Bei gewissen Funktionen, wie unserer Beispielfunktion, kann die Integration unverhältnismäßig lange dauern, wenn die Breite des Integrationsintervalls zu groß im Vergleich zu gewissen Eigenschaften der zu integrierenden Funktion ist. Betrachten Sie ein Integral, dessen Integrationsintervall groß genug ist, um unverhältnismäßig viel Rechenzeit in Anspruch zu nehmen, aber nicht so groß, daß Sie ein falsches Ergebnis erhalten. Beachten Sie, daß, im Fall $f(x) = xe^{-x}$, $f(x)$ sehr schnell gegen Null abfällt, wenn x gegen ∞ strebt, und der Beitrag großer x-Werte zum Integral vernachlässigbar wird. Deshalb können Sie die obere Integrationsgrenze ∞ durch einen Wert kleiner als 10^{99} ersetzen, zum Beispiel 10^3.

Tastenfolge	Anzeige		
0 [ENTER]	0.000	00	Eingabe der Untergrenze ins Y-Register.
[EEX] 3	1	03	Eingabe der Obergrenze ins X-Register.
[f] [∫] 1	1.000	00	Näherungswert des Integrals.
[x⇄y]	1.824	–04	Fehlerabschätzung der Approximation.

Dies ist das richtige Ergebnis, aber die Berechnung dauerte sehr lange. Um dies zu verstehen, vergleichen Sie den Graphen der Funktion auf dem Integrationsintervall, der dem auf Seite 252 sehr ähnelt, mit dem Graphen der Funktion auf dem Intervall von $x = 0$ bis $x = 10$.



Wenn Sie diese beiden Graphen vergleichen, stellen Sie fest, daß die Funktion nur für sehr kleine Werte von x «interessant» ist. Für größere Werte von x ist die Funktion «uninteressant», da sie sich stetig und glatt an die x-Achse anschmiegt.

Wie zuvor besprochen wurde, erhöht der [∫] Algorithmus die Dichte der Stützpunkte, bis der Unterschied zwischen aufeinanderfolgenden Approximationen genügend klein wird. Mit anderen Worten, der Algorithmus untersucht die Funktion an zusätzlichen Stellen solange, bis er soviel Information über die Funktion besitzt, daß das Einbeziehen weiterer Information (Stützpunkte) den Näherungswert nicht mehr wesentlich ändert.

Auf dem Integrationsintervall (0, 10), wo der Algorithmus die Funktion nur auf ihrem interessanten (aber dennoch glatten) Teil untersuchen muß, würden nach den ersten wenigen Iterationen weitere Stützstellen keine zusätzliche Information über den Verlauf der Funktion mehr liefern können. Deshalb sind auf diesem Intervall nur wenige Iterationen notwendig, bis aufeinanderfolgende Approximationen sich genügend wenig voneinander unterscheiden und der Algorithmus die Berechnung mit einer Approximation der gewünschten Genauigkeit beenden kann.

Ist andererseits das Integrationsintervall mehr von der Art des Graphen auf Seite 252, würden die meisten Stützstellen aus einem Abschnitt des Intervalls genommen, wo sich die Steigung des Graphen nicht sehr ändert. Die wenigen Stützstellen mit kleinen x-Werten würden von einer Iteration zur nächsten bemerkenswerte Änderungen des Funktionsverlaufs entdecken. Deshalb müßte die Funktion an weiteren Stützstellen berechnet werden, bevor der Unterschied zwischen aufeinanderfolgenden Approximationswerten genügend klein wird.

Um eine Approximation des Integrals mit gleicher Genauigkeit auf dem großen Intervall wie auf dem kleineren Intervall zu erhalten, muß die Dichte der Stützstellen auf dem interessanten Gebiet in beiden Fällen gleich sein. Bei gleicher Stützstellendichte müssen aber auf dem größeren Intervall viel mehr Stützstellen ausgewertet werden als auf dem kleineren Intervall. Deshalb sind auf dem größeren Intervall zusätzliche Iterationen erforderlich, um eine Approximation gleicher Genauigkeit zu erreichen, und deshalb wird zur Berechnung dieses Integrals beträchtlich mehr Zeit benötigt.

Da die Rechenzeit davon abhängt, wie schnell im interessanten Gebiet der Funktion eine bestimmte Stützstellendichte erreicht wird, erhöht sich die Zeit zur Berechnung jedes Integrals, wenn das Integrationsintervall sich über überwiegend uninteressante Teile der Funktion erstreckt. Glücklicherweise können Sie, wenn Sie ein solches Integral berechnen müssen, das Problem so umformulieren, daß die Rechenzeit beträchtlich verkürzt wird. Zwei Möglichkeiten dazu sind Unterteilung des Intervalls und Variablentransformation. Mit diesen Methoden können Sie die zu integrierende Funktion oder die Integrationsgrenzen so verändern, daß die Funktion auf dem/den Integrationsintervall/en leichter integrierbar wird. (Diese Methoden werden im Handbuch «*HP-15C – Fortgeschrittene Funktionen*» beschrieben.)

Anzeige der momentanen Approximation eines Integrals

Wenn die Berechnung eines Integrals länger dauert, als Sie warten wollen, können Sie die Ausführung anhalten und den derzeitigen Approximationswert anzeigen. Die zugehörige Fehlerabschätzung können Sie in diesem Fall allerdings nicht erhalten.

Mit der Taste [R/S] können Sie die Ausführung des [∫] Algorithmus genauso anhalten wie die Ausführung eines laufenden Programmes. Wenn Sie dies tun, stoppt der Rechner bei der derzeitigen Programmzeile Ihres Funktions-Unterprogramms und zeigt das Ergebnis der Ausführung der vorhergegangenen Programmanweisung an. Beachten Sie, daß die gewünschte Approximation beim Unterbrechen der Berechnung *nicht* die Zahl im X-Register und auch keine andere Zahl im Stack ist. Wie bei jedem anderen Programm startet ein weiteres Drücken von [R/S] die Berechnung wieder von der Programmzeile aus, an der sie unterbrochen wurde.

Nach jeder Funktionsauswertung an einer neuen Stützstelle berechnet der [∫] Algorithmus eine neue Approximation des Integrals und legt diesen Wert im LAST X-Register ab. Um die derzeitige Approximation zu erhalten, müssen Sie deshalb einfach die Ausführung stoppen und gegebenenfalls die Ausführung des Unterprogramms in Einzelschritten ([SST]) weiterführen, bis der Rechner den Funktionswert berechnet und die Approximation aktualisiert hat. Rufen Sie dann den neuen Inhalt des LAST X-Registers ab, der verfügbar ist, sobald die [RTN] Anweisung des Funktions-Unterprogrammes ausgeführt wurde.

Während der Rechner die derzeitige Approximation aktualisiert, ist die Anzeige leer und zeigt nicht **running**. (Bei der Ausführung des Funktions-Unterprogrammes wird **running** angezeigt.) Deshalb können Sie die zeilenweise Ausführung des Unterprogramms vermeiden, wenn Sie den Rechner anhalten, während die Anzeige leer ist.

