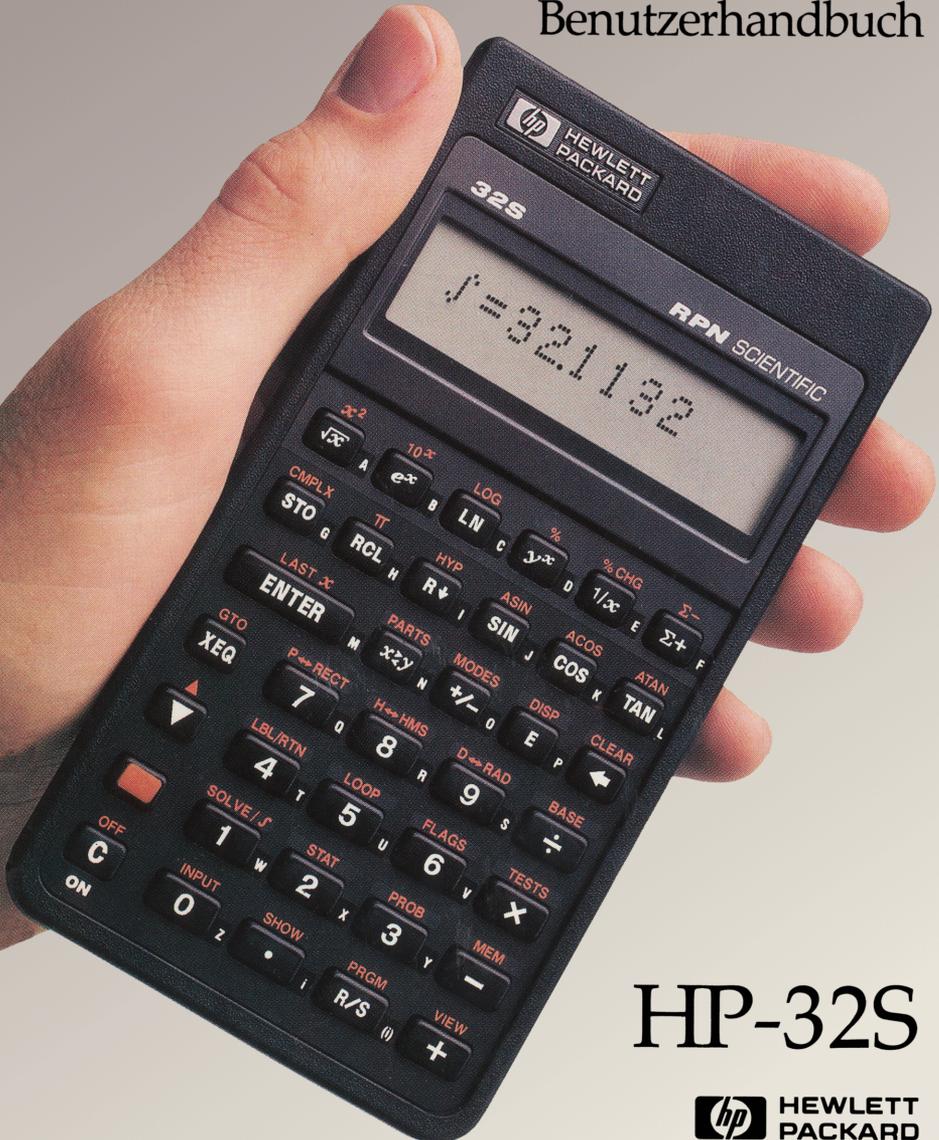


HEWLETT-PACKARD

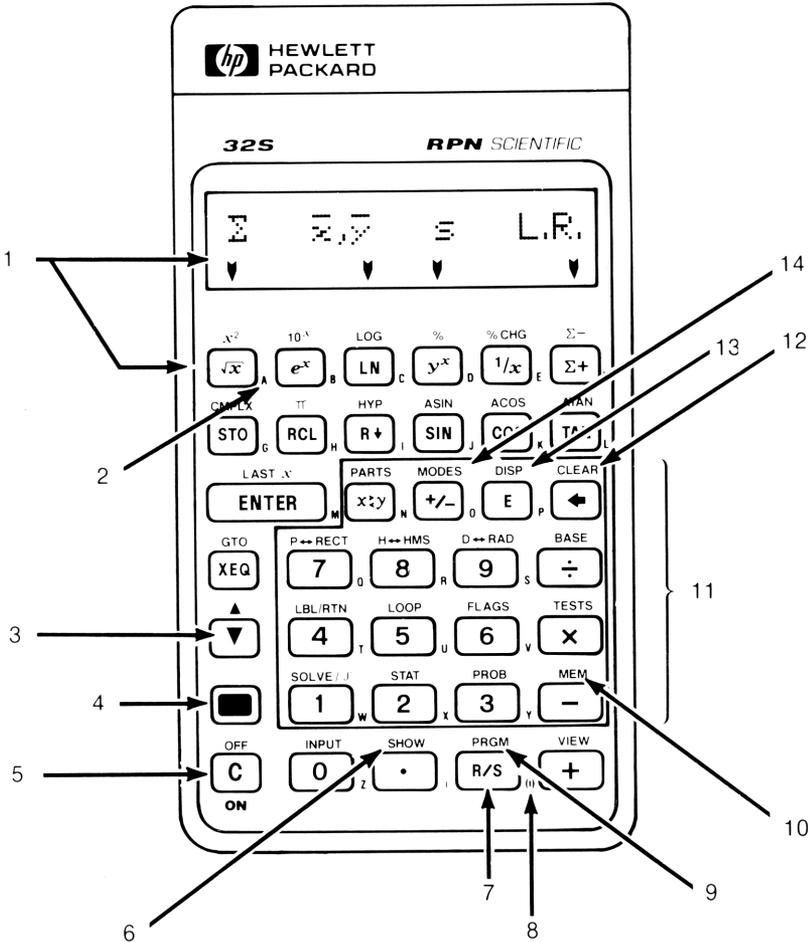
RPN Scientific Calculator

Benutzerhandbuch



HP-32S

 HEWLETT
PACKARD



- | | |
|---|---|
| 1. Menü und Menütasten. | 8. Für indirekte Adressierung über <i>i</i> . |
| 2. Buchstabetasten für Variablen und Labels. | 9. Ein- und Ausschalten von Programmeingabe-Modus. |
| 3. Durchsehen von Programmen und Listen. | 10. Benutzerspeicher; gespeicherte Variablen und Programme. |
| 4. Umschalttaste. | 11. Menütasten (eingerahmter Bereich) |
| 5. Ein; Löschen von Anzeige, Menü, Programmeingabe. | 12. Löschen des Speicherbereichs (ganz oder teilweise). |
| 6. Anzeige aller Dezimalstellen. | 13. Spezifizieren des Anzeigeformats. |
| 7. Start/Stop für Programme. | 14. Spezifizieren von Winkelmodus und Dezimalzeichen |

HP-32S **RPN Scientific Calculator**

Benutzerhandbuch



1. Ausgabe Juli 1988
Bestellnummer 00032-90041

Hinweis

Änderungen der in dieser Dokumentation enthaltenen Informationen sind vorbehalten. Allgemeine Informationen über den Rechner und zur Gewährleistung finden Sie auf den Seiten 248 und 252.

Hewlett-Packard übernimmt weder ausdrücklich noch stillschweigend irgendwelche Haftung für die in diesem Handbuch dargestellten Programme und Beispiele—weder für deren Funktionsfähigkeit noch deren Eignung für irgendeine spezielle Anwendung. Hewlett-Packard haftet nicht für direkte oder indirekte Schäden im Zusammenhang mit oder als Folge der Lieferung, Benutzung oder Leistung der Programme. (Dies gilt nicht, soweit gesetzlich zwingend gehaftet wird.)

Hewlett-Packard übernimmt keine Verantwortung für den Gebrauch oder die Zuverlässigkeit von HP Software unter Verwendung von Geräten, welche nicht von Hewlett-Packard geliefert wurden.

Diese Dokumentation enthält urheberrechtlich geschützte Informationen. Alle Rechte, insbesondere das Recht der Vervielfältigung und Verbreitung sowie der Übersetzung, bleiben vorbehalten. Kein Teil der Dokumentation darf in irgendeiner Form (durch Fotokopie, Mikrofilm oder ein anderes Verfahren) ohne vorherige schriftliche Zustimmung von Hewlett-Packard reproduziert oder unter Verwendung elektronischer Systeme verarbeitet, vervielfältigt oder verbreitet werden.

- © 1988 Hewlett-Packard GmbH
- © 1988 Hewlett-Packard Company

Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

Druckgeschichte

1. Ausgabe

Juli 1988

Fertigungsnr. 00032-90042

Vorwort

Ihr HP-32S reflektiert die hervorragende Qualität und die Aufmerksamkeit bis zum Detail bei der Entwicklung und Fertigung, wodurch sich Hewlett-Packard Produkte seit über 40 Jahren im Markt hervorheben. Hewlett-Packard steht hinter diesem Taschenrechner: Sie erhalten Unterstützung bei der Anwendung des Rechners (siehe Innenseite des Rückumschlags) und weltweiten Reparaturservice.

Hewlett-Packard Qualität

HP Taschenrechner zeichnen sich durch einfache Handhabung, Dauerhaftigkeit und Zuverlässigkeit aus.

- Der Rechner wurde so konzipiert, daß er den Beanspruchungen der täglichen Arbeitswelt hinsichtlich Mechanik, Temperatur- und Feuchtigkeitsschwankungen widersteht.
- Der Rechner und das zugehörige Handbuch wurden auf einfache Handhabung ausgelegt und getestet. Es wurde die Spiralbindung gewählt, damit Sie das Handbuch problemlos aufgeschlagen lassen können; außerdem wurden viele Beispiele aufgenommen, um die vielseitigen Einsatzmöglichkeiten des Rechners aufzuzeigen.
- Hochqualitative Materialien und permanent eingeprägte Tastenbezeichnungen sorgen für eine lange Lebenszeit und eine gute Bedienbarkeit des Tastenfelds.
- CMOS Technologie hält die Daten auch noch nach dem Ausschalten gespeichert und sorgt für eine lange Lebenszeit der Batterien.
- Der Mikroprozessor wurde hinsichtlich schneller und zuverlässiger Berechnungen optimiert. (15-stellige interne Genauigkeit!)
- Extensive Forschung führte zu einem Design, welches praktisch die Einflüsse statischer Elektrizität eliminiert (ein potentiell Risiko für Störungen und Datenverlust in Rechnern).

Leistungsmerkmale des Rechners

Die Fähigkeiten des HP-32S beruhen auf den Bedürfnissen und Wünschen vieler Kunden. Der Rechner enthält unter anderem:

- Alle Funktionen sind entweder über das Tastenfeld oder über Menüs erreichbar. Die Eingabe von Funktionsnamen ist überflüssig.
- Ausführliche Meldungen, wie z.B. `DIVIDE BY 0` anstatt von `ERR 21`. In den Variablen A bis Z ist das Speichern von Zahlen möglich.
- HP-traditionelle UPN Eingabelogik, welche Ihnen das Eintasten verkürzt.
- 390 Bytes zum Speichern von Daten und Programmen.
- Weiterentwickelte Funktionen für Statistikberechnungen, Umrechnungen zwischen Zahlensystemen, arithmetische Berechnungen mit komplexen Zahlen, Integralrechnung und Lösen der unbekanntenen Variablen in einer Gleichung.
- Extensive HP Programmierungsmöglichkeiten, einschließlich Edierfunktionen und Benennung von Ein- und Ausgabe, Unterprogramme, Schleifenbildung, bedingte Funktionen, Flags und indirekte Adressierung.

Inhaltsverzeichnis

Teil 1: Allgemeine Anwendungsweise

1	14	Bedienungsgrundlagen
	14	Die ersten Schritte
	14	Ein- und Ausschalten des Rechners
	14	Einstellen des Anzeigekontrasts
	15	Wichtige Tastenfeld- und Anzeigefunktionen
	15	Umgeschaltete Tasten
	15	Die Alphatasten
	15	Löschen von Zeichen
	16	Verwenden von Menüs
	19	Verlassen von Menüs
	20	Indikatoren
	21	Eintippen von Zahlen
	21	Umkehren des Vorzeichens
	22	Potenzen von 10
	23	Einzelheiten zur Zifferneingabe
	24	Wertebereich für Zahlen und OVERFLOW
	24	Arithmetische Berechnungen
	24	Einwertige Funktionen
	25	Zweiwertige Funktionen
	26	Kettenrechnungen
	29	Übungsaufgaben
	29	Spezifizieren des Anzeigeformats
	29	Punkt und Komma in Zahlen
	30	Anzahl der Dezimalstellen (■DISP)
	31	Temporäre Anzeige der vollen Genauigkeit
	32	Meldungen
	33	Speicherbereich des Rechners
	33	Anzeigen des freien Speicherbereichs
	34	Löschen des gesamten Speicherbereichs

- 2**
- 35 Der automatische Speicherstack**
35 Was ist ein "Stack"
36 Durchsehen des Stacks (**R↓**)
37 Austauschen des X- und Y-Registerinhalts (**xzy**)
38 Arithmetik—der Ablauf im Stack
39 Funktionsweise von ENTER
40 Funktionsweise von CLEAR X
41 Das LAST X Register
42 Fehlerkorrektur mit **LASTx**
43 Wiederverwenden von Zahlen mit **LASTx**
44 Kettenrechnungen
45 Bearbeitungsfolge in Aufgabenstellungen
46 Übungsaufgaben
- 3**
- 47 Datenspeicherung in Variablen**
48 Speichern und Zurückrufen von Zahlen
49 Ansehen von Variablen im VAR Katalog
50 Löschen von Variablen
50 Arithmetik mit Variablen
50 Speicherarithmetik
51 Rückruferarithmetik
53 Die Variable "i"
- 4**
- 54 Reellwertige Funktionen**
55 Exponential- und Logarithmusfunktionen
56 Potenzfunktion (y^x)
56 Trigonometrie
56 Eingeben von π
56 Einstellen des Winkelmodus
57 Trigonometrische Funktionen
59 Hyperbolische Funktionen
59 Prozentfunktionen (% , %CHG)
60 Konvertierungsfunktionen
60 Koordinatenkonvertierungen ($P \leftrightarrow \text{RECT}$)
63 Konvertierungen zwischen Dezimal- und Zeitformaten ($H \leftrightarrow \text{HMS}$)
64 Winkelkonvertierungen ($D \leftrightarrow \text{RAD}$)
65 Wahrscheinlichkeitsfunktionen
67 Teile von Zahlen
67 Funktionsnamen

Teil 2: Programmierung

- 5** **70** **Einfache Programme**
71 Erzeugen eines Programms
71 Programmgrenzen (LBL und RTN)
72 Programmeingabe (PRGM)
75 Aufrufen eines Programms
75 Ausführen eines Programms (XEQ)
76 Testen eines Programms
77 Ein- und Ausgabe von Daten
77 Eingabe von Daten in Variablen (INPUT)
79 Anzeigen von Variableninhalten (VIEW)
82 Anhalten oder Unterbrechen eines Programms
82 Programmieren eines Stopps oder einer Pause (STOP, PSE)
82 Unterbrechen eines gestarteten Programms
82 Fehlerbedingte Stopps
83 Edieren eines Programms
84 Der Programmspeicher
84 Ansehen des Programmspeichers
84 Speicherbelegung
85 Programmkatalog (MEM)
85 Löschen eines oder mehrerer Programme
86 Prüfsumme
87 Nicht programmierbare Funktionen
87 Polynome und Hornersches Schema
- 6** **90** **Programmierungstechniken**
90 Routinen in Programmen
91 Aufrufen von Unterprogrammen (XEQ, RTN)
92 Verschachtelte Unterprogramme
93 Verzweigungen (GTO)
95 Bedingte Funktionen
96 Vergleichsoperationen (TESTS)
97 Flags
99 Schleifen (GTO, LOOP)
100 Bedingte Schleifen (GTO)
101 Schleifenzähler (DSE, ISG)
103 Indirekte Adressierung von Variablen und Labels
103 Die Variable "i"
104 Indirekte Adresse, (i)
105 Programsteuerung mit (i)

Teil 3: Fortgeschrittene Operationen

- 7** **110** **Lösen einer Unbekannten in Gleichungen**
111 Anwenden von SOLVE
112 Entwickeln von Programmen für SOLVE
113 Beispiele zur Anwendung von SOLVE
118 Verstehen und Steuern von SOLVE
119 Überprüfen des Ergebnisses
119 Unterbrechen einer SOLVE Berechnung
120 Wählen von Anfangsnäherungen für SOLVE
124 Verwenden von SOLVE in Programmen
125 Weitere Informationen
- 8** **126** **Numerische Integration**
127 Anwenden der Integration (\int FN)
128 Entwickeln von Programmen für \int FN
128 Beispiele zur Anwendung von \int FN
131 Genauigkeit der Integration
132 Spezifizieren der Genauigkeit
132 Interpretation der Genauigkeit
134 Verwenden der Integration in Programmen
136 Weitere Informationen
- 9** **137** **Operationen mit komplexen Zahlen**
138 Der komplexe Stack
139 Komplexe Operationen
142 Verwenden von Zahlen in Polarnotation
- 10** **144** **Rechnen in verschiedenen Zahlensystemen**
146 Arithmetik im Basis-Modus 2, 8 und 16
147 Darstellung von Zahlen
148 Negative Zahlen
149 Wertebereich von Zahlen
149 Fenster für große Binärzahlen
150 Anzeigen (SHOW) von unsichtbaren
Zahlenteilen
151 Programmierung mit BASE
151 Wählen des Basis-Modus in einem Programm
151 Zahleneingabe in Programmzeilen

11	153	Statistikberechnungen
	153	Eingeben von Statistikdaten ($\Sigma+$, $\Sigma-$)
	154	Dateneingabe für eine Variable
	154	Dateneingabe für zwei Variablen
	155	Korrektur der Dateneingabe
	156	Ausführung von Statistikberechnungen
	156	Mittelwert und Standardabweichung
	158	Lineare Regression
	160	Grenzen für die Genauigkeit von Daten
	161	Summationswerte und Statistikregister
	161	Summationsstatistik
	162	Die Statistikregister im Speicherbereich

Teil 4: Applikationen

12	164	Mathematikprogramme
	164	Vektoroperationen
	175	Lösen eines linearen Gleichungssystems über Determinantenverfahren
	183	Lösen eines linearen Gleichungssystems durch Matrix-Inversion
	191	Quadratische Gleichung
	198	Koordinatentransformationen
13	204	Statistikprogramme
	204	Kurvenanpassung
	215	Normalverteilung und Verteilungsfunktion
14	222	Sonstige Programme
	222	Annuitätenrechnung (TVM)
	229	Einheitenkonvertierungen
	235	Primzahlengenerator

Teil 5: Anhänge und Index

- A**
- 240** **Kundenunterstützung, Batterien und Service**
 - 240** Unterstützung beim Anwenden des Rechners
 - 240** Antworten auf allgemeine Fragen
 - 242** Stromversorgung und Batterien
 - 242** "Schwache Batterien" Indikator
 - 243** Einsetzen der Batterien
 - 245** Umgebungsbedingungen
 - 245** Feststellen der Reparaturbedürftigkeit
 - 246** Funktionsprüfung des Rechners—der Selbsttest
 - 248** Einjährige Gewährleistungsfrist
 - 248** Gewährleistungsumfang
 - 248** Gewährleistungsausschluß
 - 249** Im Reparaturfall
 - 249** Service-Adressen
 - 250** Reparaturkosten
 - 250** Versandanweisungen
 - 251** Gewährleistung bei Reparaturen
 - 251** Servicevereinbarungen
 - 252** Sicherheitsbestimmungen
 - 252** Funkschutz
- B**
- 253** **Benutzerspeicher und Stack**
 - 253** Verwalten des Speicherbereichs
 - 254** Zurücksetzen des Rechners
 - 255** Löschen des Speicherbereichs
 - 256** Status des Stack Lifts
 - 257** Desaktivierende Operationen
 - 257** Neutrale Operationen
 - 258** Status des LAST X Registers
- C**
- 259** **Näheres zum Lösen einer Gleichung**
 - 259** Arbeitsweise von SOLVE
 - 261** Interpretieren von Ergebnissen
 - 267** Wenn SOLVE keine Nullstelle finden kann
 - 272** Rundungsfehler und "Underflow"

D

273 Näheres zur Integration

273 Auswertung des Integrals

274 Mögliche Ursachen für unkorrekte Ergebnisse

279 Bedingungen für verlängerte Rechenzeiten

281 Meldungen

286 Funktionsindex

299 Sachindex

Teil 1

Allgemeine Anwendungsweise

Seite	14	1: Bedienungsgrundlagen
	35	2: Der automatische Speicherstack
	47	3: Datenspeicherung in Variablen
	54	4: Reellwertige Funktionen

1

Bedienungsgrundlagen

Die ersten Schritte

Ein- und Ausschalten des Rechners

Drücken Sie **[C]**, um den Rechner einzuschalten. Beachten Sie ON, was unterhalb der Taste aufgedruckt ist.

Um den Rechner wieder auszuschalten, ist **[OFF]** zu drücken. Dies bedeutet, Sie müssen zuerst die Umschalttaste drücken (**[■]**), und anschließend **[C]** (wobei OFF über der Taste aufgedruckt ist). Da der Rechner über eine *andauernde Datenspeicherung* verfügt, bleiben Ihre Daten auch nach dem Ausschalten erhalten.

Um den Batteriesatz zu schonen, schaltet sich der Rechner etwa 10 Minuten nach dem letzten Tastendruck automatisch ab.

Unter normalen Betriebsbedingungen halten die Batterien über ein Jahr. Wenn der "Schwache Batterien" Indikator (**[■]**) in der Anzeige erscheint, sollten Sie die Batterien so bald wie möglich ersetzen. Eine Anleitung dazu finden Sie in Anhang A.

Einstellen des Anzeigekontrasts

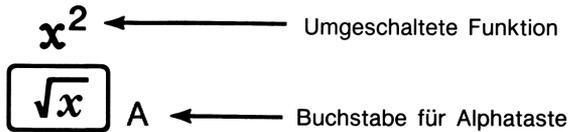
Um den Anzeigekontrast den örtlichen Lichtverhältnissen anzupassen, ist **[C]** gedrückt zu halten, während die Taste **[+]** oder **[-]** gedrückt wird.

Wichtige Tastenfeld- und Anzeigefunktionen

Umgeschaltete Tasten

Jede Taste hat 2 Funktionen: Die direkt auf der Tastenfläche angebrachte, und eine *umgeschaltete* Funktion (Zweitfunktion), welche in farbiger Schrift oberhalb der Taste aufgedruckt ist. Drücken Sie die farbige Umschalttaste (■), um eine Zweitfunktion auszuführen. Um z.B. den Rechner auszuschalten, ist zuerst ■ zu drücken und anschließend [C]. Dies wird auch als ■[OFF] dargestellt.

Das Drücken von ■ schaltet außerdem den Indikator  ein, welcher bis zum nächsten Tastendruck angezeigt bleibt. Um die Operation aufzuheben, drücken Sie einfach erneut ■.



Die Alphatasten

Die meisten der Tasten haben neben der Taste einen Buchstaben zugeordnet, wie oben abgebildet. Immer wenn Sie einen Buchstaben eintippen müssen (z.B. zur Benennung einer Variablen oder eines Labels), erscheint der **A..Z** Indikator in der Anzeige, um die Aktivierung der Alphatasten anzudeuten. (Variablen sind in Kapitel 3 behandelt.)

Löschen von Zeichen

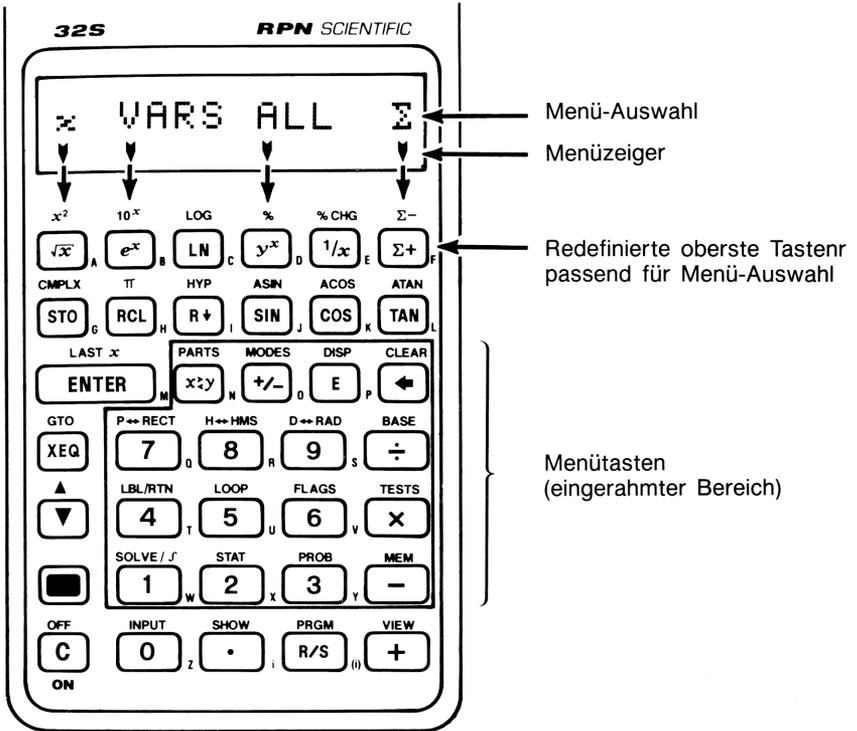
Einige der ersten Informationen, die Sie benötigen, beziehen sich auf das *Löschen*: Wie eine angezeigte Zahl korrigiert oder der Anzeigeinhalt gelöscht wird, und wie im allgemeinen mit der Eingabe neu begonnen wird.

Tasten für Löschoperationen

Taste	Beschreibung
◀	<p><i>Rückschritt-Taste.</i> Löscht das letzte Zeichen neben dem Cursor (⎵) oder verläßt das momentane Menü. Nach Abschluß einer Zahleneingabe (kein Cursor angezeigt) bewirkt ▶ das Löschen der ganzen Zahl. Außerdem werden dadurch Fehlermeldungen gelöscht.</p> <p>Während Programmeingabe: Löscht die Programmzeile.</p>
C	<p><i>Löschen oder Aufheben.</i> Setzt die angezeigte Zahl auf Null oder hebt die momentane Situation auf (wie z.B. Menü, Meldung, Eingabeaufforderung, Kataloganzeige oder Programmeingabe).</p>
■ CLEAR	<p><i>Das CLEAR Menü.</i> Bietet Ihnen die Löschoptionen: {\times}, {VARS}, {ALL} und {Σ}. Damit löschen Sie: die angezeigte Zahl, ("x" genannt), alle Variablen, den gesamten Speicherbereich und Statistikdaten.</p> <p>Während der Programmeingabe enthält das Menü auch die Funktion {PGM}, welche zur Löschung des gesamten Programmspeichers dient.</p>

Verwenden von Menüs

Der HP-32S enthält noch viel mehr Funktionen, als aus dem Tastenfeld ersichtlich ist. Dies beruht darauf, daß fast die Hälfte der Zweitfunktionen *Menütasten* darstellen, welche Ihnen (nachdem sie gedrückt wurden) verschiedene zusätzliche Funktionen—oder zusätzliche Optionen für weitere Funktionen—anbieten. Diese Art des Funktionsaufrufs ist einfacher und leichter zu finden, als wenn jede Funktion eine eigene Taste hätte.



Die Zweitfunktionen, welche auf einem helleren Hintergrund auf dem Rechner aufgedruckt sind (wie z.B. **CLEAR**) sind *Menütasten*. Das Drücken einer Menütaste bewirkt die Anzeige eines *Menüs*—Sie erhalten dadurch mehrere Optionen zur weiteren Vorgehensweise angezeigt.

HP-32S Menüs

Menü	Beschreibung	In Kapitel:
Numerische Funktionen		
PARTS	Zahlen ändernde Funktionen (ganzzahliger Anteil, Absolutbetrag, usw.).	4
P↔RECT	Konvertierung zwischen Koordinaten in Polar- und Rechtecksnotation.	4
H↔HMS	Konvertierungen zwischen Dezimalstunden und Stunden-Minuten-Sekunden (<i>Hours-Minutes-Seconds</i>).	4
D↔RAD	Konvertierungen zwischen Winkeln in Grad (Altgrad) und Bogenmaß (<i>Degrees-Radians</i>).	4
BASE	Konvertierung zwischen Zahlensystemen.	10
SOLVE/∫	Funktionen zur Bestimmung von Nullstellen und für Integration.	7, 8
STAT	Statistikfunktionen.	11
PROB	Wahrscheinlichkeitsfunktionen.	4
Programmanweisungen		
LBL/RTN	Label, Return (beenden) und Pause.	5
LOOP	Bedingte Schleifen und Zählerfunktionen.	6
FLAGS	Funktionen zum Setzen, Löschen und Testen von Flags.	6
TESTS	Vergleichstests.	6
Sonstige Funktionen		
MODES	Winkelmodi und Dezimalpunkt-Konvention.	4, 1
DISP	Anzeigeformate.	1
CLEAR	Funktionen zum Löschen von Daten.	1, 3, 5
MEM	Speicherstatus: Durch individuelle Variablen und Programme belegter Speicherplatz. Kataloge für Variablen und Programme	1

Berechnen Sie z.B. 25!:

Tastenfolge:	Anzeige	Beschreibung:
25	25_	Zeigt Zahl an.
 PROB	Cn,r Pn,r x! R	Zeigt Menü für Wahrscheinlichkeitsrechnung an.
{x!} (Taste )	1,5511E25	25! entspricht $1,5511 \times 10^{25}$.

Auf diese Weise helfen Ihnen Menüs bei der Ausführung mehrerer Funktionen—Sie werden mit Hilfe der Wahlmöglichkeiten zu den Funktionen “geführt” und müssen sich nicht sämtliche verfügbaren Funktionsnamen einprägen oder das Tastenfeld nach der benötigten Funktion absuchen.

Verlassen von Menüs

Nach jeder Ausführung einer Funktion über ein Menü wird das jeweilige Menü automatisch aufgehoben. Möchten Sie ein Menü verlassen, *ohne* eine Funktion auszuführen, so haben Sie 3 Möglichkeiten:

- Drücken von  **verläßt ein Menü**, Ebene für Ebene.

123	123_
 PROB	Cn,r Pn,r x! R
{R}	RANDOM SEED
	Cn,r Pn,r x! R
	123,0000

- Drücken von  **hebt das Menü auf**.

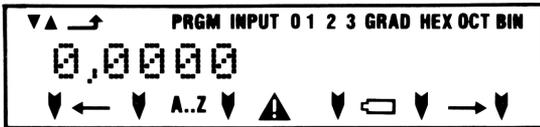
123	123_
 PROB	Cn,r Pn,r x! R
{R}	RANDOM SEED
	123,0000

■ **Drücken einer anderen Menütaste** ersetzt das seitherige Menü durch das neue Menü.

123	123_
■ PROB	Cn,r Pn,r x! R
{R}	RANDOM SEED
■ CLEAR	x VARS ALL Σ

Indikatoren

Die nachstehend abgebildeten Symbole werden als *Indikatoren* bezeichnet und kennzeichnen einen bestimmten Rechnerstatus.



Indikator	Bedeutung
▼▲	▼ and ■▲ sind aktiviert, um ein Programm oder eine Liste durchzusehen (Seiten 33, 76).
↶	Die Umschalttaste (■) wurde gedrückt (Seite 15).
PRGM	Programmeingabe ist aktiviert (Seite 72, 75).
INPUT	Programm erwartet eine Eingabe; geben Sie einen Wert ein und drücken Sie R/S , um das Programm fortzusetzen (Seite 77).
0 1 2 3	Kennzeichnet, welche Flags gesetzt sind (Seite 98).
RAD GRAD	Bogenmaß (Radiant) oder Grad (Neugrad) ist als Winkelmodus gesetzt (Seite 57).
HEX OCT BIN	Kennzeichnet vom Dezimalsystem abweichendes Zahlensystem (Seite 144).
▼	Der obersten Tastenreihe wurden neue Funktionen zugeordnet, entsprechend den Menüoptionen über den Menüzeigern (Seite 17).

Indikator	Bedeutung
 	Es gibt noch weitere Zeichen links oder rechts vom Anzeigerand. Verwenden Sie  , um den restlichen Teil einer Dezimalzahl anzuzeigen; verwenden Sie die Links-/Rechts-Verschiebetasten ( , ), um den Rest einer Binärzahl anzuzeigen (Seite 150).
A..Z	Die Alphatasten wurden aktiviert (Seite 48).
	Achtung! Kennzeichnet eine besondere Situation oder einen Fehler (Seite 21, 32).
	Schwache Batterien (Seite 242).

Eintippen von Zahlen

Sie können bis zu 12-stellige Zahlen mit einem Exponenten bis zu ± 499 eintippen. Wenn Sie versuchen, eine größere Zahl einzutippen, wird die Zifferneingabe abgebrochen und der  Indikator erscheint kurzzeitig.

Wenn Ihnen während der Zahleneingabe ein Tippfehler unterläuft, dann können Sie  zum Löschen der letzten Ziffer oder  zum Löschen der ganzen Zahl drücken.

Umkehren des Vorzeichens

Die  Taste ändert das Vorzeichen einer Zahl.

- Um eine negative Zahl einzutippen, tippen Sie zuerst die Zahl selbst ein und drücken dann .
- Um das Vorzeichen einer bereits eingegebenen Zahl zu ändern, drücken Sie einfach . (Wenn die Zahl einen Exponenten hat, so wirkt  nur auf die *Mantisse*).

Potenzen von 10

Angezeigte Exponenten. Zahlen mit 10-er Potenzen (wie z.B. $4,2 \times 10^{-5}$) werden mit einem E angezeigt, welches dem Exponenten vorangestellt ist (wie z.B. 4,2000E-5). Eine Zahl, deren Betrag zu groß oder zu klein für das Anzeigeformat ist, wird automatisch in exponentieller Form angezeigt. Beachten Sie z.B. nachstehende Tastenfolge, während das FIX 4 Anzeigeformat eingestellt ist:

Tastenfolge:	Anzeige	Beschreibung:
,000062	0,000062_	Zeigt die eingetippte Zahl an.
<input type="button" value="ENTER"/>	0,0001	Rundet die Zahl entsprechend dem Anzeigeformat.
,000042 <input type="button" value="ENTER"/>	4,2000E-5	Verwendet automatisch die wissenschaftliche Notation, da ansonsten keine signifikanten Stellen in der Anzeige erscheinen würden.

Eintippen von 10-er Potenzen. Verwenden Sie (*Exponent*), um Zahlen einzugeben, welche mit einer 10-er Potenz zu multiplizieren sind. Betrachten Sie z.B. die Plancksche Konstante $6,6262 \times 10^{-34}$.

1. Tippen Sie die *Mantisse* ein. Wenn dieser Teil negativ wäre, dann drücken Sie .

6,6262 6,6262_

2. Drücken Sie . Beachten Sie, daß der Cursor hinter E verschoben wird.

 6,6262E_

3. Tippen Sie den Exponenten ein (größter darstellbarer Exponent ist ± 499). Ist dieser negativ, so drücken Sie .

34 6,6262E-34_

Bei 10-er Potenzen ohne Mantisse, wie z.B. 10^{34} , drücken Sie einfach 34. Der Rechner zeigt daraufhin 1E34 an.

Weitere Potenzfunktionen. Drücken Sie \boxed{E} , um während der Zahleneingabe einen Exponenten von 10 zu *spezifizieren*. Drücken Sie $\boxed{10^x}$, um einen Exponenten von 10 zu *berechnen* (inverser Logarithmus zur Basis 10, siehe Kapitel 4). Um das Ergebnis einer *beliebigen* Basis zu berechnen, welche mit einem beliebigen Exponenten potenziert wird, ist $\boxed{y^x}$ zu verwenden (siehe Kapitel 4).

Einzelheiten zur Zifferneingabe

Während dem Eintippen einer Zahl ist der *Cursor* (–) in der Anzeige sichtbar. Der Cursor zeigt Ihnen, an welcher Stelle die nächste Ziffer erscheint—dies kennzeichnet jedoch gleichzeitig, daß die Zahl noch nicht vollständig ist. Aus technischer Sicht heißt dies, daß die *Zifferneingabe noch nicht abgeschlossen* ist.

Tastenfolge:	Anzeige	Beschreibung:
123	123_	Die Zifferneingabe ist noch nicht abgeschlossen—die Zahl ist noch nicht vollständig eingetippt.

Wenn Sie zur Berechnung eines *Ergebnisses* eine *Funktion ausführen*, dann verschwindet der Cursor, da die ganze Zahl eingegeben ist. Die Zifferneingabe wurde abgeschlossen.

$\boxed{\sqrt{x}}$	11,0905	Zifferneingabe abgeschlossen.
--------------------	---------	-------------------------------

Drücken von \boxed{ENTER} schließt ebenfalls die Zifferneingabe ab. Sie müssen zwei aufeinanderfolgende Zahlen durch Drücken von \boxed{ENTER} voneinander trennen, um die Zifferneingabe der ersten Zahl abzuschließen, bevor mit dem Eintippen der nächsten Zahl begonnen wird.

123 \boxed{ENTER}	123,0000	Vollständige Zahl.
5 $\boxed{+}$	128,0000	Eine weitere vollständige Zahl.

Ist die Zifferneingabe noch nicht abgeschlossen (der Cursor ist noch sichtbar), dann können Sie mit $\boxed{\leftarrow}$ das letzte Zeichen löschen. Wurde die Zifferneingabe bereits beendet (kein Cursor sichtbar), so wirkt $\boxed{\leftarrow}$ wie \boxed{C} und löscht die ganze Zahl. Probieren Sie es aus!

Wertebereich von Zahlen und OVERFLOW

Die kleinste im Rechner speicherbare Zahl ist 1×10^{-499} . Die größte im Rechner darstellbare Zahl ist $9,9999999999 \times 10^{499}$ (angezeigt als 1,0000E500, aufgrund des Rundungseffekts).

- Überschreitet eine Berechnung die größtmögliche Zahl, so wird stattdessen $9,9999999999 \times 10^{499}$ vorgegeben. Als Hinweis erscheint die Meldung OVERFLOW.
- Führt eine Berechnung zu einem Ergebnis, welches kleiner als die kleinstmögliche Zahl ist, so wird stattdessen Null vorgegeben. Eine Meldung wird nicht angezeigt.

Arithmetische Berechnungen

Wenn Sie eine Funktionstaste drücken, führt der Rechner unmittelbar danach die entsprechende Funktion aus. Es müssen daher alle Operanden vorher eingegeben sein, *bevor* Sie die Funktionstaste drücken.

Alle Berechnungen lassen sich entweder in einwertige oder zweiwertige Funktionen einteilen.

Einwertige Funktionen

Um eine einwertige Funktion (wie z.B. $\boxed{1/x}$, $\boxed{\sqrt{x}}$, $\blacksquare \boxed{x^2}$ oder $\boxed{+/-}$) zu verwenden:

1. Tippen Sie die Zahl ein. (Sie müssen nicht $\boxed{\text{ENTER}}$ drücken.)
2. Drücken Sie die Funktionstaste. (Bei Zweitfunktionen ist zuerst die Umschalttaste zu drücken.)

Berechnen Sie z.B. $1/32$ und $\sqrt{148,84}$. Quadrieren Sie danach das letzte Ergebnis und führen Sie einen Vorzeichenwechsel aus.

Tastenfolge:	Anzeige	Beschreibung:
32	32_	Operand.
$\boxed{1/x}$	0,0313	Kehrwert.
148,84 $\boxed{\sqrt{x}}$	12,2000	Quadratwurzel.



148,8400

Quadrat von 12,2.



-148,8400

Negation von 148,84.

Die einwertigen Funktionen schließen auch die trigonometrischen Funktionen, logarithmischen Funktionen, hyperbolischen Funktionen und die Funktionen für Teile von Zahlen mit ein (alle in Kapitel 4 beschrieben).

Zweiwertige Funktionen

Um eine zweiwertige Funktion (wie z.B. $+$, $-$, \times oder \div) auszuführen:

1. Tippen Sie die erste Zahl ein.
2. Drücken Sie [ENTER] zur Trennung der ersten Zahl von der zweiten.
3. Tippen Sie die zweite Zahl ein. (Drücken Sie nicht [ENTER] .)
4. Drücken Sie die Funktionstaste. (Bei einer Zweitfunktion ist zuerst die Umschalttaste zu drücken.)

Denken Sie daran, vor der Funktionsausführung beide Zahlen einzugeben.

Als Beispiel:

Aufgabe:	Tastensequenz:	Anzeige:
$12 + 3$	12 [ENTER] 3 $+$	15,0000
$12 - 3$	12 [ENTER] 3 $-$	9,0000
12×3	12 [ENTER] 3 \times	36,0000
$12 \div 3$	12 [ENTER] 3 \div	4,0000

Die Eingabereihenfolge ist natürlich entscheidend für nicht kommutative Funktionen wie $-$ und \div . Wenn die Zahlen in der falschen Reihenfolge eingegeben wurden, können Sie auch ohne erneute Eingabe das richtige Ergebnis erhalten—durch Drücken von [x↔y] , was den Austausch der 2 Zahlen bewirkt. Führen Sie anschließend die beabsichtigte Funktion aus. (Dies ist noch detaillierter in Kapitel 2 unter "Austauschen des X- und Y-Registerinhalts") beschrieben.

Kettenrechnungen

Die Geschwindigkeit und einfache Ausführung von Berechnungen mit dem HP-32S kommt deutlich bei der Durchführung von *Kettenrechnungen* (aufeinanderfolgende Berechnungen) zum Ausdruck. Selbst während der längsten Berechnung *arbeiten Sie nur mit einer oder mit zwei Zahlen je Rechenschritt*—der automatische Speicherstack sorgt für die Speicherung von Zwischenergebnissen bis zu deren Wiederverwendung und fügt sie dann in die Berechnung ein.*

- Dieses Verfahren ermöglicht eine kürzere Tastenfolge als bei der normalen Taschenrechnerlogik.
- Der Bearbeitungsprozeß eines Problems ist analog zum Bearbeiten des Problems auf Papier, allerdings übernimmt der Rechner den schwierigeren Teil.

Lösen Sie z.B. $(12 + 3) \times 7$.