Zusammengefaßt führen Sie folgende Schritte aus, um die momentane Approximation eines Integrals zu erhalten.

1. Halten Sie mit [R/S] den Rechner an, vorzugsweise bei leerer Anzeige.
2. Wenn der Rechner anhält, schalten Sie auf Programm-Modus, um die derzeitige Programmzeile festzustellen.
   - Wenn diese Zeile das Unterprogramm-Label enthält, schalten Sie in den Run-Modus zurück und rufen Sie den Inhalt des LAST X- Registers ab (Schritt 3).

- In allen anderen Fällen schalten Sie in den Run-Modus zurück und arbeiten Sie das Programm in Einzelschritten ([SST]) ab, bis Sie auf die [RTN] Anweisung (Tastencode 43 32) oder auf Zeile 000 (falls kein [RTN] eingegeben wurde) stoßen. (Halten Sie die [SST] Taste lange genug gedrückt, um die Zeilennummern und Tastencodes erkennen zu können.)

3. Drücken Sie [g] [LSTx], um die momentane Approximation abzurufen. Wenn Sie die Ausführung des Algorithmus wieder fortsetzen wollen, drücken Sie [←] [+] [R/S]. Dies füllt den Stack mit dem momentanen x-Wert und startet den Rechner wieder.

Weitere Informationen

Das Handbuch «*HP-15C – Fortgeschrittene Funktion*» behandelt mehr esoterische Gesichtspunkte des [∫] Algorithmus und seiner Anwendungen. Diese Themen sind unter anderem:

- Genauigkeit der zu integrierenden Funktion.
- Verkürzen der Rechenzeit.
- Berechnung schwieriger Integrale.
- [∫] im Komplex-Modus.

Anhang F

Batterien, Gewährleistung und Serviceinformation

Batterien

Die Stromversorgung des HP-15C erfolgt über einen Satz von drei Batterien. Bei «normalem» Gebrauch des Rechners wird mit einem Satz Alkalibatterien eine ungefähre Betriebsdauer von 3 Monaten erreicht. Der HP-15C wird mit Alkalibatterien ausgeliefert; Sie können jedoch auch Silberoxidbatterien (die die Betriebsdauer zumindest verdoppeln sollten) verwenden.

Drei frische Alkalibatterien liefern eine reine Programmrechenzeit von mindestens 60 Stunden (hierbei sei angemerkt, daß die Ausführung von Programmen die Form des Rechnerbetriebs mit dem höchsten Stromverbrauch ist)*. Bei der Verwendung von Silberoxidbatterien erhöht sich die reine Programmrechenzeit auf mindestens 135 Stunden. Diese Angaben beziehen sich, wie bereits erwähnt, auf die aufwendigste Betriebsart, nämlich die Ausführung von Programmen; bei allen anderen Betriebsarten ist der Stromverbrauch weitaus geringer. Wenn nur die Anzeige eingeschaltet ist, d.h. es laufen keine Programme ab und es werden keine Tasten gedrückt, ist der Stromverbrauch minimal.

Bei ausgeschaltetem Rechner bleibt der Inhalt des Permanentspeichers über einen Zeitraum erhalten, der dem der Haltbarkeit der Batterien außerhalb des Rechners entspricht – mindestens anderthalb Jahre bei Alkalibatterien bzw. mindestens zwei Jahre bei Silberoxidbatterien.

Die aktuelle Lebenszeit der Batterien hängt davon ab, wie oft Sie den Rechner benutzen, ob Sie ihn mehr zur Ausführung von Programmen oder für manuelle Berechnungen verwenden, und mit welchen Funktionen Sie arbeiten*.

Sowohl die zusammen mit dem Rechner gelieferten Batterien als auch die im folgenden empfohlenen Ersatzbatterien können nicht nachgeladen werden.

* Der Stromverbrauch des HP-15C wird wesentlich durch die jeweilige Betriebsart bestimmt: ausgeschaltet (bei Erhaltung des Permanentspeichers); betriebsbereit (nur die Anzeige ist eingeschaltet); rechnend (ein Programm läuft ab, eine Berechnung wird durchgeführt oder eine Taste ist gedrückt). Solange der Rechner eingeschaltet ist, besteht die typische Auslastung aus einer Mischung der beiden Betriebsarten «betriebsbereit» und «rechnend». Die aktuelle Lebensdauer der Batterien hängt daher sehr stark davon ab, wie lange der Rechner in jeder der drei Betriebsarten benutzt wird.

WARNUNG

Versuchen Sie nicht, die Batterien nachzuladen, werfen Sie sie nicht in offenes Feuer und lagern Sie sie nicht in der Nähe großer Wärmeeinstrahlung. Die Batterien können in diesen Fällen auslaufen bzw. explodieren.

Die folgenden Batterien werden als Ersatzbatterien für Ihren HP-15C empfohlen (nicht alle Batterientypen sind in allen Ländern erhältlich):

Alkalibatterien	Silberoxidbatterien
Eveready A76*	Eveready 357*
UCAR A76	UCAR 357
RAY-O-VAC RW82	RAY-O-VAC RS76 oder RW42
National oder Panasonic LR44	Duracell MS76 oder 10L14
Varta 4276	Varta 541

Anzeige abfallender Batteriespannung

Bei nachlassender Batteriespannung leuchtet in der linken unteren Ecke der Anzeige ein Stern (*) auf.

Bei Verwendung von Alkalibatterien:

- Der Rechner kann nach dem ersten Aufleuchten des Sterns noch mindestens zwei Stunden zur reinen Programmausführung verwendet werden**.
- Wenn der Rechner nach dem ersten Aufleuchten des Sterns ausgeschaltet wird, bleibt der Inhalt des Permanentspeichers noch wenigstens für einen Monat erhalten.

Bei Verwendung von Silberoxidbatterien:

- Der Rechner kann nach dem ersten Aufleuchten des Sterns noch mindestens 15 Minuten zur reinen Programmausführung verwendet werden**.
- Wenn der Rechner nach dem ersten Aufleuchten des Sterns ausgeschaltet wird, bleibt der Inhalt des Permanentspeichers noch mindestens für eine Woche erhalten.

---

* Nicht erhältlich in Großbritannien und der Republik Irland.

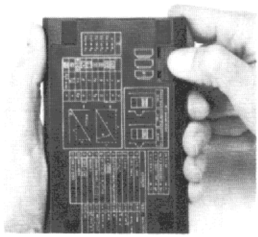
** Die angegebene Zeit bezieht sich auf den Fall, daß *laufend* Programme ausgeführt werden, d.h. auf die Betriebsart «rechnend» (siehe Fußnote auf Seite 259). Wird der Rechner für manuelle Berechnungen benutzt – eine Mischung der Betriebsarten «betriebsbereit» und «rechnend» –, so kann der Rechner noch wesentlich länger nach dem ersten Auftauchen des Sterns verwendet werden.

Einsetzen neuer Batterien

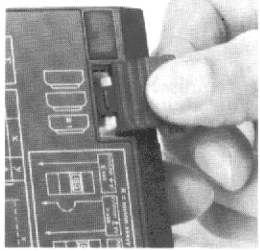
Der Inhalt des Permanentspeichers bleibt nach dem Ausbau der alten Batterien noch für einige Sekunden erhalten (sofern Sie den Rechner vor dem Entfernen der Batterien ausgeschaltet haben). Sie haben daher normalerweise genügend Zeit zum Austausch der Batterien zur Verfügung, ohne dabei Daten oder Programme zu verlieren. Der Inhalt des Permanentspeichers geht erst verloren, wenn über längere Zeit keine Batterien im Rechner sind.

Um neue Batterien einzubauen, verfahren Sie wie folgt:

1. Vergewissern Sie sich, daß der Rechner ausgeschaltet ist.
2. Halten Sie den Rechner wie in der Abbildung gezeigt und drücken Sie den Batteriefachdeckel nach außen, bis er sich leicht öffnet.



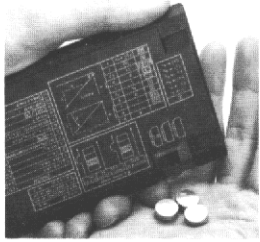
3. Greifen Sie den Batteriefachdeckel an der Außenkante und ziehen Sie ihn vom Rechner ab.



HINWEIS

Achten Sie während der nächsten beiden Schritte darauf, daß keine Tasten gedrückt werden, solange sich keine Batterien im Rechner befinden. In diesem Fall kann der Inhalt des Permanentspeichers gelöscht werden oder aber die Tastenfeldkontrolle verloren gehen (d.h. der Rechner reagiert nicht mehr auf das Drücken von Tasten).

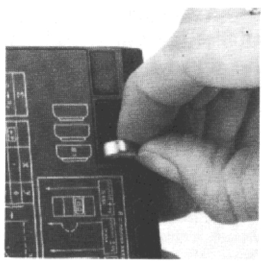
4. Drehen Sie den Rechner herum und schütteln Sie ihn leicht, bis die Batterien in Ihre Hand fallen.



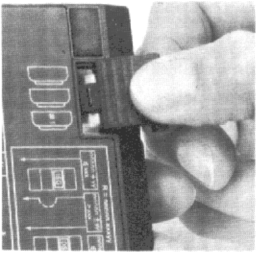
VORSICHT

Erneuern Sie im nächsten Schritt *alle drei* Batterien. Wenn Sie nur eine oder zwei Batterien austauschen, besteht die Gefahr, daß eine der alten Batterien ausläuft. Achten Sie des weiteren darauf, daß Sie die Batterien nicht verkehrt herum einsetzen. In diesem Fall kann der Inhalt des Permanentspeichers verloren gehen oder die betreffende Batterie beschädigt werden.

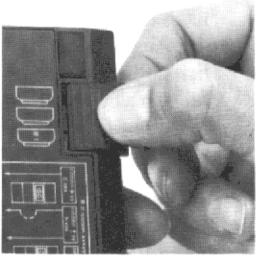
5. Halten Sie die beiden Plastikstreifen im Batteriefach nach oben und setzen Sie drei neue Batterien ein. Die Batterien müssen so eingelegt werden, daß die flachen (mit + markierten) Seiten zum nächsten Gummifuß zeigen. (Siehe Illustration auf dem Rechnergehäuse.)



6. Schieben Sie die Halterung des Batteriefachdeckels in die Aussparung im Rechnergehäuse.



7. Drücken Sie den Batteriefachdeckel nach unten, bis er mit dem Rechnergehäuse abschließt, und pressen Sie ihn anschließend nach innen, bis er leicht einrastet.
8. Schalten Sie den Rechner ein. Wenn der Permanentspeicher aus irgendeinem Grund gelöscht wurde, erscheint in der Anzeige die Meldung **Pr Error**. Sie können diese Meldung dann durch Drücken einer beliebigen Taste aus der Anzeige löschen.



Funktionsprüfung

Wenn sich Ihr HP-15C nicht einschalten läßt oder anderweitig nicht korrekt zu arbeiten scheint, sollten Sie den Rechner mit Hilfe der folgenden Schritte austesten:

Falls der Rechner *nicht* auf Tastendruck reagiert:

1. Drücken Sie gleichzeitig die Tasten [y^x] und [ON] und lassen Sie sie wieder los. Da bei dieser Operation der Inhalt des X-Registers verändert wird, sollten Sie dieses Register nach der Funktionsprüfung löschen.
2. Wenn der Rechner danach immer noch nicht auf das Drücken von Tasten reagiert, nehmen sie die Batterien heraus und setzen sie wieder ein. Achten Sie darauf, daß die Batterien korrekt in das Batteriefach eingesetzt sind.
3. Falls der Rechner auch weiterhin nicht auf Tastendruck reagiert, belassen Sie die Batterien im Batteriefach und schließen Sie die Kontakte im Batteriefach kurz. (Biegen Sie die Plastikstreifen zurück, um die Kontakte an den Seiten des Batteriefachs freizulegen.) *Sie brauchen die Kontakte nur für einen Moment kurzzuschließen.* Bei dieser Operation geht der Inhalt des Permanentspeichers verloren und Sie müssen zum Wiedereinschalten des Rechners die [ON] Taste unter Umständen mehrmals drücken.

4. Setzen Sie frische Batterien ein, wenn sich der Rechner jetzt immer noch nicht einschalten läßt. Wenn der Rechner auch danach nicht auf Tastendruck reagiert, ist er defekt.

Falls der Rechner auf Tastendruck reagiert:

1. Halten Sie bei ausgeschaltetem Rechner die Taste [ON] gedrückt und drücken Sie zusätzlich die Taste [×].
2. Lassen Sie zunächst die [ON] Taste und danach die [×] Taste los. Diese Tastenfolge löst eine vollständige Überprüfung der elektronischen Schaltkreise des Rechners aus. Bei korrekter Arbeitsweise des Rechners sollte nach einer Zeitspanne von 15 Sekunden (in der die Meldung **running** in der Anzeige blinkt) die Anzeige **–8,8,8,8,8,8,8,8,8,8** erscheinen. Zusätzlich sollten sämtliche Statusanzeigen (mit Ausnahme der Spannungsabfall-Anzeige*) eingeschaltet sein*. Wenn die Anzeige leer bleibt, die Meldung **Error 9** oder sonst in irgendeiner Form nicht das gewünschte Ergebnis zeigt, deutet dies auf einen Gerätefehler hin**.

Hinweis: Überprüfungen der Rechnerelektronik werden auch durch die Tastenkombinationen [ON]/[+] und [ON]/[÷] ausgelöst***. Diese Tests dienen der Funktionskontrolle des Rechners während der Herstellung und eventueller Reparaturen.

---

* Einige der Statusanzeigen, die am Ende dieses Tests eingeschaltet sind, werden vom HP-15C normalerweise nicht verwendet.

** Wenn der Rechner nach einem [ON]/[×] Test oder einem [ON]/[+] Test die Meldung **Error 9** anzeigt, und Sie den Rechner trotzdem weiterverwenden wollen, empfiehlt es sich, zuvor den Permanentspeicher (wie auf Seite 63 beschrieben) zu löschen.