Beginnen Sie mit dem Klammersausdruck. Würden Sie die Aufgabe mit Bleistift und Papier lösen, so würden Sie zuerst das Ergebnis von $(12 + 3)$ berechnen...

$$(12 + 3) \times 7 =$$

...und anschließend das Zwischenergebnis mit 7 multiplizieren.

$$15 \times 7 = 105$$

Lösen Sie das Problem auf die gleiche Weise mit dem HP-32S, indem Sie mit dem Klammersausdruck beginnen.

Tastenfolge:	Anzeige	Beschreibung:
12 ENTER 3 +	15,0000	Berechnet zuerst Zwischenergebnis.

Sie müssen nicht ENTER drücken, um fortfahren zu können. Da es sich um ein berechnetes Ergebnis handelt, wird es automatisch gesichert.

* Kümmern Sie sich hier nicht um den automatischen Speicherstack und dessen Funktionsweise. Der Stack ist in Kapitel 2 beschrieben.

7 $\boxed{\times}$

105,0000

Drücken einer Funktionstaste berechnet das Ergebnis, welches für weitere Berechnungen verwendet werden kann.

Beachten Sie nun folgende Beispiele. Sie werden bemerken, daß $\boxed{\text{ENTER}}$ nur zur Trennung von *nacheinander eingegebenen* Zahlen dient, wie am Anfang einer Berechnung. Nachfolgende Zahlen werden durch Operationen (wie z.B. $\boxed{+}$, $\boxed{-}$, usw.) voneinander getrennt, wobei auch Zwischenergebnisse gespeichert werden. Das zuletzt gespeicherte Ergebnis wird bei Bedarf für die Fortsetzung der Berechnung verwendet.

Berechnen Sie zuerst $\frac{2 + 3}{10}$:

Tastenfolge:**Anzeige****Beschreibung:**2 $\boxed{\text{ENTER}}$ 3 $\boxed{+}$ 10 $\boxed{\div}$

0,5000

(2 + 3) \div 10.

Berechnen Sie nun $\frac{2}{3 + 10}$:

Tastenfolge:**Anzeige****Beschreibung:**3 $\boxed{\text{ENTER}}$ 10 $\boxed{+}$

13,0000

Berechnet zuerst (2 + 10).

2 $\boxed{\text{x}\div\text{y}}$ $\boxed{\div}$

0,1538

Stellt 2 vor 13, womit die Division richtig wird: 2 \div 13.

Berechnen Sie $\frac{14 + 7 + 3 - 2}{4}$:

Tastenfolge:**Anzeige****Beschreibung**14 $\boxed{\text{ENTER}}$ 7 $\boxed{+}$ 3 $\boxed{+}$ 2 $\boxed{-}$

22,0000

Berechnet zuerst (14 + 7 + 3 - 2).

4 $\boxed{\div}$

5,5000

22 \div 4.

Berechnen Sie nun $\frac{4}{14 + (7 \times 3) - 2}$:

7 <input type="button" value="ENTER"/> 3 <input type="button" value="x"/>	21,0000	Berechnet (7 × 3).
14 <input type="button" value="+"/> 2 <input type="button" value="-"/>	33,0000	Berechnet Klammerausdruck als nächstes.
4 <input type="button" value="x÷y"/>	33,0000	Stellt 4 vor 33 für Division.
<input type="button" value="÷"/>	0,1212	Berechnet 4 ÷ 33, das Ergebnis.

Problemstellungen mit mehreren Klammerausdrücken lassen sich auf die gleiche einfache Weise lösen, indem die Zwischenergebnisse automatisch gespeichert werden. So würde $(3 + 4) \times (5 + 6)$ auf Papier wie folgt gelöst werden:

Zuerst Addition der Zahlen
in diesem Klammerausdruck...

$$(3 + 4) \times (5 + 6)$$

...danach Addition der Zahlen
in diesem Klammerausdruck...

...und schließlich Multiplikation der Zwischenergebnisse.

Sie bearbeiten das Problem mit dem HP-32S genauso, nur müssen Sie sich keine Zwischenergebnisse notieren—der Rechner übernimmt dies für Sie.

Tastensequenz:	Anzeige	Beschreibung:
3 <input type="button" value="ENTER"/> 4 <input type="button" value="+"/>	7,0000	Addiert zuerst (3 + 4).
5 <input type="button" value="ENTER"/> 6 <input type="button" value="+"/>	11,0000	Addiert dann (5 + 6).
<input type="button" value="x"/>	77,0000	Multipliziert Zwischenergebnisse und berechnet Endergebnis.

Beachten Sie: Dieses Verfahren der Zahleneingabe wird als "Umgekehrte Polnische Notation" bezeichnet:

- Sie bearbeiten nie mehr als 2 Zahlen je Rechenschritt.
- **[ENTER]** wird zum Trennen nacheinander eingetippter Zahlenwerte verwendet.
- Drücken einer Funktionstaste bewirkt die sofortige Ausführung der Funktion.
- Zwischenergebnisse werden nach der Berechnung angezeigt, was eine Überprüfung der einzelnen Rechenschritte ermöglicht.
- Zwischenergebnisse werden automatisch gespeichert; sie werden bei Bedarf automatisch zur nächsten Berechnung herangezogen, wobei der zuletzt gespeicherte Wert als erstes verwendet wird.
- Sie können das Problem in der gleichen Reihenfolge bearbeiten, wie Sie es mit Bleistift und Papier tun würden.

Übungsaufgaben

Berechne: $\frac{\sqrt{16,3805 \times 5}}{0,05} = 18,000$

Lösung: 16,3805 **[ENTER]** 5 **[x]** **[√x]** ,05 **[÷]**

Berechne: $\sqrt{[(2 + 3) \times (4 + 5)]} + \sqrt{[(6 + 7) \times (8 + 9)]} = 21,5743$

Lösung: 2 **[ENTER]** 3 **[+]** 4 **[ENTER]** 5 **[+]** **[x]** **[√x]** 6 **[ENTER]** 7 **[+]** 8 **[ENTER]** 9 **[+]** **[x]** **[√x]** **[+]**

Berechne: $(10 - 5) \div [(17 - 12) \times 4] = 0,2500$

Lösung: 17 **[ENTER]** 12 **[-]** 4 **[x]** 10 **[ENTER]** 5 **[-]** **[x]** **[÷]**

oder

10 **[ENTER]** 5 **[-]** 17 **[ENTER]** 12 **[-]** 4 **[x]** **[÷]**

Spezifizieren des Anzeigeformats

Punkt und Komma in Zahlen

Um das Dezimalzeichen mit dem Trennzeichen für Zifferngruppen in angezeigten Zahlen zu tauschen:

1. Drücken Sie **[MODES]** zur Anzeige des MODES Menüs.

2. Spezifizieren Sie das Dezimalzeichen durch Drücken von { . } oder { , }. Zum Beispiel kann eine Million wie folgt dargestellt werden:

- 1.000.000,0000 nach Drücken von { , } oder
- 1,000,000.0000 nach Drücken von { . }.

Anzahl der Dezimalstellen (DISP)

Alle Zahlen werden mit einer 12-stelligen Genauigkeit *gespeichert*,* Sie können jedoch die Anzahl der *anzuzeigenden* Dezimalstellen durch Drücken von  DISP verändern. Die anzuzeigende Zahl wird dabei entsprechend dem Anzeigeformat *gerundet*. Im DISP Menü haben Sie 4 Optionen:

FX SC EN ALL

Festkommaformat ({FX}). Im FIX Format werden Zahlen mit bis zu 11 Dezimalstellen (wenn passend) angezeigt. Auf die Eingabeaufforderung `FIX _` ist die Anzahl der anzuzeigenden Dezimalstellen einzugeben; für 10 oder 11 Stellen ist  0 oder  1 zu drücken.

Dezimalstellen

 123.456,0000

Jede Zahl, welche zu groß oder zu klein ist, um im momentanen Anzeigeformat angezeigt werden zu können, wird automatisch im wissenschaftlichen Format angezeigt.

Wissenschaftliches Anzeigeformat ({SC}). Im SCI Format wird eine Zahl in wissenschaftlicher Notation angezeigt (eine Vorkommate und bis zu 11 Nachkommastellen sowie 3-stelliger Exponent). Nach der Eingabeaufforderung `SCI _` ist die Anzahl der Dezimalstellen einzutippen; für 10 oder 11 Stellen ist  0 oder  1 zu drücken.

Dezimalstellen Potenz von 10

 1,2346E5
 Mantisse

* Während einiger komplexer internen Berechnungen verwendet der Rechner eine 15-stellige Genauigkeit für Zwischenergebnisse.

Technisches Anzeigeformat ({EN}). Im ENG Format werden Zahlen ähnlich zum wissenschaftlichen Format angezeigt, wobei der Exponent jedoch ein Vielfaches von 3 ist (weshalb die Mantisse bis zu 3 Vorkommastellen haben kann). Dieses Format ist sehr sinnvoll bei wissenschaftlichen oder technischen Berechnungen, welche Einheiten mit Vorsätzen wie z.B. Mikro, Milli oder Kilo verwenden.

Tippen Sie nach der Eingabeaufforderung ENG _ die Anzahl der anzuzeigenden Stellen *nach* der ersten signifikanten Stelle ein. (Der Vorkommateil ist immer kleiner als 1 000.) Für 10 oder 11 Stellen ist 0 oder 1 einzutippen.

Stellen nach erster signifikanter Stelle 10-er Potenz (vielfaches von 3)

123,46E3

 Mantisse

ALL Anzeigeformat ({ALL}). Im ALL Format wird eine Zahl so genau wie möglich angezeigt (max. 12 Stellen). Passen nicht alle Stellen in die Anzeige, so wird automatisch das wissenschaftliche Anzeigeformat verwendet.

123.456

Temporäre Anzeige der vollen Genauigkeit

Das Ändern der anzuzeigenden Dezimalstellen wirkt sich nur auf die Zahl in der Anzeige aus, die interne Darstellung der Zahl bleibt unverändert; eine Zahl wird intern immer mit 12 Stellen gespeichert.

14,8745632019

Sie sehen nur diese Stellen bei {FIX} 4... ...aber diese Stellen sind intern ebenfalls vorhanden.

Um temporär eine Zahl mit größtmöglicher Genauigkeit anzuzeigen, ist **[SHOW]** zu drücken. Sie erhalten damit *nur die Mantisse* (nicht den Exponenten) einer Zahl angezeigt, während Sie **[SHOW]** gedrückt halten.

Tastenfolge:	Anzeige	Beschreibung:
■ [DISP] {FX} 4		Spezifiziert 4 Dezimalstellen.
45 [ENTER] 1,3 [x]	58,5000	Zeigt 4 Dezimalstellen an.
■ [DISP] {SC} 2	5,85E1	Wissenschaftliches Format: 2 Dezimalstellen und Exponent.
■ [DISP] {ALL}	58,5	Anzeige aller signifikanten Stellen. Nachfolgende Nullen entfallen.
■ [DISP] {FX} 4	58,5000	4 Dezimalstellen, kein Exponent.
[1/x]	0,0171	
■ [SHOW] (gedrückt)	170940170940	Temporäre Anzeige der vollen Genauigkeit.

Meldungen

In bestimmten Situationen gibt der Rechner eine Meldung aus. Das **▲** Symbol erscheint dabei in der Anzeige, um Sie auf die Meldung aufmerksam zu machen.

- Um die Meldung zu löschen, ist **[C]** oder **[↵]** zu drücken.
- Um die Meldung zu löschen *und* eine andere Funktion auszuführen, ist eine beliebige andere Taste zu drücken.

Erscheint keine Meldung, sondern nur **▲**, dann haben Sie eine *inaktive Taste* gedrückt (wie z.B. **[3]** im Binär-Modus).

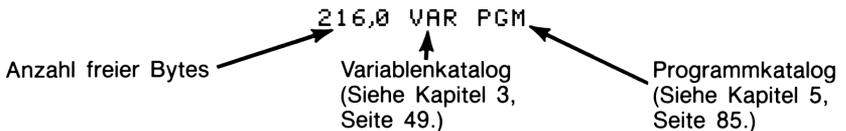
Alle angezeigten Meldungen sind auf Seite 281 erläutert.

Speicherbereich des Rechners

Der Rechner verfügt über 390 Bytes als Benutzerspeicher, in welchem Sie jede Art von Daten (Variablen oder Programmzeilen) speichern können. Die Speicheranforderungen für spezielle Anwendungsfälle sind unter "Verwalten des Speicherbereichs" in Anhang B erläutert.

Anzeigen des freien Speicherbereichs

Drücken von **MEM** zeigt Ihnen den freien Speicherbereich an:



1. Drücken Sie {VAR}, um den Variablenkatalog aufzurufen bzw. anzuzeigen; drücken Sie {PGM} zur Anzeige des Programmkatalogs.
2. Drücken Sie **▼** oder **▲**, um die Kataloge durchzusehen.
3. Um eine Variable oder ein Programm zu löschen, ist **CLEAR** zu drücken, während die Variable/das Programm im Katalog angezeigt ist.
4. Drücken Sie **C**, um den Katalog zu verlassen.

Löschen des gesamten Speicherbereichs

Löschen des gesamten Speicherbereichs bewirkt das Löschen aller Zahlenwerte und Programme, die Sie eingegeben haben. Einstellungen (für Modi und Formate) werden davon nicht berührt. (Um Einstellungen sowie Daten zu löschen, beziehen Sie sich auf "Löschen des Speicherbereichs" in Anhang B.)

Um den gesamten Speicherbereich zu löschen:

1. Drücken Sie  {ALL}. Der Rechner gibt daraufhin die Eingabeaufforderung CLR ALL? Y N aus, was als Schutzmaßnahme gegen das versehentliche Löschen des Speicherbereichs gedacht ist.
2. Drücken Sie {Y} (*yes* bzw. *ja*).

Der automatische Speicherstack

Dieses Kapitel erläutert den Ablauf der Berechnungen im automatischen Speicherstack und erklärt, weshalb hierbei die Eingabeoperationen bei komplexen Berechnungen auf ein Minimum reduziert werden. *Zum Arbeiten mit dem Rechner ist es nicht erforderlich, daß Sie dieses Thema durchlesen und verstanden haben.* Sie werden jedoch feststellen, daß nach dem Durchlesen und Verstehen dieses Kapitels Ihre Anwendungsmöglichkeiten des Rechners, besonders bei der Programmierung, wesentlich erweitert werden.

Im Teil 2, "Programmierung", werden Sie erkennen, wie der Stack bei der Manipulation und Organisation von Daten behilflich ist.

Was ist ein "Stack"

Automatisches Speichern von Zwischenergebnissen ist der Hauptgrund, weshalb der HP-32S die meisten komplexen Berechnungen auf einfache Weise und ohne Verwendung von Klammern ausführen kann. Der Schlüssel hierfür ist der *automatische UPN Speicherstack*.*

Der Speicherstack besteht aus vier Speicherplätzen, als *Register* bezeichnet, welche übereinander "gestapelt" sind; sie stellen den Arbeitsbereich für Berechnungen dar und sind mit den Buchstaben X, Y, Z und T bezeichnet. Der älteste Zahlenwert befindet sich dabei immer im T-(*top*) Register.

* Die HP Operationslogik basiert auf einem mathematischen System, das unter dem Namen "Polnische Notation" bekannt ist, welche von dem polnischen Logiker Jan Łukasiewicz (1878—1956) entwickelt wurde. Die herkömmliche mathematische Notation setzt die algebraischen Operatoren *zwischen* die relevanten Zahlen oder Variablen, während Łukasiewicz's Notation die Operatoren *vor* die Zahlen oder Variablen stellt. Für die optimale Ausnutzung beim Gebrauch des Stacks wurde die Notation von HP so modifiziert, daß die Operatoren *nach* der Angabe der Variablen erfolgt. Daher der Name *umgekehrte Polnische Notation*, oder *UPN*.

T	0,0000	"Älteste" Zahl
Z	0,0000	
Y	0,0000	
X	0,0000	Angezeigt

Die "neueste" Zahl befindet sich im X-Register: *Dies ist der angezeigte Zahlenwert.*

Beim Programmieren wird der Stack zur Ausführung von Berechnungen, temporärer Speicherung von Zwischenergebnissen, Weitergabe von Variablen zwischen Programmen und Ein-/Ausgabeoperationen verwendet.

Das X-Register. Das X-Register wird immer angezeigt, *außer* wenn ein Menü, eine Meldung oder eine Programmzeile angezeigt wird. Vielleicht haben Sie bemerkt, daß mehrere Funktionsnamen ein *x* oder *y* enthalten. Dies ist kein Zufall: Die Buchstaben verweisen auf das X- und Y-Register. So potenziert z.B. $\blacksquare \boxed{10^x}$ 10 mit der Zahl im X-Register (der angezeigten Zahl).

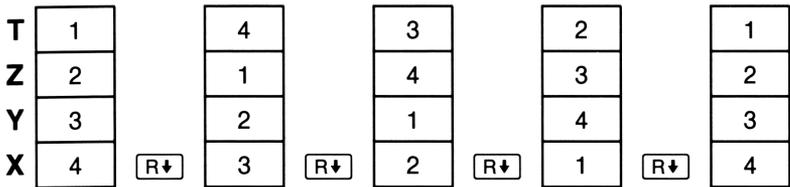
$\blacksquare \boxed{\text{CLEAR}} \{ \times \}$ **im Vergleich zu** $\boxed{\text{C}}$. Drücken von $\blacksquare \boxed{\text{CLEAR}} \{ \times \}$ setzt das X-Register *immer* auf Null, was auch als Programmanweisung gilt. Im Gegensatz dazu ist die Auswirkung von $\boxed{\text{C}}$ von der jeweiligen Situation abhängig. Sie wirkt nur dann wie $\blacksquare \boxed{\text{CLEAR}} \{ \times \}$, wenn das X-Register angezeigt ist.* Sie *löscht* sonstige Anzeigehalte: Menüs, benannte Ergebnisse und Programmeingaben.

Durchsehen des Stacks ($\boxed{\text{R}\downarrow}$)

$\boxed{\text{R}\downarrow}$ (*Rollen nach unten*) erlaubt Ihnen das Durchsehen des Stacks, indem der Inhalt Registerweise nach unten "gerollt" wird. Sie können jede Zahl sehen, nachdem Sie in das X-Register geschoben wird.

* $\boxed{\downarrow}$ wirkt auch wie $\blacksquare \boxed{\text{CLEAR}} \{ \times \}$ wenn das X-Register angezeigt ist *und* die Zifferneingabe abgeschlossen ist (kein Cursor sichtbar).

Nehmen Sie an, im Stack wäre 1, 2, 3 und 4 gespeichert (drücken Sie 1 **[ENTER]** 2 **[ENTER]** 3 **[ENTER]** 4). Viermaliges Drücken von **[R↓]** rollt die Zahlen zyklisch durch den Stack, bis sie wieder im ursprünglichen Register sind:



Was im X-Register gespeichert war, wird rotiert und kommt ins T-Register. Beachten Sie, daß der *Inhalt* der Register rotiert, nicht die Register selbst. Das X-Register bleibt immer in der Anzeige.

Austauschen des X- und Y-Registerinhalts (**[xzy]**)

Eine weitere Funktion zur Manipulation des Stackinhalts ist **[xzy]** (x mit y tauschen). Die Inhalte des X- und Y-Registers werden ausgetauscht, ohne den restlichen Stack zu beeinflussen. Zweimaliges Drücken von **[xzy]** stellt wieder den ursprünglichen Inhalt her.

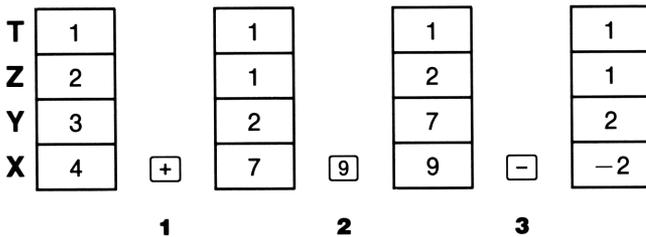
[xzy] dient hauptsächlich zwei Anwendungszwecken:

- Ansehen des Y-Registers und Rückkehr zum X-Register (zweimaliges Drücken von **[xzy]**). Manche Funktionen geben zwei Ergebnisse zurück: das erste im X-Register, das zweite im Y-Register. So konvertiert z.B. $\{ \nu, \times \rightarrow \theta, r \}$ Rechteckskordinaten im X- und Y-Register in Polarkordinaten im X- und Y-Register.
- Austauschen der Reihenfolge von Zahlen bei Berechnungen. Sie können z.B. einfach $9 \div (13 \times 8)$ berechnen, indem Sie 13 **[ENTER]** 8 **[x]** 9 **[xzy]** **[÷]** drücken.

Arithmetik—der Ablauf im Stack

Der Inhalt des Stacks wird nach oben oder unten geschoben, wenn neue Zahlen im X-Register eingegeben werden (*Stack Lift*) oder wenn 2 Zahlen durch einen Operator zu einem Ergebnis im X-Register konvertiert werden (*Stack Drop*). Nehmen Sie an, im Stack sind die Werte 1, 2, 3 und 4 gespeichert. Beachten Sie die Veränderungen im Stack, wenn diese Berechnung ausgeführt wird:

$$3 + 4 - 9:$$



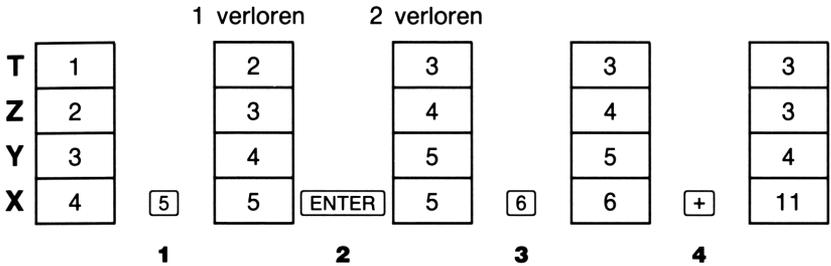
- 1** Der Stack schiebt den Inhalt nach unten (*Stack Drop*). Das oberste Register kopiert dabei seinen Inhalt.
- 2** Der Stack schiebt den Inhalt nach oben (*Stack Lift*). Der Inhalt des obersten Registers geht dabei verloren.
- 3** Stack Drop.

- Beachten Sie bei einer Verschiebung nach oben, daß der Inhalt des T-Registers verloren geht. Sie erkennen daraus, daß der Stackinhalt für Berechnungen auf vier Zahlenwerte eingeschränkt ist.
- Bei einer Verschiebung nach unten wird der Inhalt des T-Registers dupliziert.
- Aufgrund der automatischen Verschiebungen im Stack muß die Anzeige *nicht* gelöscht werden, wenn eine neue Berechnung begonnen wird.
- Die meisten Funktionen bereiten eine Verschiebung des Stackinhalts nach oben vor, *wenn die nächste Zahl* im X-Register eingegeben wird. Beziehen Sie sich auf Anhang B für eine Auflistung dieser Funktionen.

Funktionsweise von ENTER

Wie Sie bereits wissen, trennt **ENTER** beim Eingeben zwei aufeinanderfolgende Zahlen voneinander. Wie sieht hierbei der Ablauf im Stack aus? Nehmen Sie an, im Stack wäre 1, 2, 3 und 4 gespeichert. Addieren Sie nun die 2 Zahlen:

$$5 + 6:$$



- 1** Verschieben des Stacks nach oben.
- 2** Verschieben des Stacks nach oben und Duplizieren des X-Registers.
- 3** Keine Verschiebung des Stacks.
- 4** Verschieben des Stacks nach unten und Duplizieren des T-Registers.

ENTER dupliziert den Inhalt des X-Registers in das Y-Register. Die nächste eingegebene Zahl *überschreibt* die Kopie der ersten Zahl im X-Register. Der Effekt liegt darin, daß zwei nacheinander eingegebene Zahlen voneinander getrennt sind.

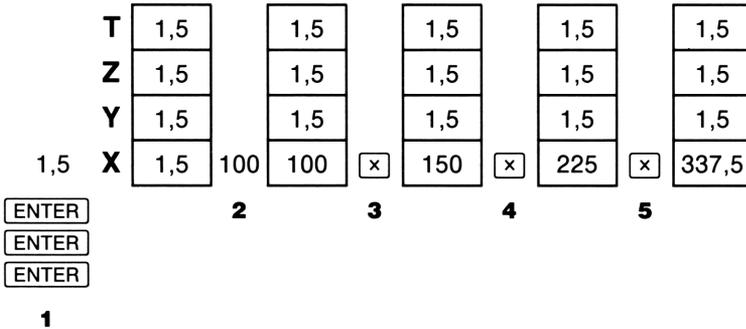
Sie können den Verdoppelungseffekt von **ENTER** auch zum Löschen des Stackinhalts verwenden: Drücken Sie 0 **ENTER** **ENTER** **ENTER**. Alle Register enthalten nun Null. Denken Sie jedoch daran, daß Sie den Stack nicht explizit löschen müssen, bevor Sie eine neue Berechnung beginnen.

Aufeinanderfolgende Verwendung einer Zahl. Der Verdoppelungseffekt von **ENTER** wirkt sich auch vorteilhaft bei anderen Anwendungen aus. Um eine Addition mit zwei gleichen Zahlen auszuführen, drücken Sie einfach **ENTER** **+**.

Auffüllen des Stacks mit einer Konstanten. Der Verdoppelungseffekt von **ENTER** und der von T in Z (bei Verschiebungen nach unten) erlaubt Ihnen das Auffüllen des Stacks mit einer numerischen Konstanten.

Beispiel: Konstantes, kumulatives Wachstum. Wie groß wäre eine Bakterienkultur nach 3 Tagen, wenn die tägliche Wachstumsrate bei 50% liegt und der Anfangsbestand 100 ist?

Dupliziert T-Register



- 1 Füllt den Stack mit der Wachstumsrate.
- 2 Eintippen des Anfangsbestands.
- 3 Berechnet die Population nach einem Tag.
- 4 Berechnet die Population nach zwei Tagen.
- 5 Berechnet die Population nach drei Tagen.

Funktionsweise von CLEAR x

Das Löschen der Anzeige (X-Register) schreibt eine Null in das X-Register. Die als nächstes eingegebene (oder zurückgerufene) Zahl überschreibt diese Null.

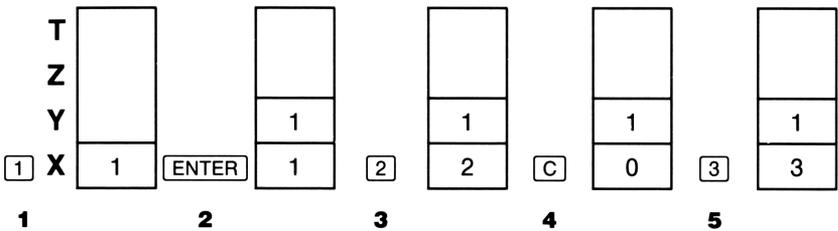
Es gibt drei Wege zum Löschen der Zahl im X-Register:

- Drücken Sie **C**.
- Drücken Sie **⇩**.
- Drücken Sie **■** **CLEAR** {**⌘**}. (Diese Tastenfolge wird hauptsächlich bei Programmeingaben benutzt.)

Beachten Sie diese Ausnahmen:

- Während der Programmeingabe löscht \blacktriangleleft Programmzeilen und $\square C$ hebt die Programmeingabe auf.
- Während der Zifferneingabe bewirkt \blacktriangleleft das Löschen der zuletzt eingegebenen Ziffer.
- Enthält die Anzeige eine *benannte* Zahl (wie z.B. $A=2,0000$), so hebt $\square C$ oder \blacktriangleleft die Anzeige auf und zeigt den Inhalt des X-Registers.

Wenn Sie z.B. 1 und 3 eingeben wollen und versehentlich 1 und 2 eingeben, so wären diese Schritte zur Korrektur erforderlich:



- 1 Hebt den Stack an.
- 2 Hebt den Stack an und dupliziert den Inhalt des X-Registers.
- 3 Überschreibt Inhalt des X-Registers.
- 4 Löscht x durch Überschreiben mit Null.
- 5 Überschreibt x (ersetzt Null).

Das LAST X Register

Das LAST X Register ist eine Ergänzung zum Stack: Es enthält den Zahlenwert, welcher sich im X-Register befand, bevor die letzte numerische Funktion ausgeführt wurde. Drücken von \blacksquare $\square LASTx$ ruft den Inhalt von LAST X in das X-Register zurück. Diese Funktion dient hauptsächlich zwei Zwecken: Fehlerkorrektur und Wiederverwenden eines Zahlenwertes für nachfolgende Rechenschritte.

Beziehen Sie sich auf Anhang B für eine detaillierte Auflistung aller Funktionen, die x im LAST X Register sichern.

Fehlerkorrektur mit $\boxed{\text{LASTx}}$

Falsche einwertige Funktion. Wenn Sie die falsche einwertige Funktion ausgeführt haben, dann verwenden Sie $\boxed{\text{LASTx}}$ zum Zurückrufen der Zahl und anschließenden Ausführung der richtigen Funktion.

Da $\boxed{\%}$ und $\boxed{\%CHG}$ keine Verschiebung nach unten verursachen, können Sie nach diesen Funktionen genauso weiterarbeiten wie bei einwertigen Funktionen.

Beispiel: Angenommen, Sie hätten gerade $\ln 4,7839 \times (3,879 \times 10^5)$ berechnet und wollten davon die Quadratwurzel ziehen, hätten jedoch aus Versehen $\boxed{e^x}$ gedrückt. Sie müssen deshalb nicht von vorne beginnen! Um das gewünschte Ergebnis zu erhalten, ist lediglich $\boxed{\text{LASTx}}$ $\boxed{\sqrt{x}}$ zu drücken.

Tipfehler bei zweiwertigen Funktionen. Wenn Ihnen bei einer zweiwertigen Operation ($\boxed{+}$, $\boxed{-}$, $\boxed{\times}$, $\boxed{+}$ oder $\boxed{y^x}$) ein Fehler unterläuft, so können Sie diesen durch $\boxed{\text{LASTx}}$ und die Umkehrfunktion der zweiwertigen Funktion ($\boxed{-}$ oder $\boxed{+}$, $\boxed{+}$ oder $\boxed{\times}$, $\boxed{1/x}$ oder $\boxed{y^x}$) korrigieren:

1. Drücken Sie $\boxed{\text{LASTx}}$ zur Rücksicherung der zweiten Zahl (x -Wert vor der Operation).
2. Führen Sie die Umkehrfunktion aus. Damit erhalten Sie die ursprüngliche erste Zahl, die zweite befindet sich noch immer im LAST X Register. Anschließend:
 - Wenn Sie die *falsche Funktion* verwendet hatten, drücken Sie erneut $\boxed{\text{LASTx}}$, um den ursprünglichen Stackinhalt wieder herzustellen. Führen Sie nun die richtige Funktion aus.
 - Wenn Sie den *falschen 2. Operanden* benutzt hatten, tippen Sie den richtigen ein und führen danach die Operation aus.

Hatten Sie den *falschen ersten Operanden* benutzt, so ist der richtige einzutippen, $\boxed{\text{LASTx}}$ zur Rücksicherung des zweiten Operanden zu drücken, und anschließend die Funktion erneut auszuführen.

Beispiel: Nehmen Sie an, bei der Berechnung von

$$16 \times 19 = 304$$

wäre Ihnen ein Fehler unterlaufen (wobei es 3 Fehlerquellen gibt):

Falsche Berechnung	Fehler	Korrektur
16 <input type="text" value="ENTER"/> 19 <input type="text" value="-"/>	Falsche Funktion.	<input type="text" value="LASTx"/> <input type="text" value="+"/> <input type="text" value="LASTx"/> <input type="text" value="x"/>
15 <input type="text" value="ENTER"/> 19 <input type="text" value="x"/>	Falsche 1. Zahl.	16 <input type="text" value="LASTx"/> <input type="text" value="x"/>
16 <input type="text" value="ENTER"/> 18 <input type="text" value="x"/>	Falsche 2. Zahl.	<input type="text" value="LASTx"/> <input type="text" value="÷"/> 19 <input type="text" value="x"/>

Wiederverwenden von Zahlen mit

Sie können über z.B. eine Konstante zurückrufen und diese für eine Berechnung heranziehen. Denken Sie daran, die Konstante kurz vor der Ausführung der Operation einzugeben, damit sie als letzte Zahl in das X-Register kommt und demzufolge im LAST X Register gesichert wird und mit zurückgerufen werden kann.

Beispiel: Berechnen Sie $\frac{96,704 + 52,3947}{52,3947}$:

	T	t	t	t
	Z	z	z	t
96,704	Y	96,704	96,704	z
<input type="text" value="ENTER"/>	X	96,704	52,3947	<input type="text" value="+"/> 149,0987
LAST X	l	l	l	52,3947

	T	t	t	
	Z	z	t	
	Y	149,0987	z	
<input type="text" value="LASTx"/>	X	52,3947	<input type="text" value="÷"/> 2,8457	
LAST X	52,3947	52,3947		

Tastenfolge:	Anzeige	Beschreibung:
96,704 [ENTER]	96,7040	
52,3947 [+]	149,0987	Zwischenergebnis.
■ [LASTx]	52,3947	Zeigt wieder Wert vor Ausführung von [+] an.
[÷]	2,8457	Endergebnis.

Beispiel: Zwei Nachbarsterne der Erde sind Rigel Centaurus (4,3 Lichtjahre entfernt) und Sirius (8,7 Lichtjahre entfernt). Berechnen Sie unter Verwendung der Lichtgeschwindigkeit c ($9,5 \times 10^{12}$ km/Jahr) die Entfernung zu diesen Sternen (in km).

$$\begin{aligned} \text{Entfernung zu Rigel Centaurus} &= 4,3 \text{ Jahre} \times (9,5 \times 10^{12} \text{ km/Jahr}) \\ \text{zu Sirius} &= 8,7 \text{ Jahre} \times (9,5 \times 10^{12} \text{ km/Jahr}) \end{aligned}$$

Tastenfolge:	Anzeige	Beschreibung:
4,3 [ENTER]	4,3000	Lichtjahre zu Rigel Centaurus.
9,5 [E] 12	9,5E12_	Lichtgeschwindigkeit c .
[x]	4,085E13	Entfernung zu Rigel Centaurus.
8,7 ■ [LASTx]	9,5000E12	Ruft c zurück.
[x]	8,2650E13	Entfernung zu Sirius.

Kettenrechnungen

Das automatische Verschieben des Stackinhalts nach oben oder unten hilft Ihnen bei der Sicherung von Zwischenergebnissen, ohne diese explizit eingeben oder Klammern verwenden zu müssen. Dies ist ein weiterer Vorteil des UPN Stacks gegenüber anderen Datenverwaltungsmethoden.

Bearbeitungsfolge in Aufgabenstellungen

In Kapitel 1 wurde empfohlen, Kettenrechnungen vom innersten Klammersymbol nach außen abzuarbeiten. Sie können aber auch das Problem von links nach rechts angehen und in dieser Reihenfolge bearbeiten.

So wurde z.B. in Kapitel 1 der Ausdruck

$$4 \div [14 + (7 \times 3) - 2]$$

berechnet, indem mit der innersten Klammer (7×3) begonnen wurde und nach außen weitergerechnet wurde—wie Sie es auf Papier tun würden:

$$7 \text{ [ENTER] } 3 \text{ [x] } 14 \text{ [+] } 2 \text{ [-] } 4 \text{ [xzy] [+]}.$$

Bei der links-rechts Vorgehensweise wäre die Lösung wie folgt:

$$4 \text{ [ENTER] } 14 \text{ [ENTER] } 7 \text{ [ENTER] } 3 \text{ [x] [+] } 2 \text{ [-] [+]},$$

wobei ein zusätzlicher Tastendruck erforderlich ist. Beachten Sie, daß das 1. Zwischenergebnis immer noch die innerste Klammer ist: (7×3) . Der Vorteil bei der links-rechts Vorgehensweise liegt darin, daß **[xzy]** nicht verwendet werden muß, um Operanden von nicht kommutativen Funktionen neu zu ordnen.

Das erste Verfahren (von innen nach außen) wird häufig bevorzugt, da:

- Eine kürzere Tastenfolge erforderlich ist.
- Weniger Stackregister benötigt werden.

Bei der links-rechts Vorgehensweise ist sicherzustellen, daß nicht mehr als 4 Zwischenergebnisse für eine Berechnung vorkommen, da der Stack nicht mehr als 4 Zahlenwerte gleichzeitig speichern kann. Im nächsten Beispiel, welches von links nach rechts bearbeitet wird, werden zu einem bestimmten Zeitpunkt alle Register benötigt.

$$4 \div [14 + (7 \times 3) - 2]$$

Tastenfolge:	Anzeige	Beschreibung:
4 [ENTER]		Speichert 4 und 14 als Zwischenwerte im Stack.
14 [ENTER]	14,0000	
7 [ENTER] 3	3_	Zu diesem Zeitpunkt ist der Stack vollständig mit Werten für diese Berechnung belegt.
[x]	21,0000	Zwischenergebnis.
[+]	35,0000	Zwischenergebnis.
2 [-]	33,0000	Zwischenergebnis.
[÷]	0,1212	Endergebnis.

Übungsaufgaben

Nachstehend finden Sie einige Aufgaben zur Gewöhnung an die Anwendung von UPN.

Berechne: $(14 + 12) \times (18 - 12) \div (9 - 7) = 78,0000$

Eine Lösung: 14 [ENTER] 12 [+] 18 [ENTER] 12 [-] [x] 9 [ENTER] 7 [-] [÷]

Berechne: $23^2 - (13 \times 9) + \frac{1}{7} = 412,1429$

Eine Lösung: 23 [x²] 13 [ENTER] 9 [x] [-] 7 [1/x] [+]

Berechne: $\sqrt{(5,4 \times 0,8) \div (12,5 - 0,7^3)} = 0,5961$

Eine Lösung: 5,4 [ENTER] ,8 [x] ,7 [ENTER] 3 [y^x] 12,5 [x_{zy}] [-] [÷] [√x]

oder

5,4 [ENTER] ,8 [x] 12,5 [ENTER] ,7 [ENTER] 3 [y^x] [-] [÷] [√x]

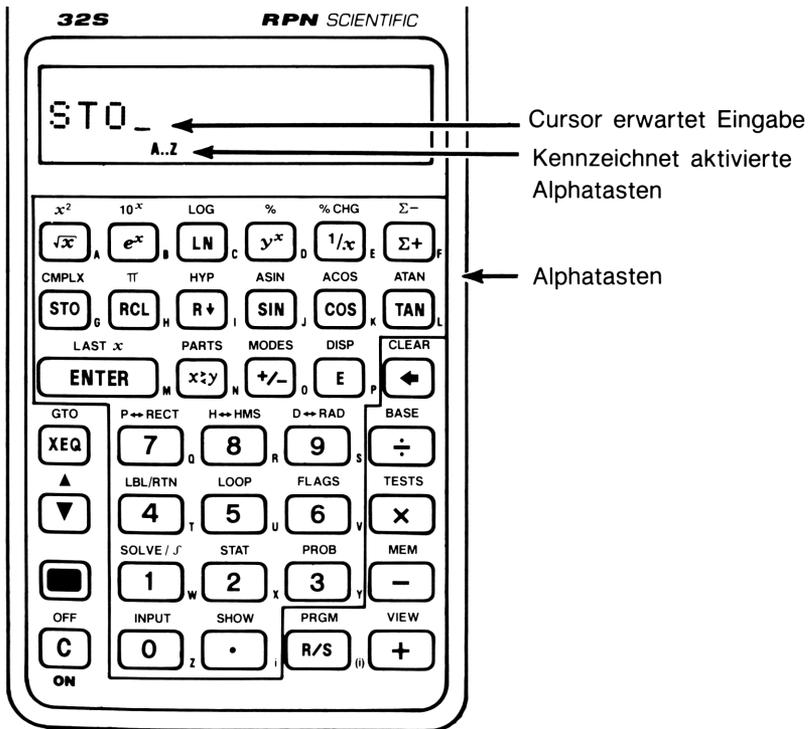
Berechne: $\sqrt{\frac{8,33 \times (4 - 5,2) \div [(8,33 - 7,46) \times 0,32]}{4,3 \times (3,15 - 2,75) - (1,71 \times 2,01)}} = 4.5728$

Eine Lösung: 4 [ENTER] 5,2 [-] 8,33 [x] [LASTx] 7,46 [-] 0,32 [x] [÷] 3,15 [ENTER] 2,75 [-] 4,3 [x] 1,71 [ENTER] 2,01 [x] [-] [÷] [√x]

3

Datenspeicherung in Variablen

Der HP-32S verfügt über 390 Bytes als *Benutzerspeicher*: Speicherplatz für Zahlen und Programme. Zahlen werden in *Variablen* gespeichert, wobei jede mit einem Buchstaben zwischen A und Z benannt wird. (Sie können den Buchstaben wählen, um eine Verbindung zum gespeicherten Inhalt herzustellen, wie z.B. G für *Fallbeschleunigung* oder C für *Lichtgeschwindigkeit*.)*



* Beachten Sie, daß die Variablen X, Y, Z und T *unterschiedliche* Speicherplätze wie das X-, Y-, Z- und T-Register im Stack darstellen.

Jeder weiße Buchstabe ist einer eindeutigen Variablen und Taste zugeordnet. Bei Bedarf werden die Alphatasten automatisch aktiviert (wobei der **A..Z** Indikator zur Kontrolle in der Anzeige erscheint).

Speichern und Zurückrufen von Zahlen

Zahlenwerte werden durch **[STO]** (*STOre*) in benannten Variablen gespeichert und über **[RCL]** (*ReCall*) aus diesen Variablen wieder zurückgerufen.

Um eine Kopie der angezeigten Zahl (X-Register) in einer Variablen zu speichern: Drücken Sie **[STO]** *Alphataste*.

Um eine Kopie eines Variablenwertes in der Anzeige zu erzeugen: Drücken Sie **[RCL]** *Alphataste*.

Beispiel: Speichern von Zahlen. Speichern Sie die Avogadro-Konstante (etwa $6,0225 \times 10^{23}$) in *A*.

Tastenfolge:	Anzeige	Beschreibung:
6,0225 [E] 23	6,0225E23_	
[STO]	STO _	Eingabeaufforderung für Variable.
A ([√x] Taste)	STO A	Zeigt Funktion während der gedrückten Taste an.
	6,0225E23	Speichert Avogadro-Konstante in <i>A</i> ; beendet gleichzeitig Zifferneingabe.
[C]	0,0000	Löscht Anzeige.
[RCL]	RCL _	Eingabeaufforderung für Variablenname.
A	6,0225E23	Ruft Avogadro-Konstante aus <i>A</i> in Anzeige.