*** Durch die Tastenkombination [ON]/[+] wird eine dem zuvor beschriebenen Test ähnliche Überprüfung ausgelöst, die jedoch nicht von selbst abbricht. Dieser Test wird innerhalb von 15 Sekunden nach Drücken einer beliebigen Taste beendet. Die Kombination [ON]/[÷] löst eine Überprüfung des Tastenfeldes und der Anzeige aus. Nach dem Loslassen der [ON] Taste leuchten zunächst bestimmte Segmente der Anzeige auf. Zur Fortsetzung des Tests müssen alle Tasten des Tastenfelds gedrückt werden, und zwar von links nach rechts, beginnend mit der ersten Reihe. Beim Drücken der einzelnen Tasten leuchten jeweils verschiedene Segmente der Anzeige auf. Wenn der Rechner korrekt arbeitet und *alle Tasten in der richtigen Reihenfolge gedrückt werden*, wird nach dem Drücken der letzten Taste die Zahl **15** angezeigt. (Die Taste [ENTER] sollte sowohl zusammen mit den Tasten der dritten Reihe als auch zusammen mit den Tasten der vierten Reihe gedrückt werden.) Wenn der Rechner nicht fehlerfrei arbeitet oder *eine Taste außerhalb der korrekten Reihenfolge gedrückt wurde*, erscheint in der Anzeige die Meldung **Error 9**. *Beachten Sie dabei, daß der Rechner nicht notwendig reparaturbedürftig ist, wenn diese Fehlermeldung durch das Drücken einer Taste in der falschen Reihenfolge ausgelöst wurde.* Dieser Test kann durch Drücken einer beliebigen Taste außerhalb der Reihenfolge (was natürlich eine **Error 9** Meldung auslöst) vorzeitig beendet werden. Sowohl die Anzeige **Error 9** als auch die Anzeige **15** kann durch Drücken einer beliebigen Taste gelöscht werden.

Wenn sich Ihr Verdacht auf Fehlfunktion des Rechners aufgrund einer korrekten Anzeige in Schritt 2 nicht bestätigt hat, so besteht die Möglichkeit, daß Sie bei der Bedienung des Rechners einen Fehler gemacht haben. In diesem Fall empfehlen wir, den auf Ihr spezielles Problem zutreffenden Abschnitt dieses Handbuchs noch einmal durchzuarbeiten. Wenn Sie danach immer noch Schwierigkeiten feststellen, sollten Sie sich mit Hewlett-Packard unter einer der im Abschnitt «Service» (siehe Seite 267) gelisteten Adressen in Verbindung setzen.

Gewährleistung

Hewlett-Packard gewährleistet, daß das Produkt frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn das Produkt – direkt oder über einen autorisierten Hewlett-Packard Vertragshändler – an eine Hewlett-Packard Serviceniederlassung eingeschickt wird. Die Gewährleistungsfrist beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die auf unsachgemäße Veränderungen des Produkts durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt.

Die Gewährleistung gilt nur bei Vorlage der vollständig ausgefüllten Service-Karte in Verbindung mit entweder

a) dem von einem autorisierten HP-Vertragshändler ausgestellten und unterschriebenen Original-Kaufbeleg aus dem das Kaufdatum und. die Seriennummer des Produkts hervorgehen

oder

b) der Originalrechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Regelung unberührt.

Änderungsverpflichtung

Die Produkte von Hewlett-Packard werden auf der Basis der zum Zeitpunkt der Herstellung gegebenen technischen Spezifikationen verkauft. Hewlett-Packard übernimmt keine Verpflichtung zur nachträglichen Anpassung oder Modifikation einmal verkaufter Produkte.

Gewährleistungsinformation

Wenn Sie Fragen zu dieser Gewährleistungserklärung haben, nehmen Sie bitte Kontakt mit einem autorisierten Hewlett-Packard Händler oder einer Hewlett-Packard Verkaufs- oder Serviceniederlassung auf. Sollte dies nicht möglich sein, wenden Sie sich bitte an:

- In Europa:

Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box
CH-1217 Meyrin 2
Genf
Schweiz

Hinweis: Bitte senden Sie an diese Adresse *keine* Geräte zur Reparatur.

- In den Vereinigten Staaten:

Hewlett-Packard
Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Telefon: (503) 758-1010

- In allen anderen Ländern:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, California 94304
U.S.A.
Telefon: (415) 857-1501

Hinweis: Bitte senden Sie an diese Adresse *keine* Geräte zur Reparatur.

Service

Hewlett-Packard unterhält Serviceniederlassungen in vielen Ländern der Welt. Diese Niederlassungen stehen Ihnen jederzeit für eine eventuelle Reparatur zur Verfügung, auch wenn die Gewährleistungsfrist von einem Jahr bereits abgelaufen sein sollte. Reparaturen nach Ablauf der Gewährleistungsfrist sind kostenpflichtig.

Normalerweise erfolgt die Reparatur und der Rückversand von Hewlett-Packard Produkten innerhalb von 5 Werktagen. In Abhängigkeit von der Auslastung der Serviceniederlassung kann diese Zeitspanne im Einzelfall überschritten werden.

Service-Zentrale in den Vereinigten Staaten

In den Vereinigten Staaten erreichen Sie die Service-Zentrale von Hewlett-Packard für Taschenrechner- und Taschencomputer-Produkte unter der folgenden Anschrift:

Hewlett-Packard Company
Corvallis Division Service Department
P.O. Box 999/1000 N.E. Circle Blvd.
Corvallis, Oregon 97330, U.S.A.
Telefon: (503) 757-2000

Serviceniederlassungen in Europa

Hewlett-Packard unterhält Serviceniederlassungen in den folgenden Ländern. Wenn das Land, in dem Sie sich befinden, nicht aufgeführt ist, sollten Sie sich mit dem HP-Händler, bei dem Sie Ihr Gerät erworben haben, in Verbindung setzen.

BELGIEN
HEWLETT-PACKARD BELGIUM SA/NV
Boulevard de la Woluwe 100
Woluwelaan
B-1200 BRÜSSEL
Tel.: (2) 762 32 00

DÄNEMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 BIRKEROD (Kopenhagen)
Tel.: (02) 81 66 40

FINNLAND
HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 ESPOO 10 (Helsinki)
Tel.: (90) 455 02 11

FRANKREICH
HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 LES ULIS CEDEX
Tel.: (6) 907 78 25

DEUTSCHLAND
HEWLETT-PACKARD GmbH
Vertriebszentrale
Berner Straße 117
Postfach 560 140
D-6000 FRANKFURT 56
Tel.: (0611) 5004-1

NIEDERLANDE
HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
NL-1181 KK AMSTELVEEN (Amsterdam)
P.O. Box 67
Tel.: (020) 47 20 21

ITALIEN
HEWLETT-PACKARD S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 CERNUSCO SUL NAVIGLIO (Milan)
Tel.: (2) 90 36 91

NORWEGEN
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 OESTERAAS (Oslo)
Tel.: (2) 17 11 80

ÖSTERREICH
HEWLETT-PACKARD GmbH
Wagramerstrasse-Lieblgasse
A-1220 WIEN
Tel.: (0222) 23 65 11

GROSSBRITANNIEN
HEWLETT-PACKARD Ltd.
King Street Lane
Winnersh, Wokingham
GB BERKSHIRE RG11 5AR
Tel.: (734) 78 47 74

OSTEUROPA
Wenden Sie sich an die unter
Österreich angegebene Adresse.

SCHWEDEN
HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9
Box 19
S-16393 SPÅNGA
Tel.: (0046) 8 750 20 00

SCHWEIZ
HEWLETT-PACKARD (SCHWEIZ) AG
Allmend 2
CH-8967 WIDEN
Tel.: (057) 31 21 11

SPANIEN
HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
E MADRID 16
Tel.: (1) 458 2600

Internationale Serviceinformation

Nicht jede Hewlett-Packard Serviceniederlassung bietet Service für alle Hewlett-Packard Produkte. Wenn Sie jedoch Ihr Gerät bei einem autorisierten Hewlett-Packard Händler gekauft haben, können Sie sicher sein, daß in dem Land des Erwerbs auch Servicemöglichkeiten bestehen.

Wenn Sie sich nicht in dem Land befinden, in dem Sie Ihr Gerät erworben haben, befragen Sie die lokale Hewlett-Packard Serviceniederlassung nach den dort verfügbaren Servicemöglichkeiten. Wenn kein Service verfügbar ist, senden Sie Ihr Gerät an die zuvor aufgeführte Adresse der Service- Zentrale in den Vereinigten Staaten. Unter der gleichen Adresse können Sie auch eine Liste der Serviceniederlassungen in anderen Ländern anfordern.

Sämtliche durch den Versand entstehende Kosten gehen zu Ihren Lasten.

Reparaturkosten

Bei Reparaturen außerhalb der Gewährleistungsfrist werden Standardsätze zugrunde gelegt, die den Arbeitslohn und den Materialaufwand beinhalten. In den Vereinigten Staaten unterliegt der gesamte Rechnungsbetrag der lokalen Umsatzsteuer des Kunden. In europäischen Ländern ist der Rechnungsbetrag mehrwertsteuerpflichtig bzw. unterliegt ähnlichen Steuern. Der jeweilige Steueranteil wird in der Rechnung getrennt ausgewiesen.

Für Produkte, die durch Gewalteinwirkung oder sonstigen Mißbrauch beschädigt worden sind, gelten diese festen Reparatursätze nicht. In diesen Fällen wird die Reparatur individuell nach Arbeitszeit und Materialaufwand berechnet.

Service-Garantie

Bei Reparaturen außerhalb der Gewährleistungsfrist wird eine Garantie auf das Material und die Arbeitsleistung von 90 Tagen gegeben. Diese Garantiefrist gilt ab dem Reparaturdatum.

Versandanweisungen

Wenn Sie Ihr defektes Gerät einsenden, fügen Sie bitte bei:

- Eine vollständig ausgefüllte Service-Karte, einschließlich einer Fehlerbeschreibung.
- Die Originalrechnung oder einen sonstigen Kaufnachweis, sofern die Reparatur in die einjährige Gewährleistungsfrist fällt.

Zur Vermeidung von Transportschäden sollte das Gerät (zusammen mit der Service-Karte, einer kurzen Beschreibung des Fehlers sowie, falls erforderlich, dem Kaufnachweis) nur in der Originalverpackung oder einer adäquaten Schutzverpackung versandt werden. Transportschäden fallen nicht unter die einjährige Gewährleistung. Das verpackte Gerät sollte entweder direkt oder über Ihren HP-Vertragshändler zur nächsten Hewlett-Packard Serviceniederlassung geschickt werden. (Wenn Sie sich nicht in dem Land befinden, in dem Sie Ihr Gerät erworben haben, beziehen Sie sich bitte auf den Abschnitt «Internationale Serviceinformation».) Hewlett-Packard empfiehlt Ihnen, den Transport gegebenenfalls versichern zu lassen.

Die Versandkosten zur Hewlett-Packard Serviceniederlassung gehen zu Ihren Lasten, unabhängig davon, ob sich das Gerät noch in der Gewährleistungsfrist befindet oder nicht.

Bei Reparaturen innerhalb der Gewährleistungsfrist übernimmt die Serviceniederlassung die Kosten für den Rückversand. Bei Reparaturen außerhalb dieser Frist sind die Rücksendungskosten im Rechnungsbetrag enthalten.

Sonstiges

Hewlett-Packard bietet keine Service-Verträge an. Entwurf und Ausführung des Produkts und der Elektronik sind geistiges Eigentum von Hewlett-Packard; Service-Handbücher können daher nicht an Kunden abgegeben werden.

Sollten Sie weitere servicebezogene Fragen haben, setzen Sie sich bitte mit Ihrem HP-Vertragshändler oder der nächstgelegenen Hewlett-Packard Serviceniederlassung in Verbindung.

Benutzerberatung

Sollten beim Einsatz Ihres Geräts in bestimmten Anwendungsfällen Fragen auftauchen, so rufen Sie einfach unsere Kundenberatung an (siehe Verzeichnis der Niederlassungen) oder schreiben direkt an:

HEWLETT-PACKARD GmbH
Vertriebszentrale/Werbeabteilung
Bernerstraße 117
Postfach 560140
D-6000 FRANKFURT/M 56

Händler- und Produktinformation

Informationen betreffend des Händlernetzes, der Produkte und Preise erhalten Sie in der Bundesrepublik Deutschland über die Telefonnummer 0611/50041.

Temperaturspezifikationen

- Betriebstemperatur: 0° bis 55° C
- Lagertemperatur: –40° bis 65° C

Index Funktionstasten

ON

ON schaltet die Anzeige ein und aus **(Seite 18)**; wird auch verwendet zum Löschen des Permanentspeichers **(Seite 63)**, zum Ändern der Ziffer- und Dezimaltrennzeichen **(Seite 61)** und in verschiedenen Tests zur Funktionsfähigkeit des Rechners **(Seiten 263–264)**.

Komplexe Funktionen

Re≷Im. Vertauschen von Realteil und Imaginärteil. Aktiviert den Komplex-Modus (baut den imaginären Stack auf) und vertauscht den Real- und Imaginärteil des X-Registers **(Seite 124)**.

I wird zur Eingabe von komplexen Zahlen verwendet (baut einen imaginären Stack auf) **(Seite 121)**. Zusammen mit DIM wird es zur direkten Dimensionierung von Matrizen verwendet **(Seite 174)**. (Für die Verwendung mit Indexregister Funktionen siehe unter Indexregister-Kontrolle, **Seite 274**.)

(i) zeigt den Inhalt des imaginären X-Registers an, solange die Taste gedrückt wird **(Seite 124)**.

SF 8 setzt Flag 8; der Komplex-Modus wird aktiviert **(Seite 121)**.

CF 8 löscht Flag 8; der Komplex-Modus wird desaktiviert **(Seite 121)**.