Ansehen des Variableninhalts ohne Zurückrufen in die

Anzeige. Die **VIEW** Funktion zeigt den Variableninhalt an, ohne eine Kopie im X-Register zu erzeugen. Der Variablenname erscheint dabei mit in der Anzeige, wie z.B.:

A=1234,5678

VIEW wird sehr häufig beim Programmieren verwendet, ist aber auch immer dann hilfreich, wenn Sie den Variableninhalt ansehen möchten, ohne den Stackinhalt zu verändern.

Um die VIEW Anzeige aufzuheben, drücken Sie einmal **↓** oder **C**.

Ansehen von Variablen im VAR Katalog

Die **MEM** (*MEMory*) Funktion bietet Informationen über die Speicherbelegung:

Anzahl freier Bytes im Speicher



Um den Inhalt aller Variablen (oder aller ungleich Null) anzusehen:

1. Drücken Sie **MEM** {VAR}.
2. Drücken Sie **↓** oder **▲** zum Verschieben des Listeninhalts bzw. zur Anzeige der gewünschten Variablen. (Beachten Sie den **▼▲** Indikator, der auf die aktivierten Pfeiltasten hinweist.)

Um alle signifikanten Stellen einer im {VAR} Katalog gelisteten Zahl anzuzeigen, ist **SHOW** zu drücken. (Bei einer Dualzahl mit mehr als 12 Stellen können mit **√x** und **Σ+** die restlichen Stellen angesehen werden.)

3. Um den Inhalt einer angezeigten Variablen in das X-Register zu kopieren, ist **ENTER** zu drücken.
4. Um den Variableninhalt auf Null zu setzen, ist **CLEAR** zu drücken, während die Variable im Katalog angezeigt ist.
5. Drücken Sie **C** zum Verlassen des Katalogs (oder **↓** zum Übergang auf die nächsthöhere Menüebene).

Löschen von Variablen

Der Inhalt einer Variablen bleibt aufgrund der andauernden Datenspeicherung so lange erhalten, bis er ersetzt oder die Variable gelöscht wird. *Löschen* einer Variablen bedeutet das Speichern von Null als Inhalt (wobei kein Speicherplatz benötigt wird).

Um eine einzelne Variable zu löschen: Speichern Sie Null in ihr.

Um ausgewählte Variablen zu löschen:

1. Drücken Sie   {VAR} und verwenden Sie  oder   zur Anzeige der Variablen.
2. Drücken Sie  .
3. Drücken Sie  zum Verlassen des Katalogs, oder  zum Übergang auf die nächsthöhere Ebene.

Um alle Variablen gleichzeitig zu löschen: Drücken Sie   {VARS}.

Arithmetik mit Variablen

Speicherarithmetik und *Rückrufarithmetik* ermöglichen Ihnen die Ausführung von Berechnungen mit Variablen, *ohne den Variableninhalt in den Stack zu kopieren*. Für die Berechnung wird eine Zahl aus dem X-Register und ein Wert aus einer Variablen verwendet.

Speicherarithmetik

Bei der *Speicherarithmetik* wird , ,  oder  zur Ausführung der Berechnung in der Variablen selbst und zum Speichern des Ergebnisses in ihr verwendet; dabei wird der Wert im X-Register benutzt, wobei der Stack nicht verändert wird.

$$\text{Neuer Variableninhalt} = \text{seitheriger Inhalt} \{+, -, \times, \div\} x$$

Nehmen Sie z.B. an, Sie möchten den Inhalt von A (15) um die Zahl im X-Register (3, angezeigt) reduzieren. Drücken Sie  A. Sie erhalten $A = 12$, während 3 noch immer angezeigt wird.

A 15

A 12

Ergebnis: $15 - 3$,
das heißt, $A - x$.

T t
Z z
Y y
X 3

STO - A

T t
Z z
Y y
X 3

Rückrufarithmetik

Die *Rückrufarithmetik* verwendet $\boxed{\text{RCL}} \boxed{+}$, $\boxed{\text{RCL}} \boxed{-}$, $\boxed{\text{RCL}} \boxed{\times}$ oder $\boxed{\text{RCL}} \boxed{\div}$ für Berechnungen im X-Register unter Verwendung eines zurückgerufenen Variableninhalts, wobei das Ergebnis in der Anzeige bleibt. Es ist nur das X-Register betroffen, der restliche Stackinhalt bleibt unverändert.

Neues $x =$ seitheriges $x \{+, -, \times, \div\}$ Variable

Nehmen Sie z.B. an, Sie möchten die Zahl im X-Register (3, angezeigt) durch den Wert in A (12) dividieren. Drücken Sie $\boxed{\text{RCL}} \boxed{\div}$ A. Sie erhalten $x = 0,25$, während 12 unverändert in A gespeichert ist.*

A 12

A 12

T t
Z z
Y y
X 3

RCL \div A

T t
Z z
Y y
X 0,25

Ergebnis: $3 \div 12$,
das heißt, $x \div A$.

* Rückrufarithmetik hilft in Programmen zur Einsparung von Speicherplatz. Die Verwendung von $\boxed{\text{RCL}} \boxed{\div}$ A (eine Anweisung) benötigt nur die Hälfte des Speicherbedarfs von $\boxed{\text{RCL}} \boxed{A}$, $\boxed{\div}$ (zwei Anweisungen).

Weitere Beispiele. Nehmen Sie an, die Variablen D , E und F enthalten die Werte 1, 2 und 3. Verwenden Sie Speicherarithmetik zur Addition von 1 zu jeder der Variablen.

Tastenfolge:	Anzeige	Beschreibung:
1 STO D		Speichert die angenommenen Werte in den Variablen.
2 STO E		
3 STO F	3,0000	
1 STO + D STO + E STO + F	1,0000	Addiert 1 zu D , E und F .
VIEW D	D=2,0000	Zeigt den momentanen Inhalt von D an.
VIEW E	E=3,0000	
VIEW F	F=4,0000	
◀	1,0000	Löscht die VIEW Anzeige, zeigt X-Register an.

Nehmen Sie an, die Variablen D , E und F enthalten die Werte 2, 3 und 4 vom letzten Beispiel. Dividieren Sie 3 durch D , multiplizieren Sie das Ergebnis mit E und addieren Sie F .

Tastenfolge:	Anzeige	Beschreibung:
3 RCL ÷ D	1,5000	Berechnet $3 \div D$.
RCL x E	4,5000	$3 \div D \times E$.
RCL + F	8,5000	$3 \div D \times E + F$.

Die Variable “i”

Es gibt eine 27. Variable—die Variable *i*. (Das **i** befindet sich neben der Taste **.**.) Neben ihrer normalen Verwendung (Speichern von Zahlen) hat diese Variable noch einen speziellen Verwendungszweck: Über die **(i)** Funktion kann sie auf *andere Variablen verweisen*—eine Technik, die als *indirekte Adressierung* bezeichnet wird. Da es sich hierbei um eine Programmierungstechnik handelt, ist diese Variable in Kapitel 6 unter “Indirektes Adressieren von Variablen und Labels” behandelt.

4

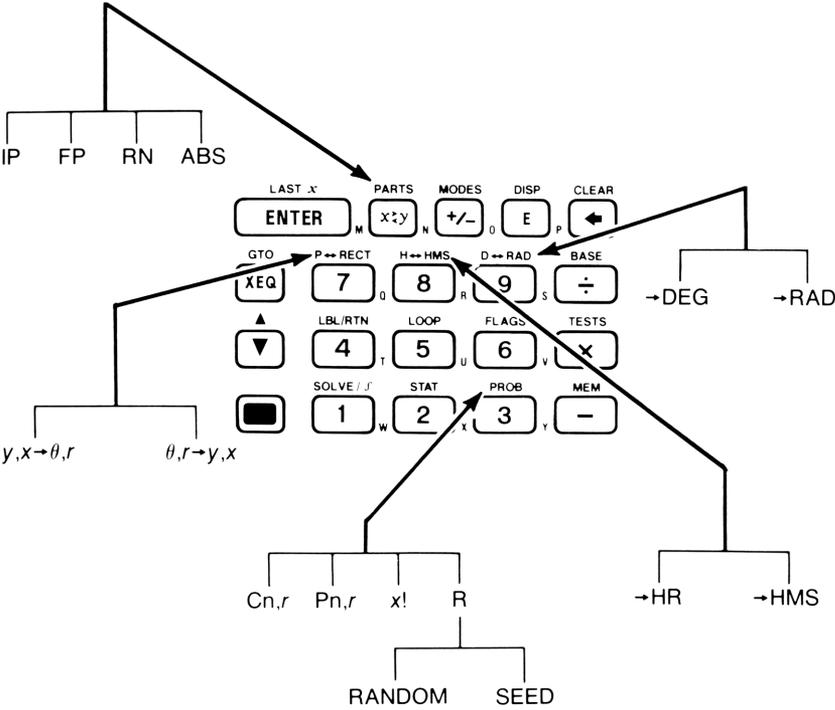
Reellwertige Funktionen

Dieses Kapitel behandelt die meisten Rechnerfunktionen, welche sich auf Berechnungen mit reellen Zahlen beziehen (einschließlich einigen numerischen Funktionen im Zusammenhang mit Programmen, wie z.B. die Absolutbetragsfunktion):

- Exponential- und Logarithmusfunktionen.
- Trigonometrische Funktionen.
- Hyperbolische Funktionen.
- Prozentrechnungsfunktionen.
- Konvertierungsfunktionen für Koordinaten, Winkel und Bruchteile.
- Wahrscheinlichkeitsfunktionen.
- Teile von Zahlen (Manipulationsfunktionen).

Arithmetische Funktionen und Berechnungen sind in Kapitel 1 und 2 behandelt. Die höheren numerischen Operationen (Nullstellenbestimmung, Integration, komplexe Zahlen, Konvertierung zwischen Zahlensystemen und Statistikberechnungen sind in Teil 3 dieses Handbuchs behandelt.

Viele der numerischen Funktionen sind über die Tasten der obersten zwei Tastenreihen zugänglich. Die restlichen Funktionen sind in einem dieser Menüs enthalten:



Exponential- und Logarithmusfunktionen

Tippen Sie zuerst die Zahl ein und führen Sie danach die Funktion aus. Es ist nicht erforderlich, **ENTER** zu drücken.

Auszuführende Funktion:	Tastensequenz:
Natürlicher Logarithmus (Basis e)	LN
Dekadischer Logarithmus (Basis 10)	LOG
Natürliche Exponentialfunktion	e^x
Dekadische Exponentialfunktion	10^x

Potenzfunktion (y^x)

Um eine Zahl y mit einem Exponenten x zu potenzieren, ist y **ENTER** x **y^x** einzutippen.

Für $y > 0$ kann x eine beliebige rationale Zahl darstellen. Für $y < 0$ muß x eine ganze Zahl sein. Für $y = 0$ muß x ein positives Vorzeichen annehmen.

Als Beispiel:

Aufgabenstellung:	Eingabefolge:	Ergebnis:
15^2	15 x²	225,0000
$2^{-1,4}$	2 ENTER 1,4 +/- y^x	0,3789
$(-1,4)^3$	1,4 +/- ENTER 3 y^x	-2,7440
$\sqrt[3]{2}$ oder $2^{1/3}$	2 ENTER 3 1/x y^x	1,2599

Trigonometrie

Eingeben von π

Drücken Sie **π** zur Eingabe der ersten 12 Stellen von π in das X-Register. (Die angezeigte Zahl hängt vom Anzeigeformat ab.) Da es sich hier um eine Funktion handelt, muß π nicht durch Drücken von **ENTER** von einer anderen Zahl getrennt werden.

Beachten Sie, daß der Rechner π nicht genau darstellen kann, da π eine irrationale Zahl ist.

Einstellen des Winkelmodus

Der Winkelmodus spezifiziert, in welcher Winkeleinheit die Zahlenwerte bei trigonometrischen Berechnungen interpretiert werden sollen. Der Modus allein konvertiert *noch nicht* bereits gespeicherte Zahlen (siehe "Konvertierungsfunktionen" später in diesem Kapitel).

$$360 \text{ Altgrad} = 2\pi \text{ Radiant} = 400 \text{ Neugrad}$$

Um den Winkelmodus zu spezifizieren, drücken Sie . Wählen Sie danach eine Option aus.

Option	Beschreibung	Indikator
{DG}	Stellt Grad-Modus (DEG) ein; verwendet Dezimaldarstellung, nicht Minuten und Sekunden.	keiner
{RD}	Stellt Bogenmaß (RAD) ein.	RAD
{GR}	Stellt Neugrad-Modus (GRAD) ein.	GRAD

Trigonometrische Funktionen

Während x angezeigt ist:

Auszuführende Berechnung:	Tastenfolge:
Sinus x .	
Cosinus x .	
Tangens x .	
Arcussinus x .	
Arcuscosinus x .	
Arcustangens x .	



Hinweis

Berechnungen mit der irrationalen Zahl π können bei der internen 12-stelligen Genauigkeit nicht *exakt* dargestellt werden. Dies ist insbesondere bei trigonometrischen Berechnungen bemerkbar; der berechnete Sinus π ist z.B. nicht genau Null, sondern $-2,0676 \times 10^{-13}$, eine sehr kleine Zahl nahe Null.

Beispiel: Weisen Sie nach, daß der Cosinus von $(\frac{5}{7})\pi$ und der Cosinus von $128,57^\circ$ identisch sind.*

Tastenfolge:	Anzeige	Beschreibung:
■ MODES {RD}		Stellt Bogenmaß als Winkelmodus ein; RAD Indikator wird angezeigt.
5 ENTER 7 ÷ ■ π x COS	-0,6235	Cos $(\frac{5}{7})\pi$.
■ MODES {DG} 128,57 COS	-0,6235	Schaltet in Grad als Winkelmodus um (kein Indikator) und berechnet $\cos 128,57^\circ$, was mit $\cos (\frac{5}{7})\pi$ übereinstimmt.

Programmierungshinweis. Gleichungen, welche eine inverse trigonometrische Funktion zur Bestimmung eines Winkels θ benutzen, haben oft eine Form wie:

$$\theta = \arctan (y/x).$$

Wenn $x = 0$, dann ist $y \div x$ nicht definiert, was zu der Fehlermeldung **DIVIDE BY 0** führt. Für ein Programm wäre es daher zweckmäßiger, einen Winkel θ durch eine Transformation von *Rechtecks- nach Polarkoordinaten*, welche x, y in r, θ konvertiert, zu bestimmen. Beziehen Sie sich auf "Koordinatentransformationen" an späterer Stelle in diesem Kapitel.

* Tatsächlich sind die berechneten Ergebnisse aufgrund der ungenauen Darstellung von π nur bis zu vier signifikanten Stellen gleich. (Drücken Sie ■ SHOW, um weitere Stellen anzuzeigen.)

Hyperbolische Funktionen

Während x angezeigt ist:

Auszuführende Berechnung:	Tastenfolge:
Sinus hyperbolicus von x (SINH).	■ [HYP] [SIN]
Cosinus hyperbolicus von x (COSH).	■ [HYP] [COS]
Tangens hyperbolicus von x (TANH).	■ [HYP] [TAN]
Arcuscosinus hyperbolicus von x (ASINH).	■ [HYP] ■ [ASIN]
Arcuscosinus hyperbolicus von x (ACOSH).	■ [HYP] ■ [ACOS]
Arcustangens hyperbolicus von x (ATANH).	■ [HYP] ■ [ATAN]

Prozentfunktionen (% , %CHG)

Die Prozentfunktionen stellen einen Sonderfall dar (verglichen mit \boxed{x} und $\boxed{+}$), da sie den Wert der Basis erhalten (im Y-Register), wenn Sie das Ergebnis der Prozentrechnung in das X-Register zurückgeben. Sie können also aufeinanderfolgende Berechnungen ausführen und dabei Basis und Prozentwert verwenden, ohne erneut die Basis eingeben zu müssen.

Auszuführende Berechnung:	Eingabefolge:
$x\%$ von y	y [ENTER] x ■ [%]
Prozentuale Differenz zwischen y und x . ($y \neq 0$)	y [ENTER] x ■ [%CHG]

Beispiel: Berechnen Sie die Mehrwertsteuer i.H.v. 14% und den Gesamtpreis für einen Artikel, der netto DM 15,76 kostet. Verwenden Sie FIX 2 als Anzeigeformat, um auf ganze Pfennige zu runden.

Tastenfolge:

Anzeige

Beschreibung:

■ [DISP] {FX} 2

Rundet angezeigten Wert auf 2 Dezimalstellen.

15,76 14 % 2,21

Berechnet 14%
Mehrwertsteuer.

17,97

Gesamtpreis (Netto-
preis + Steuer).

Nehmen Sie an, daß ein DM 15,76 teurer Artikel letztes Jahr DM 16,12 kostete. Wie groß ist die prozentuale Änderung zwischen dem jetzigen Preis und dem des vergangenen Jahres?

Tastenfolge:

Anzeige

Beschreibung:

16,12 15,76

%CHG

-2,23

Der Preis ist um 2,23%
im Vergleich zum Vor-
jahr gesunken.

DISP {FX} 4

-2,2333

Stellt wieder 4 Dezi-
malstellen ein.

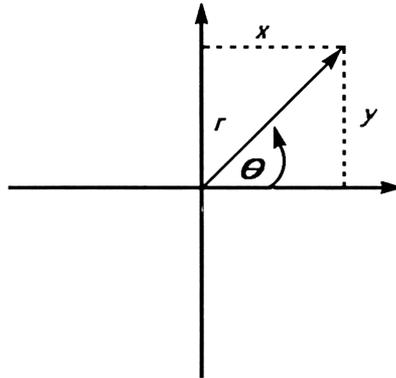
Beachten Sie, daß die Reihenfolge der zwei Zahlen entscheidend für die %CHG Funktion ist. Die Reihenfolge entscheidet, ob die prozentuale Änderung positiv oder negativ ausgedrückt wird.

Konvertierungsfunktionen

Es gibt drei Arten von Konvertierungen: Koordinaten (Polar-/Rechtecksnotation), Winkel (Grad/Bogenmaß) und gebrochener Zahlenteil (Dezimal/Minuten-Sekunden).

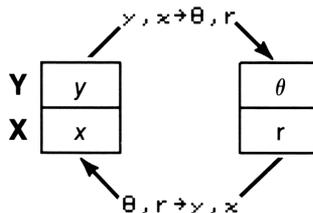
Koordinatentransformationen (P↔RECT)

Rechteckskoordinaten (x, y) und Polarkoordinaten (r, θ) werden wie in der nachfolgenden Abbildung angegeben. Funktionen im P↔RECT (Polar- von/nach Rechtecksnotation) Menü konvertieren zwischen den beiden. Der Winkel θ wird entsprechend dem spezifizierten Winkelmodus interpretiert. Ein berechnetes Ergebnis für θ liegt zwischen -180° und 180° , zwischen $-\pi$ und π oder zwischen -200 und 200 Neugrad.

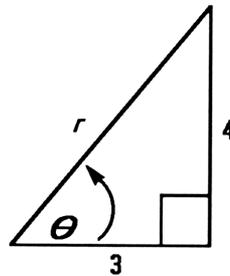
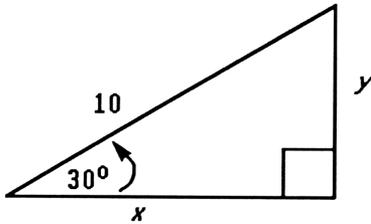


Um zwischen Rechteckskordinaten (x, y) und Polarkordinaten (r, θ) zu konvertieren:

1. Geben Sie zuerst die Koordinaten (in Rechtecks- oder Polarnotation) ein, welche Sie konvertieren möchten. Die Reihenfolge ist y x oder θ r .
2. Drücken Sie .
3. Führen Sie die gewünschte Konvertierung durch: $\{y, x \rightarrow \theta, r\}$ (Rechtecks- nach Polarkordinaten) oder $\{\theta, r \rightarrow y, x\}$ (Polar- nach Rechteckskordinaten). Die konvertierten Koordinaten belegen das X- und Y-Register.
4. In der Anzeige finden Sie nun entweder r (Polarnotation) oder x (Rechtecksnotation). Drücken Sie , um θ oder y angezeigt zu erhalten.



Beispiel: Umrechnung von Polar- nach Rechteckskordinaten. Bestimmen Sie x und y im links abgebildeten Dreieck; bestimmen Sie r und θ im rechts abgebildeten Dreieck.



Tastenfolge:

Anzeige

Beschreibung:

MODES {DG}

Stellt Grad-Modus ein.

30 ENTER 10

Berechnet x .

P↔RECT { $\theta, r \rightarrow y, x$ } 8,6603

x↔y

5,0000

Anzeige von y .

4 ENTER 3

Berechnet die Hypotenuse (r).

P↔RECT { $y, x \rightarrow \theta, r$ } 5,0000

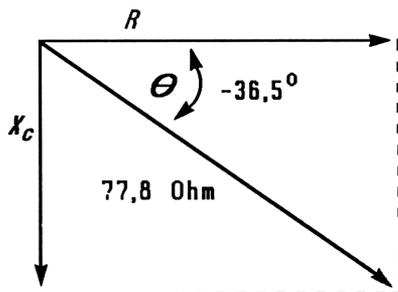
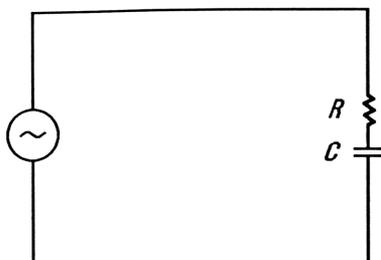
x↔y

53,1301

Anzeige von θ .

Beispiel: Konvertierung von Vektoren. Ingenieur D. Düsentrieb ermittelte für das nachfolgend abgebildete RC-Glied eine Gesamtimpedanz von $77,8 \text{ Ohm}$ und eine Phasenverschiebung von $36,5^\circ$, um die die Spannung dem Strom nacheilt. Wie groß ist der Widerstand R und wie groß ist der Blindwiderstand X_C im vorliegenden Schaltkreis?

Verwenden Sie ein Vektordiagramm, wie nachfolgend abgebildet, in welchem die Impedanz dem Radius r und die Phasenverschiebung dem Winkel θ in Polarnotation entspricht. Sie erhalten die gesuchten Größen, indem Sie die gegebenen Werte in Rechteckskordinaten umwandeln: Der x -Wert entspricht R und y entspricht X_C (beide in Ohm).



Tastenfolge:	Anzeige	Beschreibung:
MODES {DG}		Stellt Grad-Modus ein.
36,5 +/- ENTER	-36,5000	Eingabe von θ , negative Phasenverschiebung für Spannung.
77,8	77,8_	Eingabe von r , Gesamtimpedanz in Ohm.
P\leftrightarrowRECT { $\theta, r \rightarrow y, x$ }	62,5401	Berechnet x , Widerstand R in Ohm.
x\leftrightarrowy	-46,2772	Zeigt y an, den Blindwiderstand X_C .

Weitere anspruchsvollere Operationen mit Vektoren (Addition, Subtraktion, Kreuzprodukt und Skalarprodukt) finden Sie unter dem Programm "Vektoroperationen" in Kapitel 12 ("Mathematikprogramme").

Konvertierungen zwischen Dezimal- und Zeitformaten (H \leftrightarrow HMS)

Zeitangaben (in Stunden bzw. *Hours, H*) oder Winkelmaße (in Grad bzw. *Degrees, D*) können zwischen einer dezimalen Darstellung (*H.h* oder *D.d*) und einem Stunden-Minuten-Sekunden Format (*H.MMSSss* oder *D.MMSSss*) umgerechnet werden, indem Sie das H \leftrightarrow HMS Menü (*Stunden von/nach Stunden-Minuten-Sekunden*) verwenden.

Um zwischen Dezimal- und Minuten-Sekunden Format umzurechnen:

1. Tippen Sie die Zeit oder den Winkel (in Dezimal- oder Minuten-Sekunden Format) ein, welchen Sie konvertieren möchten.
2. Drücken Sie \blacksquare $[H\leftrightarrow HMS]$.
3. Wählen Sie $\{\leftarrow HR\}$ (Stunden-Minuten-Sekunden nach Stunden) oder $\{\rightarrow HMS\}$ (Stunden nach Stunden-Minuten-Sekunden). Anschließend wird das Ergebnis angezeigt.

Beispiel: Konvertieren von Zeitangaben. Wieviel Minuten und Sekunden entspricht $\frac{1}{7}$ einer Stunde? Verwenden Sie FIX 6 als Anzeigeformat.

Tastenfolge:	Anzeige	Beschreibung:
\blacksquare $[DISP]$ $\{FX\}$ 6		
7 $[1/x]$	0,142857	$\frac{1}{7}$ als Dezimalzahl.
\blacksquare $[H\leftrightarrow HMS]$ $\{\rightarrow HMS\}$	0,083429	Entspricht 8 Minuten und 34,29 Sekunden.
\blacksquare $[DISP]$ $\{FX\}$ 4		Stellt wieder FIX 4 Format ein.

Winkelkonvertierungen (D \leftrightarrow RAD)

Das D \leftrightarrow RAD Menü (Grad bzw. Degrees von/nach Radiant) arbeitet unabhängig vom eingestellten Winkelmodus. Bei der Konvertierung in Bogenmaß wird angenommen, daß das X-Register einen Wert in Grad darstellt; bei der Konvertierung in Grad wird angenommen, daß es sich im X-Register um einen Wert im Bogenmaß handelt.

Um Winkel zwischen Grad und Bogenmaß (Radiant) umzurechnen:

1. Tippen Sie den Winkel ein (als Dezimalgrad oder Radiant), welcher konvertiert werden soll.
2. Drücken Sie \blacksquare $[D\leftrightarrow RAD]$.
3. Wählen Sie $\{\rightarrow DEG\}$ (Radiant nach Grad) oder $\{\rightarrow RAD\}$ (Grad nach Radiant). Anschließend wird das Ergebnis angezeigt.

Wahrscheinlichkeitsfunktionen

Das PROB Menü (*Probability*) enthält Funktionen zur Berechnung von Kombinationen, Permutationen, Fakultäten und Zufallszahlen.

PROB Menü

Menüoption	Beschreibung
{Cn,r}	<i>Kombinationen.</i> Zuerst Eingabe von n , danach r (nur ganze positive Zahlen). Berechnet alle <i>Möglichkeiten</i> , n verschiedene Elemente zu Mengen mit jeweils r Elementen zusammenzufassen. Jedes Element darf nur einmal vorkommen, wobei Mengen, die die gleichen Elemente in unterschiedlicher Reihenfolge enthalten, <i>nicht</i> mitgezählt werden.
{Pn,r}	<i>Permutationen.</i> Zuerst Eingabe von n , danach r (nur ganze positive Zahlen). Berechnet alle <i>Möglichkeiten</i> , n verschiedene Elemente zu Mengen mit jeweils r Elementen zusammenzufassen. Jedes Element darf nur einmal vorkommen, wobei Mengen, die die gleichen Elemente in unterschiedlicher Reihenfolge enthalten, <i>mitgezählt</i> werden.
{x!}	<i>Fakultät und Gammafunktion.</i> Berechnet die Fakultät der angezeigten ganzen Zahl ($0 \leq x \leq 253$). Um die Gammafunktion von a , $\Gamma(a)$ zu berechnen, ist $(a - 1)$ einzutippen und \blacksquare [PROB] {x!} zu drücken. (Die Funktion {x!} berechnet $\Gamma(x + 1)$; x kann nicht negativ sein.)
{R}	<i>Zufallszahlengenerator.</i> Enthält 2 Optionen. Drücken von {RANDOM} erzeugt eine Zufallszahl im Bereich $0 \leq x < 1$.* Drücken von {SEED} beginnt eine neue Zufallszahlenreihe mit der Zahl im X-Register als Startwert.

* Der Zufallszahlengenerator des HP-32S erzeugt eigentlich eine Zahl, welche Teil einer gleichverteilten Pseudo-Zufallszahlenfolge ist; sie erfüllt den Spektraltest (D. Knuth, *Seminumerical Algorithms*, vol. 2. London: Addison Wesley, 1981).

{RANDOM} verwendet einen Startwert zum Erzeugen einer Zufallszahl, welche wieder als Startwert für die nächste Zufallszahl verwendet wird. Sie können daher die gleiche Folge von Zufallszahlen wiederholen, indem Sie den gleichen Startwert verwenden; {SEED} erlaubt Ihnen das Eingeben eines neuen Startwertes. Beim Löschen des Speicherbereichs wird der Startwert auf 0 gesetzt.

Beispiel: QA-Team. Eine Abteilung mit 14 Frauen und 10 Männern gründet ein 6-köpfiges Team zur Qualitätssicherung der gefertigten Produkte. Wieviel unterschiedliche Kombinationen von Personen sind möglich?

Tastenfolge:	Anzeige	Beschreibung:
24 [ENTER] 6	6_	Gruppen zu 6 bei einer Grundgesamtheit von 24.
■ [PROB]	$C_{n,r} P_{n,r} \times! R$	PROB Menü.
{C _{n,r} }	134.596,0000	Anzahl aller möglichen Kombinationen.

Wenn die Teammitglieder zufällig ausgewählt werden, mit welcher Wahrscheinlichkeit würden dann 6 Frauen das Team bilden? Um die *Wahrscheinlichkeit* eines Ereignisses zu bestimmen, ist die Anzahl der Kombinationen *für dieses Ereignis* durch die *gesamte Anzahl* von Kombinationen zu dividieren.

Tastenfolge:	Anzeige	Beschreibung:
14 [ENTER] 6	6_	14 Frauen in Gruppen zu je 6.
■ [PROB] {C _{n,r} }	3,003,0000	Anzahl von Kombinationen mit 6 femininen Teammitgliedern.
[x÷y]	134.596,0000	Ruft die gesamte Anzahl von Kombinationen zurück.
[÷]	0,0223	Dividiert Kombinationen der nur femininen Gruppen durch gesamte Anzahl Kombinationen von 6-er Gruppen zur Berechnung der Wahrscheinlichkeit für ein rein feminines Team.

Teile von Zahlen

Die im PARTS Menü enthaltenen Funktionen ändern die Zahl im X-Register auf einfache Weise; diese Funktionen werden hauptsächlich beim Programmieren verwendet.

PARTS Menü

Menüoption	Beschreibung
{ IP }	<i>Ganzzahliger Teil.</i> Entfernt den Nachkommateil von x und ersetzt ihn mit Nullen (z.B. ist der ganzzahlige Teil von 14,2300 gleich 14,0000).
{ FP }	<i>Gebrochener Teil.</i> Entfernt den Vorkommateil von x und ersetzt ihn mit Nullen (z.B. ist der gebrochene Teil von 14,2300 gleich 0,2300).
{ RN }	<i>Rundung.</i> Rundet x intern auf die im Anzeigeformat spezifizierten Nachkommastellen (ohne Rundung wird eine Zahl intern mit 12-stelliger Genauigkeit dargestellt).
{ ABS }	<i>Absolutbetrag.</i> Ersetzt x durch seinen Absolutbetrag.

Funktionsnamen

Sie haben vielleicht schon bemerkt, daß der Name einer Funktion in der Anzeige erscheint, wenn Sie die entsprechende Funktionstaste gedrückt halten (er bleibt solange angezeigt, bis die Taste wieder freigegeben wird). So erscheint, während Sie z.B. \sqrt{x} gedrückt halten, in der Anzeige die Bezeichnung SQRT. "SQRT" (*Square Root*) ist der Name der Funktion, wie er auch in Programmzeilen verwendet wird. Außerdem wird die Funktion unter dem gleichen Namen in alphabetischer Reihenfolge im Funktionsindex gelistet.

Teil 2

Programmierung

Seite	70	5: Einfache Programme
	90	6: Programmierungstechniken

5

Einfache Programme

In Teil 1 des Handbuchs wurden Ihnen die Funktionen und Operationen vorgestellt, welche Sie *manuell*, d.h. durch Drücken einer Taste für jede einzelne Operation, ausführen können. Ein *Programm* erlaubt Ihnen die Wiederholung von Operationen oder Berechnungen, ohne daß Sie jeweils die ganze Tastenfolge wiederholen müssen. In diesem Kapitel erfahren Sie, wie Sie eine Folge von Operationen in einem Programm zusammenfassen und wie ein Programm gestartet wird. Das nächste Kapitel, "Programmierungstechniken", erläutert, wie Unterprogramme und Programmabfragen beim Entwickeln von Programmen eingesetzt werden können.

Einführung: Einfaches Programmbeispiel. Zur Berechnung der Fläche eines Kreises mit einem Radius von 5 cm wäre die Gleichung $A = \pi r^2$ anzuwenden, welche über die Tastenfolge

5 ■ x^2 ■ π ■ \times

realisiert wird, um das Ergebnis 78.5398 angezeigt zu erhalten.

Was wäre aber, wenn Sie die Fläche mehrerer Kreise mit unterschiedlichen Radien berechnen wollten? Anstatt die erforderliche Tastenfolge jedesmal zu wiederholen (wobei sich nur "5" für den neuen Radius ändern würde), könnten Sie die sich wiederholende Tastenfolge in einem Programm zusammenfassen:

001 x^2
002 π
003 \times

Dieses sehr einfache Programm geht davon aus, daß sich der Wert für den Radius im X-Register (der Anzeige) befindet, wenn das Programm gestartet wird. Es berechnet die Kreisfläche und zeigt das Ergebnis an.

Um das Programm in den Programmspeicher einzutippen:

Tastensequenz	Anzeige	Beschreibung
■ PRGM ■ GTO ◦ ◦	PRGM TOP	Stellt Programmzeiger an Anfang des Programmspeichers.
■ x ² ■ π ×	001 x ² 002 π 003 ×	
■ PRGM		

Versuchen Sie nun, das Programm zu starten und die Fläche für einen Kreis mit $r = 5$ zu berechnen.

Tastensequenz	Anzeige	Beschreibung
■ GTO ◦ ◦		Setzt Programmzeiger an Anfang zurück.
5 R/S	78,5398	Das Ergebnis!

Erzeugen eines Programms

Zur Veranschaulichung von Programmierungstechniken und -Konzepten wird in den folgenden Abschnitten wiederholt auf das obenstehende Programm zur Berechnung der Kreisfläche verwiesen.

Programmgrenzen (LBL und RTN)

Wenn Sie mehr als ein Programm im Programmspeicher ablegen möchten, dann benötigt ein Programm ein *Label* zur Kennzeichnung des Programmbeginns (z.B. A01 LBL A) und ein *Return* zur Kennzeichnung des Programmendes (z.B. A05 RTN). Beachten Sie, daß die Zeilennummern jeweils mit dem gewählten Label beginnen.

Programm-Labels. Programme und Segmente von Programmen (auch als *Routinen* bezeichnet) sollten mit einem Label beginnen. Drücken Sie folgende Tasten, um ein Label einzugeben:

■ LBL/RTN { LBL } Buchstabentaste

Das Label dient zur Identifikation bei der Ausführung eines bestimmten Programms bzw. Routine und besteht aus einem Buchstaben von A bis Z. Die Alphatasten werden wie bei der Anwendung von Variablen benutzt (siehe Kapitel 3). Sie können ein Label nur einmal vergeben (bei Nichtbeachtung wird `DUPLICAT. LBL` angezeigt), es kann jedoch der gleiche Buchstabe vergeben werden, welcher für eine Variable benutzt wurde.

Es ist möglich, das oberste Programm im Programmspeicher ohne Label einzugeben. Die nachfolgenden Programme erfordern jedoch ein Label, um die einzelnen Programme abzugrenzen.

Zeilennummern. Zeilennummern ist der gleiche Buchstabe vorangestellt, welcher als Label verwendet wurde (z.B. `A01`). Besitzt eine Routine mehr als 99 Zeilen, so erscheint die Zeilennummer mit einem Punkt anstatt der linken Ziffer (z.B. `A.01` für Zeile 101 in A). Bei mehr als 199 Zeilen wird in der Zeilennummer ein Komma anstatt der linken Ziffer verwendet (z.B. `A,01` für Zeile 201).

Programmrücksprung. Programme und Unterprogramme sollten mit einer Rücksprunganweisung abgeschlossen werden. Sie erreichen dies mit der Tastenfolge:

■ `[LBL/RTN]` {RTN}

Ist das Programm bis zum Ende durchgelaufen, so wird durch die RTN Anweisung der Programmzeiger auf `PRGM TOP` gesetzt, was den Anfang des Programmspeichers darstellt.

Programmeingabe(PRGM)

Drücken von ■ `[PRGM]` schaltet den Programmeingabe-Modus (**PRGM** Indikator ein) ein und aus. In diesem Modus bewirkt das Drücken von Tasten die Speicherung als Programmzeilen im Speicherbereich anstatt der unmittelbaren Ausführung der Operation. Jede Anweisung bzw. Zahl belegt eine Programmzeile, wobei es außer dem freien Speicherbereich keine Eingrenzung für die Anzahl von Zeilen innerhalb eines Programms gibt.

Um ein Programm zu speichern:

1. Drücken Sie ■ `[PRGM]` zum Umschalten in Programmeingabe.

2. Drücken Sie **█** **[GTO]** **◻** **◻** zur Anzeige von PRGM TOP. Damit wird der *Programmzeiger* an den Anfang des Programmspeichers gesetzt. Bei der Eingabe von Programmzeilen erscheinen diese *vor* anderen gespeicherten Programmen.

Wenn Sie Programme nicht mehr benötigen, können Sie diese durch Drücken von **█** **[CLEAR]** {PGM} löschen. Drücken Sie nach der Meldung CL PGMS? Y N zur Bestätigung {Y}, wonach *alle* Programme gelöscht werden.

3. Versehen Sie das Programm mit einem *Label*—ein einzelner Buchstabe von A bis Z. Drücken Sie **█** **[LBL/RTN]** {LBL} *Buchstabe*. Verwenden Sie einen Buchstaben, welcher eine Beziehung zum Programminhalt herstellt, z.B. "V" für "Volumen".
4. Drücken Sie die gleichen Tasten wie bei der manuellen Bearbeitung einer Aufgabenstellung, wenn Sie Rechenoperationen als Programmanweisungen speichern möchten.
5. Beenden Sie das Programm mit einer *Return* Anweisung, wodurch der Programmzeiger am Programmende auf PRGM TOP gesetzt wird; drücken Sie dazu **█** **[LBL/RTN]** {RTN}.
6. Drücken Sie **◻** (oder **█** **[PRGM]**) zum Aufheben der Programmeingabe.

In Programmzeilen enthaltene Zahlen werden so genau wie bei der Eingabe angezeigt, wobei das ALL oder SCI Anzeigeformat verwendet wird. (Sind einzelne Ziffern nicht angezeigt, so drücken Sie **█** **[SHOW]** zur Anzeige der verborgenen Stellen.)

Datenein- und Ausgabe. Für Programme mit mehr als einem Eingabe- oder Ausgabewert gibt es Anweisungen, welche zur Eingabe einer bestimmten Variablen auffordern (INPUT) und eine benannte Variable anzeigen (VIEW). Näheres dazu erfahren Sie später in diesem Kapitel unter "Ein- und Ausgabe von Daten".

◻, **◄** und **█** **[CLEAR]** {×} **bei Programmeingabe.** Beachten Sie nachstehende Sonderfälle während der Programmeingabe:

- **◻** hebt immer die Programmeingabe auf; es wird nie eine Zahl auf Null gesetzt.
- **◄** löscht die momentane Programmzeile; ist der Cursor angezeigt, so wird die zuletzt eingegebene Ziffer gelöscht.
- Zur *Programmierung* einer Funktion, welche den Inhalt des X-Registers löscht, ist **█** **[CLEAR]** {×} zu verwenden.

Funktionsnamen in Programmen. Der Name einer Funktion, welche in einem Programm verwendet wird, ist nicht notwendigerweise der gleiche Name, wie er auf dem Tastenfeld oder im Menü erscheint. Der im Programm verwendete Funktionsname ist gewöhnlich eine längere Ausführung einer Abkürzung, die auf eine Funktionstaste oder in ein Menü paßt. Die längere Version der Abkürzung erscheint bei der Ausführung der Funktion kurz in der Anzeige, während Sie die Funktionstaste gedrückt halten.

Beispiel: Eingabe eines benannten Programms. Die nachstehende Tastenfolge löscht das vorherige Programm zur Berechnung der Kreisfläche und gibt ein neues ein, welches eine Benennung (*Label*) und eine Rücksprunganweisung (*Return*) enthält. Falls Ihnen ein Tippfehler unterläuft, dann drücken Sie  zum Löschen der momentanen Programmzeile; anschließend können Sie die korrekte Anweisung eingeben.

Tastenfolge:	Anzeige	Beschreibung:
		Aktiviert Programm-eingabe-Modus (PRGM an).
 {PGM} {Y}	PRGM TOP	Löscht Programmspeicher.
 {LBL} A	A01 LBL A	Benennt dieses Programm mit "A" (für <i>Area</i>).
	A02 x ²	Eingabe von 3 Programmzeilen.
	A03 π	
	A04 x	
 {RTN}	A05 RTN	Schließt Programm ab.
		Hebt Programmeingabe auf (PRGM Indikator aus).

Aufrufen eines Programms

Wenn Sie ein Programm ablaufen bzw. *ausführen* möchten, darf der Programmeingabemodus nicht aktiv sein (keine Programmzeilennummern angezeigt; **PRGM** aus). Drücken von \boxed{C} hebt die Programmeingabe auf.

Ausführen eines Programms (XEQ)

Drücken Sie \boxed{XEQ} *Label*, um das mit dem entsprechenden Label versehene Programm auszuführen.* Der **PRGM** Indikator fängt dabei zu blinken an, bis die Programmausführung beendet ist.

Falls das Programm bestimmte Eingabewerte erfordert, so geben Sie die Zahlen vor dem Start des Programms ein.

Beispiel: Führen Sie das Programm mit dem Label A aus, um die Flächen von drei unterschiedlichen Kreisen mit den Radien 5, 2,5 und 2π zu berechnen. Denken Sie daran, den Radius jeweils vor der Programmausführung einzugeben.

Tastenfolge:	Anzeige	Beschreibung:
5 \boxed{XEQ} A	78,5398	Eingabe des Radius, danach Ausführung von A. Das Ergebnis erscheint in der Anzeige.
2,5 \boxed{XEQ} A	19,6350	Berechnet Fläche des 2. Kreises.
2 $\boxed{\pi}$ $\boxed{\times}$ \boxed{XEQ} A	124,0251	Berechnet Fläche des 3. Kreises.