Konvertierungen

→R transformiert die im X- und Y-Register gespeicherten Polarkoordinaten *r* (Betrag) und θ (Winkel) in die entsprechenden Rechteckskoordinaten *x* und *y* **(Seite 52)**. Operationen im Komplex-Modus, siehe Seite 134.

→P transformiert die im X- und Y-Register gespeicherten Rechteckskoordinaten *x* und *y* in Polarkoordinaten *r* (Betrag) und θ (Winkel) **(Seite 30)**. Operationen im Komplex-Modus, siehe Seite 134.

→H.MS konvertiert dezimale Stunden (oder Grad) in Stunden, Minuten und Sekunden (oder Grad, Minuten, Sekunden) **(Seite 27)**.

→H konvertiert Stunden, Minuten, Sekunden (oder Grad, Minuten, Sekunden) in dezimale Stunden (oder Grad) **(Seite 27)**.

→RAD konvertiert Grad in Radiant (Bogenmaß) **(Seite 27)**.

→DEG konvertiert Radiant in Grad **(Seite 27)**.

Zifferneingabe

ENTER kopiert den Inhalt des angezeigten X-Registers in das Y-Register; dient zur Trennung von aufeinanderfolgenden Zahleneingaben **(Seiten 22, 37)**.

CHS kehrt das Vorzeichen der Mantisse oder des Exponenten der im angezeigten X-Register gespeicherten Zahl um **(Seiten 19, 124)**.

EEX gibt den Exponenten ein; die nächsten eingetasteten Ziffern werden als Exponent zur Basis 10 interpretiert **(Seite 19)**.

0 bis 9 Zifferntasten **(Seite 22)**.

· Dezimalpunkt **(Seite 22)**.

Anzeige-Kontrolle

FIX schaltet die Anzeige in das Festkommaformat **(Seite 58)**.

SCI schaltet die Anzeige in die wissenschaftliche Notation **(Seite 58)**.

ENG schaltet die Anzeige in die technische Notation **(Seite 59)**.

Mantisse. Durch f CLEAR PREFIX werden alle zehn signifikanten Stellen der Zahl im X-Register angezeigt (solange die PREFIX Taste gedrückt ist) **(Seite 60)**; löscht außerdem alle unvollständigen Eingaben **(Seite 19)**.

Hyperbolische Funktionen

HYP SIN, HYP COS, HYP TAN berechnen den Sinus hyperbolicus, den Cosinus hyperbolicus bzw. den Tangens hyperbolicus **(Seite 28)**.

HYP^{-1} SIN, HYP^{-1} COS, HYP^{-1} TAN berechnen den inversen Sinus hyperbolicus, den inversen Cosinus hyperbolicus bzw. den inversen Tangens hyperbolicus **(Seite 28)**.

Indexregister-Kontrolle

[I] Indexregister (R_I). Speicherregister für indirekte Programmsteuerung (Programmverzweigung mit [GTO] und [GSB], Programmschleifen mit [ISG] und [DSE]), indirekte Flagsteuerung und indirekte Anzeigenformatsteuerung **(Seite 107)**; dient zusätzlich zur Eingabe komplexer Zahlen und zur Aktivierung des Komplex-Modus **(Seite 121)**.

[(i)] indirekte Operationen. Dient zur Adressierung eines *anderen* Speicherregisters *mittels* R_I für Speicher-, Rückruf- und arithmetische Operationen sowie zur Programmschleifensteuerung **(Seite 107)**. Dient zusammen mit [DIM] auch zur Aufteilung der Speicherregister **(Seite 215)**.

Logarithmische und Exponentialfunktionen

[LN] berechnet den natürlichen Logarithmus **(Seite 28)**.

[e^x] natürliche Exponentialfunktion: berechnet *e* hoch der Zahl in der Anzeige (dem X-Register) **(Seite 28)**.

[LOG] berechnet den dekadischen Logarithmus (Basis 10) **(Seite 28)**.

[10^x] dekadische Exponentialfunktion: berechnet 10 hoch der Zahl in der Anzeige (dem X-Register) **(Seite 28)**.

[y^x] berechnet die Potenz der im Y-Register gespeicherten Zahl zu der in der Anzeige (X-Register) gespeicherten Basis (geben Sie zuerst *y* ein, dann *x*); bewirkt einen Stack Drop **(Seite 29)**.

Mathematische Operationen

[+], [−], [×], [÷] arithmetische Operatoren; bewirken einen Stack Drop **(Seite 29)**.

[$\sqrt{x}$] berechnet die Quadratwurzel von *x* **(Seite 25)**.

[x^2] berechnet das Quadrat von *x* **(Seite 25)**.

[x!] berechnet die Fakultät (*x!*) von *x* oder den Funktionswert der Gammafunktion (Γ) von $(1+x)$ **(Seite 25)**.

[1/x] berechnet den Reziprokwert von *x* **(Seite 25)**. (Siehe Seite 275 für die Verwendung im Zusammenhang mit Matrizen.)

[π] lädt die Zahl π in die Anzeige **(Seite 24)**.

[SOLVE] berechnet die reellen Nullstellen der Funktion $f(x)$, wobei $f(x)$ vom Benutzer in einem Unterprogramm spezifiziert werden muß **(Seite 180)**.

[$\int_y^x$] Integration: berechnet das bestimmte Integral von $f(x)$, wobei $f(x)$ vom Benutzer in einem Unterprogramm spezifiziert werden muß **(Seite 194)**.

Matrizenfunktionen

[DIM] dimensioniert die spezifizierte Matrix {[A] bis [E], [I]} **(Seite 141)**.

[RESULT] spezifiziert die Matrix, die das Ergebnis bestimmter Matrizenoperationen aufnehmen soll **(Seite 148)**.

[USER] User-Modus: Zeilen- und Spaltenindex in R_0 und R_1 werden nach jeder Ausführung von [STO] oder [RCL] {[A] bis [E], [(i)]} automatisch inkrementiert **(Seite 144)**.

gespeicherte Matrix in die «geteilte» Form ($\mathbf{Z}^P$) **(Seite 162)**.

[x=0] [TEST] 0 [TEST] 5 [TEST] 6 Vergleichsoperationen mit Matrixdeskriptoren im X-Register oder in den Registern X und Y. [x=0] und [TEST] 0 ($x \neq 0$) vergleichen den Inhalt des X-Registers mit 0. Matrixdeskriptoren werden als ungleich 0 betrachtet. [TEST] 5 ($x = y$) und [TEST] 6 ($x \neq y$) vergleichen die Deskriptoren in X und Y. Das Ergebnis beeinflußt die Programmausführung; bei «falsch» (false) wird die nächste Zeile übersprungen **(Seite 174)**.

Zahlenmanipulation

[ABS] berechnet den Absolutwert der Zahl in der Anzeige **(Seite 24)**.

[FRAC] berechnet den dezimalen Anteil der Zahl in der Anzeige (dem X-Register) durch Abschneiden des ganzzahligen Anteils **(Seite 24)**.

[INT] berechnet den ganzzahligen Anteil der Zahl in der Anzeige (dem X-Register) durch Abschneiden des ganzzahligen Anteils **(Seite 24)**.

[RND] rundet die zehnstellige Mantisse der Zahl im X-Register auf das aktuelle Anzeigeformat **(Seite 24)**.

Prozentrechnung

[%] Prozent: berechnet x% (Zahl in der Anzeige) der Zahl im Y-Register **(Seite 29)**. Im Gegensatz zu den meisten anderen Funktionen zweier Variablen bedingt [%] keinen Stack Drop.

[Δ%] prozentuale Differenz: berechnet die prozentuale Differenz zwischen dem Wert des Y-Registers und dem Wert des angezeigten X-Registers **(Seite 30)**. Es erfolgt kein Stack Drop.

Vorwahltasten

[f]. Vor einer Funktionstaste bewirkt [f], daß die goldfarben oberhalb dieser Taste aufgedruckte Funktion gewählt wird **(Seite 18)**.

[g]. Vor einer Funktionstaste bewirkt [g], daß die blaue auf die Vorderseite der Taste aufgedruckte Funktion gewählt wird **(Seite 21)**.

Mehr über andere Vorwahltasten finden Sie unter Anzeige-Kontrolle **(Seite 273)**, unter Speicherung **(Seite 278)** und im Index Programmtasten **(Seite 278)**.

CLEAR [PREFIX] löscht jede Vorwahltaste und unvollständige Eingaben wie z.B. [f] [SCI] **(Seite 19)**. Zusätzlich wird die zehnstellige Mantisse der Zahl im X-Register angezeigt **(Seite 60)**.

Wahrscheinlichkeitsrechnung

[Cx,y] Kombination: berechnet die Anzahl aller Möglichkeiten y verschiedene Elemente zu Mengen mit jeweils x Elementen ohne Beachtung der Reihenfolge und ohne Wiederholungen zusammenzufassen. Bewirkt einen Stack Drop **(Seite 47)**. (Für die Verwendung im Zusammenhang mit Matrizen siehe Seite 276.)

[Py,x] Permutation: berechnet die Anzahl aller

adressierte Speicherregister {0 bis 9, .0 bis .9, [I], [(i)]} **(Seite 42)**. Wird auch zur Speicherregisterarithmetik verwendet: neuer Registerinhalt = alter Registerinhalt {[+], [−], [×], [÷]} Wert in der Anzeige **(Seite 44)**.

[RCL] ruft eine Kopie des Werts im adressierten {0 bis 9, .0 bis .9, [I], [(i)]} Speicherregisters in die Anzeige zurück **(Seite 42)**. Wird auch zur Speicherregisterarithmetik verwendet: neue Anzeige = alte Anzeige {[+], [−], [×], [÷]} Registerinhalt **(Seite 44)**.

CLEAR [REG] ersetzt alle Inhalte der Datenspeicherregister durch Null **(Seite 43)**.

[LSTx] ruft die vor der letzten Operation angezeigte Zahl in das angezeigte X-Register zurück **(Seite 28)**.

Trigonometrische Funktionen

[DEG] setzt den trigonometrischen Modus auf (dezimale) Altgrad – dies wird durch das Fehlen der Statusanzeigen **GRAD** oder **RAD** gekennzeichnet **(Seite 26)**. [DEG] kann nicht im Komplex-Modus verwendet werden.

[RAD] setzt den trigonometrischen Modus auf Radiant (Bogenmaß) – dies wird durch die Statusanzeige **RAD** gekennzeichnet **(Seite 26)**.

[GRD] setzt den trigonometrischen Modus auf Neugrad – dies wird durch die Statusanzeige **GRD** gekennzeichnet **(Seite 26)**. [GRD] kann nicht im Komplex-Modus verwendet werden.

[SIN], [COS], [TAN] berechnet den Sinus, Cosinus und den Tangens der Zahl in der Anzeige (X-Register) **(Seite 26)**.

[SIN⁻¹], [COS⁻¹], [TAN⁻¹] berechnet den Arcussinus, Arcuscosinus und den Arcustangens der Zahl in der Anzeige (X-Register) **(Seite 26)**.

Index Programmtasten

[P/R] Programm/Run-Modus: schaltet den Rechner in den Programm-Modus (**PRGM** Statusanzeige an) oder in den Run-Modus (**PRGM** Statusanzeige ist aus) **(Seite 66)**.

CLEAR [PRGM] im Programm-Modus: löscht alle Programme aus dem Speicher und gibt alle Programmregister frei. Im Run-Modus wird der Rechner nur auf Zeile 000 zurückgesetzt **(Seite 67)**.

[MEM] zeigt die augenblickliche Speicheraufteilung des Rechners an (Anzahl der zur Datenspeicherung verwendeten Register, der Register im Common Pool und der Register im Programmspeicher) **(Seite 215)**.

[←] löschen: im Programm-Modus wird die angezeigte Anweisung aus dem Programmspeicher gelöscht. Alle nachfolgenden Anweisungen werden um eine Zeile nach oben verschoben **(Seite 83)**.

[LBL] *Label*: dient, gefolgt von einer Labelbezeichnung, zur Kennzeichnung des Beginns einer Routine **(Seite 83)**.

[A] [B] [C] [D] [E] 0 1 2 3 4 5 6 7 8 9 .0 .1 .2 .3 .4 .5 .6 .7 .8 .9 Labelbezeichnungen: definieren den Beginn einer Programmroutine, wenn [LBL] voransteht **(Seite 67)**. Wird auch ohne [LBL] verwendet, um die Ausführung der spezifizierten Routine zu starten **(Seite 69)**.

[USER] aktiviert und desaktiviert den User-Modus, der die (weißen) Primärfunktionen und die goldenfarbenen Alternativfunktionen der fünf Funktionen oben links (die Tasten [A] bis [E]) vertauscht **(Seite 69)**. Der User-Modus verändert auch den Gebrauch von [STO] und [RCL] bei Matrizenrechnungen {[A] bis [E]}. Im User-Modus werden automatisch R_0 (Anzahl der Zeilen) oder R_1 (Anzahl der Spalten) beim Speichern oder Zurückrufen von Matrixelementen erhöht **(Seite 144)**.

[GTO] Sprunganweisung: wird mit einer Labelbezeichnung (siehe oben) oder [I] verwendet, um den Rechner auf das spezifizierte Label zu positionieren. Bei Ausführung von [GTO] innerhalb eines Programms wird die Kontrolle an die bezeichnete Anweisung übergeben. Bei Ausführung im Run-Modus wird der Rechner nur neu positioniert. **(Seite 90)**. Wenn eine negative Zahl in R_I gespeichert ist, bewirkt [GTO] [I] einen Sprung zu einer *Zeilennummer* **(Seite 109)**.

[GTO] [CHS] *nnn* Sprunganweisung zu einer Zeilennummer: setzt den Rechner in die existierende Zeile mit der Nummer *nnn*. Nicht programmierbar **(Seite 82)**.

[GSB] Aufruf eines Unterprogramms: wird mit einer Labelbezeichnung (siehe oben) oder [I] verwendet, um die Ausführung einer gegebenen, mit einem Label versehenen Routine zu beginnen. Kann sowohl innerhalb eines Programms als auch über das Tastenfeld (im Run-Modus) verwendet werden. Eine [RTN] Anweisung bewirkt, daß die Ausführung mit der ersten auf die [GSB] Anweisung folgenden Zeile fortgesetzt wird **(Seite 101)**.