* Wenn Sie nur ein Programm im Programmspeicher abgelegt haben, können Sie dieses auch durch Drücken von \boxed{GTO} $\boxed{\cdot}$ $\boxed{R/S}$ (*Run/Stop*) ausführen.

Testen eines Programms

Wenn Sie wissen, daß ein Programm einen Fehler enthält, andererseits aber nicht genau wissen, an welcher Stelle der Fehler ist, so können Sie durch die schrittweise Ausführung des Programms auf einfache Weise einen Programmtest ausführen. Dies ist außerdem empfehlenswert, wenn Sie ein längeres oder kompliziertes Programm auf Fehlerfreiheit testen möchten. Bei der schrittweisen Programmausführung können Sie das Ergebnis nach jedem Programmschritt ansehen. Damit ist eine Überprüfung bzw. ein Vergleich mit bekannten Daten möglich und eine fehlerbedingte Abweichung erkennbar.

1. Stellen Sie sicher, daß—wie bei der normalen Programmausführung—der Programmeingabe-Modus nicht aktiv ist (**PRGM** Indikator aus).
2. Drücken Sie   *Label*, um den Programmzeiger an den Anfang des Programms zu setzen (d.h. auf die LBL Anweisung). Die *Go TO* Anweisung verschiebt den Programmzeiger, ohne jedoch das Programm gleich zu starten. (Handelt es sich bei dem zu testenden Programm um das erste oder einzige Programm im Speicher, dann können Sie auch     drücken, um den Programmzeiger an den Anfang zu stellen.)
3. Halten Sie  gedrückt. Dies zeigt die momentane Programmzeile an. Wenn Sie  freigeben, wird diese Zeile ausgeführt. Das Ergebnis der Ausführung erscheint danach im X-Register.
Wenn Sie zur *vorangehenden* Zeile zurückkehren möchten, so drücken Sie  . Es erfolgt keine Ausführung.
4. Der Programmzeiger wird an die nächste Programmzeile bewegt. Wiederholen Sie Schritt 3, bis Sie auf einen Fehler stoßen (falsches Ergebnis wird angezeigt) oder das Ende des Programms erreicht haben.

Wenn der Programmeingabe-Modus aktiv ist, dann bewegt  (oder  ) nur den Programmzeiger, ohne die jeweilige Zeile auszuführen. Halten Sie die entsprechende Pfeiltaste gedrückt, wenn Sie die Programmzeilen automatisch nach oben oder unten "rollen" möchten.

Beispiel: Programmtest. Führen Sie Programm A schrittweise aus. Verwenden Sie dabei für den Radius 5 als Testwert. Stellen Sie sicher, daß der Programmeingabe-Modus nicht aktiv ist, bevor Sie beginnen.

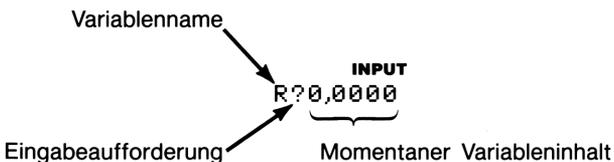
Tastenfolge:	Anzeige	Beschreibung:
5	5_	
■ GTO A	5,0000	Verschiebt Programmzeiger an Label A.
▼ (gedrückt halten) (freigeben)	A01 LBL A 5,0000	
▼ (gedrückt halten) (freigeben)	A02 \times^2 25,0000	Quadriert Eingabe.
▼ (gedrückt halten) (freigeben)	A03 π 3,1416	Wert von π .
▼ (gedrückt halten) (freigeben)	A04 \times 78,5398	25π .
▼ (gedrückt halten) (freigeben)	A05 RTN 78,5398	Programmende. Das Ergebnis ist korrekt.

Ein- und Ausgabe von Daten

Die in Programmen verwendbaren *Variablen* dienen zum Speichern von Eingabedaten, Zwischen- und Endergebnissen. (Variablen werden, wie in Kapitel 3 erklärt, über einen Buchstaben von A bis Z identifiziert.) *Der Variablenname selbst hat keine Verbindung mit Programm-Labels.*

Eingabe von Daten in Variablen (INPUT)

Die INPUT Anweisung (■ INPUT Variable) hält das Programm an und zeigt eine Eingabeaufforderung für die jeweilige Variable an, zusammen mit dem momentanen Variableninhalt. Als Beispiel:



Drücken Sie $\boxed{R/S}$ (*Run/Stop*) oder $\boxed{\nabla}$ zur Fortsetzung des Programms. Der eingegebene Wert überschreibt dabei den Inhalt des X-Registers und den der Variablen. Wenn Sie den angezeigten Wert nicht geändert haben, bleibt er im X-Register sowie in der Variable erhalten.

Das Programm zur Berechnung der Kreisfläche hat folgende Form, nachdem eine INPUT Anweisung aufgenommen wurde:

```
A01 LBL A
A02 INPUT R
A03  $\times^2$ 
A04  $\pi$ 
A05  $\times$ 
A06 RTN
```

Verwenden von INPUT in einem Programm

1. Entscheiden Sie, welche Werte benötigt werden und weisen Sie diesen eine Variable zu (im vorliegenden Beispiel ist der Radius einzugeben, weshalb eine Variable "R" verwendet wurde).
2. Fügen Sie am Programmanfang für jede benötigte Variable eine INPUT Anweisung ein. An späterer Stelle im Programm, wenn der Eingabewert tatsächlich benötigt wird, fügen Sie dann eine \boxed{RCL} Variable Anweisung ein, wodurch der Variableninhalt zurück in den Stack gebracht wird.*

Beispiel: Beziehen Sie sich auf das Programm "Annuitätenrechnung" auf Seite 222 in Teil 4. Als erstes werden dort die Eingabewerte für die Variablen *N*, *I*, *B*, *P* und *F* (Zeilen T02 bis T06) abgefragt.

* Da INPUT den soeben eingetippten Wert im X-Register läßt, *müssen Sie nicht* die Variable an späterer Stelle zurückerufen—Sie könnten ihn auch immer dann eingeben, wenn er benötigt wird, wodurch Speicherplatz gespart wird. Allerdings empfiehlt es sich bei längeren Programmen, die Eingabewerte am Programmanfang einzugeben und sie bei späterem Bedarf wieder in den Stack zurückzufufen.

Denken Sie auch daran, daß Sie Berechnungen ausführen können, während das Programm angehalten ist und auf Eingaben wartet. Dadurch kann sich der Stackinhalt ändern, was sich wiederum auf die nächste vom Programm ausgeführte Berechnung auswirken könnte. Das Programm sollte deshalb nicht davon ausgehen, daß sich der Stack vor und nach der INPUT Anweisung nicht geändert hat. Wenn Sie die Eingabeaufforderung für die benötigten Variablen gleich zu Programmbeginn und das Zurückerufen bei der anstehenden Berechnung vorsehen, verhindern Sie auch, daß vor einer Berechnung der Stackinhalt geändert wurde.

Der Programmablauf. Wenn Sie das Programm starten, wird bei jeder INPUT Anweisung der **INPUT** Indikator eingeschaltet und zur Eingabe eines Wertes aufgefordert, wie z.B. $R \text{ ? } 0,0000$. Der angezeigte Wert stellt dabei den momentanen Inhalt von R dar.

- **Um die Zahl zu ändern**, ist der neue Wert einzutippen und $\boxed{R/S}$ zu drücken.* Wenn Sie einen Wert berechnen müssen, so ist dies vor dem Drücken von $\boxed{R/S}$ möglich.
- **Um die Zahl unverändert zu lassen**, drücken Sie einfach $\boxed{R/S}$.
- **Um Berechnungen mit der angezeigten Zahl auszuführen**, drücken Sie \boxed{ENTER} vor der Eingabe einer anderen Zahl.
- **Um die INPUT Anweisung aufzuheben**, drücken Sie \boxed{C} .[†] Der momentane Variableninhalt bleibt im X-Register. Wenn Sie $\boxed{R/S}$ zur Fortsetzung des Programms drücken, wird die aufgehobene INPUT Aufforderung wiederholt.
- **Um alle Stellen** bei einer Eingabeaufforderung anzuzeigen, ist \boxed{SHOW} zu drücken. (Bei einem binären Wert mit mehr als 12 Stellen führt $\boxed{\sqrt{x}}$ und $\boxed{\Sigma+}$ zur Anzeige der restlichen Stellen.)

Anzeigen von Variableninhalten (VIEW)

Die programmierte VIEW Anweisung ($\blacksquare \boxed{VIEW}$ Variable) unterbricht den Programmablauf und zeigt den Inhalt der Variablen an, wie z.B.

$$R=78,5398$$

Hiebei handelt es sich *nur um die Anzeige* des Wertes, nicht um das Kopieren des Variableninhaltes in das X-Register.

- Drücken von \boxed{ENTER} kopiert die Zahl in das X-Register.
- Hat die Zahl mehr als 10 signifikante Stellen, so bewirkt das Drücken von $\blacksquare \boxed{SHOW}$ die Anzeige der ganzen Zahl. (Handelt es sich um einen binären Wert mit mehr als 12 Stellen, so verwenden Sie $\boxed{\sqrt{x}}$ und $\boxed{\Sigma+}$ zur Anzeige der restlichen Stellen.)
- Drücken von \boxed{C} (oder $\boxed{\blacktriangleleft}$) löscht die VIEW Anzeige und zeigt den Inhalt des X-Registers an.

* Dieser neue Wert überschreibt den seitherigen Wert im X-Register.

[†] Wenn Sie \boxed{C} während der Zahleneingabe drücken, wird der Wert auf Null gesetzt. Drücken Sie erneut \boxed{C} , um INPUT aufzuheben.

- Drücken von **CLEAR** löscht den Inhalt der angezeigten Variablen.

Drücken Sie **R/S** zur Fortsetzung des Programms.

Zum Beispiel: Beziehen Sie sich auf das Programm "Lösen eines linearen Gleichungssystems über Determinantenverfahren" auf Seite 175 in Teil 4. Die Zeilen S24 bis S29 am Ende der S Routine zeigen die Ergebnisse für X, Y und Z. Beachten Sie auch, daß jeder VIEW Anweisung in diesem Programm—wie in allen Applikationen—eine RCL Anweisung vorangeht. RCL ist nicht notwendig, ist aber eine Erleichterung, da die jeweilige Variable in das X-Register gebracht wird und dort für Berechnungen zur Verfügung ist. (Drücken von **ENTER** während der Ausführung von VIEW hätte den gleichen Effekt.)

Beispiel: Eingeben und Ansehen von Variablen in einem Programm. Schreiben Sie eine Gleichung zur Berechnung der Oberfläche und des Volumens von einem Zylinder bei gegebenem Radius und Höhe. Benennen Sie das Programm C (für *Cylinder*) und benutzen Sie die Variablen S (*Surface*), V (*Volumen*), R (*Radius*) und H (*Höhe*). Verwenden Sie folgende Gleichungen:

$$V = \pi R^2 H$$

$$S = 2\pi R^2 + 2\pi RH = 2(\pi R^2 + \pi RH).$$

Tastensequenz	Anzeige	Beschreibung	
PRGM GTO . .	PRGM TOP	Programmeingabe; stellt Zeiger an Speicheranfang.	
LBL/RTN {LBL} C (C ist LN Taste)	C01 LBL C	Benennt Programm.	
INPUT R	C02 INPUT R	Anweisungen zur Eingabe von Radius und Höhe, zum Berechnen und Speichern des Volumens in V.	
INPUT H	C03 INPUT H		
RCL R	C04 RCL R		
x²	C05 x ²		
RCL x H	C06 RCLx H		
π	C07 π		
x	C08 x		
STO V	C09 STO V		
RCL ÷ H	C10 RCL÷ H		Konvertiert $\pi R^2 H$ nach πR^2 .

$\boxed{\text{RCL}}$ R	C11 RCL R	
$\boxed{\text{RCL}}$ $\boxed{\times}$ H	C12 RCL \times H	
\blacksquare $\boxed{\pi}$	C13 π	
$\boxed{\times}$	C14 \times	Berechnet πRH .
$\boxed{+}$	C15 $+$	Berechnet $(\pi R^2 + \pi RH)$.
2	C16 2_	Berechnet $2 (\pi R^2 + \pi RH)$, speichert Oberfläche in S.
$\boxed{\times}$	C17 \times	
$\boxed{\text{STO}}$ S	C18 STO S	
\blacksquare $\boxed{\text{VIEW}}$ V	C19 VIEW V	Anzeigen von Volumen und Oberfläche.
\blacksquare $\boxed{\text{VIEW}}$ S	C20 VIEW S	
\blacksquare $\boxed{\text{LBL/RTN}}$ {RTN}	C21 RTN	Schließt Programm ab.
$\boxed{\text{C}}$		Hebt Programmeingabe auf.
\blacksquare $\boxed{\text{MEM}}$ {PGM}	LBL C 031,5	Zeigt Speicherbedarf und Prüfsumme; eine abweichende Prüfsumme weist auf Abweichungen bei der Eingabe hin.
\blacksquare $\boxed{\text{SHOW}}$ (gedrückt halten)	CHKSUM=4602	

Bestimmen Sie nun das Volumen und die Oberfläche eines Zylinders mit einem Radius von 2,5 cm und einer Höhe von 8,0 cm.

Tastenfolge:	Anzeige	Beschreibung
$\boxed{\text{XEQ}}$ C (C ist die $\boxed{\text{LN}}$ Taste)	R?0,0000	Startet Ausführung von C; Eingabeaufforderung für R. (Seitheriger Inhalt von R wird angezeigt.)
2,5 $\boxed{\text{R/S}}$	H?0,0000	Eingabeaufforderung für H. (Seitheriger Inhalt wird angezeigt.)
8 $\boxed{\text{R/S}}$	V=157,0796	Volumen in cm^3 .
$\boxed{\text{R/S}}$	S=164,9336	Oberfläche in cm^2 .

Anhalten oder Unterbrechen eines Programms

Programmieren eines Stopps oder einer Pause (STOP, PSE)

- Drücken von **[R/S]** (*Run/Stop*) während der Programmeingabe fügt eine STOP Anweisung ein. Damit wird ein Programm angehalten, bis Sie es durch Drücken von **[R/S]** wieder fortsetzen. Sie können STOP anstatt RTN verwenden, wenn nach Programmabschluß der Programmzeiger nicht an den Anfang des Programmspeichers zurückgesetzt werden soll.
- Drücken von **[LBL/RTN]** {PSE} während der Programmeingabe fügt eine PSE (*Pause*) Anweisung ein. Dadurch wird das Programm für etwa eine Sekunde unterbrochen und es wird der Inhalt des X-Registers angezeigt.

Unterbrechen eines gestarteten Programms

Sie können ein laufendes Programm jederzeit unterbrechen, indem Sie **[C]** oder **[R/S]** drücken. Das Programm schließt die momentane Anweisung ab, bevor es anhält. Drücken Sie wieder **[R/S]** (*Run/Stop*), um das Programm fortzusetzen.

Werden **[XEQ]**, **[GTO]** oder {RTN} bei einem unterbrochenen Programm gedrückt, so kann das Programm *nicht* durch Drücken von **[R/S]** fortgesetzt werden—es ist erneut **[XEQ]** *Label* auszuführen.

Fehlerbedingte Stopps

Tritt ein Fehler während des Programmablaufs auf, so wird das Programm angehalten und eine Meldung ausgegeben. (Sie finden eine Auflistung von Meldungen und Fehlerursachen ab Seite 285.)

Sie können die fehlerverursachende Programmzeile ansehen, indem Sie **[PRGM]** drücken. Das Programm hat an dieser Zeile angehalten. (Es kann sich z.B. um eine \div Anweisung handeln, welche eine unzulässige Division durch Null verursachte.)

Edieren eines Programms

Durch Einfügen oder Löschen von Programmzeilen können Sie ein im Programmspeicher abgelegtes Programm modifizieren. Selbst wenn nur eine kleine Änderung erforderlich ist, müssen Sie die ganze Zeile löschen und eine neue Zeile einfügen.

Um eine Programmzeile zu löschen:

1. Wählen Sie das entsprechende Programm oder die Routine (■ *Label*), aktivieren Sie den Programmeingabe-Modus (■) und drücken Sie oder ■ zum Auffinden der zu ändernden Zeile. Halten Sie die Pfeiltaste gedrückt, um das Programm durchzublätern. (Wenn Sie die Zeilennummer bereits kennen, so können Sie durch Drücken von ■ *Label nn* den Programmzeiger an die gewünschte Stelle setzen.)
2. Löschen Sie die zu modifizierende Zeile durch Drücken von . Der Programmzeiger wird daraufhin an die *vorangehende* Zeile gestellt. (Wenn Sie mehrere aufeinanderfolgende Programmzeilen löschen möchten, sollten Sie daher mit der *letzten* Zeile beginnen.)
3. Tippen Sie die neue Anweisung ein, falls erforderlich. Damit wird die gelöschte Anweisung ersetzt.
4. Heben Sie die Programmeingabe auf (oder ■).

Um eine Programmzeile einzufügen: Finden Sie die Zeile auf, welche der Stelle *vorangeht*, wo die neue Zeile eingefügt werden soll. Tippen Sie die neue Anweisung ein; Sie wird *nach* der momentan angezeigten Zeile eingefügt.

Wenn Sie z.B. eine neue Zeile vor den Zeilen A04 und A05 eines Programms einfügen möchten, so würden Sie zuerst Zeile A04 anzeigen und dann die Anweisung(en) eintippen. Die nachfolgenden Programmzeilen, beginnend mit der ursprünglichen Zeile A05, werden nach unten verschoben und neu durchnummeriert.

Der Programmspeicher

Ansehen des Programmspeichers

Drücken von **PRGM** aktiviert/desaktiviert die Programmeingabe; bei aktiver Programmeingabe wird der Inhalt des Programmspeichers angezeigt.

Der Programmspeicher beginnt mit **PRGM TOP**. Die Auflistung der Programmzeilen ist zyklisch, womit Sie den Programmzeiger über den Anfang direkt an das Ende der Liste bewegen können, und umgekehrt. Während die Programmeingabe aktiv ist, kann der Programmzeiger (die angezeigte Zeile) wie folgt verschoben werden:

- Verwenden Sie die Pfeiltasten **▼** und **▲**. Drücken von **▼** an der letzten Zeile verschiebt den Zeiger an **PRGM TOP**, während das Drücken von **▲** bei **PRGM TOP** den Zeiger an die letzte Programmzeile verschiebt.

Um mehrere Zeilen pro Tastendruck durchzusehen ("zu rollen"), ist die Taste **▼** oder **▲** gedrückt zu halten.

- Drücken Sie **GTO** **◻◻**, um den Zeiger an **PRGM TOP** zu stellen.
- Drücken Sie **GTO** **◻** *Label nn*, um zu einer benannten Zeile (mit einer Zeilennummer < 100) zu springen.

Ist die Programmeingabe nicht aktiv (keine Zeilen angezeigt), so kann der Programmzeiger auch durch Drücken von **GTO** *Label* an die betreffende Zeile verschoben werden.

Das Aufheben der Programmeingabe *ändert nicht* die Position des Programmzeigers.

Speicherbelegung

Jede Programmzeile belegt entweder 1,5 oder 9,5 Bytes.

- Zahlen belegen 9,5 Bytes, *außer* für ganze Zahlen von 0 bis 99, welche nur 1,5 Bytes belegen.
- Alle anderen Anweisungen belegen 1,5 Bytes.

Wenn während der Programmeingabe die Meldung MEMORY FULL angezeigt wird, so steht für die soeben eingetippte Zeile kein Speicherplatz mehr zur Verfügung. Sie können freien Speicherplatz erzeugen, indem Sie andere Programme oder Daten löschen. (Siehe "Löschen eines oder mehrerer Programme", unten, oder "Verwalten des Speicherbereichs" in Anhang B.)

Programmkatalog (MEM)

Der Programmkatalog stellt eine Auflistung aller Programm-Labels mit der entsprechend belegten Anzahl von Bytes dar. Drücken Sie {PGM} zur Anzeige des Katalogs, wobei Sie mit oder den Listenzeiger verschieben können. Der Katalog bietet folgende Anwendungsmöglichkeiten:

- Ansehen der im Programmspeicher enthaltenen Labels und die Speicherbelegung jedes benannten Programms bzw. Routine.
- Ausführen eines benannten Programms. (Drücken Sie oder , während das Label angezeigt ist.)
- Anzeigen eines benannten Programms. (Drücken Sie , während das Label angezeigt ist.)
- Löschen eines bestimmten Programms. (Drücken Sie , während das Label angezeigt ist.)
- Ansehen der Prüfsumme, welche zu einem bestimmten Programmsegment gehört. (Drücken Sie .)

Der Katalog zeigt Ihnen, wieviel Speicherplatz von einem benannten Programmsegment belegt wird. Die Programme/Segmente sind jeweils durch Labels gekennzeichnet:

LBL C 031,5

Von Programm C belegte Anzahl Bytes.



Löschen eines oder mehrerer Programme

Um ein bestimmtes Programm zu löschen:

1. Drücken Sie {PGM} und lassen Sie sich das Label des Programms anzeigen (über und .

2. Drücken Sie **■** **CLEAR**.
3. Drücken Sie **C** zum Aufheben der Kataloganzeige oder **⬇** zum Zurückgehen zur nächsthöheren Ebene.

Um alle Programme im Programmspeicher zu löschen:

1. Drücken Sie **■** **PRGM** zum Aktivieren der Programmeingabe (**PRGM** Indikator an).
2. Drücken Sie **■** **CLEAR** {PGM} zum Löschen des Speichers.
3. CL PGMS? Y N erwartet eine Bestätigung. Drücken Sie {Y}.
4. Drücken Sie **■** **PRGM** zum Aufheben der Programmeingabe.

Durch (**■** **CLEAR** {ALL}) werden ebenfalls alle Programme gelöscht.

Prüfsumme

Die *Prüfsumme* stellt einen eindeutigen hexadezimalen Wert für jedes Programm-Label und dessen zugehörigen Programmzeilen (bis zum nächsten Label) dar. Sie ermöglicht den Vergleich zwischen einer bekannten Prüfsumme und der eines eingetippten Programms. Sind die beiden Werte nicht gleich, so wurde bei der Programmeingabe die vorgegebene Tastenfolge für die bekannte Prüfsumme nicht eingehalten. Um die Prüfsumme anzuzeigen:

1. Drücken Sie **■** **MEM** {PGM} zur Anzeige der Programm-Labels.
2. Lassen Sie sich das gewünschte Label anzeigen.
3. Drücken Sie **■** und halten Sie **SHOW** gedrückt, um **CHKSUM=Wert** anzuzeigen.

Um z.B. die Prüfsumme für das momentane Programm anzuzeigen:

Tastenfolge:	Anzeige	Beschreibung
■ MEM {PGM}	LBL C 031,5	Zeigt Label C, das 31,5 Bytes belegt.
■ SHOW (gedrückt halten)	CHKSUM=4602	Stimmt Ihre Prüfsumme <i>nicht</i> mit diesem Wert überein, so liegen Abweichungen bei der Eingabe vor.

Alle in Teil 4 enthaltenen Applikationen enthalten CHKSUM Werte für jede benannte Routine, was zur Überprüfung der korrekten Programmeingabe dienen soll.

Nicht programmierbare Funktionen

Nachstehende HP-32S Funktionen lassen sich nicht in Programmen verwenden:

■ CLEAR { PGM }	■ GTO ◦ ◦
■ CLEAR { ALL }	■ GTO ◦ Label nn
◀	■ MEM
▼, ■ ▲	■ SHOW
■ PRGM	

Polynome und Hornersches Schema

Einige Ausdrücke, wie z.B. Polynome, enthalten mehrmals die gleiche Variable als Unbekannte. In dem Ausdruck

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

kommt z.B. die Variable x an vier verschiedenen Stellen vor. Ein Programm zur Lösung dieser Gleichung könnte wiederholt eine gespeicherte Kopie von x aus einer Variablen abrufen. Allerdings besteht eine kürzere Lösung dieses Problems darin, den Stack mit einer Konstanten aufzufüllen (siehe "Auffüllen des Stacks mit einer Konstanten" auf Seite 39 in Kapitel 2).

Das Hornersche Schema ist ein hilfreiches Verfahren zum Umordnen eines Polynoms, um Rechenschritte und Rechenzeit einzusparen. Es ist besonders nützlich bei den Funktionen SOLVE und fFN, deren Ausführung etwas länger dauert, da sie Unterprogramme verwenden.

Das Horner'sche 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 zur Berechnung von $5x^4 + 2x^3$ in der Form $((5x + 2)x)x$, und berechnen Sie den Ausdruck für den Wert $x = 7$.

Tastenfolge:	Anzeige	Beschreibung
■ PRGM ■ GTO ◊ ◊	PRGM TOP	Wird PRGM TOP bereits angezeigt, so können Sie GTO überspringen.
■ LBL/RTN {LBL} P	P01 LBL P	
■ INPUT X ENTER ENTER ENTER 5 x	P02 INPUT X P03 ENTER P04 ENTER P05 ENTER P06 5 P07 x	Füllt den Stack mit x , berechnet $5x$.
2 + x	P08 2 P09 + P10 x	$(5x + 2)x$.
x x	P11 x P12 x	$(5x + 2)x^3$.
■ LBL/RTN {RTN}	P13 RTN	
C		Hebt Programmeingabe auf.

Werten Sie nun das Polynom für $x = 7$ aus.

XEQ P	$x?$ Wert	Eingabeaufforderung für x .
--------------	-----------	-------------------------------

7 R/S	12.691,0000	Ergebnis.
--------------	-------------	-----------

Eine allgemeinere Form dieses Programms zum Lösen jeder Gleichung $((Ax + B)x + C)x + Dx + E$ wäre dieses:

```
P01 LBL P
P02 INPUT A
P03 INPUT B
P04 INPUT C
P05 INPUT D
P06 INPUT E
P07 INPUT X
P08 ENTER
P09 ENTER
P10 ENTER
P11 RCL X A
P12 RCL + B
P13 ×
P14 RCL + C
P15 ×
P16 RCL + D
P17 ×
P18 RCL + E
P19 RTN
```

Programmierungstechniken

Kapitel 5 behandelte Programmierungsgrundlagen des HP-32S. Das vorliegende Kapitel beschäftigt sich mit anspruchsvolleren, aber nützlichen Techniken:

- Verwendung von Unterprogrammen zur Gliederung von Programmen. Programmteile, welche zur Bearbeitung einer speziellen Aufgabe dienen, werden mit einem Label versehen und vom Hauptprogramm getrennt. Unterprogramme dienen auch zur Verkürzung eines Programms, welches eine Reihe von Rechenschritten wiederholt ausführen muß.
- Verwendung von bedingten Funktionen (Vergleiche und Flags) zur Entscheidung, welche Anweisungen oder Unterprogramme für eine bestimmte Aufgabe benutzt werden soll.
- Verwendung von Schleifen mit Zählern, um einen Satz von Anweisungen n -mal auszuführen.
- Verwendung von indirekter Adressierung, um auf unterschiedliche Variablen über die gleiche Programmanweisung zuzugreifen.

Routinen in Programmen

Ein Programm setzt sich aus einer oder mehreren *Routinen* zusammen, die eine ablauffähige Einheit zum Ausführen einer speziellen Aufgabe darstellen. Wird ein Programm zu komplex, so ist eine Aufgliederung in mehrere kleinere Einheiten vorteilhaft. Dadurch können Sie das Programm einfacher schreiben, modifizieren und testen.

Sehen Sie sich z.B. das Programm für die Normalverteilung auf Seite 215 in Teil 4 an. Dieses Programm besitzt 4 Routinen mit den Labels S, D, N und F. Routine S "initialisiert" das Programm durch Ansammeln der Eingabedaten für den Mittelwert und die Standardabweichung. Routine D spezifiziert die Integrationsgrenzen, führt Routine N aus und zeigt das Ergebnis an. Routine N selbst integriert die in Routine F definierte Funktion und schließt die Berechnung von $Q(x)$ ab.

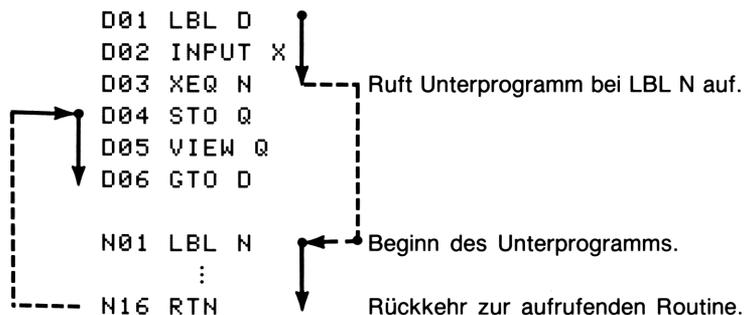
Normalerweise beginnt eine Routine mit einem Label (LBL) und endet mit einer Anweisung, welche die Programmausführung steuert (wie z.B. RTN, GTO, STOP, oder ein unterschiedliches Label).

Aufrufen von Unterprogrammen (XEQ, RTN)

Ein *Unterprogramm* besteht aus einer Routine, die von einer anderen Routine *aufgerufen bzw. ausgeführt* wird und zur aufrufenden Routine *zurückkehrt*, nachdem sie abgeschlossen wurde. Das Unterprogramm *muß* mit LBL beginnen und mit RTN enden. Es kann selbst wieder ein anderes Unterprogramm aufrufen.

- XEQ muß für ein Unterprogramm zu einem Label (LBL) verzweigen (nicht zu einer Zeilennummer).
- Nachdem das nächste RTN festgestellt wird, kehrt das Programm zu der Zeile zurück, welche dem XEQ folgt.

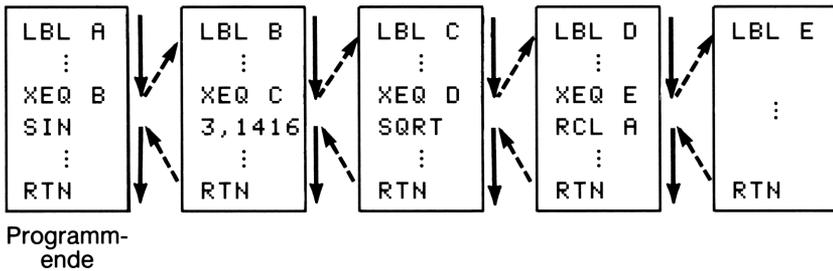
Zum Beispiel stellt die oben erwähnte Routine N ein Unterprogramm dar (zur Berechnung von $Q(x)$), welches von Routine D in Zeile D03 XEQ N aufgerufen wird. Routine N endet mit einer RTN Anweisung, wodurch das Programm mit Routine D (zum Speichern und Anzeigen des Ergebnisses) in Zeile D04 fortfährt.



Verschachtelte Unterprogramme

Ein Unterprogramm kann ein zweites Unterprogramm aufrufen und dieses ein weiteres, usw.; diese "Verschachtelung" von Unterprogrammen—das Aufrufen eines Unterprogramms innerhalb eines Unterprogramms—ist auf vier Ebenen (ohne Berücksichtigung der Hauptprogrammebene) beschränkt. Der Ablauf bei verschachtelten Unterprogrammen ist nachfolgend dargestellt:

Hauptprogramm
(Oberste Ebene)



Der Versuch zur Ausführung eines über mehr als 7 Ebenen verschachtelten Unterprogramms führt zur Fehlermeldung XEQ OVERFLOW.

Beispiel: Verschachteltes Unterprogramm. Das nachfolgende Unterprogramm mit dem Label S berechnet den Ausdruck

$$\sqrt{a^2 + b^2 + c^2 + d^2}$$

als Teil einer längeren Berechnung in einem größeren Programm. Das Unterprogramm ruft ein *weiteres* Unterprogramm (verschachtelt) mit dem Label Q auf, um das sich wiederholende Quadrieren und Addieren auszuführen. Dadurch wird das Hauptprogramm kürzer als bei der Entwicklung ohne Unterprogramm.

Rückkehr zum
Hauptprogramm

Vom Hauptprogramm,
XEQ S

	Programmzeilen:	Beschreibung
	S01 LBL S ←	Beginnt das "Haupt"-Unterprogramm.
	S02 INPUT A	Eingabe von A.
	S03 INPUT B	Eingabe von B.
	S04 INPUT C	Eingabe von C.
	S05 INPUT D	Eingabe von D.
	S06 RCL D	Ruft die Daten für die nachfolgende Berechnung zurück.
	S07 RCL C	
	S08 RCL B	
	S09 RCL A	
	S10 x^2	Berechnet A^2 .
	S11 XEQ Q	Berechnet B^2 , dann $A^2 + B^2$.
1	→ S12 XEQ Q	Berechnet $A^2 + B^2 + C^2$.
2	→ S13 XEQ Q	Berechnet $A^2 + B^2 + C^2 + D^2$.
3	→ S14 SQRT	Berechnet
		$\sqrt{A^2 + B^2 + C^2 + D^2}$
	S15 RTN	Beendet "Haupt"-Unterprogramm; Rückkehr zum Hauptprogramm.
	Q01 LBL Q ←	Start des verschachtelten Unterprogramms.
	Q02 $x \langle \rangle y$	Quadriert Zahl und addiert Ergebnis zur aktuellen Summe der Quadrate.
	Q03 x^2	
	Q04 +	
	Q05 RTN	Beendet verschachteltes Unterprogramm Q; Rückkehr zum "Haupt"-Unterprogramm S.

Verzweigungen (GTO)

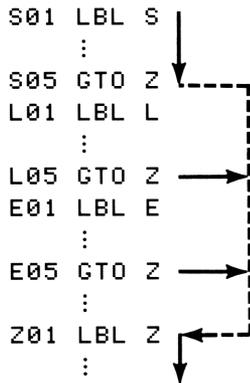
Wie Sie bei der Anwendung von Unterprogrammen bereits gesehen haben, ist es oft wünschenswert, die Programmausführung an einer anderen Stelle als der unmittelbar folgenden Programmzeile fortzusetzen. Dieser Vorgang wird als *Verzweigen* bezeichnet.

Unbedingte Verzweigungen verwenden die Anweisung GTO (*Go TO*), um zu einem *Programm-Label* zu verzweigen. Es ist nicht möglich, während des Programmablaufs zu einer bestimmten Programmzeile zu verzweigen.

■ Label

Programmierte GTO Anweisung. Die Anweisung GTO Label verlangt die Programmausführung an die Programmzeile, welche das jeweilige Label enthält. Von dieser Zeile aus wird das Programm fortgesetzt, ohne daß eine automatische Rückkehr an den Ausgangspunkt der Verzweigung erfolgt (daher nicht für Unterprogramme geeignet).

Betrachten Sie z.B. das Kurvenanpassungsprogramm auf Seite 204 in Teil 4. Die Anweisung GTO Z bewirkt eine Verzweigung von jeder der drei unabhängigen Initialisierungsroutinen zu LBL Z—der Routine, welche den gemeinsamen Eingangspunkt zum Hauptteil des Programms bildet:



Verwenden von ■ über Tastenfeld. Wenn die Programmeingabe *nicht* aktiviert wurde (keine Programmzeilen angezeigt; **PRGM** aus), können Sie ■ zum Verschieben des Programmzeigers an ein spezifiziertes Label oder eine Zeilennummer benutzen, *ohne* dabei das Programm zu starten.

- Zu einem Label: ■ Label (Beispiel: ■ A)
- Zu einer Zeilennummer: ■ Label nn (nn < 100. Beispiel: ■ A05)
- Zu PRGM TOP: ■

Vergleichsoperationen (TESTS)

Für Programmierzwecke gibt es 8 Vergleichsabfragen, welche über das TEST Menü zugänglich sind. Drücken von **TESTS** zeigt nachstehende Testkategorien:



Denken Sie daran, daß x sich auf den Inhalt des X-Registers und y sich auf den Inhalt des Y-Registers bezieht. Es wird *nicht* die Variable X mit der Variablen Y verglichen.

Wählen Sie die gewünschte Kategorie und drücken Sie anschließend die Menütaste für den entsprechenden Vergleich:

Menütasten in TESTS

$\{x?y\}$	$\{x?0\}$
$\{\neq y\}$ für $x \neq y?$	$\{\neq 0\}$ für $x \neq 0?$
$\{> y\}$ für $x > y?$	$\{< 0\}$ für $x < 0?$
$\{> y\}$ für $x > y?$	$\{> 0\}$ für $x > 0?$
$\{= y\}$ für $x = y?$	$\{= 0\}$ für $x = 0?$

Obwohl Sie sich diese Menüs auch außerhalb der Programmeingabe anzeigen lassen können, haben diese Funktionen außerhalb von Programmen keinen Anwendungszweck.

Zum Beispiel: Das Programm für quadratische Gleichungen auf Seite 191 verwendet die Vergleiche $x=0?$ und $x<0?$ in Routine Q.

```

Q01 LBL Q
Q02 INPUT A
Q03  $x=0?$ 
Q04 GTO Q
Q05 INPUT B
    :
```

Überprüft die Zulässigkeit von A, welches nicht Null sein darf.
 Wenn $A = 0$, erneuter Programmstart.
 Wenn $A \neq 0$, Fortsetzung des Programms.

Die Zeilen Q14 bis Q19 berechnen $B^2 - 4AC$. Die nachfolgenden Zeilen testen, ob ein negativer Wert vorliegt (welcher zu einer imaginären Nullstelle führen würde).

```
      :  
Q20  x<0?           Ergebnis negativ?  
Q21  GTO I          Wenn ja, verzweige zu anderer Routine.  
Q22  SQRT           Wenn positiv, ziehe Wurzel.  
      :
```

Flags

Ein Flag ist ein Statusindikator. Er ist entweder *gesetzt (wahr)* oder *gelöscht (falsch)*. Das *Testen eines Flags* ist eine weitere Vergleichsoperation, die der "Do if true" Regel folgt: Die Programmausführung wird sequentiell fortgesetzt, wenn der Flag gesetzt ist, oder es wird eine Zeile übersprungen, wenn der Flag gelöscht ist.

Bedeutung von Flags. Der HP-32S besitzt sieben Flags, von 0 bis 6 durchnummeriert. Jeder dieser Flags kann durch eine entsprechende Anweisung gesetzt, gelöscht oder getestet werden, wobei das Setzen und Löschen auch vom Tastenfeld aus möglich ist.*

- Die Flags 0, 1, 2, 3 und 4 haben keine vordefinierte Bedeutung, d.h. deren Status kann zur Kennzeichnung einer beliebigen Situation dienen (siehe nachstehendes Beispiel).
- Ist Flag 5 gesetzt, so wird der Programmablauf unterbrochen, wenn im Programm ein Überlauf eintritt, und es wird OVERFLOW und ▲ angezeigt.† Ist Flag 5 gelöscht, so erfolgt keine Programmunterbrechung, aber es wird kurz OVERFLOW angezeigt, wenn das Programm anhalten sollte.
- Flag 6 wird *automatisch* vom Rechner gesetzt, wenn ein Bereichsüberlauf eintritt (Sie können trotzdem Flag 6 auch selbst setzen). Er hat keine besondere Auswirkung, außer daß der Status abgefragt werden kann.

* Die einzige andere Operation, welche das Löschen von Flags zur Folge hat, ist das Löschen des Speicherbereichs. Sehen Sie dazu in Anhang B nach.

† Ein *Overflow* tritt ein, wenn das Ergebnis einer Rechenoperation die größte im Rechner darstellbare Zahl überschreitet. In diesem Fall wird das Ergebnis durch die größte Zahl substituiert.

Die Flags 5 und 6 ermöglichen Ihnen die Steuerung von Überlauf-Situationen, welche während des Programmablaufs eintreten können. Das Setzen von Flag 5 bewirkt einen Programmstopp gleich nach der Zeile, welche den Bereichsüberlauf verursachte. Durch Testen von Flag 6 in einem Programm können Sie bei jedem Überlauf den Programmablauf oder das Ergebnis ändern.

Indikatoren für gesetzte Flags. Flags 0, 1, 2 und 3 haben korrespondierende Indikatoren in der Anzeige, welche angezeigt werden, sofern der jeweilige Flag gesetzt ist. Die Anzeige (bzw. das Fehlen) von **0**, **1**, **2** oder **3** teilt Ihnen jederzeit mit, ob einer der vier Flags gesetzt oder gelöscht ist. Allerdings gibt es diese Einrichtung nicht für die Flags 4, 5 und 6; deren Status kann nur über eine programmierte FS? Anweisung ermittelt werden. (Sehen Sie dazu "Testen von Flags (FS?)" unten.)

Funktionen für Flags. Drücken von  **FLAGS** bewirkt die Anzeige des FLAGS Menüs:

SF CF FS?

Nachdem Sie die gewünschte Funktion gewählt haben, werden Sie zur Eingabe der Flagnummer (0–6) aufgefordert. Um z.B. Flag 0 zu setzen, ist  **FLAGS** {SF} 0 zu setzen.

FLAGS Menü

Menütaste	Bedeutung
{SF} <i>n</i>	Setze Flag. Setzt Flag <i>n</i> .
{CF} <i>n</i>	Lösche Flag. Löscht Flag <i>n</i> .
{FS?} <i>n</i>	Flag gesetzt? Testet Status von Flag <i>n</i> .

Testen von Flags (FS?). Das Testen eines Flags stellt eine Abfrage dar, genauso wie eine Vergleichsoperation. Die Anweisung FS? *n* überprüft, ob der jeweilige Flag gesetzt oder gelöscht ist. Wenn er gesetzt ist, dann wird die nächste Zeile ausgeführt—ansonsten wird sie übersprungen. Diese entspricht der "Do if True" Regel, welche auf Seite 95 unter "Bedingte Funktionen" erläutert ist.

Obwohl Sie {FS?} nicht nur während der Programmeingabe ausführen können, ist das Testen von Flags nur innerhalb eines Programms sinnvoll.

Sie sollten sich beim Programmieren zur Gewohnheit machen, für das Testen von Flags von einem bekannten Status auszugehen. Der aktuelle Status eines Flags hängt davon ab, wie vorangehende Programme abgelaufen und welche Auswirkungen diese auf den jeweiligen Flag gehabt haben. Sie dürfen nicht *annehmen*, daß z.B. der entsprechende Flag gelöscht ist und daß er nur innerhalb des Programms gesetzt werden kann. *Stellen Sie sicher*, daß der Flag gelöscht ist (durch eine CF Anweisung), bevor die Bedingung eintritt, welche das Setzen des Flags zur Folge hat. Sehen Sie dazu nachstehendes Beispiel:

Beispiel: Verwenden von Flags. Das Programm zum Lösen einer quadratischen Gleichung auf Seite 191 in Teil 4 verwendet Flag 0 in Verbindung mit dem Vergleich $x < 0?$, um das Vorzeichen von B festzustellen. Beachten Sie, daß Zeile Q11 Flag 0 löscht, um sicherzustellen, daß er nur für den beabsichtigten Zweck gesetzt ist.

Q11	CF 0	Stellt sicher, daß Flag 0 gelöscht ist.
Q12	$x < 0?$	Ist B (im X-Register) negativ?
Q13	SF 0	Setzt Flag 0, <i>wenn</i> B negativ ist.
	:	
Q23	FS? 0	Ist Flag 0 gesetzt (ist B negativ)?
Q24	+/-	Wenn ja, ändere Vorzeichen.
Q25	+	Führe in jedem Fall Addition aus.
	:	

Andere Programme in Teil 4, die von Flags Gebrauch machen, sind "Kurvenanpassung" und "Einheitenkonvertierungen". Beide verwenden Flags zur Abfrage, welche Bedingung der Anwender lösen möchte (welche Art von Kurve, welcher Konvertierungstyp) und steuern dadurch die jeweilige Option bzw. Berechnung.

Programmschleifen (GTO, LOOP)

Das Zurückspringen auf ein Label in einer vorhergehenden Zeile ermöglicht die mehrfache Ausführung eines Programmteils. Dieses Verfahren wird auch als *Programmschleife* bezeichnet.

```

D01 LBL D
D02 INPUT M
D03 INPUT N
D04 INPUT T
D05 GTO D

```

Diese Routine (Teil des Programms "Koordinatentransformationen" auf Seite 198 in Teil 4) ist ein Beispiel für eine *unbestimmte Schleife*. Sie wird dazu benutzt, die Anfangswerte vor Ausführung der Koordinatentransformation zu sammeln. Nach der Eingabe der drei Werte bleibt es dem Anwender überlassen, die Schleife zu unterbrechen, indem die gewünschte Konvertierung gewählt wird (Drücken von XEQ) N für Altes-nach-neuem-System oder XEQ) A für Neues-nach-altem-System).

Bedingte Schleifen (GTO)

Wenn Sie eine Operation so lange ausführen möchten, bis eine bestimmte Bedingung erfüllt ist (wobei nicht bekannt ist, wie oft die Schleife dabei wiederholt werden muß), können Sie eine Schleife mit einer Abfrage und einer GTO Anweisung erzeugen.

Zum Beispiel benutzt die nachstehende Routine eine Schleife zur Reduzierung des Wertes A um einen konstanten Betrag B , bis A kleiner oder gleich B ist.

Programmzeilen:	Beschreibung
A01 LBL A	
A02 INPUT A	
A03 INPUT B	
S01 LBL S	
S02 RCL A	Das Abrufen von A ist einfacher als das Kontrollieren, wo es sich im Stack aufhält.
S03 RCL- B	Berechnet $A - B$.
S04 STO A	Ersetzt altes A mit neuem Wert.
S05 RCL B	Ruft Konstante für Vergleich ab.
S06 <?>	Ist $B < \text{neues } A$?
S07 GTO S	Ja: Rücksprung, um Subtraktion zu wiederholen.
S08 VIEW A	Nein: Zeigt neuen Inhalt von A an.
S09 RTN	

Schleifen mit Zähler (DSE, ISG)

Wenn Sie genau wissen, wie oft eine Schleife durchlaufen werden soll, stehen Ihnen zwei Spezialfunktionen im LOOP Menü (■ LOOP) zur Verfügung: DSE (*decrement; skip if less than or equal to*) reduziert bei jedem Schleifendurchgang einen Schleifenzähler und überspringt die nächste Programmzeile, wenn der Zähler einen bestimmten Vergleichswert erreicht oder unterschreitet. ISG (*increment; skip if greater than*) erhöht bei jedem Schleifendurchgang einen Schleifenzähler und überspringt die nächste Programmzeile, wenn der Zähler einen Vergleichswert überschreitet.

Verwenden Sie für eine absteigende Schleife:

■ LOOP {DSE} Variable

Verwenden Sie für eine aufsteigende Schleife:

■ LOOP {ISG} Variable

Diese Funktionen entsprechen einer FOR-NEXT Schleife in BASIC:

```
FOR Variable = Anfangswert TO Endwert STEP Schrittweite
  :
NEXT Variable
```

Eine DSE Anweisung ist äquivalent zu einer FOR-NEXT Schleife mit einer negativen Schrittweite.

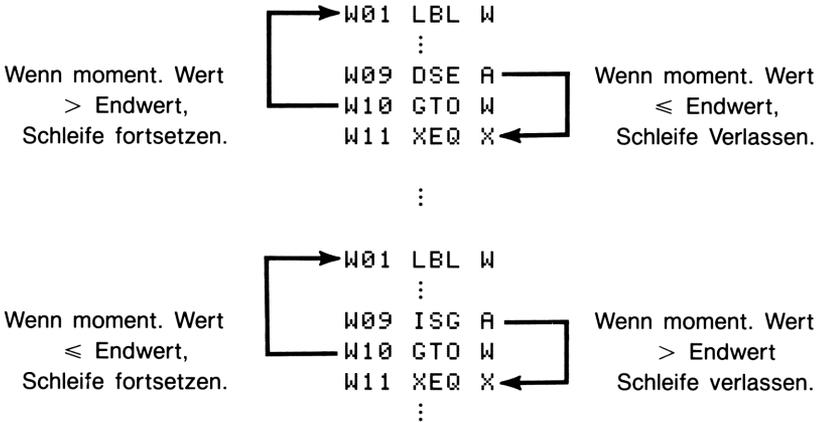
Nachdem die Menütaste für {DSE} oder {ISG} gedrückt wurde, werden Sie zur Eingabe einer Variablen aufgefordert, welche die *Schleifensteuerzahl* darstellt.

Die Schleifensteuerzahl. Die spezifizierte Variable dient zur Speicherung der Schleifensteuerzahl, welche das Format $\pm cccccc,fffii$ besitzt, wobei:

- $\pm cccccc$ ist der momentane Wert des Zählers (1 bis 12 Stellen); er *ändert* sich während der Schleifenausführung.
- *fff* ist der Endwert des Zählers (muß 3-stellig sein); er bleibt während der Schleifenausführung *gleich*.
- *ii* ist die Schrittweite (muß 2-stellig sein oder nicht spezifiziert); sie *ändert* sich *nicht*. Wird kein Wert spezifiziert, dann benutzt der Rechner den Wert 01 für *ii*.

Bei einer gegebenen Schleifensteuerzahl $cccccc,ffffi$ reduziert DSE $cccccc$ auf $cccccc - ii$, vergleicht das neue $cccccc$ mit fff und setzt die Programmausführung mit der übernächsten Programmzeile fort, falls $cccccc \leq fff$.

Bei einer gegebenen Schleifensteuerzahl $cccccc,ffffi$ erhöht ISG $cccccc$ auf $cccccc + ii$, vergleicht das neue $cccccc$ mit fff und setzt die Programmausführung mit der übernächsten Programmzeile fort, falls $cccccc > fff$.



Zum Beispiel bedeutet eine Schleifensteuerzahl von 0,050 für ISG: Der Zähler läuft von 0 bis 50 und wird je Schleifendurchlauf um 1 erhöht.

Das nachstehende Programm verwendet ISG zum 10-maligen Durchlaufen einer Schleife. Die Schleifensteuerzahl (0000001,01000) ist in der Variablen Z gespeichert. Führende und nachfolgende Nullen können weggelassen werden.

```

L01 LBL L
L02 1,01
L03 STO Z
M01 LBL M
M02 ISG Z
M03 GTO M
M04 RTN

```

Drücken Sie **VIEW** Z zur Anzeige der Schleifensteuerzahl, welche inzwischen den Wert 11,0100 angenommen hat.

Indirekte Adressierung von Variablen und Labels

Indirekte Adressierung wird bei fortgeschrittener Programmierung zum Spezifizieren einer Variablen oder eines Labels verwendet, *ohne vorab den Inhalt/Variablennamen festzulegen*. Dies erfolgt erst beim Programmablauf, z.B. durch ein Zwischenergebnis (oder eine Eingabe).

Indirekte Adressierung benutzt zwei verschiedene Tasten: \boxed{i} (mit $\boxed{\cdot}$) und $\boxed{(i)}$ (mit $\boxed{R/S}$).^{*} Diese Tasten sind für viele Funktionen aktiv, welche A bis Z als Variablen oder Labels verwenden.

- i ist eine Variable, deren Inhalt auf eine *andere* Variable oder Label Bezug nehmen kann. Wie jede andere Variable (von A bis Z) enthält sie nur eine Zahl.
- $\boxed{(i)}$ ist eine Programmierungsfunktion mit der Anweisung: "Verwende die Zahl in i zur Bestimmung, welche Variable oder welches Label adressiert werden soll". Dies stellt eine *indirekte Adresse* dar. (A bis Z sind *direkte Adressen*.)

Beide Variablen, \boxed{i} und $\boxed{(i)}$, werden zusammen zum Erzeugen einer indirekten Adresse benutzt (siehe Beispiel unten). Nur für sich betrachtet ist i eine Variable wie jede andere. Für sich allein ist $\boxed{(i)}$ entweder undefiniert (keine Zahl in i) oder unkontrolliert (Verwendung, was zufällig in i geblieben ist).

Die Variable "i"

Der Inhalt von i kann gespeichert, zurückgerufen und manipuliert werden, wie jeder andere Variableninhalt. Sie können sogar nach i lösen und über i integrieren.

Funktionen, die i direkt verwenden

STO i	INPUT i	DSE i
RCL i	VIEW i	ISG i
STO $+, -, \times, \div i$	\int FN i	
RCL $+, -, \times, \div i$	SOLVE i	

^{*} Die Variable i ist völlig unabhängig von $\boxed{(i)}$ oder der Variablen i .

Indirekte Adresse, (i)

Viele Funktionen, die A bis Z (als Variablen oder Labels) verwenden, können $\boxed{(i)}$ zur *indirekten* Bezugnahme auf A bis Z (Variablen oder Labels) verwenden. Die Funktion $\boxed{(i)}$ benutzt den Wert in *i* zur Bestimmung, welche Variable oder welches Label zu adressieren ist:

Indirekte Adressierung

Inhalt von <i>i</i> :	Durch (i) definierte Adresse:
± 1	Variable A oder Label A
\vdots	\vdots
± 26	Variable Z oder Label Z
≥ 27 oder ≤ -27 oder 0	Fehler: INVALID (i)

Es wird nur der Absolutbetrag des ganzzahligen Teils des Wertes in *i* zur Adressierung verwendet.

Nachstehend finden Sie die Funktionen, welche (i) als Adresse benutzen können. Für GTO, XEQ und FN= verweist (i) auf ein Label; für alle anderen Funktionen wird auf eine Variable verwiesen.

Funktionen, die (i) für indirekte Adressierung verwenden:

STO(i)	INPUT(i)
RCL(i)	VIEW(i)
STO+, -, ×, ÷ (i)	DSE(i)
RCL+, -, ×, ÷ (i)	ISG(i)
XEQ(i)	SOLVE(i)
GTO(i)	fFN(i)
	FN=(i)

Programmsteuerung mit (i)

Da der Inhalt von i sich bei jedem Programmablauf ändern kann—sogar in verschiedenen Teilen des gleichen Programms—kann eine Programmanweisung wie `GTO(i)` zu unterschiedlichen Zeitpunkten zu verschiedenen Labels verzweigen. Durch das Offenlassen (bis zum Programmablauf), welche Variable oder welches Label benötigt wird, ist eine gewisse Flexibilität gewährleistet (siehe erstes Beispiel unten).

Indirekte Adressierung ist sehr hilfreich bei der Steuerung von Schleifen. Die Variable i dient als *Index*, der die Adresse der Variable enthält, welche die Schleifensteuerzahl für die Funktionen DSE und ISG enthält (siehe zweites Beispiel unten).

Beispiel: Wählen eines Unterprogramms mit (i). Das "Kurvenanpassungsprogramm" auf Seite 204 in Teil 4 verwendet indirekte Adressierung zur Bestimmung des Kurvenmodells. (Unterschiedliche Unterprogramme berechnen x und y für die verschiedenen Modelle.) Beachten Sie, daß i gespeichert wird und später in weit auseinander liegenden Teilen des Programms indirekt adressiert wird.

Die ersten vier Routinen (S, L, E, P) des Programms spezifizieren die Anpassungsmodelle und weisen jedem eine Zahl zu (1, 2, 3, 4). Diese Zahl wird in Routine Z, der gemeinsamen Eingangsstelle für alle Modelle, gespeichert.

```
Z03 STO i
```

Routine Y verwendet i zum Aufruf des entsprechenden Unterprogramms zur Berechnung der x - und y -Näherungen. Zeile Y03 ruft das Unterprogramm zur Berechnung von \hat{y} auf:

```
Y03 XEQ(i)
```

und Zeile Y08 ruft ein anderes Unterprogramm zum Berechnen von \hat{x} auf, nachdem i um 6 erhöht wurde:

```
Y06 6  
Y07 STO+ i  
Y08 XEQ(i)
```

Wenn i :	Aufruf durch XEQ(i):	Zum:
1	LBL A	Berechnen von y für lineares Modell.
2	LBL B	Berechnen von y für logarithmisches Modell.
3	LBL C	Berechnen von y für exponentielles Modell.
4	LBL D	Berechnen von y für Potenz-Modell.
7	LBL G	Berechnen von x für lineares Modell.
8	LBL H	Berechnen von x für logarithmisches Modell.
9	LBL I	Berechnen von x für exponentielles Modell.
10	LBL J	Berechnen von x für Potenz-Modell.

Beispiel: Schleifensteuerung mit (i). Im Programm "Lösen eines linearen Gleichungssystems über Determinantenverfahren" auf Seite 175 in Teil 4 wird ein Indexwert in i verwendet. Dieses Programm benutzt die Schleifenanweisungen ISG i und DSE i in Verbindung mit den indirekten Anweisungen RCL(i) und STO(i) zum Auffüllen und manipulieren einer Matrix.

Routine A stellt den ersten Teil des Programms dar, in welchem die ursprüngliche Schleifensteuerzahl in i gespeichert wird.

Programmzeilen:	Beschreibung
A01 LBL A	Beginn für Dateneingabe.
A02 1,012	Schleifensteuerzahl: Durchlaufe Schleife von 1 bis 12 mit einer Schrittweite von 1.
A03 STO i	Speichert Schleifensteuerzahl in i .

Die nächste Routine ist L, eine Schleife zur Eingabe aller 12 bekannten Werte einer 3×3 Koeffizientenmatrix (Variablen A-I) und den drei Konstanten (J-L) für die Gleichungen.

Programmzeilen:

L01 LBL L

L02 INPUT(i)

L03 ISG i

L04 GTO L

L05 GTO A

Beschreibung

Diese Routine sammelt alle bekannten Werte in drei Gleichungen.

Eingabeaufforderung für eine Zahl und Speicherung derselben in der durch i adressierten Variablen.

Addiert 1 zu i und wiederholt die Schleife, bis i 13,012 erreicht hat.

Wenn i den Endwert des Zählers überschreitet, verzweigt das Programm zurück zu A.

Teil 3

Fortgeschrittene Operationen

Seite 110	7: Lösen einer Unbekannten in Gleichungen
126	8: Numerische Integration
137	9: Operationen mit komplexen Zahlen
144	10: Rechnen in verschiedenen Zahlensystemen
153	11: Statistikberechnungen

7

Lösen einer Unbekannten in Gleichungen

Die SOLVE Routine dient zum Lösen einer beliebigen Unbekannten in einer Gleichung. Betrachten Sie z.B. die Funktion

$$x^2 - 3y.$$

Diese Funktion kann gleich Null gesetzt werden, wodurch eine Gleichung erzeugt wird:

$$x^2 - 3y = 0.*$$

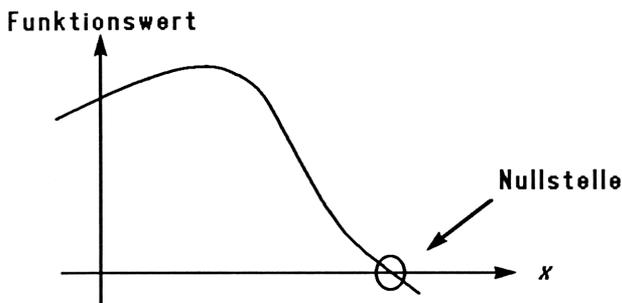
Wenn Sie den Wert von y in dieser Gleichung kennen, dann kann SOLVE die Unbekannte x lösen. Kennen Sie den Wert von x , so ist es SOLVE möglich, die Unbekannte y zu lösen. Dies funktioniert für "Textaufgaben" genauso:

$$\begin{aligned} \text{Aufschlag} \times \text{Kosten} &= \text{Preis} \\ \text{Aufschlag} \times \text{Kosten} - \text{Preis} &= 0. \end{aligned}$$

Sind zwei der drei Variablen bekannt, so können Sie über SOLVE die dritte Variable (Unbekannte) lösen.

Wenn die Gleichung nur eine Variable enthält, oder wenn für alle Variablen bis auf eine die Werte vorgegeben sind, dann bedeutet das Lösen für x das Auffinden der Nullstelle(n) der Gleichung. Eine *Nullstelle* tritt an der Stelle auf, wo der Graph der Funktion die x -Achse schneidet. Der Funktionswert nimmt hier den Wert Null an.

* Eigentlich können Sie die Funktion mit jedem reellen Wert gleichsetzen, wie zum Beispiel $x^2 - y = 10$. Dies kann dann als $x^2 - y - 10 = 0$ ausgedrückt werden, wenn SOLVE verwendet werden soll.



Anwenden von SOLVE

Um eine unbekannte Variable zu lösen:

1. Geben Sie ein Programm ein, welches die Funktion definiert. (Siehe "Entwickeln von Programmen für SOLVE", nachfolgend.)
2. Wählen Sie das Programm, das die zu lösende Funktion enthält. Drücken Sie **▀** **SOLVE/∫** {FN} *Label*.
3. Lösen Sie die Unbekannte. Drücken Sie **▀** **SOLVE/∫** {SOLVE} *Variable*.

Durch Drücken von **C** oder **R/S** können Sie eine laufende Berechnung anhalten.

Anfangsnäherungen. Bei bestimmten Funktionen ist es empfehlenswert, ein oder zwei *Anfangsnäherungen* (in der Variablen und im X-Register) vorzugeben, bevor die Berechnung gestartet wird. Sie können dadurch eine kürzere Rechenzeit erreichen, das Ergebnis in einem reellen Lösungsbereich suchen, und mehrere Lösungen—sofern vorhanden—auffinden. Sehen Sie dazu "Wählen von Anfangsnäherungen für SOLVE" auf Seite 120.

Ergebnisse. Das X-Register und die Variable selbst enthalten die Endnäherungen der Nullstelle; das Y-Register enthält die vorangehende Näherung und das Z-Register enthält den Funktionswert an der letzten Näherung für die Nullstelle (welcher Null sein sollte).

Für sehr schwierige mathematische Bedingungen kann keine definitive Lösung gefunden werden (siehe "Interpretieren von Ergebnissen" und "Wenn SOLVE keine Nullstelle finden kann" in Anhang C).

Lösen einer anderen Unbekannten in derselben Gleichung:

Geben Sie einfach die neue Unbekannte vor: \blacksquare [SOLVE/] {SOLVE} Variable. Es wird wieder das gleiche Programm verwendet, welches zuletzt spezifiziert wurde (FN= Label).

Entwickeln von Programmen für SOLVE

Bevor Sie eine unbekannte Variable berechnen können, müssen Sie ein Programm (oder Unterprogramm) schreiben, welches die Funktion* auswertet.

Schreiben einer Funktion für eine Gleichung. Vereinfachen Sie zuerst die Gleichung, indem Sie alle gleichen Variablen und Konstanten zusammenfassen. Bringen Sie danach alle Terme auf eine Seite, womit nur 0 auf der anderen Seite übrig bleibt.

Zum Beispiel wird das Volumen eines Quaders durch

$$\text{Länge} \times \text{Breite} \times \text{Höhe} = \text{Volumen}$$

bestimmt. Umordnen der Terme führt zu

$$\text{Länge} \times \text{Breite} \times \text{Höhe} - \text{Volumen} = 0, \text{ oder}$$

$$L \times B \times H - V = 0.$$

Schreiben eines Programms zur Auswertung einer Funktion:

1. Beginnen Sie mit einem Label, damit das Programm von SOLVE aufgerufen werden kann.
2. Schließen Sie eine INPUT Anweisung für jede Variable mit ein (einschließlich der Unbekannten). (Kommt nur eine Variable in der Funktion vor, so lassen Sie INPUT weg, da die Anweisung für die Unbekannte sowieso ignoriert wird.)†

* Es sind nur reelle Zahlen für SOLVE zulässig. Liegt jedoch eine komplexe Funktion vor, welche so umgeschrieben werden kann, daß die Real- und Imaginärteile isoliert sind, dann kann SOLVE die Lösung getrennt berechnen.

† Die INPUT Anweisungen sind bei Funktionen mit mehreren Variablen sinnvoll und hilfreich. Da INPUT für die Unbekannte ignoriert wird, brauchen Sie *nur ein* Programm zur Eingabe *aller* Variablen schreiben. Sie können das gleiche Programm immer wieder verwenden, unabhängig von der jeweils gewählten Unbekannten.

3. Geben Sie die Anweisungen zur Auswertung der Funktion ein. Verwenden Sie jeweils eine RCL Anweisung, wenn ein Variableninhalt für eine Berechnung benötigt wird.
4. Beenden Sie das Programm mit RTN. Das Programm sollte dabei so abgeschlossen werden, daß der Funktionswert sich im X-Register (der Anzeige) befindet.

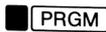
Bei jeder Ausführung Ihres Programms durch SOLVE ändert sich der Wert der Unbekannten, ebenso der vom Programm erzeugte Funktionswert. Wenn das Programm Null zurückgibt, dann wurde eine Lösung für die Unbekannte gefunden.

Beispiele zur Anwendung von SOLVE

Beispiel: Bestimmen der Seitenlänge einer Schachtel. Verwenden Sie das nachstehende Programm zur Berechnung der Dimensionen einer Schachtel ($L \times B \times H - V$). Beachten Sie, daß das Programm *Rückrufarithmetik* verwendet, welche weniger Speicherplatz erfordert, als durch das Aufrufen einer Variablen und das Ausführen arithmetischer Berechnungen als separate Operationen erforderlich wäre.

```
S01 LBL S
S02 INPUT L
S03 INPUT B
S04 INPUT H
S05 INPUT V
S06 RCL L
S07 RCL× B
S08 RCL× H
S09 RCL- V
S10 RTN
```

Geben Sie zuerst das mit S benannte Programm ein:

Tastenfolge:	Anzeige	Beschreibung
		Beginnt Programmeingabe. Verschiebt Programmzeiger an Speicheranfang (falls notwendig).
	PRGM TOP	

<input type="checkbox"/> <input type="checkbox"/> LBL/RTN	{LBL} S	S01 LBL S
<input type="checkbox"/> <input type="checkbox"/> INPUT	L	S02 INPUT L
<input type="checkbox"/> <input type="checkbox"/> INPUT	B	S03 INPUT B
<input type="checkbox"/> <input type="checkbox"/> INPUT	H	S04 INPUT H
<input type="checkbox"/> <input type="checkbox"/> INPUT	V	S05 INPUT V
<input type="checkbox"/> <input type="checkbox"/> RCL	L	S06 RCL L
<input type="checkbox"/> <input type="checkbox"/> RCL	<input type="checkbox"/> x B	S07 RCL x B
<input type="checkbox"/> <input type="checkbox"/> RCL	<input type="checkbox"/> x H	S08 RCL x H
<input type="checkbox"/> <input type="checkbox"/> RCL	<input type="checkbox"/> - V	S09 RCL - V
<input type="checkbox"/> <input type="checkbox"/> LBL/RTN	{RTN}	S10 RTN

Eingabe der Programmzeilen.

An dieser Stelle enthält das X-Register den Funktionswert für $L \times B \times H - V$.

C

Beendet Programmeingabe. Zeigt Inhalt des X-Registers an.

Lösen Sie das Volumen einer Schachtel, welche 25 cm lang, 10 cm breit und 8,5 cm hoch ist. Anschließend wird die gleiche Funktion zum Lösen einer anderen Variablen benutzt.

SOLVE/f {FN} FN= _

Eingabeaufforderung für das Label des Programms, das die Funktion definiert.

S

Spezifiziert S.

SOLVE/f {SOLVE} SOLVE _

Eingabeaufforderung für die Unbekannte.

V L?Wert
 25 R/S B?Wert
 10 R/S H?Wert
 8,5

Startet Programm S; Eingabeaufforderung für alle Daten, *außer* V.

R/S SOLVING
 V=2.125,0000

Das Volumen beträgt $2\,125\text{ cm}^3$.

Bestimmen Sie nun die Länge der Schachtel, wenn das Volumen verändert wird (auf $3\,000\text{ cm}^3$), Höhe und Breite jedoch gleich bleiben. Denken Sie daran, daß Sie das Programm-Label erneut spezifizieren müssen, da das zuletzt benutzte Label wieder verwendet werden soll.

■ **SOLVE/f** {SOLVE}

L B?10,0000

Startet Programm S zum Lösen von L. Eingabe der Unbekannten.

R/S H?8,5000

R/S V?2.125,0000

3000 **R/S** L=35,2941

Um einen Wert beizubehalten, ist lediglich **R/S** zu drücken.

Beispiel: Gleichung für gleichförmige Bewegung. Die Gleichung für die Bewegung eines freifallenden Objekts lautet:

$$s = v_0 t + \frac{1}{2} g t^2$$

wobei s die Strecke, v_0 die Anfangsgeschwindigkeit, t die Zeit und g die Fallbeschleunigung darstellt. Durch Gleichsetzen mit Null und eine einfache Umformung erhalten Sie:

$$0 = t(v_0 + gt/2) - s.$$

Das nachstehende Programm wertet diese Funktion aus:

```
M01 LBL M
M02 INPUT V
M03 INPUT T
M04 INPUT G
M05 INPUT S
M06 RCL G
M07 2
M08 ÷
M09 RCL× T
M10 RCL+ V
M11 RCL× T
M12 RCL- S
M13 RTN
```

Die Fallbeschleunigung g ist als Variable mit aufgenommen, um Ihnen das Arbeiten mit unterschiedlichen Einheiten zu ermöglichen:

$$g = 9,8 \text{ m/s}^2 = 32,2 \text{ ft/s}^2.$$

Geben Sie obiges Programm (LBL M) ein. Berechnen Sie, wie tief ein Objekt innerhalb von 5 Sekunden fällt, ausgehend vom Ruhezustand.

Tastenfolge:	Anzeige	Beschreibung
\blacksquare $\boxed{\text{SOLVE}/f}$ {FN} M	FN= M (<i>kurz</i>)	Spezifiziert LBL M für die Funktion.
\blacksquare $\boxed{\text{SOLVE}/f}$ {SOLVE} S	V?3,000,0000	Spezifiziert S als Unbekannte. Eingabeaufforderung für V, während der seitherige Inhalt von V angezeigt wird (aus letztem Beispiel).
0 $\boxed{R/S}$	T? Wert	
5 $\boxed{R/S}$	G? Wert	
9,8 $\boxed{R/S}$	S=122,5000	Gesuchte Strecke in Meter.

Versuchen Sie eine andere Berechnung mit der gleichen Gleichung: Wie lange dauert es, bis ein Objekt 500 Meter tief fällt? Da v_0 und g bereits gespeichert sind, müssen Sie diese Werte nicht mehr eingeben.

\blacksquare $\boxed{\text{SOLVE}/f}$ {SOLVE} T	V?0,0000	Spezifiziert eine neue Unbekannte; Eingabeaufforderung für V.
$\boxed{R/S}$	G?9,8000	
$\boxed{R/S}$	S?122,5000	
500 $\boxed{R/S}$	T=10,1015	Gesuchte Zeit in Sekunden.

Beispiel: Auffinden der Nullstellen einer Gleichung. Betrachten Sie die Gleichung

$$x^3 - 5x^2 - 10x = -20.$$

Das Umformen der Gleichung führt zu

$$x^3 - 5x^2 - 10x + 20 = 0.$$

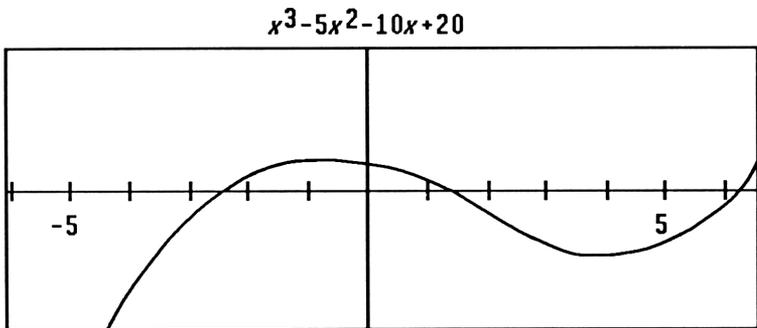
Das Hornerische Schema (siehe Kapitel 5) vereinfacht die Gleichung, um weniger Speicherplatz zu belegen:

$$x(x(x - 5) - 10) + 20 = 0.$$

Das nachstehende Programm wertet diese Funktion aus:

```
N01 LBL N
N02 RCL X
N03 5
N04 -
N05 RCL× X
N06 10
N07 -
N08 RCL× X
N09 20
N10 +
N11 RTN
```

Eine Abbildung dieser Funktion sieht wie folgt aus:



Die Abbildung läßt erkennen, daß drei Nullstellen existieren, da die Kurve dreimal die x -Achse schneidet. Der Rechner kann alle drei Nullstellen auffinden, wenn Sie SOLVE dreimal starten und dabei *unterschiedliche Anfangsnäherungen* vorgeben. (Weitere Informationen dazu finden Sie unter "Wählen von Anfangsnäherungen für SOLVE.")

Geben Sie obiges Programm ein (LBL N). Die Abbildung zeigt, daß die erste Nullstelle zwischen $x = -3$ und $x = -2$, die zweite Nullstelle zwischen 1 und 2, und die dritte Nullstelle zwischen 6 und 7 liegt. Speichern Sie jedes Paar der Anfangsnäherungen in der Variablen X und im X-Register und lösen Sie anschließend für X.

Tastenfolge:	Anzeige	Beschreibung
■ SOLVE/ {FN} N		Wählt Programm N.
3 +/- STO X	-3,0000	Erstes Paar von Anfangsnäherungen.
2 +/-	-2_	
■ SOLVE/ {SOLVE} X	X=-2,4433	Spezifiziert die Unbekannte; gibt erste Nullstelle zurück.
1 STO X	1,0000	Zweites Paar von Anfangsnäherungen.
2	2_	
■ SOLVE/ {SOLVE} X	X=1,3416	Die zweite Nullstelle.
6 STO X	6,0000	Drittes Paar von Anfangsnäherungen.
7	7_	
■ SOLVE/ {SOLVE} X	X=6,1017	Die dritte Nullstelle.

Wenn Sie *keine* Anfangsnäherungen eingegeben haben, dann konnte nur *eine* dieser Nullstellen aufgefunden werden; welche, hängt davon ab, was zufällig in der Variablen X und im X-Register gespeichert war, da der Rechner diese Werte immer als Anfangsnäherungen benutzt.

Verstehen und Steuern von SOLVE

SOLVE verwendet eine iterative Prozedur zum Auffinden einer Lösung für die unbekannte Variable. Die Prozedur substituiert zuerst zwei Anfangsnäherungen für die Unbekannte in der Gleichung. Basierend auf dem Ergebnis durch diese Näherungen erzeugt SOLVE eine neue, bessere Näherung. Dies wiederholt sich, bis SOLVE einen Wert gefunden hat, welcher Null als Funktionswert ergibt.

Einige Gleichungen sind schwieriger zu lösen als andere. Manchmal kann es erforderlich sein, daß Sie Anfangsnäherungen eingeben müssen, um eine Lösung aufzufinden (siehe "Wählen von Anfangsnäherungen für SOLVE", unten). Wenn SOLVE keine Nullstelle auffinden kann, wird NO ROOT FND angezeigt (siehe Anhang C).

Überprüfen des Ergebnisses

Nachdem SOLVE die Berechnung beendet hat, können Sie durch Ansehen der übrigen Werte im Stack überprüfen, ob das angezeigte Ergebnis tatsächlich eine Nullstelle von $f(x)$ darstellt:

- Die Anzeige (X-Register) und die Variable selbst enthalten die Lösung (Nullstelle) für die Unbekannte, d.h. der Wert, bei welchem sich $f(x) = 0$ ergibt.
- Das Y-Register (drücken Sie $\boxed{R\downarrow}$) enthält die vorangehende Näherung für die Nullstelle. Dieser Wert sollte identisch mit dem Wert im X-Register sein. Falls *nicht*, so ist die angezeigte Nullstelle nur eine *Approximation* der Nullstelle und die Werte im X- und Y-Register grenzen die Nullstelle ein. Diese Werte sollten sehr nahe zusammenliegen.
- Das Z-Register (drücken Sie erneut $\boxed{R\downarrow}$ zur Anzeige von Z) enthält den Wert von $f(x)$ für den gegebenen Wert im X-Register. Bei einer exakten Nullstelle muß dieser Wert Null sein. Falls nicht, so ist die angezeigte Nullstelle nur eine *Approximation* und dieser Wert sollte sehr nahe bei Null liegen.

Endet eine Berechnung mit `NO ROOT FND`, so bedeutet dies, daß die Berechnung nicht in Richtung einer Nullstelle konvergierte und die Werte im X- und Y-Register liegen wahrscheinlich nicht nahe beieinander. Sie können den Wert im X-Register—die Endnäherung der Nullstelle—durch Drücken von \boxed{C} oder \blacktriangledown (zum Löschen der Meldung) ansehen. Diese zwei Werte begrenzen das Intervall, das zuletzt bei der Suche nach einer Nullstelle benutzt wurde. Das Z-Register enthält den Wert von $f(x)$ für die Endnäherung der Nullstelle; er liegt wahrscheinlich nicht nahe bei Null.

Unterbrechen einer SOLVE Berechnung

Um die Berechnung zu unterbrechen, ist \boxed{C} oder $\boxed{R/S}$ zu drücken. Die momentan beste Näherung der Nullstelle ist in der Variablen gespeichert. Verwenden Sie $\blacksquare\boxed{VIEW}$ zum Ansehen, ohne dabei den Stackinhalt zu verändern. Möchten Sie die Berechnung fortsetzen, so drücken Sie $\boxed{R/S}$.*

* Drücken von \boxed{XEQ} , \boxed{GTO} , oder $\{RTN\}$ hebt die SOLVE Berechnung auf. In diesem Fall ist erneut das Programm zu starten, anstatt die Berechnung fortzusetzen.

Wählen von Anfangsnäherungen für SOLVE

Die zwei Anfangsnäherungen kommen von:

- Der Zahl, die momentan in der Unbekannten gespeichert ist.
- Die Zahl im X-Register (der Anzeige).

Diese Werte dienen als Anfangsnäherungen, *unabhängig davon, ob Sie selbst Werte vorgeben oder nicht*. Wenn Sie nur eine Näherung vorgeben und in der Variablen speichern, dann wird der gleiche Wert als zweite Näherung verwendet, da die Anzeige noch die soeben gespeicherte Zahl enthält. Das Eingeben Ihrer eigenen Schätzwerte hat folgende Vorteile:

- Durch Eingrenzen des Suchbereichs kann die Rechenzeit verkürzt werden.
- Sind mehrere Lösungen möglich, so können die Anfangsnäherungen bei der Wahl der gewünschten Lösung behilflich sein. So kann z.B. die Gleichung:

$$s = s_0 + v_0t + \frac{1}{2}gt^2$$

2 Lösungen für T besitzen. Sie können das physikalisch sinnvolle Ergebnis ($t > 0$) durch Vorgabe der geeigneten Anfangsnäherungen berechnen.*

- Erlaubt eine Gleichung nicht alle Werte für die Unbekannte, so können die richtigen Schätzwerte helfen, diese Werte zu vermeiden. Die Gleichung:

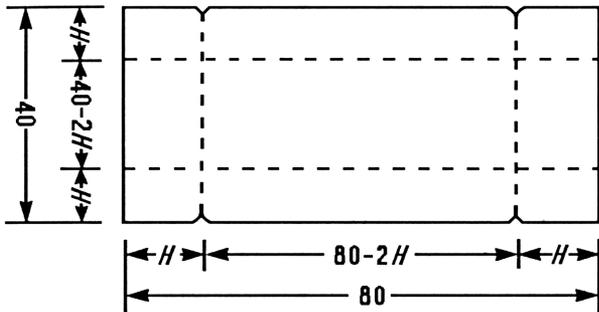
$$y = t + \log x$$

erlaubt z.B. keine Werte $x \leq 0$. Geeignete Schätzwerte helfen, daß SOLVE nicht auf die Fehler LOG(Ø) oder LOG(NEG) stößt.

Das Beispiel im vorherigen Abschnitt ("Auffinden der Nullstellen einer Gleichung") hatte Anfangsnäherungen zur Berechnung aller drei Nullstellen einer Gleichung verwendet. Nachstehend ein weiteres Beispiel, welches die Dimensionen einer Schachtel berechnet (analog zum Beispiel auf Seite 113), wobei jedoch mehrere Restriktionen zu beachten sind.

* Das Beispiel auf Seite 116 erforderte keine Eingabe von Anfangsnäherungen für t , da im ersten Teil ein Wert für T zur Berechnung von S gespeichert wurde. Der in T verbliebene Wert stellte eine gute (realistische) Näherung dar und wurde automatisch beim Lösen von T verwendet.

Beispiel: Falten einer Schachtel. Unter Verwendung eines rechteckigen Kartonstücks ($80 \text{ cm} \times 40 \text{ cm}$) ist eine nach oben geöffnete Schachtel mit einem Volumen von 7500 cm^3 zu bilden. Sie müssen dabei die Höhe bestimmen (das Maß, um welches jede Seite umgefaltet werden muß), welche zu dem gewünschten Volumen führt. Eine höhere Schachtel wird dabei einer kürzeren vorgezogen.



Wenn H die Höhe darstellt, dann ergibt sich die Länge der Schachtel zu $(80 - 2H)$ und die Breite zu $(40 - 2H)$. Das Volumen berechnet sich wie folgt:

$$V = (80 - 2H) \times (40 - 2H) \times H$$

wobei die mit Null gleichgesetzte Funktion die Form

$$\begin{aligned} 0 &= (80 - 2H) \times (40 - 2H) \times H - V \\ &= 4H [(40 - H) (20 - H)] - V \end{aligned}$$

annimmt.

Ein Programm, welches die umseitige Funktion definiert, wäre:

```
V01 LBL V
V02 INPUT H
V03 INPUT V
V04 40
V05 RCL- H
V06 20
V07 RCL- H
V08 ×
V09 4
V10 ×
V11 RCL× H
V12 RCL- V
V13 RTN
```

Es kann angenommen werden, daß entweder eine hohe, schmale Schachtel oder eine flache, weite Schachtel mit dem gewünschten Volumen hergestellt werden kann. Da eine höhere Schachtel bevorzugt wird, sollten größere Werte als Anfangsnäherungen für die Höhe gemacht werden. Andererseits sind größere Werte als 20 physikalisch nicht möglich, da der Karton nur 40 cm breit ist. Werte von 10 und 20 cm scheinen daher geeignet zu sein.

Tastenfolge:	Anzeige	Beschreibung
■ [SOLVE/] {FN} V		Wählt Programm V als die zu lösende Funktion.
10 [STO] H	10,0000	Speichert Ober- und Untergrenze.
20	20_	
■ [SOLVE/] {SOLVE}		Eingabeaufforderung für Volumen.
H	V?Wert	
7500 [R/S]	SOLVING H=15,0000	Dies ist die gewünschte Höhe.

Überprüfen Sie nun die Qualität dieses Ergebnisses—d.h., ob eine genaue Nullstelle vorliegt—indem Sie die Werte der vorhergehenden Näherung (im Y-Register) und $f(x)$ an der Nullstelle (im Z-Register) ansehen.

R↓

15,0000

Dieser Wert aus dem Y-Register stellt die letzte Näherung vor dem Endergebnis dar. Da die Zahlen gleich sind, liegt eine exakte Nullstelle vor...

R↓

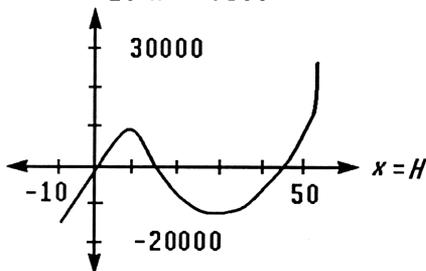
0,0000

...und, wie der Wert aus dem Z-Register zeigt, ist an der Nullstelle $f(x) = 0$.

$50 \times 10 \times 15$ cm ist die Dimension der gewünschten Schachtel. Wenn Sie die Obergrenze für die Höhe (20 cm) ignoriert und die Anfangsnäherungen von 30 und 40 cm benutzt haben, erhielten Sie eine Höhe von 42,0256 cm—eine Lösung ohne physikalische Bedeutung. Falls kleine Anfangsnäherungen verwendet wurden (0 und 10 cm), dann haben Sie eine Höhe von 2,9744 cm erhalten—was zu einer unerwünschten, flachen Schachtel führt.