[BST] Rückschritt: der Rechner wird im Programmspeicher um eine oder mehrere Zeilen zurückpositioniert. (Dies gilt auch im Programm-Modus.) Die Zeilennummer und der Inhalt der vorherigen Programmzeile werden angezeigt **(Seite 83)**.

[SST] Einzelschritt: im Programm-Modus wird der Rechner im Programmspeicher eine oder mehrere Zeilen vorwärtspositioniert. Im Run-Modus wird die augenblickliche Programmzeile angezeigt und ausgeführt und danach zur nächsten Zeile gesprungen **(Seite 82)**.

PSE Pause: unterbricht die Programmausführung für ungefähr eine Sekunde, um den Inhalt des X-Registers anzuzeigen **(Seite 68)**.

R/S Run/Stop: wechselweises Starten und Anhalten der Programmausführung. Die Programmausführung beginnt immer mit der Zeile, auf die der Rechner momentan positioniert ist **(Seite 68)**.

RTN Rücksprung: bewirkt innerhalb eines Hauptprogramms, daß der Rechner in Zeile 000 verzweigt und die Ausführung anhält **(Seite 68)**. Innerhalb eines Unterprogramms erfolgt ein Rücksprung in die Zeile nach der GSB Anweisung **(Seite 101)**.

SF Setzen eines Flags (= true): setzt den spezifizierten Flag (0 bis 9). Flag 0 bis Flag 7 sind Benutzer-Flags, Flag 8 deutet den Komplex-Modus und Flag 9 eine Overflow-Bedingung an **(Seite 92)**.

CF Löschen eines Flags (= false): löscht den spezifizierten Flag (0 bis 9) **(Seite 92)**.

F? Abfrage eines Flags. Ist der Flag gesetzt, wird die Ausführung mit der nächsten Zeile fortgesetzt; andernfalls wird diese übersprungen **(Seite 92)**.

$x \leqslant y$ $x=0$ TEST {0 bis 9}. Vergleichsoperationen. Jede dieser Operationen vergleicht den Wert im X-Register mit Null oder mit dem Wert im Y-Register in der angedeuteten Weise. Ist das Ergebnis «wahr» (true), wird die Programmausführung mit der nächsten Zeile im Programmspeicher fortgesetzt; bei «falsch» (false) wird die nächste Programmzeile übersprungen **(Seite 91)**.

$x=0$ und TEST 0, 5 und 6 können auch mit komplexen Zahlen und Matrix-Deskriptoren verwendet werden **(Seiten 132, 174)**.

TEST 0 $x \neq 0$
TEST 1 $x > 0$
TEST 2 $x < 0$
TEST 3 $x \geqslant 0$
TEST 4 $x \leqslant 0$
TEST 5 $x = y$
TEST 6 $x \neq y$
TEST 7 $x > y$
TEST 8 $x < y$
TEST 9 $x \geqslant y$

DSE erniedrigt einen Zähler in einem festgesetzten Register und überspringt eine Programmzeile, wenn der neue Zählerwert gleich oder kleiner als ein spezifizierter Testwert ist **(Seite 109)**.

ISG erhöht einen Zähler in einem festgesetzten Register und überspringt eine Programmzeile, wenn der neue Zählerwert größer als ein spezifizierter Testwert ist **(Seite 109)**.

Sachindex

Die Seitenangaben in **Fett**schrift beziehen sich auf die Hauptreferenz; die Seitenangaben in Normalschrift sind Hinweise auf weitere Referenzen.

A

B

C

D

K

N

O

P

Q

R

S

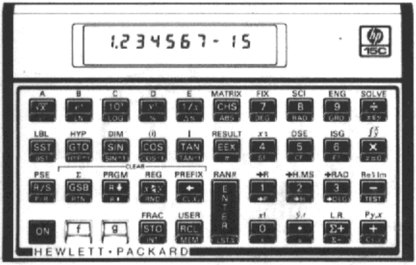
W

X

Y

Z

Tastenfeld und Permanentspeicher des HP-15C



SPEICHERSTACK

	reeller	imaginärer
T		
Z		
Y		
X		

Anzeige des reellen X-Registers.

LAST X		

DATENSPEICHERPOOL

R_I	

R_0	0		$R_{.0}$	10
R_1	1		$R_{.1}$	11
R_2	2	n	$R_{.2}$	12
R_3	3	Σx	$R_{.3}$	13
R_4	4	Σx^2	$R_{.4}$	14
R_5	5	Σy	$R_{.5}$	15
R_6	6	Σy^2	$R_{.6}$	16
R_7	7	Σxy	$R_{.7}$	17
R_8	8		$R_{.8}$	18
R_9	9		$R_{.9}$	19

COMMON POOL

R_{20} … R_{65}

Matrizenspeicher

Imaginärer Stack

SOLVE und $\int_y^x$

Freie Register

Programmspeicher (maximal sieben Programmzeilen pro Register)

In der Standardkonfiguration sind R_0 bis R_{19} der Datenspeicherung zugewiesen.

Die aktuelle Aufteilung kann mittels der DIM (i) Funktion verändert werden.

Jedes Register besteht aus sieben Bytes; jede Programmanweisung belegt ein bzw. zwei Bytes. Bei Bedarf wird jeweils ein Register in Programmspeicher umgewandelt; beginnend mit dem Register mit der höchsten Adresse.

Die Zuteilung von Speicher für Programmzeilen erfolgt *innerhalb* des Common Pool automatisch.

In der Standardkonfiguration besteht der Common Pool aus den Registern R_{20} bis R_{65}, von denen die oben aufgeführten Funktionen und Benutzerprogramme Speicherplatz abziehen.

VERKAUFSNIEDERLASSUNGEN:

Hewlett-Packard GmbH:
6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1
7030 Böblingen, Herrenbergerstraße 110, Tel. (07031) 667-1
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 5971-1
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 63804-1
8028 Taufkirchen, Eschenstraße 5, Tel. (089) 6117-1
3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 466001
8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 522083/87
1000 Berlin 30, Keithstraße 2–4, Tel. (030) 249086
6800 Mannheim, Roßlauer Weg 2–4, Tel. (0621) 70050
7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 70241
7517 Waldbronn 2, Hewlett-Packard Straße, Tel. (07243) 602-1

Hewlett-Packard (Schweiz) AG:
Allmend 2, CH-8967 Widen, Tel. (057) 312111

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:
Wagramerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 236511

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

SERVICENIEDERLASSUNGEN:

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

Hewlett-Packard (Schweiz) AG:
Allmend 2, CH-8967 Widen, Tel. (057) 312111

Hewlett-Packard Ges. m. b. H., für Österreich / für sozialistische Staaten:
Wagramerstraße-Lieblgasse 1, A-1220 Wien, Tel. (0222) 236511

HEWLETT PACKARD

00015-90002 CP

Printed in Singapore 8/82