Verwendung eines Diagramms zur Wahl von Anfangsnäherungen. Als Hilfe zur Veranschaulichung einer Funktion können Sie diese auch grafisch darstellen. Verwenden Sie Ihre Programmroutine zur Auswertung der Funktion für verschiedene Werte der Unbekannten. Dabei ist für jeden Punkt des Graphen die x -Koordinate in der Variablen zu speichern und durch Drücken von $\boxed{\text{XEQ}}$ *Label* die korrespondierende y -Koordinate zu berechnen. Für das obige Problem wäre immer $V = 7500$ zu setzen und der Wert von H zu variieren, um verschiedene Funktionswerte zu erhalten. Die Abbildung der Funktion sieht wie folgt aus:

$$4H(40-H)(20-H) - 7500$$



Verwenden von SOLVE in einem Programm

Sie können SOLVE auch als Teil eines Programms verwenden. Falls erforderlich, so sind Werte oder Eingabeaufforderungen für Anfangsnäherungen (in die Unbekannte und das X-Register) in das Programm mit aufzunehmen, bevor `SOLVE Variable` ausgeführt wird. Die zwei Anweisungen zum Lösen einer Unbekannten in einer Gleichung erscheinen im Programm als:

```
FN= Label  
SOLVE Variable
```

Benannte Ausgabe. Die *programmierte* SOLVE Anweisung erzeugt keine Benennung für das Ergebnis (*Variable=Wert*), da dies eventuell nicht das signifikante Ergebnis des Programms darstellt (ggf. möchten Sie weitere Berechnungen mit dem Wert anstellen, bevor eine Anzeige erfolgen soll). Wenn Sie das Ergebnis *anzeigen möchten*, so fügen Sie nach SOLVE eine `VIEW Variable` Anweisung ein.

Bedingte Ausführung, falls keine Lösung. Wenn keine Lösung für die Unbekannte gefunden werden konnte, wird die nächste Programmzeile übersprungen (in Übereinstimmung mit der "Do if True" Regel, welche in Kapitel 6 erklärt ist). Das Programm sollte dann eine Routine ansteuern, welche den Fall einer fehlenden Lösung abdeckt (z.B. durch Wahl neuer Anfangsnäherungen oder Ändern eines Eingabewertes).

Beispiel: Annuitätenrechnung (TVM). Das Programm zur Annuitätenrechnung in Kapitel 14 löst finanzmathematische Problemstellungen durch Lösen einer der Unbekannten in der gegebenen TVM Gleichung. Diese Gleichung ist als Funktion in der Routine T definiert und enthält die Variablen für den Barwert, Endwert, periodische Zahlung, Zinssatz sowie Anzahl aller Zahlungen.

Sofern die Werte von vier bekannten Variablen gegeben sind, berechnet SOLVE (Zeile L04) die Lösung für die fünfte Variable (Unbekannte):

L01 LBL L	
L02 STO i	Speichert einen Indexwert, der kennzeichnet, welche der Variablen als Unbekannte spezifiziert wurde.
L03 FN= T	Wählt die in Programm T definierte Funktion.
L04 SOLVE(i)	Berechnet eine Lösung für die gekennzeichnete Unbekannte in Programm T.
L05 VIEW(i)	Zeigt das Ergebnis an.
L06 GTO(i)	Kehrt zum ursprünglichen Unterprogramm zurück, um für eine weitere Berechnung vorbereitet zu sein.

Diese Anwendungsweise von SOLVE funktioniert problemlos für Fälle, in denen der Anwender keine Anfangsnäherungen vorgeben muß.

Einschränkungen. Die SOLVE Anweisung kann keine Routine aufrufen, welche ebenfalls eine SOLVE Anweisung enthält; d.h. es ist keine rekursive Verwendung möglich (SOLVE(SOLVE) Fehler). Ebenso kann SOLVE keine Routine aufrufen, welche eine FN= *Label* Anweisung enthält (SOLVE ACTIVE Fehler). SOLVE kann keine Routine aufrufen, die eine fFN Anweisung enthält (SOLVE(fFN) Fehler), genauso, wie fFN keine Routine mit SOLVE aufrufen kann (f(SOLVE) Fehler).

Die SOLVE *Variable* Anweisung verwendet eine der sieben ausstehenden Unterprogramm-Rückspungadressen im Rechner (beziehen Sie sich auf "Verschachtelte Unterprogramme" in Kapitel 6).

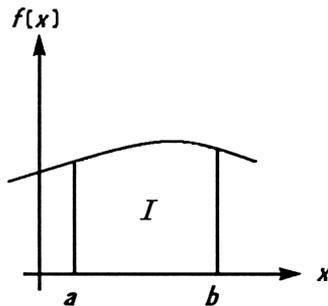
Weitere Informationen

Dieses Kapitel enthält Hinweise zum Lösen einer Unbekannten oder von Nullstellen über einen weiten Applikationsbereich. Anhang C enthält noch weitere Informationen über den Lösungsalgorithmus in SOLVE, wie Ergebnisse zu interpretieren sind, was geschieht, wenn keine Lösung aufgefunden werden konnte, und Bedingungen, welche zu einem Fehler führen können.

Numerische Integration

Viele Aufgaben, die in der Mathematik, der Wissenschaft und im Ingenieurwesen zu lösen sind, erfordern die Berechnung des bestimmten Integrals einer Funktion. Das Integral einer Funktion $f(x)$ mit einer unteren Integrationsgrenze a und der oberen Integrationsgrenze b wird mathematisch wie folgt ausgedrückt:

$$I = \int_a^b f(x) dx.$$



Die Größe I kann geometrisch als die Fläche, welche von der Kurve $f(x)$, der x -Achse und den Grenzen $x = a$ und $x = b$ begrenzt wird, ausgelegt werden (vorausgesetzt, daß $f(x)$ im Integrationsintervall nicht negativ ist).

Die \int FN Operation integriert eine spezifizierte Funktion über eine definierte Integrationsvariable.* Die Funktion muß vorab in einem benannten Programm definiert sein, wobei mehr als nur eine Variable enthalten sein kann.

* Für \int FN sind nur reelle Zahlen zulässig.

Anwendung der Integration (\int FN)

Um eine Funktion zu integrieren:

1. Geben Sie ein Programm ein, welches die zu integrierende Funktion definiert. (Siehe "Entwickeln eines Programms für \int FN", unten.)
2. Wählen Sie das Programm, das die zu integrierende Funktion enthält: Drücken Sie   {FN} Label.
3. Geben Sie die Integrationsgrenzen ein: Tippen Sie die *Untergrenze* ein und drücken Sie , tippen Sie danach die *Obergrenze* ein.
4. Wählen Sie die Integrationsvariable: Drücken Sie   { \int FN} Variable.

Damit wird die Berechnung gestartet.

Diese Operation benötigt weit mehr Speicherplatz als jede andere im Rechner enthaltene Operation. Sollte das Ausführen von { \int FN} zu der Meldung MEMORY FULL führen, so beziehen Sie sich auf Anhang B.

Sie können einen laufenden Integrationsprozeß anhalten, indem Sie  oder  drücken. (Allerdings ist im Normalfall keine Information über die Integration verfügbar, bis die Berechnung abgeschlossen ist.) Um die Berechnung fortzusetzen, ist  zu drücken.*

Genauigkeit. Die Einstellung des Anzeigeformats beeinflusst die Tiefe der Genauigkeit, die für die Funktion und das Ergebnis verwendet wird. Die Integration ist genauer, benötigt jedoch *viel mehr* Zeit im {ALL} und höheren {F \times }, {SC} und {EN} Modus. Die

Fehlerabschätzung des Ergebnisses erscheint im Y-Register, wodurch die Integrationsgrenzen in die T- und Z-Register hochgeschoben werden. Weitere Informationen finden Sie unter "Genauigkeit der Integration" auf Seite 131.

Ergebnisse. Das X-Register enthält das Integral, das Y-Register die Fehlerabschätzung, das Z-Register die obere Integrationsgrenze und das T-Register die untere Integrationsgrenze. (Die Integrationsvariable enthält einen bedeutungslosen Wert.)

* Drücken von ,  oder {RTN} hebt die \int FN Operation auf. In diesem Fall ist die Operation erneut zu starten, anstatt sie einfach fortzusetzen.

Um die gleiche Funktion mit unterschiedlichen Werten zu integrieren: Überspringen Sie die oben erwähnten ersten zwei Schritte. Wenn Sie die gleichen Integrationsgrenzen verwenden, so drücken Sie $\boxed{R\downarrow}$ $\boxed{R\downarrow}$ zum Rückruf in das X- und Y-Register. (Falls *nicht* die gleichen Grenzen verwendet werden, ist Schritt 3 zu wiederholen.) Führen Sie anschließend \blacksquare $\boxed{\text{SOLVE}/\int}$ {fFN} Variable aus. (Zur Lösung einer anderen Aufgabenstellung unter Verwendung einer *anderen* Funktion ist neu zu beginnen, indem die Funktion innerhalb eines Programms definiert wird.)

Entwickeln von Programmen für \int FN

Um ein Programm zu schreiben, das die Funktion des Integranden definiert:

1. Beginnen Sie mit einem Label, damit das Programm von \int FN aufgerufen werden kann.
2. Schließen Sie eine INPUT Anweisung für jede Variable mit ein (einschließlich der Integrationsvariablen). (Kommt nur eine Variable in der Funktion vor, so können Sie die INPUT Anweisung weglassen.)*
3. Geben Sie die Anweisungen zur Definition der Funktion ein. Verwenden Sie jeweils eine RCL Anweisung, wenn ein Variableninhalt für eine Berechnung benötigt wird.
4. Beenden Sie das Programm mit RTN. Das Programm sollte dabei so abgeschlossen werden, daß der Funktionswert sich im X-Register (der Anzeige) befindet.

Beispiele zur Anwendung von \int FN

Beispiel: Besselsche Funktion. Die Besselsche Funktion erster Art 0-ter Ordnung hat die Form:

$$J_0(x) = 1/\pi \int_0^\pi \cos(x \sin t) dt.$$

Berechnen Sie die Funktion für x -Werte von 2 und 3.

* Die INPUT Anweisungen sind bei Funktionen mit mehreren Variablen sinnvoll und hilfreich. Da INPUT für die Integrationsvariable ignoriert wird, brauchen Sie *nur ein* Programm zur Eingabe *aller* Variablen schreiben. Sie können das gleiche Programm immer wieder verwenden, unabhängig von der jeweils gewählten Integrationsvariablen.

Nachstehendes Programm wertet die Funktion $f(t) = \cos(x \sin t)$ aus:

```
J01 LBL J
J02 RAD
J03 INPUT X
J04 INPUT T
J05 RCL T
J06 SIN
J07 RCL X X
J08 COS
J09 RTN
```

Erforderliche Tastenfolge zur Eingabe des Programms:

Tastensequenz	Anzeige	Beschreibung
PRGM		Beginnt Programmeingabe.
GTO \cdot \cdot	PRGM TOP	
LBL/RTN {LBL} J	J01 LBL J	Eingabe des Programms.
MODES {RD}	J02 RAD	
INPUT X	J03 INPUT X	
INPUT T	J04 INPUT T	
RCL T SIN	J06 SIN	
RCL \times X COS	J08 COS	
LBL/RTN {RTN}	J09 RTN	An dieser Stelle übernimmt das X-Register den Funktionswert.
C		Beendet Programmeingabe.

Integrieren Sie nun die Funktion über t von 0 bis π ; $x = 2$.

SOLVE/f {fFN}	FN= _	Wählt Routine J für die Funktion.
J		
0 ENTER π	3,1416	Eingabe der Integrationsgrenzen (untere zuerst).
SOLVE/f {fFN}	fFN d _	Spezifiziert T als Integrationsvariable. Eingabeaufforderung für x.
T	X?Wert	

2 [R/S]

 $\int = 0,7034$

$x = 2$. Startet Integration und ermittelt Ergebnis für $\int_0^x f(t)$.

Um die Berechnung abzuschließen, sollten Sie daran denken, den Wert des Integrals mit der Konstante $(1/\pi)$ zu multiplizieren. (Sie könnten diesen Schritt auch mit in das Programm aufnehmen.)

■ [π] [÷]

0,2239

Endergebnis für $J_0(2)$.

Berechnen Sie nun $J_0(3)$ mit den gleichen Integrationsgrenzen. Sie müssen nicht erneut die Funktion spezifizieren (Routine J), aber die Integrationsgrenzen $(0, \pi)$ müssen erneut angegeben werden, da sie durch die abschließende Division durch π aus dem Stack geschoben wurden.

0 [ENTER] ■ [π]

3,1416

Zeigt Obergrenze an.

■ [SOLVE/∫] {JFN} T

X?2,0000

Startet Integration; Eingabeaufforderung für x .

3 [R/S]

 $\int = -0,8170$ Integral von $f(t)$.

■ [π] [÷]

-0,2601

Ergebnis für $J_0(3)$.

Beispiel: Integralsinus. In der Nachrichtentechnik wird für manche Zwecke (z.B. die Stromübertragung in idealisierten Netzwerken) ein Integral der folgenden Form (auch *Integralsinus* genannt) benötigt:

$$Si(t) = \int_0^t \left(\frac{\sin x}{x} \right) dx$$

Berechnen Sie $Si(2)$.

Tippen Sie nachstehendes Programm zur Auswertung der Funktion $f(x) = (\sin x) \div x$ ein.*

* Der Versuch, die Funktion an der Stelle $x = 0$, der unteren Integrationsgrenze, zu berechnen, würde die Fehlermeldung **DIVIDE BY 0** zur Folge haben. Der für die Integration verwendete Algorithmus wertet jedoch die Funktion normalerweise *nicht* an den 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.

S01 LBL S
 S02 RAD
 S03 RCL X
 S04 SIN
 S05 RCL ÷ X
 S06 RTN

Integrieren Sie nun die Funktion über x (d.h. X) von 0 bis 2 ($t = 2$).

Tastensequenz	Anzeige	Beschreibung
■ SOLVE/∫ {FN} S		Wählt Routine S als Funktion.
0 ENTER 2	2_	Eingabe der Integrationsgrenzen (untere zuerst).
■ SOLVE/∫ {∫FN} X	∫=1,6054	Ergebnis für $S_i(2)$.

Genauigkeit der Integration

Da der Rechner ein Integral nicht exakt berechnen kann, bestimmt er eine *Approximation* für das Integral. Die Genauigkeit dieser Näherung hängt von der Genauigkeit der Funktion selbst ab, wie sie mit Ihrem Programm berechnet wird.* Dies wird von Rundungsfehlern des Rechners und der Genauigkeit von empirischen Konstanten beeinflusst.

* 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.* Die allgemeinen Charakteristika von Funktionen, welche zu Problemen führen können, sowie Techniken, welche in diesen Fällen Abhilfe schaffen können, sind in Anhang D beschrieben.

Spezifizieren der Genauigkeit

Das Anzeigeformat entscheidet mit über die *Genauigkeit* der Integration: Je mehr Stellen angezeigt werden, desto größer ist die Genauigkeit des berechneten Integrals (und desto länger die Rechenzeit); je weniger Stellen angezeigt werden, desto kürzer die Rechenzeit, wobei jedoch davon ausgegangen werden kann, daß die Funktion nur eine Genauigkeit besitzt, die durch die Anzahl der Stellen des Anzeigeformats angegeben ist.

Zur Spezifikation der *Integrationsgenauigkeit* ist das Anzeigeformat so zu wählen, daß *nicht mehr* Stellen angezeigt werden, als im *Wert des Integranden* als genau angesehen werden. Die gleiche Genauigkeitstiefe wird im Integrationsergebnis wiedergegeben.

Interpretation der Genauigkeit

Nach dem Berechnen des Integrals speichert der Rechner eine *Fehlerabschätzung* der Approximation des Integrals im Y-Register. Drücken Sie einfach \boxed{xzy} , um die Abschätzung anzuzeigen.

Ist beispielsweise das Integral von $Si(2)$ gleich $1,6054 \pm 0,0001$, dann stellt 0,0001 die Fehlerabschätzung dar.

Beispiel: Spezifizieren der Genauigkeit. Wählen Sie als Anzeigeformat SCI 2 und berechnen Sie das Integral im Ausdruck für $Si(2)$ (aus vorherigem Beispiel).

Tastenfolge:	Anzeige	Beschreibung
\blacksquare $\boxed{\text{DISP}}$ {SC} 2	1,61E0	Stellt wissenschaftliche Notation mit 2 Dezimalstellen ein.
0 $\boxed{\text{ENTER}}$ 2	2_	Integrationsgrenzen.
\blacksquare $\boxed{\text{SOLVE}/f}$ {FN} S	2,00E0	Wählt Routine S für die Funktion.
\blacksquare $\boxed{\text{SOLVE}/f}$ {fFN} X	f=1,61E0	Auf 2 Dezimalstellen angenähertes Integral.

$x \rightarrow y$	1,00E-3	Die Fehlerabschätzung für die Approximation des Integrals.
-------------------	---------	--

Das Integral ist $1,61 \pm 0,00100$. Die Fehlerabschätzung läßt erkennen, daß das Ergebnis wenigstens bis auf 2 Dezimalstellen genau ist.

Wenn die Fehlerabschätzung für eine Approximation größer ausfällt, als von Ihnen als akzeptierbar angenommen wird, so können Sie die Anzahl der Dezimalstellen im Anzeigeformat erhöhen und die Integration wiederholen (vorausgesetzt, daß $f(x)$ immer noch auf die angezeigten Stellen genau berechnet wird). Im allgemeinen verkleinert sich die Fehlerabschätzung um den Faktor 10 für jede zusätzlich spezifizierte Stelle im Anzeigeformat.

Beispiel: Ändern der Genauigkeit. Das soeben berechnete Integral von $Si(2)$ soll jetzt bis zu 4 Stellen anstatt nur 2 genau sein.

Tastenfolge:	Anzeige	Beschreibung
\blacksquare [DISP] {SC} 4	1,0000E-3	Spezifiziert Genauigkeit auf 4 Dezimalstellen. Die Fehlerabschätzung des letzten Beispiels wird noch immer angezeigt.
[R \downarrow] [R \downarrow]	2,0000E0	Schiebt die Integrationsgrenzen aus dem Z- und T-Register in das X- und Y-Register.
\blacksquare [SOLVE/∫] {∫FN} X	∫=1,6054E0	Ergebnis.
$x \rightarrow y$	1,0000E-5	Beachten Sie, daß die Fehlerabschätzung etwa $\frac{1}{100}$ des Wertes im SCI 2 Anzeigeformat der vorherigen Berechnung ist.
\blacksquare [DISP] {FX} 4		Stellt wieder FIX 4 Format ein.

Aus dieser Fehlerabschätzung läßt sich schließen, daß das Ergebnis nur bis auf vier Dezimalstellen korrekt sein *könnte*. In Wirklichkeit ist das vorliegende Ergebnis jedoch bis auf *sieben* Dezimalstellen genau, verglichen mit dem tatsächlichen Wert des Integrals. Da die Fehlerabschätzung durchweg sehr vorsichtig berechnet wird, *weisen die berechneten Approximationen im allgemeinen eine höhere Genauigkeit auf, als die Fehlerabschätzung andeutet*.

Weitere Informationen finden Sie in Anhang D.

Verwenden der Integration in einem Programm

Integration kann auch innerhalb eines Programms ausgeführt werden. Denken Sie daran, Eingabeaufforderungen für die Integrationsgrenzen oder Werte dafür im Programm aufzunehmen, bevor die Integration ausgeführt wird. Beachten Sie außerdem, daß zur Ausführungszeit Genauigkeit und Rechenzeit durch das Anzeigeformat beeinflußt werden. Die zwei Integrationsanweisungen erscheinen im Programm als:

```
FN= Label
∫FN d Variable
```

Benannte Ergebnisse. Die *programmierte* ∫FN Anweisung erzeugt keine Benennung für das Ergebnis (∫=Wert), da dies eventuell nicht das signifikante Ergebnis des Programms darstellt (ggf. möchten Sie weitere Berechnungen mit dem Wert anstellen, bevor eine Anzeige erfolgen soll). Wenn Sie das Ergebnis *anzeigen möchten*, so fügen Sie nach ∫FN eine PSE (■ LBL/RTN {PSE}) oder STOP (R/S) Anweisung ein.

Beispiel: Normalverteilung. Das Programm zur "Normalverteilung und Verteilungsfunktion" auf Seite 215 in Teil 4 schließt eine Integration der Gleichung für die Verteilungsfunktion der Normalverteilung ein:

$$\frac{1}{S \sqrt{2\pi}} \int_M^D e^{-\frac{1}{2} \left(\frac{D-M}{S} \right)^2} dD$$

Diese Funktion ist in Routine F definiert:

F01 LBL F
F02 RCL D
F03 RCL- M
F04 RCL÷ S
F05 x^2
F06 2
F07 ÷
F08 +/-
F09 e^x
F10 RTN

Andere Routinen fordern zur Eingabe für die bekannten Variablen auf und führen Berechnungen zur Bestimmung von $Q(D)$, der rechtsseitigen Fläche unter der Glockenkurve. Die Integration selbst wird in Routine Q vorbereitet und ausgeführt.

Q01 LBL Q
Q02 RCL M Ruft untere Integrationsgrenze ab.
Q03 RCL X Ruft obere Integrationsgrenze ab. ($X = D$)
Q04 FN= F Spezifiziert die durch LBL F definierte
 Funktion für die Integration.
Q05 ∫FN d D Integriert die Funktion über die Variable D .
 ⋮

Einschränkungen. Die ∫FN *Variable* Anweisung kann keine Routine aufrufen, welche ebenfalls eine ∫FN Anweisung enthält; d.h. es ist keine rekursive Verwendung möglich (∫⟨∫FN⟩ Fehler). Ebenso kann ∫FN keine Routine aufrufen, welche eine FN= *Label* Anweisung enthält (∫FN ACTIVE Fehler). ∫FN kann keine Routine aufrufen, die eine SOLVE Anweisung enthält (∫⟨SOLVE⟩ Fehler), genauso, wie SOLVE keine Routine mit einer ∫FN Anweisung aufrufen kann (SOLVE⟨∫FN⟩ Fehler).

Die ∫FN d *Variable* Anweisung verwendet eine der sieben ausstehenden Unterprogramm-Rücksprungadressen im Rechner (beziehen Sie sich auf "Verschachtelte Unterprogramme" in Kapitel 6).

Weitere Informationen

Dieses Kapitel bietet Ihnen Hinweise zur Anwendung der Integration im HP-32S über einen weiten Applikationsbereich. Anhang D enthält noch weitere Informationen über den für die Integration verwendeten Algorithmus, mögliche Ursachen für unkorrekte Ergebnisse, Bedingungen, welche zu einer Verlängerung der Rechenzeit führen können, und wie Sie die momentane Approximation für ein Integral erhalten können.

9

Operationen mit komplexen Zahlen

Der HP-32S ermöglicht Ihnen Berechnungen mit komplexen Zahlen, d.h. Zahlen, die in der Form

$$x + iy$$

vorliegen. Er enthält Operationen für komplexe Arithmetik (+, −, ×, ÷), komplexe Trigonometrie (sin, cos, tan) und die mathematischen Funktionen $-z$, $1/z$, $z_1^{z_2}$, $\ln z$ und e^z (wobei z_1 und z_2 komplexe Zahlen darstellen).

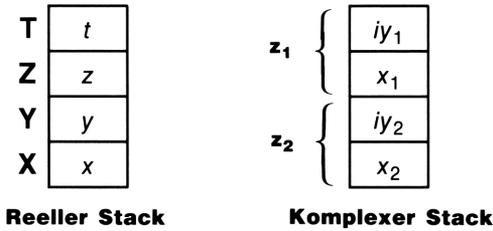
Komplexe Zahlen werden vom HP-32S als separate Werte behandelt, d.h. Real- und Imaginärteil einer komplexen Zahl sind getrennt einzugeben. Um zwei komplexe Zahlen einzugeben, sind vier Werte einzutippen. Wenn Sie eine komplexe Operation ausführen möchten, so drücken Sie **CMPLX**, bevor Sie die Operatortaste drücken. Soll beispielsweise

$$(2 + i4) + (3 + i5)$$

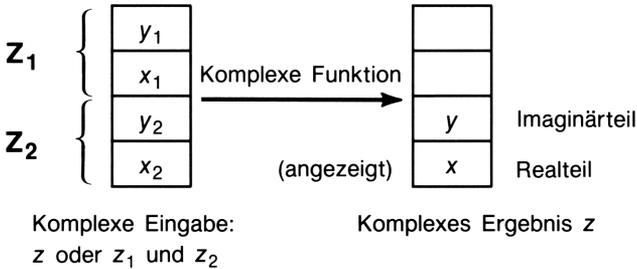
berechnet werden, so drücken Sie 4 **ENTER** 2 **ENTER** 5 **ENTER** 3 **CMPLX** **+**. Das Ergebnis ist $5 + i9$. (Drücken Sie **x↔y** zur Anzeige des Imaginärteils.)

Der komplexe Stack

Der komplexe Stack besteht in Wirklichkeit aus dem normalen Speicherstack, welcher zur Speicherung zweier komplexer Zahlen $z_{1x} + iz_{1y}$ und $z_{2x} + iz_{2y}$ in zwei Doppelregister aufgeteilt ist:



Da der Imaginär- und Realteil einer komplexen Zahl getrennt eingegeben und gespeichert werden, können Sie leicht jeden Teil für sich manipulieren.



Geben Sie immer den Imaginärteil (der y -Teil) einer komplexen Zahl zuerst ein. Nach einer komplexen Operation erscheint der Realteil des Ergebnisses (z_x) in der Anzeige; drücken Sie xzy zur Anzeige des Imaginärteils (z_y).

Komplexe Operationen

Führen Sie komplexe Operationen genauso wie reelle Operationen aus, nur mit dem Unterschied, daß  **CMPLX** vor der Operatortaste gedrückt wird.

Um Operationen mit einer komplexen Zahl auszuführen:

1. Geben Sie die komplexe Zahl z , welche sich aus $x + iy$ zusammensetzt, durch Eintippen von y  x ein.
2. Wählen Sie die komplexe Funktion:

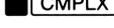
Funktionen für eine komplexe Zahl z

Zur Berechnung von:	Drücken Sie
Vorzeichenwechsel, $-z$	 CMPLX 
Kehrwert, $1/z$	 CMPLX 
Natürlicher Logarithmus, $\ln z$	 CMPLX 
Natürliche Exponentialfunktion, e^z	 CMPLX 
Sin z	 CMPLX 
Cos z	 CMPLX 
Tan z	 CMPLX 

Um arithmetische Operationen mit zwei komplexen Zahlen auszuführen:

1. Geben Sie die erste komplexe Zahl, z_1 (zusammengesetzt aus $x_1 + iy_1$) durch Drücken von y_1  x_1  ein. (Für $z_1^{z_2}$ ist zuerst die Basis, z_1 , einzutippen.)
2. Geben Sie die zweite komplexe Zahl z_2 durch Drücken von y_2  x_2 ein. (Für $z_1^{z_2}$ ist der Exponent, z_2 , als zweites einzutippen.)
3. Wählen Sie die gewünschte arithmetische Operation:

Arithmetik mit zwei komplexen Zahlen z_1 und z_2

Zur Berechnung von:	Drücken Sie
Addition, $z_1 + z_2$	 
Subtraktion, $z_1 - z_2$	 
Multiplikation, $z_1 \times z_2$	 
Division, $z_1 \div z_2$	 
Potenzfunktion, $z_1^{z_2}$	 

Beispiele. Nachstehend einige Beispiele für trigonometrische und arithmetische Berechnungen mit komplexen Zahlen:

Berechnen Sie $\sin(2 + i3)$.

Tastensequenz:

Anzeige

Beschreibung

3  2
 

9,1545

Realteil des Arguments.



-4,1689

Als Ergebnis erhalten Sie $9,1545 - i4,1689$.

Werten Sie den Ausdruck

$$z_1 \div (z_2 + z_3)$$

aus, wobei $z_1 = 23 + i13$, $z_2 = -2 + i$, $z_3 = 4 - i3$.

Da der Stack nur zwei komplexe Zahlen gleichzeitig gespeichert halten kann, ist die Berechnung in der Form

$$z_1 \times [1 \div (z_2 + z_3)]$$

auszuführen.

Tastenfolge:	Anzeige	Beschreibung
1 [ENTER] 2 [+/-] [ENTER] 3 [+/-] [ENTER] 4 [Cmplx] +	2,0000	Addiert $z_2 + z_3$; zeigt Realteil an.
[Cmplx] 1/x	0,2500	$1 \div (z_2 + z_3)$.
13 [ENTER] 23 [Cmplx] x	2,5000	$z_1 \div (z_2 + z_3)$.
[xzy]	9,0000	Ergebnis: $2,5 + i9$.

Berechnen Sie $(4 - i2/5)(3 - i2/3)$. Verwenden Sie keine komplexe Operation, wenn Sie nur einen Teil einer komplexen Zahl manipulieren.

Tastenfolge:	Anzeige	Beschreibung
2 [ENTER] 5 [+/-] [+/-]	-0,4000	Berechnet Imaginärteil über reelle Operation.
4 [ENTER]	4,0000	Eingabe des Realteils der 1. komplexen Zahl.
2 [ENTER] 3 [+/-] [+/-]	-0,6667	Berechnet Imaginärteil der 2. komplexen Zahl.
3 [Cmplx] x	11,7333	Schließt Eingabe der 2. komplexen Zahl ab und multipliziert anschließend die 2 Zahlen.
[xzy]	-3,8667	Ergebnis: $11,7333 - i3,8667$.

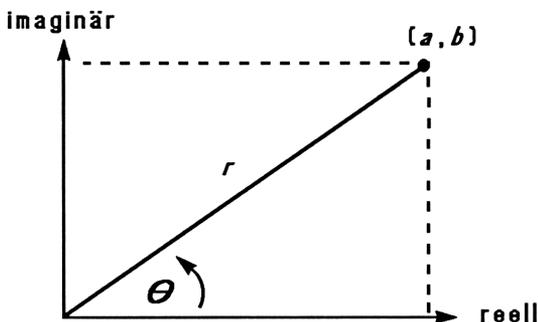
Berechnen Sie e^{z^2} , wobei $z = (1 + i)$. Verwenden Sie [Cmplx] [y^x] zur Auswertung von z^{-2} ; geben Sie -2 als $-2 + i0$.

Tastenfolge:	Anzeige	Beschreibung
1 [ENTER] 1 [ENTER] 0 [ENTER] 2 [+/-]		Zwischenergebnis von $(1 + i)^{-2}$.
■ [CMPLX] [y ^x]	0,0000	
■ [CMPLX] [e ^x]	0,8776	Realteil von Endergebnis.
[y ^x]	-0,4794	Endergebnis: $0,8776 - i0,4794$.

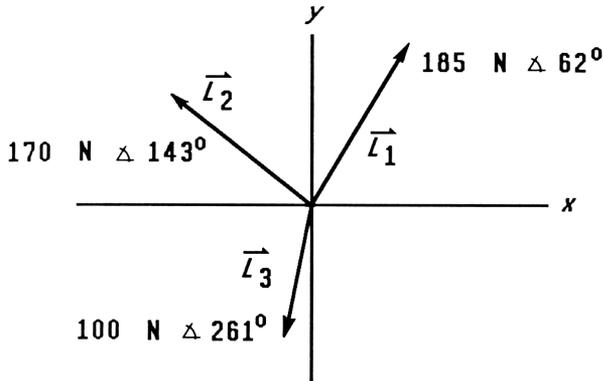
Verwenden von Zahlen in Polarnotation

In vielen Applikationen sind komplexe Zahlen in *Polarnotation* bzw. *Phasennotation* angegeben. Diese Formen benutzen Zahlenpaare, analog zur Darstellung von komplexen Zahlen, wodurch Sie über die komplexen Operationen arithmetische Berechnungen mit diesen Zahlenpaaren anstellen können. Da für komplexe Operationen im HP-32S die Zahlen in *Rechtecksnotation* vorliegen müssen, sind vor der komplexen Operation die Polarkoordinaten (über ■ [P↔RECT]) zuerst in Rechteckskordinaten und anschließend wieder zurück in Polarkoordinaten zu konvertieren.

$$\begin{aligned}
 a + ib &= r(\cos \theta + i \sin \theta) = re^{i\theta} \\
 &= r \angle \theta \quad \text{Polar- oder Phasennotation}
 \end{aligned}$$



Beispiel: Vektoraddition. Addieren Sie die nachstehenden drei Kräfte. Sie müssen dazu zuerst die Koordinaten von Polar- in Rechteckskordinaten konvertieren.



Tastensequenz	Anzeige	Beschreibung
\blacksquare MODES {DG}		Stellt Grad-Modus ein.
62 ENTER 185		Eingabe von F_1 und Konvertierung in Rechtecksnotation.
\blacksquare P \leftrightarrow RECT { $\theta, r \rightarrow y, x$ }	86,8522	
143 ENTER 170		Eingabe und Konvertierung von F_2 .
\blacksquare P \leftrightarrow RECT { $\theta, r \rightarrow y, x$ }	-135,7680	
\blacksquare CMLPX +	-48,9158	Addition der Vektoren.
261 ENTER 100		Eingabe und Konvertierung von F_3 .
\blacksquare P \leftrightarrow RECT { $\theta, r \rightarrow y, x$ }	-15,6434	
\blacksquare CMLPX +	-64,5592	Addition von $F_1 + F_2 + F_3$.
\blacksquare P \leftrightarrow RECT { $y, x \rightarrow \theta, r$ }	178,9372	Konvertiert Vektor zurück in Polarnotation; zeigt r an.
\square xzy	111,1489	Zeigt θ an.

10

Rechnen in verschiedenen Zahlensystemen

Das BASE Menü (■ **BASE**) ermöglicht Ihnen die Wahl zwischen vier verschiedenen Zahlensystemen (*Basis-Modus*) zur Eingabe von Zahlen und zur Ausführung von Berechnungen (einschließlich Programmierung). Beim Wechsel in einen anderen Basis-Modus wird automatisch die *angezeigte* Zahl entsprechend konvertiert.

Das BASE Menü

Menüoption	Beschreibung
{DEC}	<i>Dezimale Basis.</i> Kein Indikator angezeigt. Konvertiert Zahlen zur Basis 10, mit Vor- und Nachkommateil.
{HX}	<i>Hexadezimale Basis.</i> HEX Indikator wird angezeigt. Konvertiert Zahlen zur Basis 16; verwendet nur ganze Zahlen. Der obersten Tastenreihe werden die Buchstaben A bis F zugewiesen.
{OC}	<i>Oktale Basis.</i> OCT Indikator wird angezeigt. Konvertiert Zahlen zur Basis 8; verwendet nur ganze Zahlen. Die Tasten 8 , 9 und die Erstfunktionen der oberen Tastenreihe werden deaktiviert.
{BN}	<i>Binäre Basis.</i> BIN Indikator wird angezeigt. Konvertiert Zahlen zur Basis 2; verwendet nur ganze Zahlen. Alle Zifferntasten außer 0 und 1 und die Erstfunktionen der oberen Tastenreihe werden deaktiviert. Wenn eine Zahl mehr als 12 Stellen enthält, werden die Tasten \sqrt{x} und $\Sigma+$ zum Anzeigen von "Fenstern" aktiviert.

Beispiele: Zahlenkonvertierung in verschiedene Zahlensysteme. Die nachstehende Tastenfolge zeigt die Konvertierung von Zahlen in verschiedene Basis-Modi.

Konvertieren Sie $125,99_{10}$ in HEX, OCT und BIN Darstellung.

Tastenfolge:	Anzeige	Beschreibung
125,99 ■ [BASE] {HX}	7D	Konvertiert nur den ganzzahligen Teil (125) der Dezimalzahl als Hexadezimalzahl und zeigt das Ergebnis an.
■ [BASE] {OC}	175	Basis 8.
■ [BASE] {BN}	1111101	Basis 2.
■ [BASE] {DEC}	125,9900	Stellt wieder Basis 10 ein; es wurde der ursprüngliche dezimale Wert erhalten, einschließlich des Nachkommateils.

Konvertieren Sie $24FF_{16}$ in eine Binärzahl. Das Ergebnis wird mehr als 12 Stellen (Anzeigebreite) enthalten.

■ [BASE] {HX} 24FF	24FF_	Verwenden Sie [Σ+] zur Eingabe von "F".
■ [BASE] {BN}	010011111111	Die ganze Binärzahl paßt nicht in die Anzeige. Der ← Indikator deutet an, daß die Zahl auf der linken Seite weitere Stellen enthält; der ▾ Indikator zeigt auf [√x].
[√x]	10	Zeigt den restlichen Teil der Zahl an; die vollständige Zahl ist 10010011111111_2 .
[Σ+]	010011111111	Zeigt erneut die ersten 12 Stellen an.
■ [BASE] {DEC}	9.471,0000	Wechsel zurück zum Dezimalsystem.

Arithmetik im Basis-Modus 2, 8 und 16

Sie können die arithmetischen Operationen $\boxed{+}$, $\boxed{-}$, $\boxed{\times}$ und $\boxed{\div}$ in jedem Zahlensystem anwenden.* In der Basis 2, 8 und 16 werden die Operationen im Zweierkomplement der jeweiligen Zahl ausgeführt, wobei jeweils nur ganzzahlige Werte zulässig sind:

- Besitzt eine Zahl einen Nachkommateil, so wird nur der Vorkommateil für die arithmetische Berechnung verwendet.
- Das Ergebnis einer Operation ist immer ganzzahlig (ggf. wird der Wert gekürzt).

Während bei Konvertierungen nur die angezeigte Zahl verändert wird, nicht aber der Wert im X-Register, wird durch *arithmetische Operationen* auch der Inhalt des X-Registers geändert.

Kann das Ergebnis einer Operation nicht mit 36 Bits dargestellt werden, dann zeigt der Rechner **OVERFLOW** und danach die größte darstellbare positive oder negative Zahl an.

Beispiele. Nachfolgend einige Beispiele für arithmetische Operationen bei hexadezimalen, oktalem und binärem Basis-Modus:

$$12F_{16} + E9A_{16} = ?$$

Tastenfolge:	Anzeige	Beschreibung
$\boxed{\text{BASE}}$ {HX}		Stellt Basis 16 ein; HEX Indikator an.
12F $\boxed{\text{ENTER}}$ E9A $\boxed{+}$	FC9	Ergebnis.

* Die einzigen Funktionstasten, welche tatsächlich außerhalb des Dezimalsystems deaktiviert sind, sind $\boxed{\sqrt{x}}$, $\boxed{e^x}$, $\boxed{\text{LN}}$, $\boxed{y^x}$, $\boxed{1/x}$ und $\boxed{\Sigma+}$. Andererseits sollten Sie daran denken, daß außer bei arithmetischen Operationen keine sinnvollen Ergebnisse erzielt werden, da die Werte jeweils auf den ganzzahligen Teil gekürzt werden.

Wertebereich von Zahlen

Die Wortlänge von 36 Bits legt die Größe des Zahlenbereichs fest, in welchem hexadezimale (9 Stellen), oktale (12 Stellen) und binäre Zahlen (36 Stellen) dargestellt werden können, sowie den Bereich der Dezimalzahlen, welche in ein anderes Zahlensystem konvertiert werden können.

Bereich für Zahlensystem-Konvertierungen

Basis	Größe pos. ganze Zahl	Größe neg. ganze Zahl
HEX	7FFFFFFF	80000000
DEC	34.359.738.367	-34.359.738.368
OCT	37777777777	40000000000
BIN	111111111111	100000000000
	111111111111	000000000000
	111111111111	000000000000

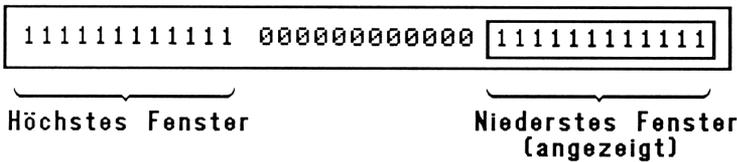
Bei der Eingabe von Zahlen in HEX, OCT oder BIN Basis wird die Eingabe abgebrochen, wenn Sie zu viele Stellen eintippen möchten. So verursacht z.B. der Eingabeversuch einer 10-stelligen hexadezimalen Zahl nach der neunten Stelle die Anzeige von  und den Abbruch der Zahleneingabe.

Enthält die Anzeige eine Dezimalzahl außerhalb des erlaubten Bereichs, so erscheint nach dem Wechsel in ein anderes Zahlensystem die Meldung `TOO BIG`. Jede Operation, die zur Anzeige von `TOO BIG` führt, verursacht eine Überlauf-Bedingung, wobei die zu große Zahl durch die größtmögliche positive oder negative Zahl substituiert wird.

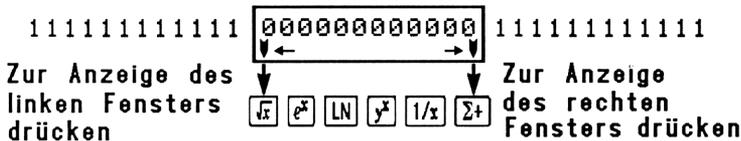
Fenster für große Binärzahlen

Die größte Binärzahl umfaßt 36 Stellen—dreimal so viel Stellen, wie in die Anzeige passen. Jede 12-stellige Anzeige einer großen Binärzahl wird als *Fenster* bezeichnet.

36-Bit Zahl



Umfaßt eine Binärzahl mehr als 12 Stellen, dann erscheint der ← oder → Indikator (oder beide) in der Anzeige, um anzudeuten, auf welcher Seite noch weitere Stellen (momentan unsichtbar) zu finden sind. Drücken Sie die gekennzeichnete Taste (\sqrt{x} oder $\Sigma+$), um das gewünschte Fenster anzuzeigen.



Anzeigen (SHOW) von unsichtbaren Zahlenteilen

Die Funktionen \blacksquare [VIEW] und \blacksquare [INPUT] arbeiten mit nichtdezimalen Zahlen genauso, wie Sie es von Dezimalzahlen her kennen. Paßt allerdings die oktale oder binäre Zahl nicht vollständig in die Anzeige, so werden die linken Stellen durch eine Ellipse (...) ersetzt. Drücken Sie \blacksquare [SHOW] zur Anzeige der durch die Labels A=... oder A?... verdeckten Stellen.

Tastenfolge:	Anzeige	Beschreibung
\blacksquare [BASE] {0C} 123456712345 [STO] A	23456712345_ 123456712345	Eingabe einer großen Oktalzahl.
\blacksquare [VIEW] A	A=...456712345	Entfernt drei Stellen am linken Rand.
\blacksquare [SHOW] (gedrückt halten)	123456712345	Zeigt alle Stellen an.

Programmierung mit BASE

Sie können Anweisungen programmieren, um unter Verwendung von **BASE** den Basis-Modus zu ändern. Diese Einstellungen funktionieren in Programmen genauso wie Sie es von den Operationen über das Tastenfeld gewohnt sind. Sie haben dadurch die Möglichkeit, Programme zu schreiben, welche das Eingeben von Zahlenwerten, das Ausführen von Berechnungen und die Anzeige des Ergebnisses in einem beliebigen Zahlensystem erlauben.

Wenn Sie ein Programm entwickeln, welches Zahlen außerhalb des Dezimalsystems verwendet, so ist der Basis-Modus zweimal—*als momentane Einstellung für den Rechner und in Ihrem Programm* (als Anweisung)—zu spezifizieren.

Wählen des Basis-Modus in einem Programm

Fügen Sie am Anfang des Programms eine BIN, OCT oder HEX Anweisung ein. Sie sollten normalerweise am Ende des Programms eine DEC Anweisung vorsehen, damit der Rechner wieder zum Dezimalsystem zurückkehrt, wenn das Programm abgeschlossen wird.

Eine Programmanweisung zum Ändern des Basis-Modus legt fest, wie eine eingegebene Zahl interpretiert und in welcher Basis die Ausgabe *während und nach der Programmausführung* angezeigt wird, *beeinflusst aber nicht* die Programmzeilen während der Eingabe.

Die SOLVE und fFN Operationen stellen automatisch den DEC Modus ein.

Zahleneingabe in Programmzeilen

Spezifizieren Sie den Basis-Modus, bevor Sie mit der Programmeingabe beginnen. Die momentane Einstellung bestimmt die Basis, in welcher die Zahlen in Programmzeilen eingegeben werden. Die Anzeige dieser Zahlenwerte ändert sich, wenn Sie den Basis-Modus ändern.

Programmzeilennummern werden immer als Dezimalzahlen angezeigt.

In der Anzeige erscheint ein Indikator (außer in Basis 10) zur Kennzeichnung, welche Zahlenbasis Sie momentan spezifiziert haben. Vergleichen Sie z.B. die nachstehenden Programmzeilen in der linken und rechten Spalte. Beachten Sie, daß die hexadezimale Zahl, wie alle nichtdezimalen Zahlen, rechtsbündig angezeigt wird.

Dezimal-Modus eingestellt

```
      :  
      PRGM  
A09 HEX  
  
      PRGM  
A10 23  
      :
```

Hexadezimal-Modus eingestellt

```
      :  
      PRGM  
A09 HEX  
  
      PRGM  
A10  
      :  
      HEX  
      17
```

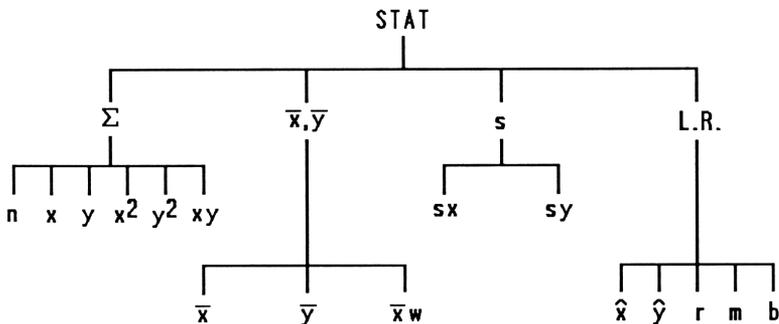
↑
Programmzeilennummern
sind immer dezimal.

←
←
Momentan eingestellter
Basis-Modus.

Statistikberechnungen

Das STAT Menü (*STATistik*) bietet Ihnen Funktionen zur statistischen Analyse von Datensätzen, welche aus einer oder zwei Variablen bestehen können.

- Daten einer Variablen: Mittelwert und Standardabweichung.
- Daten zweier Variablen (x,y): Lineare Regression und lineare Näherung (\hat{x} und \hat{y}).
- Gewogenes Mittel (x gewichtet mit y).
- Summationsstatistik: n , Σx , Σy , Σx^2 , Σy^2 und Σxy .



Eingeben von Statistikdaten ($\Sigma+$, $\Sigma-$)

Statistikdaten von einer oder zwei Variablen werden auf ähnliche Weise eingegeben. Die Werte werden als Summen in sechs *Statistikregistern* akkumuliert, wobei der Inhalt der Register über **STAT** { Σ } angesehen werden kann.

Dateneingabe einer Variablen

1. Drücken Sie  {Σ} zum Löschen zuvor gespeicherter Statistikdaten.
2. Tippen Sie jeden x -Wert ein und drücken Sie .
3. Die Anzeige enthält n , die Anzahl von Datenwerten, welche inzwischen akkumuliert wurden.*
4. Setzen Sie die Eingabe der Werte fort; n wird dabei nach jeder Eingabe aktualisiert.

Um einen Wert unmittelbar nach der Eingabe wieder in die Anzeige zurückzurufen, ist  zu drücken.

Dateneingabe zweier Variablen

Bestehen Ihre Statistikdaten aus zwei Variablen, so stellt x die *unabhängige* und y die *abhängige Variable* dar. (Denken Sie daran, ein (x, y) Datenpaar in umgekehrter Reihenfolge einzugeben, damit y im Y-Register und x im X-Register gespeichert wird.

1. Drücken Sie  {Σ} zum Löschen zuvor gespeicherter Statistikdaten.
2. Tippen Sie *zuerst* den y -Wert ein und drücken Sie .
3. Tippen Sie den korrespondierenden x -Wert ein und drücken Sie .
4. Die Anzeige enthält n , die Anzahl von Datenwerten, welche inzwischen akkumuliert wurden.
5. Setzen Sie die Eingabe der x,y -Paare fort. n wird dabei nach jeder Eingabe aktualisiert.

Um einen x -Wert unmittelbar nach der Eingabe wieder in die Anzeige zurückzurufen, ist  zu drücken.

* Bei diesem Verfahren werden eigentlich zwei Variablen in die Statistikregister eingegeben, da der bereits im Y-Register gespeicherte Wert als y -Wert akkumuliert wird. Aus diesem Grund führt der Rechner auch eine lineare Regression durch und zeigt auf y basierende Werte an, auch wenn nur x -Werte eingegeben wurden—oder wenn Sie eine ungleiche Anzahl von x - und y -Werten eingegeben haben. Es wird keine Fehlermeldung angezeigt, wobei jedoch die erzielten Ergebnisse offensichtlich nicht sinnvoll sind.

Korrektur der Dateneingabe

Wenn Ihnen bei der Eingabe von Statistikdaten ein Fehler unterlaufen ist, so können Sie die falschen Werte löschen und die richtigen Werte eingeben. Selbst wenn nur ein Wert eines x, y -Paares falsch ist, müssen Sie beide Werte löschen und die 2 richtigen Werte eingeben.

Um eine Korrektur von Statistikdaten vorzunehmen:

1. Geben Sie die fehlerhaften Werte erneut ein, wobei Sie jedoch anstatt $\Sigma+$ die Taste $\Sigma-$ drücken müssen. Dadurch werden die Werte gelöscht und n reduziert.
2. Geben Sie über $\Sigma+$ die richtigen Werte ein.

Handelt es sich bei den fehlerhaften Werten um die, welche zuletzt eingegeben wurden, so können Sie einfach LASTx drücken und anschließend $\Sigma-$, wodurch diese gelöscht werden. (Der falsche y -Wert war noch immer im Y-Register und sein x -Wert wurde im LASTx Register gesichert.)

Beispiel: Tippen Sie die linken x, y -Werte ein und korrigieren Sie diese auf die rechten Werte.

Ursprüngliche x, y	Korrigierte x, y
20, 4	20, 5
400, 6	40, 6

Tastenfolge:	Anzeige	Beschreibung
CLEAR $\{\Sigma\}$		Löscht alten Inhalt der Statistikregister und gibt zwei Datenpaare ein. Die Anzeige enthält n , die Anzahl der Datenpaare.
4 ENTER 20 $\Sigma+$	1,0000	
6 ENTER 400 $\Sigma+$	2,0000	
LASTx	400,0000	Ruft letzten x -Wert zurück. Letztes y ist noch im Y-Register gespeichert (xzy xzy zur Überprüfung).

\blacksquare Σ^- 1,0000
 6 ENTER 40 Σ^+ 2,0000

Löscht und ersetzt letztes Datenpaar (400, 6 durch 40, 6).

4 ENTER 20 \blacksquare Σ^- 1,0000
 5 ENTER 20 Σ^+ 2,0000

Löscht und ersetzt das erste Paar (20, 4 auf 20, 5). Es sind noch immer 2 Paare gespeichert.

Ausführung von Statistikberechnungen

Nachdem die Statistikdaten in den Statistikregistern gespeichert sind, können Sie auf diese die im STAT Menü enthaltenen Funktionen anwenden. Drücken Sie \blacksquare STAT zur Anzeige des Menüs.

Das STAT Menü

Menüoption	Beschreibung
{ Σ }	Summationsmenü: n , Σx , Σy , Σx^2 , Σy^2 , Σxy . Siehe "Summationsstatistik".
{ \bar{x}, \bar{y} }	Mittelwert-Menü: \bar{x} , \bar{y} und gewogenes \bar{x} ($\bar{x}w$). Siehe "Mittelwert und Standardabweichung".
{ s }	Standardabweichung-Menü: s_x und s_y . Siehe "Mittelwert und Standardabweichung".
{L.R.}	Lineare-Regression-Menü: Kurvenanpassung, (r , m , b) und lineare Näherung (\hat{x} , \hat{y}). Siehe "Lineare Regression".

Mittelwert und Standardabweichung

Mittelwert-Menü (\bar{x} , \bar{y}):

- Drücken Sie \blacksquare STAT { \bar{x}, \bar{y} } { \bar{x} } zur Berechnung des arithmetischen Mittelwerts (Durchschnitt) der x -Werte.
- Drücken Sie \blacksquare STAT { \bar{x}, \bar{y} } { \bar{y} } zur Berechnung des arithmetischen Mittelwerts (Durchschnitt) der y -Werte.

- Drücken Sie \blacksquare $\boxed{\text{STAT}}$ $\{\bar{x}, \bar{y}\}$ $\{\bar{x}\omega\}$ zur Berechnung des gewogenen Mittels der x -Werte, wobei die y -Werte als Gewichtung oder Häufigkeit der x -Werte dienen.

Standardabweichung-Menü (s). Die Standardabweichung gibt an, mit welcher Streuung die Zahlen um den Mittelwert liegen.

- Drücken Sie \blacksquare $\boxed{\text{STAT}}$ $\{s\}$ $\{s_x\}$ zur Berechnung der Standardabweichung für die x -Werte.*
- Drücken Sie \blacksquare $\boxed{\text{STAT}}$ $\{s\}$ $\{s_y\}$ zur Berechnung der Standardabweichung für die y -Werte.*

Beispiel: Mittelwert und Standardabweichung mit einer Variablen. Die Zeiterfassung für einen bestimmten Produktionsprozeß ergab folgende sechs Meßwerte:

15,5 9,25 10,0
12,5 12,0 8,5

Berechnen Sie den Mittelwert und die Standardabweichung der ermittelten Zeiten.

Tastenfolge:	Anzeige	Beschreibung
\blacksquare $\boxed{\text{CLEAR}}$ $\{\Sigma\}$		Löscht Inhalt der Statistikregister.
15,5 $\boxed{\Sigma+}$	1,0000	Eingabe der ersten Zeit.
9,25 $\boxed{\Sigma+}$ 10 $\boxed{\Sigma+}$	3,0000	Eingabe der restlichen Daten.
12,5 $\boxed{\Sigma+}$ 12 $\boxed{\Sigma+}$	5,0000	
8,5 $\boxed{\Sigma+}$	6,0000	
\blacksquare $\boxed{\text{STAT}}$ $\{\bar{x}, \bar{y}\}$ $\{\bar{x}\}$	11,2917	Berechnet Mittelwert.

* Damit wird die Standardabweichung einer Stichprobe berechnet, wobei davon ausgegangen wird, daß die vorliegenden Daten die Stichprobe einer größeren Grundgesamtheit darstellen. Liegt in Wirklichkeit die Grundgesamtheit vor, so können Sie die *wahre Standardabweichung* berechnen, indem Sie den Mittelwert der ursprünglichen Daten berechnen, diesen mit $\boxed{\Sigma+}$ den Daten hinzufügen und danach die Standardabweichung berechnen.

STAT { \bar{s} } { $\bar{s}x$ } 2,5808 Berechnet Standardabweichung.

Beispiel: Gewogenes Mittel. Ein Fertigungsbetrieb kauft ein bestimmtes Teil vier Mal pro Jahr ein; im letzten Jahr wurden dabei folgende Einkäufe vorgenommen:

Preis je Teil	DM 4,25	DM 4,60	DM 4,70	DM 4,10
Anzahl Teile	250	800	900	1000

Berechnen Sie das gewogene Mittel der Preise. Denken Sie daran, y (Gewicht bzw. Häufigkeit) vor x (Preis) einzugeben.

Tastenfolge:	Anzeige	Beschreibung
CLEAR { Σ }		Löscht Statistikregister.
250 ENTER 4,25 Σ+	1,00000	Eingabe der Preise und deren Gewichtungen.
800 ENTER 4,6 Σ+	2,00000	
900 ENTER 4,7 Σ+	3,00000	
1000 ENTER 4,1 Σ+	4,00000	
STAT { \bar{x}, \bar{y} } { $\bar{x}w$ }	4,4314	Berechnet Mittelwert der Preise (nach Einkaufsmenge gewichtet).

Lineare Regression

Lineare Regression ist ein statistisches Verfahren zum Auffinden derjenigen Geraden, die die Quadrate der Abstände von zwei oder mehreren x,y -Datenpaaren von der Geraden minimiert und damit einen Zusammenhang zwischen den x - und y -Werten schafft.

- Um einen Näherungswert (bzw. Vorhersagewert) für x (oder y) zu berechnen, ist *zuerst* ein gegebener hypothetischer Wert für y (oder x) einzutippen und *danach* STAT {L.R.} { \hat{x} } (oder { \hat{y} }) zu drücken.
- Um die Werte der Kurvenanpassung zu finden, ist STAT {L.R.} gefolgt von { r }, { m } oder { b } zu drücken.

Menü für lineare Regression (L.R.)

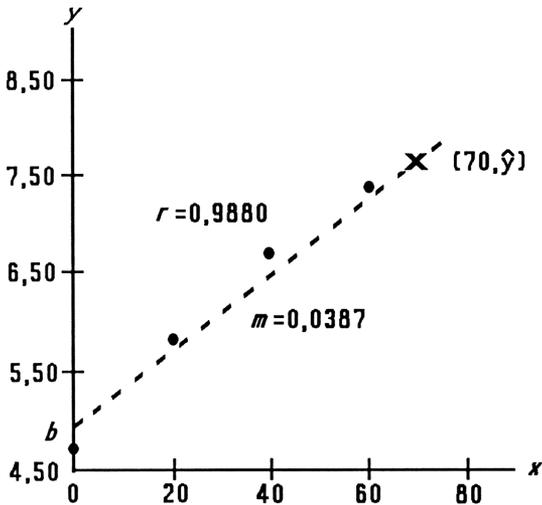
Menüoption	Beschreibung
{ \hat{x} }	Berechnet (schätzt) x für einen gegebenen Wert von y , basierend auf den Werten zur Kurvenanpassung.
{ \hat{y} }	Berechnet (schätzt) y für einen gegebenen Wert von x , basierend auf den Werten zur Kurvenanpassung.
{ r }	Korrelationskoeffizient der (x,y) Daten. Er ist eine Zahl zwischen -1 und $+1$ und gibt Auskunft darüber, wie nahe die Daten an der berechneten Geraden liegen; $+1$ kennzeichnet eine perfekte positive Korrelation, -1 eine perfekte negative Korrelation. Ein Wert nahe 0 bedeutet, daß die Kurve eine schlechte Anpassung darstellt.
{ m }	Steigung der berechneten Geraden.
{ b }	y -Achsenabschnitt der berechneten Geraden.

Beispiel: Kurvenanpassung. Der Ertrag einer neuen Reissorte hängt von der entsprechenden Düngung mit Stickstoff ab. Bestimmen Sie für die nachstehenden Daten über lineare Regression: Korrelationskoeffizient, Steigung und y -Achsenabschnitt.

X, Verwendete Stickstoffmenge (kg/ha)	0,00	20,00	40,00	60,00	80,00
Y, Erzielter Ertrag (t/ha)	4,63	5,78	6,61	7,21	7,78

Tastenfolge:	Anzeige	Beschreibung
■ [CLEAR] { Σ }		Löscht Statistikregister.
4,63 [ENTER] 0 [$\Sigma+$]		Dateneingabe, Anzeige von n : 5 Datenpaare wurden eingegeben.
5,78 [ENTER] 20 [$\Sigma+$]		
6,61 [ENTER] 40 [$\Sigma+$]		
7,21 [ENTER] 60 [$\Sigma+$]		
7,78 [ENTER] 80 [$\Sigma+$]	5,0000	
■ [STAT] { L.R. }	\hat{x} \hat{y} r m b	Zeigt Menü für lineare Regression an.
{ r }	0,9880	Korrelationskoeffizient; die Daten liegen eng gestreut um die berechnete Gerade.

■ [STAT] {L.R.} {m}	0,0387	Steigung der Geraden.
■ [STAT] {L.R.} {b}	4,8560	y-Achsen­schnitt­punkt.



Mit welchem Ertrag ist zu rechnen, wenn 70 kg Stickstoff zur Düngung verwendet wird. Verwenden Sie die Ergebnisse aus der vorherigen Berechnung.

Tastenfolge:	Anzeige	Beschreibung
70	70_	Eingabe von hypothetischem x -Wert.
■ [STAT] {L.R.} { \hat{y} }	7,5615	Vorhergesagter Wert in t/ha.

Grenzen für die Genauigkeit von Daten

Da der Rechner mit einer endlichen Genauigkeit rechnet (12 bis 15 Stellen), ist es unumgänglich, daß bei Berechnungen Rundungsfehler auftreten. Hier zwei Beispiele:

Normalisieren von großen und nahe zusammenliegenden Zahlen.

Der Rechner kann u.U. die Standardabweichung und lineare Regression von einer Variablen nicht korrekt berechnen, wenn deren Werte sich nur durch einen relativ kleinen Betrag unterscheiden. Um diese Fehlerquelle zu vermeiden, sollten Sie die Daten normalisieren, indem Sie jeden Wert als die Differenz von einem Zentralwert (wie z.B. dem Mittelwert) eingeben. Für solche x -Werte muß die Differenz dann wieder zur Berechnung von \bar{x} und \hat{x} addiert werden, wobei \hat{y} und b ebenso angepaßt werden müssen. Würden z.B. als x -Werte 7776999, 7777000 und 7777001 vorliegen, so sollten Sie die Daten als -1, 0 und 1 eingeben; addieren Sie dann 7777000 wieder zu \bar{x} und \hat{x} . Für b ist $7777000 \times m$ zu addieren. Bei der Berechnung von \hat{y} ist sicherzustellen, daß ein x -Wert kleiner als 7777000 vorgegeben wird.

Ähnliche Ungenauigkeiten können auftreten, wenn die x - und y -Werte betragsmäßig große Unterschiede aufweisen. Auch in diesen Fällen kann das Normalisieren der Daten Abhilfe herbeiführen.

Auswirkung von gelöschten Daten. Die Ausführung von   hebt keine Rundungsfehler auf, welche sich vielleicht bei der Eingabe der Originaldaten in die Statistikregister ergeben haben. Dieser Unterschied ist ohne große Auswirkung, außer wenn die falschen Daten einen relativ großen Betrag haben im Vergleich mit den richtigen Daten. In solchen Fällen ist es ratsam, die Statistikregister zu löschen und alle Daten erneut einzugeben.

Summationswerte und Statistikregister

Die Statistikregister stellen sechs eindeutige Speicherorte im Rechner dar und dienen zur Akkumulation der sechs Summationswerte.

Summationsstatistik

Durch Drücken von   { Σ } erhalten Sie Zugriff auf die Inhalte der Statistikregister:

- Drücken Sie { n } zur Anzeige der Anzahl von akkumulierten Datensätzen.
- Drücken Sie { \times } zur Anzeige der Summe der x -Werte.
- Drücken Sie { \surd } zur Anzeige der Summe der y -Werte.

- Drücken Sie $\{x^2\}$, $\{y^2\}$ und $\{xy\}$ zur Anzeige der Summe der Quadrate und des Produkts. Diese Werte können von Interesse bei der Ausführung von weiteren statistischen Berechnungen sein, die nicht direkt im Rechner enthalten sind.

Die Statistikregister im Speicherbereich

Der erforderliche Speicherplatz (48 Bytes) für die Statistikregister wird automatisch reserviert (falls er nicht schon existiert), wenn Sie $\boxed{\Sigma+}$ oder $\boxed{\Sigma-}$ drücken. Die Registerinhalte werden gelöscht und der belegte Speicherbereich wird wieder freigegeben, nachdem $\blacksquare \boxed{\text{CLEAR}}$ $\{\Sigma\}$ ausgeführt wird.

Wenn Sie zum ersten Mal $\boxed{\Sigma+}$ (oder $\boxed{\Sigma-}$) drücken und es ist nicht genügend freier Speicherplatz vorhanden, gibt der Rechner die Meldung MEMORY FULL aus. In dieser Situation sind Variablen oder Programme zu löschen (oder beide), um entsprechend freien Speicherplatz zurück zu erhalten, bevor Sie mit der Eingabe von Statistikdaten beginnen können. Beziehen Sie sich auf "Verwalten des Speicherbereichs" in Anhang B.

Teil 4

Applikationen

Seite 164 12: Mathematikprogramme

204 13: Statistikprogramme

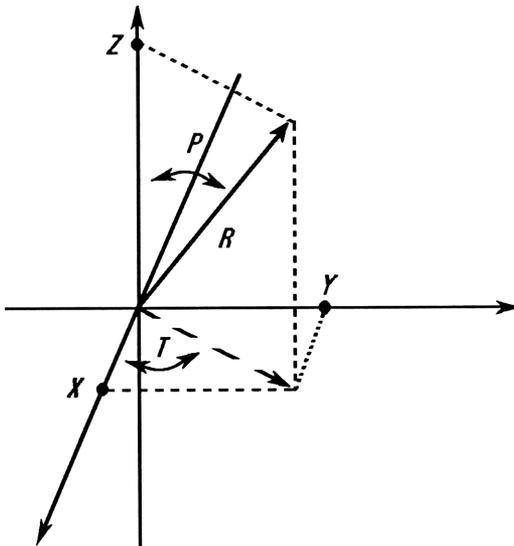
222 14: Sonstige Programme

Mathematikprogramme

Die Speicheranforderung und Prüfsumme kann für jedes Programm-Label mit Hilfe des Programmkatalogs (auf Seite 85) überprüft werden.

Vektoroperationen

Dieses Programm ermöglicht die Basis-Vektoroperationen Addition, Subtraktion, Kreuzprodukt und Skalarprodukt. Das Programm verwendet dazu dreidimensionale Vektoren und erlaubt die Ein- und Ausgabe in Rechtecks- oder Polarnotation. Außerdem können die Winkel zwischen den Vektoren berechnet werden.



Vektor-Koordinatensystem

Für das Programm werden die nachstehenden Gleichungen verwendet.

Koordinatentransformation:

$$\begin{aligned}X &= R \sin(P) \cos(T) & R &= \sqrt{X^2 + Y^2 + Z^2} \\Y &= R \sin(P) \sin(T) & T &= \arctan \frac{Y}{X} \\Z &= R \cos(P) & P &= \arctan \frac{Z}{\sqrt{X^2 + Y^2}}\end{aligned}$$

Vektoraddition- und Subtraktion:

$$\begin{aligned}\mathbf{v}_1 + \mathbf{v}_2 &= (X + U)\mathbf{i} + (Y + V)\mathbf{j} + (Z + W)\mathbf{k} \\ \mathbf{v}_2 - \mathbf{v}_1 &= (U - X)\mathbf{i} + (V - Y)\mathbf{j} + (W - Z)\mathbf{k}\end{aligned}$$

Kreuzprodukt:

$$\mathbf{v}_1 \times \mathbf{v}_2 = (YW - ZV)\mathbf{i} + (ZU - XW)\mathbf{j} + (XV - YU)\mathbf{k}$$

Skalarprodukt:

$$D = XU + YV + ZW$$

Winkel zwischen Vektoren (γ):

$$G = \arccos \frac{D}{R_1 \times R_2},$$

wobei

$$\mathbf{v}_1 = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$$

und

$$\mathbf{v}_2 = U\mathbf{i} + V\mathbf{j} + W\mathbf{k}$$

Von den Eingaberoutinen (LBL P und LBL R) wird der Vektor V_1 angezeigt.

Programmliste:

Programmzeilen:

Beschreibung

R01 LBL R	Definiert Anfang der Anzeige/Eingaberroutine für Rechteckskoordinaten.
R02 INPUT X	Zeigt oder akzeptiert Eingabe von X.
R03 INPUT Y	Zeigt oder akzeptiert Eingabe von Y.
R04 INPUT Z	Zeigt oder akzeptiert Eingabe von Z.
Bytes und Prüfsumme: 006,0; 80FB	
Q01 LBL Q	Definiert Anfang der Transformation von Rechtecks-nach-Polarnotation.
Q02 RCL Y	
Q03 RCL X	
Q04 $\gamma, x \rightarrow \theta, r$	Berechnet $\sqrt{X^2 + Y^2}$ und $\arctan(Y/X)$.
Q05 $x \langle \rangle y$	
Q06 STO T	Sichert $T = \arctan(Y/X)$.
Q07 R+	Holt $\sqrt{X^2 + Y^2}$ zurück.
Q08 RCL Z	
Q09 $\gamma, x \rightarrow \theta, r$	Berechnet $\sqrt{X^2 + Y^2 + Z^2}$ und P .
Q10 STO R	Sichert R.
Q11 $x \langle \rangle y$	
Q12 STO P	Sichert P.
Bytes und Prüfsumme: 018,0; D6D5	
P01 LBL P	Definiert Anfang der Anzeige/Eingaberroutine für Polarkoordinaten.
P02 INPUT R	Zeigt oder akzeptiert Eingabe von R.
P03 INPUT T	Zeigt oder akzeptiert Eingabe von T.
P04 INPUT P	Zeigt oder akzeptiert Eingabe von P.
P05 RCL T	
P06 RCL P	
P07 RCL R	
P08 $\theta, r \rightarrow \gamma, x$	Berechnet $R \cos(P)$ und $R \sin(P)$.
P09 STO Z	Speichert $Z = R \cos(P)$.
P10 R+	
P11 $\theta, r \rightarrow \gamma, x$	Berechnet $R \sin(P) \cos(T)$ und $R \sin(P) \sin(T)$.
P12 STO X	Sichert $X = R \sin(P) \cos(T)$.
P13 $x \langle \rangle y$	

P14 STO Y Sichert $Y = R \sin(P) \sin(T)$.
P15 GTO P Verzweigt zurück zur weiteren Anzeige in
Polarnotation.
Bytes und Prüfsumme: 022,5; AA98

E01 LBL E Definiert den Anfang der Vektoreingabe-
Routine.
E02 RCL X Kopiert Inhalt von X, Y und Z entsprechend
nach U, V und W.
E03 STO U
E04 RCL Y
E05 STO V
E06 RCL Z
E07 STO W
E08 GTO Q Verzweigt zurück zur Konvertierung in
Polarnotation und Anzeige/Eingabe.
Bytes und Prüfsumme: 012,0; 7137

X01 LBL X Definiert Anfang von Vektoraustausch-
Routine.
X02 RCL X Tauscht X, Y und Z entsprechend mit U V
und W.
X03 RCL U
X04 STO X
X05 $\times \langle \rangle$
X06 STO U
X07 RCL Y
X08 RCL V
X09 STO Y
X10 $\times \langle \rangle$
X11 STO V
X12 RCL Z
X13 RCL W
X14 STO Z
X15 $\times \langle \rangle$
X16 STO W
X17 GTO Q Verzweigt zurück zur Konvertierung in
Polarkoordinaten und Anzeige/Eingabe.
Bytes und Prüfsumme: 025,5; EAD8

A01	LBL A	Definiert Anfang der Vektoradditions-Routine.
A02	RCL X	
A03	RCL+ U	
A04	STO X	Sichert $X + U$ in X .
A05	RCL V	
A06	RCL+ Y	
A07	STO Y	Sichert $V + Y$ in Y .
A08	RCL Z	
A09	RCL+ W	
A10	STO Z	Sichert $Z + W$ in Z .
A11	GTO Q	Verzweigt zurück zur Konvertierung in Polarkoordinaten und Anzeige/Eingabe.
Bytes und Prüfsumme: 016,5; F888		
S01	LBL S	Definiert Anfang der Vektorsubtraktions-Routine.
S02	-1	Multipliziert X Y und Z mit (-1) zur Änderung des Vorzeichens.
S03	STO× X	
S04	STO× Y	
S05	STO× Z	
S06	GTO A	Verzweigt zur Vektoradditions-Routine.
Bytes und Prüfsumme: 017,0; 250B		
C01	LBL C	Definiert den Anfang der Kreuzprodukt-Routine.
C02	RCL Y	
C03	RCL× W	
C04	RCL Z	
C05	RCL× V	
C06	-	Berechnet $(YW - ZV)$, was der X Komponente entspricht.
C07	RCL Z	
C08	RCL× U	
C09	RCL X	
C10	RCL× W	
C11	-	Berechnet $(ZU - WX)$, was der Y Komponente entspricht.

C12 RCL X	
C13 RCL× V	
C14 RCL Y	
C15 RCL× U	
C16 -	
C17 STO Z	Speichert $(XV - YU)$, was der Z Komponente entspricht.
C18 R+	
C19 STO Y	Speichert Y Komponente.
C20 R+	
C21 STO X	Speichert X Komponente.
C22 GTO 0	Verzweigt zurück zur Konvertierung in Polarkoordinaten und Anzeige/Eingabe.
Bytes und Prüfsumme: 033,0; D74B	
D01 LBL D	Definiert Anfang der Skalarprodukt- und Vektorwinkel-Routine.
D02 RCL X	
D03 RCL× U	
D04 RCL Y	
D05 RCL× V	
D06 +	
D07 RCL Z	
D08 RCL× W	
D09 +	
D10 STO D	Speichert das Skalarprodukt von $XU + YV + ZW$.
D11 VIEW D	Zeigt Skalarprodukt an.
D12 RCL D	
D13 RCL R	
D14 ÷	Dividiert das Skalarprodukt durch den Betrag des X-, Y-, Z-Vektors.
D15 RCL W	
D16 RCL V	
D17 RCL U	
D18 $\gamma, x \rightarrow \theta, r$	
D19 $x \langle \rangle y$	
D20 R+	
D21 $\gamma, x \rightarrow \theta, r$	Berechnet den Betrag des U-, V-, W-Vektors.
D22 $x \langle \rangle y$	
D23 R+	

D24 ÷	Dividiert vorheriges Ergebnis durch den Betrag.
D25 ACOS	Berechnet den Winkel.
D26 STO G	
D27 VIEW G	Zeigt den Winkel an.
D28 GTO P	Verzweigt zurück zur Konvertierung in Polarkoordinaten und Anzeige/Eingabe.

Bytes und Prüfsumme: 042,0; 739F

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 280,5 Bytes: 192,5 für Programm, 88 für Variablen.

Anmerkungen: Die Länge von Routine S kann um 6,5 Bytes verkürzt werden. Der angezeigte Wert -1 belegt 9,5 Bytes. Wenn dies als 1, gefolgt von $+/-$, dargestellt wird, werden nur 3 Bytes belegt. Für diese Darstellung müssen Sie einen Pseudoschritt zwischen 1 und $+/-$ einfügen und diesen anschließend löschen.

Anwendung des Programms:

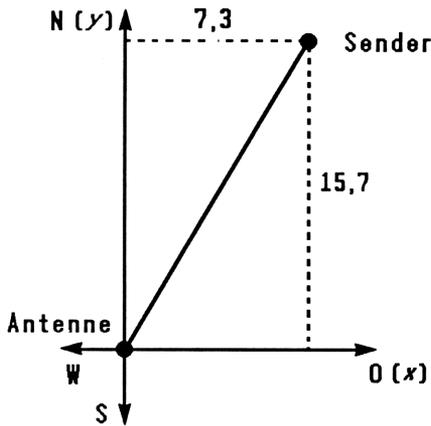
1. Tippen Sie die Programmroutinen ein; drücken Sie \boxed{C} , wenn Sie die Eingabe beendet haben.
2. Liegt Ihr Vektor in Rechtecksform vor, so drücken Sie \boxed{XEQ} R und fahren mit Schritt 4 fort. Ist der Vektor in Polarnotation angegeben, dann drücken Sie \boxed{XEQ} P und fahren mit Schritt 3 fort.
3. Tippen Sie R ein und drücken Sie $\boxed{R/S}$; tippen Sie T ein und drücken Sie $\boxed{R/S}$; tippen Sie P ein und drücken Sie $\boxed{R/S}$. Setzen Sie die Ausführung mit Schritt 5 fort.
4. Tippen Sie X ein und drücken Sie $\boxed{R/S}$; tippen Sie Y ein und drücken Sie $\boxed{R/S}$; tippen Sie Z ein und drücken Sie $\boxed{R/S}$.
5. Wenn Sie einen weiteren Vektor eingeben möchten, so drücken Sie \boxed{XEQ} E (für Eingabe) und fahren mit Schritt 2 fort.

6. Führen Sie die gewünschte Vektoroperation aus:
 - a. Addieren Sie die Vektoren durch Drücken von A;
 - b. Subtrahieren Sie Vektor 1 von Vektor 2 durch Drücken von S;
 - c. Berechnen Sie das Kreuzprodukt durch Drücken von C;
 - d. Berechnen Sie das Skalarprodukt durch Drücken von D und den Winkel zwischen den Vektoren durch Drücken von .
7. Optional: Zum Nachsehen von \mathbf{v}_1 in Polarnotation ist P zu drücken, und danach wiederholt , um die individuellen Elemente anzuzeigen.
8. Optional: Zum Nachsehen von \mathbf{v}_1 in Rechtecksnotation ist R zu drücken, und danach wiederholt , um die individuellen Elemente anzuzeigen.
9. Wenn Sie das Kreuzprodukt addiert, subtrahiert oder berechnet haben, dann wurde \mathbf{v}_1 durch das Ergebnis ersetzt. \mathbf{v}_2 wird nicht verändert. Bei der Fortsetzung der auf dem momentanen Ergebnis basierenden Berechnung ist zu beachten, daß vor der Eingabe eines neuen Vektors E gedrückt werden muß.
10. Gehen Sie zu Schritt 2 über, um die Vektorberechnungen fortzusetzen.

Benutzte Variablen:

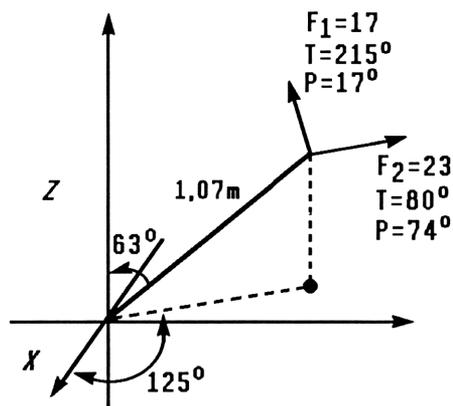
X, Y, Z	Die Komponenten von \mathbf{v}_1 in Rechtecksnotation.
U, V, W	Die Komponenten von \mathbf{v}_2 in Polarnotation.
R, T, P	Radius und Winkel in der x - y Ebene (θ) sowie der Winkel der Z Achse von \mathbf{v}_1 (Φ).
D	Das Skalarprodukt.
G	Der Winkel zwischen den Vektoren (γ).

Beispiel 1: Eine Kurzwellenantenne wird auf einen Sender ausgerichtet, welcher sich in einer Entfernung von 15,7 km nördlich, 7,3 km östlich und 0,76 km tiefer als der Antennenstandpunkt befindet. Verwenden Sie die Rechtecks-nach-Polarnotation Transformation zur Berechnung der Entfernung sowie der Richtung zum Sender.



Tastenfolge:	Anzeige	Beschreibung
$\boxed{\text{XEQ}}$ R	$\times ? \text{Wert}$	Beginnt die Eingabe/Anzeige-Routine für Rechtecksnotation.
7,3 $\boxed{\text{R/S}}$	$Y ? \text{Wert}$	Setzt X gleich mit 7,3.
15,7 $\boxed{\text{R/S}}$	$Z ? \text{Wert}$	Setzt Y gleich mit 15,7.
,76 $\boxed{+/-}$ $\boxed{\text{R/S}}$	$R = 17,3308$	Setzt Z gleich mit $-0,76$ und berechnet R, den Radius.
$\boxed{\text{R/S}}$	$T = 65,0631$	Berechnet T, den Winkel in der x/y Ebene.
$\boxed{\text{R/S}}$	$P = 92,5134$	Berechnet P, den Winkel der z-Achse.

Beispiel 2: Wie groß ist das Drehmoment des nachstehend abgebildeten Hebels in seiner Ausgangslage? Wie groß ist die Kraftkomponente entlang des Hebels? Wie groß ist der Winkel zwischen der Resultante der Kräfte und dem Hebel?



Addieren Sie zuerst die Kräfte.

Tastenfolge:	Anzeige	Beschreibung
<input type="button" value="XEQ"/> P	R?Wert	Beginnt Eingaberoutine für Polarnotation.
17 <input type="button" value="R/S"/>	T?Wert	Setzt Radius gleich 17.
215 <input type="button" value="R/S"/>	P?Wert	Setzt T gleich 215.
17 <input type="button" value="R/S"/>	R?17,0000	Setzt P gleich 17.
<input type="button" value="XEQ"/> E	R?17,0000	Eingabe des Vektors durch Kopieren in v_2 .
23 <input type="button" value="R/S"/>	T?-145,0000	Setzt Radius von v_1 gleich 23.
80 <input type="button" value="R/S"/>	P?17,0000	Setzt T gleich 80.
74 <input type="button" value="R/S"/>	R?23,0000	Setzt P gleich 74.
<input type="button" value="XEQ"/> A	R?29,4741	Addiert Vektoren und zeigt Ergebnisvektor R an.
<input type="button" value="R/S"/>	T?90,7032	Zeigt T von Ergebnisvektor.

R/S	P?39,9445	Zeigt P von Ergebnisvektor.
XEQ E	R?29,4741	Eingabe des Ergebnisvektors.

Da das Drehmoment sich aus dem Kreuzprodukt des Radiusvektors und dem Kraftvektor ergibt ($\mathbf{r} \times \mathbf{F}$), ist der Hebelvektor einzugeben und das Kreuzprodukt zu berechnen.

Tastenfolge:	Anzeige	Beschreibung
1,07 R/S	T?90,7032	Setzt R gleich 1,07.
125 R/S	P?39,9445	Setzt T gleich 125.
63 R/S	R?1,0700	Setzt P gleich 63.
XEQ C	R?18,0209	Berechnet das Kreuzprodukt und zeigt R des Ergebnisses an.
R/S	T?55,3719	Zeigt T von Kreuzprodukt an.
R/S	P?124,3412	Zeigt P von Kreuzprodukt an.
XEQ R	X?8,4554	Zeigt Rechtecksform von Kreuzprodukt an.
R/S	Y?12,2439	
R/S	Z?-10,1660	

Das Skalarprodukt kann zum Berechnen der Kraft (noch immer in \mathbf{v}_2) entlang der Hebelachse benutzt werden.

Tastenfolge:	Anzeige	Beschreibung
$\boxed{\text{XEQ}}$ P	R?18,0209	Beginnt Eingaberoutine für Polarnotation.
1 $\boxed{\text{R/S}}$	T?55,3719	Definiert den Radius als Einheitsvektor.
125 $\boxed{\text{R/S}}$	P?124,3412	Setzt T gleich 125.
63 $\boxed{\text{R/S}}$	R?1,0000	Setzt P gleich 63.
$\boxed{\text{XEQ}}$ D	D=24,1882	Berechnet Skalarprodukt.
$\boxed{\text{R/S}}$	G=34,8490	Berechnet Winkel zwischen resultierendem Kraftvektor und Hebel.
$\boxed{\text{R/S}}$	R?1,0000	Rückkehr zur Eingaberoutine.

Lösen eines linearen Gleichungssystems über Determinantenverfahren

Dieses Programm dient zum Lösen eines linearen Gleichungssystems mit zwei oder drei Unbekannten. Als Lösungsverfahren dient die Cramersche Regel, welche auch als Determinantenverfahren bekannt ist.

Bei einem gegebenen System von drei linearen Gleichungen

$$\begin{aligned} AX + DY + GZ &= J \\ BX + EY + HZ &= K \\ CX + FY + IZ &= L \end{aligned}$$

können die drei Unbekannten X, Y und Z über die Determinanten berechnet werden.

$$X = \frac{Det_x}{Det} \qquad Y = \frac{Det_y}{Det} \qquad Z = \frac{Det_z}{Det}$$

$$Det = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \qquad Det_x = \begin{bmatrix} J & D & G \\ K & E & H \\ L & F & I \end{bmatrix}$$

$$Det_y = \begin{bmatrix} A & J & G \\ B & K & H \\ C & L & I \end{bmatrix} \qquad Det_z = \begin{bmatrix} A & D & J \\ B & E & K \\ C & F & L \end{bmatrix}$$

Programmliste:

Programmzeilen:	Beschreibung
A01 LBL A	Startpunkt für Eingabe aller bekannten Werte.
A02 1,012	Schleifensteuerzahl: Durchläuft Schleife von 1 bis 12, wobei der Schleifenzähler jeweils um 1 erhöht wird.
A03 STO i	Speichert Steuerwert in Indexvariable.
Bytes und Prüfsumme: 012,5; 7878	
L01 LBL L	Beginn der Eingabeschleife.
L02 INPUT(i)	Eingeben und Speichern der Variablen, die durch <i>i</i> adressiert wird.
L03 ISG i	Erhöhung von <i>i</i> um 1.
L04 GTO L	Wenn <i>i</i> kleiner als 13, Rücksprung zu LBL L und Eingabe des nächsten Wertes.
L05 GTO A	Rückkehr zu LBL A zum Durchsehen der Werte.
Bytes und Prüfsumme: 007,5; C1DE	
S01 LBL S	Startpunkt für Lösung des linearen Gleichungssystems.
S02 9	Indexwert von <i>I</i> für indirekte Adressierung.
S03 STO i	Speichert Indexwert.

S04 XEQ E	Austausch der Lösungsspalte mit der Koeffizientenspalte, die I enthält.
S05 XEQ D	Berechnet Determinante.
S06 STO Z	Sichert Determinante in Z .
S07 XEQ E	Speichert Determinante wieder in ursprünglicher Form.
S08 6	Indexwert von F für indirekte Adressierung.
S09 STO i	Speichert Indexwert.
S10 XEQ E	Austausch von Lösungsspalte und Spalte, welche F enthält.
S11 XEQ D	Berechnet Determinante.
S12 STO Y	Sichert Determinante in Y .
S13 XEQ E	Speichert Determinante wieder in ursprünglicher Form.
S14 3	Indexwert v. C für indirekte Adressierung.
S15 STO i	Speichert Indexwert in Indexvariable.
S16 XEQ E	Austausch von Lösungsspalte und Spalte, welche F enthält.
S17 XEQ D	Berechnet Determinante.
S18 STO X	Sichert Determinante in X .
S19 XEQ E	Speichert Determinante wieder in ursprünglicher Form.
S20 XEQ D	Berechnet Determinante von ursprünglichen Koeffizienten.
S21 STO+ X	Dividiert durch urspr. Determinante.
S22 STO+ Y	
S23 STO+ Z	
S24 RCL X	Rückrufen und Anzeigen der Ergebnisse.
S25 VIEW X	
S26 RCL Y	
S27 VIEW Y	
S28 RCL Z	
S29 VIEW Z	
S30 RTN	Rückkehr zum aufrufenden Programm oder zu PRGM TOP.

Bytes und Prüfsumme: 045,0; 3971

E01 LBL E	Diese Routine tauscht die Spalten für Cramersche Regel aus.
E02 RCL(i)	Holt letztes Element aus Spalte der Koeffizientendeterminante.
E03 RCL L	Holt letztes Element aus Lösungsvektor.
E04 STO(i)	Sichert Vektorelement in Determinante.

E05	$x \langle \rangle y$	Holt das Koeffizientenelement zurück.
E06	STO L	Sichert das Element im Vektor.
E07	DSE i	Ändert Index, um auf mittleres Element der Determinantenspalte zu zeigen.
E08	RCL(i)	Ruft mittleres Element der Determinantenspalte ab.
E09	RCL K	Ruft mittleres Element von Vektor ab.
E10	STO(i)	Sichert Vektorelement in Determinante.
E11	$x \langle \rangle y$	Holt das Koeffizientenelement zurück.
E12	STO K	Sichert das Element im Vektor.
E13	DSE i	Ändert Indexwert, um auf oberstes Element der Determinantenspalte zu zeigen.
E14	RCL(i)	Ruft oberstes Element der Determinantenspalte ab.
E15	RCL J	Ruft oberstes Element des Vektors ab.
E16	STO(i)	Sichert Vektorelement in Determinante.
E17	$x \langle \rangle y$	Holt das Element der Determinante zurück.
E18	STO J	Sichert das Determinantenelement im Vektor.
E19	2	
E20	STO+ i	Speichert für i den ursprünglichen Wert zurück.
E21	RTN	Rückkehr zum aufrufenden Programm oder zu PRGM TOP.

Bytes und Prüfsumme: 031,5; 8420

D01	LBL D	Diese Routine berechnet Determinante.
D02	RCL A	
D03	RCL× E	
D04	RCL× I	Berechnet $A \times E \times I$.
D05	RCL D	
D06	RCL× H	
D07	RCL× C	
D08	+	Berechnet $(A \times E \times I) + (D \times H \times C)$.
D09	RCL G	
D10	RCL× F	
D11	RCL× B	
D12	+	Berechnet $(A \times E \times I) + (D \times H \times C) + (G \times F \times B)$.
D13	RCL G	
D14	RCL× E	

D15 RCL× C
 D16 - $(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C).$
 D17 RCL A
 D18 RCL× F
 D19 RCL× H
 D20 - $(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H).$
 D21 RCL D
 D22 RCL× B
 D23 RCL× I
 D24 - $(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H) - (D \times B \times I).$
 D25 RTN Rückkehr zum aufrufenden Programm oder zu PRGM TOP.

Bytes und Prüfsumme: 037,5; 152E

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 262 Bytes: 134 für Programm, 128 für Variable.

Anwendung des Programms:

1. Tippen Sie die Programmroutinen ein; drücken Sie am Ende \boxed{C} .
2. Drücken Sie \boxed{XEQ} A zur Eingabe der Koeffizienten (d.h. A bis L) des linearen Gleichungssystems.
3. Tippen Sie nach Aufforderung jeden Koeffizienten (A bis L) ein und drücken Sie $\boxed{R/S}$.
4. Optional: Zur Berechnung der Determinante eines 3×3 Systems ist \boxed{XEQ} D zu drücken.
5. Berechnen Sie die Systemlösung durch Drücken von \boxed{XEQ} S.
6. Sehen Sie sich den Wert von X an und drücken Sie $\boxed{R/S}$ zur Anzeige von Y.
7. Drücken Sie $\boxed{R/S}$ zur Anzeige des Werts für Z.
8. Gehen Sie zurück zu Schritt 2, um neues System zu berechnen.

Benutzte Variablen:

A bis I	Koeffizienten der Gleichungen.
J bis L	Rechte Seite der Gleichungen.
X bis Z	Unbekannte.
i	Schleifensteuerzahl (Indexvariable).

Anmerkungen: Dieses Programm ist für ein System von zwei oder drei Gleichungen (d.h. einer Matrix mit $n \leq 3$) geeignet.

Für 2×2 Lösungen ist für die Koeffizienten C, F, G, H und L Null vorzugeben. Verwenden Sie 1 für Koeffizient I; bei rechteckigen Matrizen ist Null für die "fehlenden" Koeffizienten vorzugeben.

Es gibt nicht für jedes Gleichungssystem eine Lösung. In diesen Fällen wird DIVIDE BY 0 bei Zeile S21 angezeigt.

Beispiel 1: Berechnen Sie für nachstehendes Gleichungssystem die Determinante und die Systemlösung. Setzen Sie danach die Werte in die erste Gleichung ein, um die Gleichung zu überprüfen.

$$\begin{aligned} 23X + 15Y + 17Z &= 1 \\ 8X + 11Y - 6Z &= 1 \\ 4X + 15Y + 12Z &= 1 \end{aligned}$$

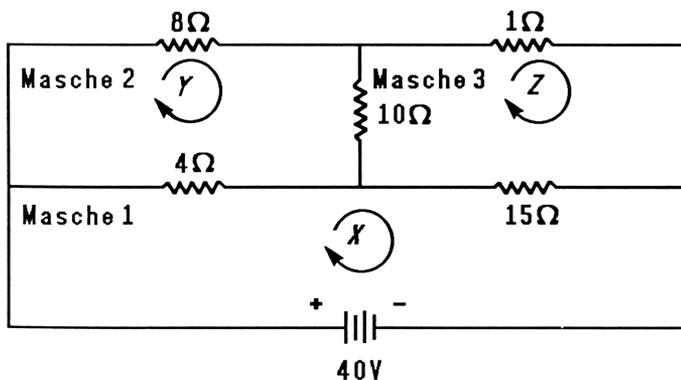
Tastenfolge:	Anzeige	Beschreibung
<input type="text" value="XEQ"/> A	A?Wert	Beginnt Eingabe.
23 <input type="text" value="R/S"/>	B?Wert	Speichert 23 für ersten Koeffizient, A.
8 <input type="text" value="R/S"/>	C?Wert	Speichert 8 für B.
4 <input type="text" value="R/S"/>	D?Wert	Speichert 4 für C.
15 <input type="text" value="R/S"/>	E?Wert	Speichert 15 für D.
:	:	Setzt Eingabe für alle Variablen (E bis L) fort.
1 <input type="text" value="R/S"/>	A?23,0000	Rücksprung zum ersten Koeffizienten.
<input type="text" value="XEQ"/> D	4,598,0000	Berechnet die Determinante.

<input type="button" value="XEQ"/> S	$X=0,0043$	Löst Gleichungssystem und zeigt X an.
<input type="button" value="R/S"/>	$Y=0,0787$	Zeigt Y an.
<input type="button" value="R/S"/>	$Z=-0,0165$	Zeigt Z an.

Um die erhaltene Lösung zu überprüfen:

Tastenfolge:	Anzeige	Beschreibung
23 <input type="button" value="RCL"/> <input type="button" value="x"/> X	0,1000	Multipliziert X mit 23.
15 <input type="button" value="RCL"/> <input type="button" value="x"/> Y	1,1809	Multipliziert Y mit 15.
<input type="button" value="+"/>	1,2810	Addiert die zwei letzten Ergebnisse.
17 <input type="button" value="RCL"/> <input type="button" value="x"/> Z	-0,2810	Multipliziert Z mit 17.
<input type="button" value="+"/>	1,0000	Schließt die linke Seite der Gleichung ab. Da die linke und rechte Seite gleich 1 ist (bei 11 signifikanten Stellen), liegt eine korrekte Lösung vor.

Beispiel 2: Lösen Sie die Maschenströme im nachstehend abgebildeten Netzwerk:



Schreiben Sie zuerst die Gleichungen für den Spannungsabfall in jeder Masche des Netzwerks.

Für Masche 1: $4X - 4Y + 15X - 15Z - 40 = 0$

Für Masche 2: $4Y - 4X + 8Y + 10Y - 10Z = 0$

Für Masche 3: $10Z - 10Y + Z + 15Z - 15X = 0$

Das Zusammenfassen ähnlicher Terme innerhalb jeder Gleichung führt zu

$$\begin{aligned} 19X - 4Y - 15Z &= 40 \\ -4X + 22Y - 10Z &= 0 \\ -15X - 10Y + 26Z &= 0 \end{aligned}$$

Tastenfolge:	Anzeige	Beschreibung
XEQ A	A?Wert	Beginnt Eingaberoutine.
19 R/S	B?Wert	Speichert 19 für ersten Koeffizienten, A.
4 +/- R/S	C?Wert	Speichert -4 für B.
15 +/- R/S	D?Wert	Speichert -15 für C.
:		Setzt die Eingabe für D bis L fort.
0 R/S	A?19,0000	Eingabe von L und Rückkehr zur Eingabestelle für ersten Koeffizienten.
XEQ S	X=7,8601	Löst Gleichungssystem und zeigt X an.
R/S	Y=4,2298	Zeigt Y an.
R/S	Z=6,1615	Zeigt Z an.

Lösen eines linearen Gleichungssystems durch Invertieren der Matrix

Dieses Programm löst ein lineares Gleichungssystem mit zwei oder drei Unbekannten; zur Berechnung der Lösung wird die entsprechende Matrix invertiert und anschließend multipliziert.

Ein System aus drei linearen Gleichungen

$$\begin{aligned}AX + DY + GZ &= J \\ BX + EY + HZ &= K \\ CX + FY + IZ &= L\end{aligned}$$

kann durch nachstehende Matrixgleichung dargestellt werden:

$$\begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} J \\ K \\ L \end{bmatrix}$$

Die Matrixgleichung kann für X , Y und Z gelöst werden, indem die Ergebnismatrix mit der Inversen der Koeffizientenmatrix multipliziert wird.

$$\begin{bmatrix} A' & D' & G' \\ B' & E' & H' \\ C' & F' & I' \end{bmatrix} \begin{bmatrix} J \\ K \\ L \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Einzelheiten zur Ermittlung der Inversen finden Sie bei den Erläuterungen über die Invertierungsroutine I.

Programmliste:

Programmzeilen:	Beschreibung
A01 LBL A	Eingabebeginn für Koeffizienten.
A02 1,012	Schleifensteuerzahl: Durchlaufen der Schleife von 1 bis 12, Erhöhung des Schleifenzählers um jeweils 1.
A03 STO i	Speichert Steuerwert in Indexvariable.
Bytes und Prüfsumme: 012,5; 7878	
L01 LBL L	Beginn der Eingabeschleife.
L02 INPUT(i)	Eingeben und Speichern der Variablen, die durch i adressiert wird.
L03 ISG i	Addiert 1 zu i .
L04 GTO L	Wenn i kleiner als 13 ist, Rücksprung zu LBL L und Eingabe des nächsten Werts.
L05 GTO A	Sprung zu LBL A, um Werte anzusehen.
Bytes und Prüfsumme: 007,5; C1DE	
I01 LBL I	Diese Routine invertiert eine 3×3 Matrix.
I02 XEQ D	Berechnet Determinante und sichert den Wert für die Divisionsschleife J.
I03 STO W	
I04 RCL A	
I05 RCL \times I	
I06 RCL C	
I07 RCL \times G	
I08 -	
I09 STO X	Berechnet $E' \times$ Determinante = $AI - CG$.
I10 RCL C	
I11 RCL \times D	
I12 RCL A	
I13 RCL \times F	
I14 -	
I15 STO Y	Berechnet $F' \times$ Determinante = $CD - AF$.
I16 RCL B	
I17 RCL \times G	
I18 RCL A	
I19 RCL \times H	
I20 -	
I21 STO Z	Berechnet $H' \times$ Determinante = $BG - AH$.
I22 RCL A	

I23 RCL× E	
I24 RCL B	
I25 RCL× D	
I26 -	
I27 STO i	Berechnet $I' \times$ Determinante = $AE - BD$.
I28 RCL E	
I29 RCL× I	
I30 RCL F	
I31 RCL× H	
I32 -	
I33 STO A	Berechnet $A' \times$ Determinante = $EI - FH$.
I34 RCL C	
I35 RCL× H	
I36 RCL B	
I37 RCL× I	
I38 -	Berechnet $B' \times$ Determinante = $CH - BI$.
I39 RCL B	
I40 RCL× F	
I41 RCL C	
I42 RCL× E	
I43 -	
I44 STO C	Berechnet $C' \times$ Determinante = $BF - CE$.
I45 R+	
I46 STO B	Speichert B' .
I47 RCL F	
I48 RCL× G	
I49 RCL D	
I50 RCL× I	
I51 -	Berechnet $D' \times$ Determinante = $FG - DI$.
I52 RCL D	
I53 RCL× H	
I54 RCL E	
I55 RCL× G	
I56 -	
I57 STO G	Berechnet $G' \times$ Determinante = $DE - EG$.
I58 R+	
I59 STO D	Speichert D' .
I60 RCL i	
I61 STO I	Speichert I' .
I62 RCL X	
I63 STO E	Speichert E' .

I64 RCL Y
 I65 STO F Speichert F' .
 I66 RCL Z
 I67 STO H Speichert H' .
 I68 9
 I69 STO i Stellt Index auf letztes Matrixelement ein.
 I70 RCL W Ruft Wert der Determinante zurück.
 Bytes und Prüfsumme: 105,0; E5C1

J01 LBL J Diese Routine schließt das Invertieren mit
 Division durch die Determinante ab.
 J02 STO+(i) Dividiert Element.
 J03 DSE i Reduziert Indexwert, damit dieser näher auf
 A zeigt.
 J04 GTO J Rücksprung zur Verarbeitung des nächsten
 Werts.
 J05 RTN Rückkehr zum aufrufenden Programm oder
 zu PRGM TOP.
 Bytes und Prüfsumme: 007,5; A354

M01 LBL M Diese Routine multipliziert eine
 Spaltenmatrix und eine 3×3 Matrix.
 M02 7 Stellt Index auf letztes Element in der ersten
 Zeile ein.
 M03 XEQ N
 M04 8 Stellt Index auf letztes Element in zweiter
 Zeile ein.
 M05 XEQ N
 M06 9 Stellt Index auf letztes Element in dritter
 Zeile ein.
 Bytes und Prüfsumme: 009,0; 0A85

N01 LBL N Diese Routine berechnet das Produkt des
 Spaltenvektors und einer durch den Index
 adressierten Zeile.
 N02 STO i Sichert Indexwert in i .
 N03 RCL J Ruft J aus Spaltenmatrix zurück.
 N04 RCL K Ruft K aus Spaltenmatrix zurück.
 N05 RCL L Ruft L aus Spaltenmatrix zurück.
 N06 RCL*(i) Multiplikation mit letztem Zeilenelement.
 N07 XEQ P Multiplikation mit mittlerem Zeilenelement
 und Addition.

N08 XEQ P	Multiplikation mit drittem Zeilenelement und Addition.
N09 23	Stellt Index ein, um X, Y oder Z anzuzeigen, in Abhängigkeit der Eingabezeile.
N10 STO+ i	
N11 R+	Holt Ergebnis zurück.
N12 STO(i)	Speichert Ergebnis.
N13 VIEW(i)	Zeigt Ergebnis an.
N14 RTN	Rückkehr zum aufrufenden Programm oder zu PRGM TOP.
Bytes und Prüfsumme: 021,0; BBBF	
P01 LBL P	Diese Routine multipliziert und addiert Werte innerhalb einer Zeile.
P02 x<>y	Holt nächsten Spaltenwert.
P03 DSE i	Stellt Index auf den nächsten Zeilenwert ein.
P04 DSE i	
P05 DSE i	
P06 RCLx(i)	Multipliziert Spaltenwert mit Zeilenwert.
P07 +	Addiert Produkt zu vorhergehender Summe.
P08 RTN	Rückkehr zum aufrufenden Programm.
Bytes und Prüfsumme: 012,0; 520E	
D01 LBL D	Diese Routine berechnet die Determinante.
D02 RCL A	
D03 RCLx E	
D04 RCLx I	Berechnet $A \times E \times I$.
D05 RCL D	
D06 RCLx H	
D07 RCLx C	
D08 +	Berechnet $(A \times E \times I) + (D \times H \times C)$.
D09 RCL G	
D10 RCLx F	
D11 RCLx B	
D12 +	Berechnet $(A \times E \times I) + (D \times H \times C) + (G \times F \times B)$.
D13 RCL G	
D14 RCLx E	
D15 RCLx C	
D16 -	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C)$.

D17 RCL A
 D18 RCL× F
 D19 RCL× H
 D20 - $(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H)$.
 D21 RCL D
 D22 RCL× B
 D23 RCL× I
 D24 - $(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H) - (D \times B \times I)$.
 D25 RTN Rückkehr zum aufrufenden Programm oder zu PRGM TOP.
 Bytes und Prüfsumme: 037,5; 152E

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 348 Bytes: 212 für Programm, 136 für Variablen.

Anwendung des Programms:

1. Tippen Sie die Programmroutinen ein; drücken Sie am Ende \boxed{C} .
2. Drücken Sie \boxed{XEQ} A zur Eingabe der Matrixkoeffizienten und des Spaltenvektors.
3. Tippen Sie den Koeffizienten oder das Vektorelement (A bis L) bei jeder Aufforderung ein und drücken Sie $\boxed{R/S}$.
4. Optional: Drücken Sie \boxed{XEQ} D zur Berechnung der Determinante eines 3×3 Systems.
5. Drücken Sie \boxed{XEQ} zur Berechnung der Inversen der 3×3 Matrix.
6. Optional: Drücken Sie \boxed{XEQ} A und wiederholt $\boxed{R/S}$, um die Werte der invertierten Matrix anzusehen.
7. Drücken Sie \boxed{XEQ} M zur Multiplikation der Inversen mit dem Spaltenvektor und anschließender Anzeige des Wertes von X . Drücken Sie $\boxed{R/S}$, um den Wert von Y anzusehen, danach erneut $\boxed{R/S}$, um Z anzusehen.
8. Um ein neues Gleichungssystem zu lösen, ist wieder mit Schritt 2 zu beginnen.

Benutzte Variablen:

<i>A</i> bis <i>I</i>	Koeffizienten der Matrix.
<i>J</i> bis <i>L</i>	Werte des Spaltenvektors.
<i>W</i>	Hilfsvariable zum Speichern der Determinante.
<i>X</i> bis <i>Z</i>	Werte des Ausgabevektors; auch als Hilfsvariablen benutzt.
<i>i</i>	Schleifensteuerzahl (Indexvariable); auch für Zwischenspeicherungen benutzt.

Anmerkungen: Für Lösungen einer 2×2 Matrix ist Null für die Koeffizienten *C*, *F*, *H* *G* und *L* vorzugeben. Speichern Sie für den Koeffizient *I* den Wert 1.

Nicht alle Gleichungssysteme besitzen eine Lösung.

Beachten Sie, daß die Routinen *A*, *L* und *D* gemeinsam in diesem Programm und im Programm "Lösen eines linearen Gleichungssystem über Determinantenverfahren" verwendet werden.

Beispiel: Berechnen Sie für das nachstehende System die Inverse sowie die Systemlösung. Sehen Sie sich die invertierte Matrix an; invertieren Sie die Matrix erneut und überprüfen Sie, ob wieder die ursprüngliche Matrix erhalten wurde.

$$\begin{aligned} 23X + 15Y + 17Z &= 31 \\ 8X + 11Y - 6Z &= 17 \\ 4X + 15Y + 12Z &= 14 \end{aligned}$$

Tastensequenz:	Anzeige	Beschreibung
<input type="text" value="XEQ"/> <i>A</i>	<i>A</i> ?Wert	Beginnt Eingabe.
23 <input type="text" value="R/S"/>	<i>B</i> ?Wert	Speichert 23 für ersten Koeffizienten, <i>A</i> .
8 <input type="text" value="R/S"/>	<i>C</i> ?Wert	Speichert 8 für <i>B</i> .
4 <input type="text" value="R/S"/>	<i>D</i> ?Wert	Speichert 4 für <i>C</i> .
15 <input type="text" value="R/S"/>	<i>E</i> ?Wert	Speichert 15 für <i>D</i> .
:	:	Fortsetzung der Eingabe für <i>D</i> bis <i>L</i> .

14	R/S	A?23,0000	Rückkehr zum ersten Koeffizienten.
	XEQ I	4.598,0000	Berechnet die Inverse und zeigt Determinante an.
	XEQ M	X=0,9306	Multiplikation mit Spaltenvektor, um X zu berechnen.
	R/S	Y=0,7943	Berechnung und Anzeige von Y.
	R/S	Z=-0,1364	Berechnung und Anzeige von Z.
	XEQ A	A?0,0483	Beginnt Durchsehen der invertierten Matrix.
	R/S	B?0,0261	Zeigt nächsten Wert an, ...
:		:	... und so weiter ...
	R/S	I?0,0289	zeigt nächsten Wert an.
	XEQ I	0,0002	Invertiert Inverse, um wieder ursprüngliche Matrix zu erzeugen.
	XEQ A	A?23,0000	Beginnt Durchsicht der invertierten Inversen.
	R/S	B?8,0000	Zeigt nächsten Wert an, ...
:		:	... und so weiter.

Quadratische Gleichung

Dieses Programm benutzt die Lösungsformel für die quadratische Gleichung, um die reellen und komplexen Wurzeln eines Polynoms 2. Grades zu berechnen.

Ein Polynom 2. Grades

$$ax^2 + bx + c = 0$$

kann mit Hilfe der Lösungsformel

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

für x gelöst werden, wobei $b^2 - 4ac$ die Diskriminante ist. Liegt eine komplexe Wurzel vor (negative Diskriminante), so ist der Realteil

$$R = \frac{-b}{2a}$$

während der Imaginärteil die Form

$$I = \pm \frac{i \sqrt{|b^2 - 4ac|}}{2a}$$

hat. Bei reellen Wurzeln wird immer zuerst die Lösung mit dem größeren Absolutbetrag berechnet. Dadurch werden Ungenauigkeiten, die eingeführt werden können, wenn die Quadratwurzel der Diskriminante fast gleich mit b ist, auf ein Minimum beschränkt. Nachdem x_1 berechnet wurde, läßt sich x_2 mit Hilfe der Beziehung

$$x_2 = \frac{c}{ax_1}$$

berechnen. Numerische Fehler, wie z.B. der vom Programm vermiedenen, sind nicht ungewöhnlich für Computerprogramme. Jeder Computer, welcher zwangsläufig eine endliche Stellenzahl bei der Ausführung von Berechnungen verwendet, erzeugt keine numerisch absolut korrekten Ergebnisse, sofern nicht bei der Wahl und Implementation des Lösungsverfahrens Vorsicht ausgeübt wird.

Vom Computer erzeugte ungenaue Ergebnisse lassen sich oft vermeiden, wenn beim Entwickeln der Software solche numerischen Probleme miteinbezogen werden. In Beispiel 4 wird dieser Unterschied aufgezeigt.

Programmliste:

Programmzeilen:	Beschreibung
Q01 LBL Q	Definiert den Anfang der Routine für die quadratische Lösungsformel.
Q02 INPUT A	Eingeben und Speichern des Werts von A.
Q03 $x=0?$	Wenn A gleich Null, Rücksprung zwecks erneuter Eingabe für A.
Q04 GTO Q	
Q05 INPUT B	Eingeben und Speichern des Werts von B.
Q06 INPUT C	Eingeben und Speichern des Werts von C.
Q07 $x=0?$	Wenn C gleich Null, Rücksprung und erneute Eingabe für alle Variablen.
Q08 GTO Q	
Q09 RCL B	Ruft B zurück.
Q10 +/-	-B.
Q11 CF 0	Löscht Flag 0. (Annahme, daß (-B) positiv ist.)
Q12 $x<0?$	Ist (-B) negativ?
Q13 SF 0	Setzt Flag 0, wenn Antwort "ja" ist.
Q14 RCL B	
Q15 x^2	Berechnet B^2 .
Q16 4	
Q17 RCL× A	
Q18 RCL× C	
Q19 -	Berechnet $B^2 - 4AC$.
Q20 $x<0?$	Test, ob imaginäre Wurzeln vorliegen.
Q21 GTO I	Verzweigung zur Imaginär-Routine, falls ja.
Q22 SQRT	$\sqrt{B^2 - 4AC}$.
Q23 FS? 0	Test, ob (-B) negativ ist.
Q24 +/-	Wählt Wurzel mit größerem Absolutbetrag.
Q25 +	$-B - \sqrt{B^2 - 4AC}$ oder $-B + \sqrt{B^2 - 4AC}$.
Q26 2	
Q27 ÷	
Q28 RCL÷ A	Berechnet X von größerem Absolutbetrag.
Q29 STO X	Speichert Wert von X mit größerem Absolutbetrag.

Q30 VIEW X	Zeigt X an.
Q31 RCL C	Berechnet zweiten Wert von X.
Q32 RCL÷ A	
Q33 RCL÷ X	Berechnet $X = C \div AX$.
Q34 STO X	Speichert zweiten Wert von X.
Q35 VIEW X	Zeigt X an.
Q36 GTO Q	Rücksprung zur Berechnung einer neuen Aufgabe.

Bytes und Prüfsumme: 054,0; A04D

I01 LBL I	Definiert Beginn der Routine zur Berechnung einer imaginären Wurzel.
I02 ABS	
I03 SQRT	
I04 2	
I05 RCL× A	
I06 ÷	Berechnet <u>Absolutbetrag</u> von $\sqrt{B^2 - 4AC} \div 2A$.
I07 STO I	Speichert den Imaginärteil.
I08 RCL B	
I09 +/-	
I10 LAST×	Rücksicherung von 2A.
I11 ÷	
I12 STO R	Speichert den Realteil in R.
I13 RCL I	Ruft den Imaginärteil von X zurück.
I14 RCL R	Ruft den Realteil von X zurück.
I15 VIEW R	Zeigt den Realteil an.
I16 VIEW I	Zeigt den Imaginärteil an.
I17 GTO Q	Rücksprung zur Berechnung einer neuen Aufgabe.

Bytes und Prüfsumme: 025,5; DA3B

Benutzte Flags: Flag 0 kennzeichnet das Vorzeichen von $(-B)$. Falls $(-B)$ negativ ist, so wird Flag 0 gesetzt; er wird später im Programm wieder getestet, um sicherzustellen, daß die als erstes berechnete Wurzel die Wurzel mit dem größeren Absolutbetrag ist. Wenn $(-B)$ negativ ist (Flag 0 gesetzt), dann subtrahiert die Routine die Quadratwurzel der Diskriminante von $(-B)$. Ist $(-B)$ positiv (Flag 0 gelöscht), dann wird die Quadratwurzel zur Diskriminante addiert.

Erforderlicher Speicherplatz: 127,5 Bytes: 79,5 für Programm, 48 für Variablen.

Anmerkungen: Eine Erweiterung dieses Programms zum Lösen kubischer Gleichungen wäre relativ einfach. Da eine kubische Gleichung immer mindestens eine reelle Wurzel hat, könnte die SOLVE Funktion zum Lösen der Wurzel benutzt werden. Durch anschließende synthetische Division könnte die kubische Gleichung auf eine quadratische Gleichung reduziert werden, welche vom vorliegenden Programm gelöst werden könnte.

Anwendung des Programms:

1. Tippen Sie die Programmroutinen ein; drücken Sie am Ende \boxed{C} .
2. Drücken Sie \boxed{XEQ} Q, um die Routine zum Lösen einer quadratischen Gleichung zu starten.
3. Tippen Sie A ein und drücken Sie $\boxed{R/S}$.
4. Tippen Sie B ein und drücken Sie $\boxed{R/S}$.
5. Tippen Sie C ein und drücken Sie $\boxed{R/S}$.
6. Sehen Sie sich den ersten Wert von X an, falls reelle Wurzeln vorliegen, oder sehen Sie sich den Realteil R an, falls imaginäre Wurzeln vorliegen.
7. Liegen imaginäre Wurzeln vor, so drücken Sie $\boxed{R/S}$, um den zweiten Wert von X oder den Imaginärteil I anzusehen.
8. Um eine neue Gleichung zu berechnen, ist $\boxed{R/S}$ zu drücken und wieder mit Schritt 3 fortzufahren.

Benutzte Variablen:

- A Koeffizient von x^2 .
B Koeffizient von x .
C Konstante.
X Die erste oder die zweite Lösung x .
R Der Realteil der komplexen Wurzel.
I Der positive Imaginärteil der komplexen Wurzel.

Beispiel 1: Berechnen Sie die Wurzel von $3x^2 + 5x - 3 = 0$.

Tastensequenz	Anzeige	Beschreibung
$\boxed{\text{XEQ}}$ Q	A?Wert	Beginn des Programms zur Lösung einer quadratischen Gleichung.
3 $\boxed{\text{R/S}}$	B?Wert	Speichert 3 in A.
5 $\boxed{\text{R/S}}$	C?Wert	Speichert 5 in B.
3 $\boxed{+/-}$ $\boxed{\text{R/S}}$	X=-2,1350	Speichert -3 in C und berechnet die erste Lösung für X.
$\boxed{\text{R/S}}$	X=0,4684	Berechnet zweite Lösung von X.

Beispiel 2: Bestimmen Sie die Wurzeln von $3x^2 + 5x + 3 = 0$. Beachten Sie, daß der einzige Unterschied zur vorherigen Aufgabenstellung im Vorzeichen von C liegt. Wenn Sie Beispiel 1 bereits bearbeitet haben, müssen Sie nur noch das Vorzeichen von C austauschen:

Tastensequenz	Anzeige	Beschreibung
$\boxed{\text{R/S}}$	A?3,0000	Fortsetzung des Programms.
$\boxed{\text{R/S}}$	B?5,0000	Behält A bei.
$\boxed{\text{R/S}}$	C?-3,0000	Behält B bei.
$\boxed{+/-}$ $\boxed{\text{R/S}}$	R=-0,8333	Ändert das Vorzeichen von C und berechnet den Realteil der komplexen Wurzel.
$\boxed{\text{R/S}}$	I=0,5528	Berechnet den positiven Teil der imaginären Wurzel.

Beispiel 3: Ein Gegenstand wird aus einer Höhe von 2 m mit einer Anfangsgeschwindigkeit von 20 m/s senkrecht nach oben geworfen. Wann trifft er wieder auf den Boden auf? (Vernachlässigen Sie den Luftwiderstand und gehen Sie von einer Fallbeschleunigung von $9,81 \text{ m/s}^2$ aus.)

Entsprechend den Kinematik-Gesetzen kann diese Problemstellung als Polynom 2. Grades ausgedrückt werden, wobei T die Zeit in Sekunden darstellt:

$$f(t) = (-9,81 \div 2)t^2 + 20t + 2$$

Tastenfolge:	Anzeige	Beschreibung
$\boxed{\text{XEQ}}$ Q	A?Wert	Startet Programm zum Lösen einer quadratischen Gleichung.
9,81 $\boxed{+/-}$ $\boxed{\text{ENTER}}$ 2 $\boxed{\div}$ $\boxed{\text{R/S}}$	B?Wert	Speichert $(-9,81/2)$ in A.
20 $\boxed{\text{R/S}}$	C?Wert	Speichert 20 in B.
2 $\boxed{\text{R/S}}$	X=4,1751	Speichert 2 in C und berechnet X bzw. t .
$\boxed{\text{R/S}}$	X=-0,0977	Berechnet die andere Wurzel.

Da eine negative Zeit in der vorliegenden Aufgabe keinen Sinn macht, ist das erste Ergebnis, 4,1751 s, das einzige sinnvolle Ergebnis.

Beispiel 4: Lösen Sie das nachstehende Polynom 2. Grades unter Verwendung des vorliegenden Programms. Ändern Sie anschließend den Sinn in der Abfrage in Zeile Q12 so, daß die zweite Wurzel zuerst berechnet wird; vergleichen Sie danach die Ergebnisse. Denken Sie daran, die ursprüngliche Programmzeile zurückzuspeichern, oder löschen Sie das Programm, wenn Sie das Beispiel abgeschlossen haben.

$$x^2 + (3 \times 10^6)x + 1 = 0$$

Tastenfolge:	Anzeige	Beschreibung
$\boxed{\text{XEQ}}$ Q	A?Wert	Startet Programm.
1 $\boxed{\text{R/S}}$	B?Wert	Speichert 1 in A.
3 $\boxed{\text{E}}$ 6 $\boxed{\text{R/S}}$	C?Wert	Speichert 3×10^6 in B.
1 $\boxed{\text{R/S}}$	X=-3.000.000,00	Speichert 1 in C und berechnet 1. Wurzel.

<input type="text" value="R/S"/>	$X=-3,3333E-7$	Berechnet die zweite Wurzel.
<input type="checkbox"/> <input type="text" value="PRGM"/>	Q36 GTO Q	Schaltet in Programm-eingabe um.
<input type="checkbox"/> <input type="text" value="GTO"/> <input type="checkbox"/> Q12	Q12 $x<0?$	Stellt Programmzeiger auf Zeile Q12.
<input type="checkbox"/> <input type="text" value="↵"/>	Q11 CF 0	Löscht Zeile Q12.
<input type="checkbox"/> <input type="text" value="TESTS"/> { $x?0$ } { >0 }	Q12 $x>0?$	Fügt die Abfrage $x>0?$ ein.
<input type="text" value="C"/>	$-3,3333E-7$	Hebt Programmein-gabe auf.
<input type="text" value="XEQ"/> Q	R?1,0000	Startet Programm.
<input type="text" value="R/S"/> <input type="text" value="R/S"/>		Überspringt Datenein-gabe, da die Werte bereits gespeichert sind.
<input type="text" value="R/S"/>	$X=0,0000$	Berechnet erste Wurzel unter Verwendung vorheriger Eingaben.
<input type="text" value="R/S"/>	DIVIDE BY 0	Versuch, zweite Wurzel zu berechnen.

Wie Sie sehen können, kann die Auswirkung einer einfachen Änderung in der Folge der Berechnungen ziemlich signifikant sein.

Wenn Sie die zuerst berechneten Werte in die Gleichung einsetzen, werden Sie feststellen, daß die linke Seite der Gleichung für die Wurzel des kleineren Absolutbetrags gleich Null ist (was theoretisch richtig ist), und 1 für die Wurzel des größeren Absolutbetrags. Bedeutet dies, daß das Ergebnis von $-3\ 000\ 000,0000$ falsch ist? Die Antwort hierauf ist ein bedingtes "Nein". Wenn Sie den Wert in der niederwertigsten Stelle um 1 erhöhen oder reduzieren und diesen Wert dann in der Gleichung einsetzen, ist die linke Seite gleich 31 oder -29 . Demnach ist $-3\ 000\ 000,0000$, obwohl nicht absolut richtig, das *bestmögliche 12-stellige Ergebnis*, welches berechnet werden konnte.

Koordinatentransformationen

Dieses Programm ermöglicht eine zweidimensionale Koordinatentranslation und -Rotation.

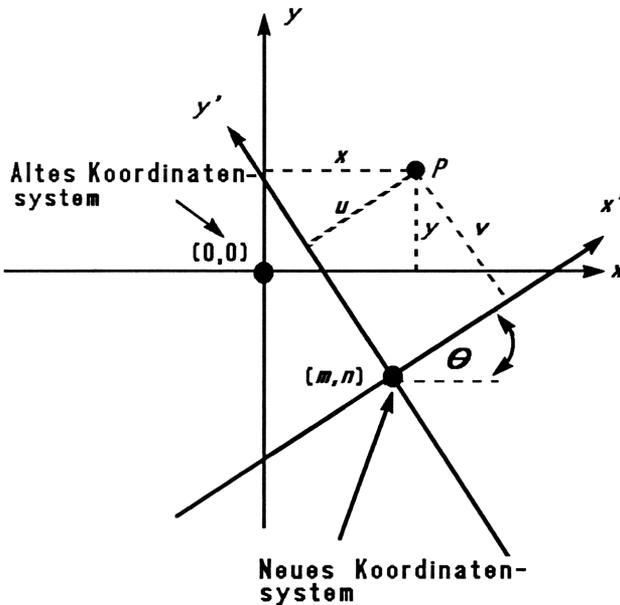
Die nachstehenden Gleichungen werden dazu verwendet, den Punkt P mit den kartesischen Koordinaten (x, y) im alten System in die Koordinaten (u, v) des neuen, verschobenen und rotierten Systems zu konvertieren.

$$\begin{aligned}u &= (x - m) \cos\theta + (y - n) \sin\theta \\v &= (y - n) \cos\theta - (x - m) \sin\theta\end{aligned}$$

Die umgekehrte Transformation wird durch die Gleichungen

$$\begin{aligned}x &= u \cos\theta - v \sin\theta + m \\y &= u \sin\theta + v \cos\theta + n\end{aligned}$$

erreicht. Die im HP-32S enthaltenen Funktionen machen diese Berechnungen relativ unkompliziert.



Zweidimensionale Rotation um die Achse

Programmliste:

Programmzeilen:

	Beschreibung
D01 LBL D	Diese Routine definiert das neue Koordinatensystem.
D02 INPUT M	Eingeben/Speichern von M , die x -Koordinate des neuen Koordinatenursprungs.
D03 INPUT N	Eingeben/Speichern von N , die y -Koordinate des neuen Koordinatenursprungs.
D04 INPUT T	Eingeben/Speichern von T , dem Winkel θ .
D05 GTO D	Durchläuft Schleife zum Anzeigen der Eingabewerte.

Bytes und Prüfsumme: 007,5; 1CD9

N01 LBL N	Diese Routine konvertiert die Koordinaten aus dem alten System in das neue System.
N02 INPUT X	Eingeben/Speichern von X , der alten x -Koordinate.
N03 INPUT Y	Eingeben/Speichern von Y , der alten y -Koordinate.
N04 RCL X	Schiebt Y nach oben, ruft X in X -Register.
N05 RCL N	Schiebt X, Y nach oben, ruft N in X -Register.
N06 RCL M	Schiebt N, X, Y nach oben, ruft M zurück.
N07 CMPLX-	Berechnet $(X-M)$ und $(Y-N)$.
N08 RCL T	Schiebt $(X-M)$ und $(Y-N)$ nach oben und ruft T zurück.
N09 +/-	Ändert das Vorzeichen von T , da $\sin(-T)$ gleich $-\sin(T)$.
N10 1	Speichert 1 für Radius zur Berechnung von $\cos(T)$ und $-\sin(T)$.
N11 $\theta, r \rightarrow y, x$	Berechnet $\cos(T)$ und $-\sin(T)$ in X - und Y -Register.
N12 CMPLX \times	Berechnet $(X-M) \cos(T) + (Y-N) \sin(T)$ und $(Y-N) \cos(T) - (X-M) \sin(T)$.
N13 STO U	Speichert x -Koordinate in Variable U .
N14 $x \leftrightarrow y$	Tauscht Positionen der Koordinaten aus.
N15 STO V	Speichert y -Koordinate in Variable V .
N16 $x \leftrightarrow y$	Speichert urspr. Koordinatenpositionen.
N17 VIEW U	Hält das Programm zur Anzeige von U an.
N18 VIEW V	Hält das Programm zur Anzeige von V an.
N19 GTO N	Rücksprung zur Ausführung einer neuen Berechnung.

Bytes und Prüfsumme: 028,5; 6078

A01	LBL A	Diese Routine konvertiert Koordinaten vom neuen System in das alte System.
A02	INPUT U	Eingeben und Speichern von U .
A03	INPUT V	Eingeben und Speichern von V .
A04	RCL U	Schiebt V nach oben und ruft U zurück.
A05	RCL T	Schiebt U, V nach oben, ruft T zurück.
A06	1	Setzt für die Berechnung von $\sin(T)$ und $\cos(T)$ den Radius auf 1.
A07	$\theta, r \rightarrow y, x$	Berechnet $\cos(T)$ und $\sin(T)$.
A08	CMPLX \times	Berechnet $U \cos(T) - V \sin(T)$ und $U \sin(T) + V \cos(T)$.
A09	RCL N	Schiebt vorherige Ergebnisse nach oben und ruft N zurück.
A10	RCL M	Schiebt Ergebnisse nach oben und ruft M zurück.
A11	CMPLX+	Schließt Berechnung durch Addition von M und N zu vorherigen Ergebnissen ab.
A12	STO X	Speichert x -Koordinate in variable X .
A13	$x \leftrightarrow y$	Tauscht Positionen der Koordinaten aus.
A14	STO Y	Speichert y -Koordinate in Variable Y .
A15	$x \leftrightarrow y$	Bringt Koordinaten zurück in ursprüngliche Positionen.
A16	VIEW X	Hält das Programm zur Anzeige von X an.
A17	VIEW Y	Hält das Programm zur Anzeige von Y an.
A18	GTO A	Rücksprung zum Programmanfang für weitere Berechnungen.

Bytes und Prüfsumme: 027,0; 9AE6

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 119 Bytes: 63 für Programm, 56 für Variablen.

Anwendung des Programms:

1. Tippen Sie die Routine ein; drücken Sie zum Abschluß \square .
2. Drücken Sie \square D zum Starten der Definition für die Koordinatentransformation.
3. Tippen Sie die x -Koordinate des neuen Koordinatenursprungs M ein und drücken Sie \square .
4. Tippen Sie die y -Koordinate des neuen Koordinatenursprungs N ein und drücken Sie \square .

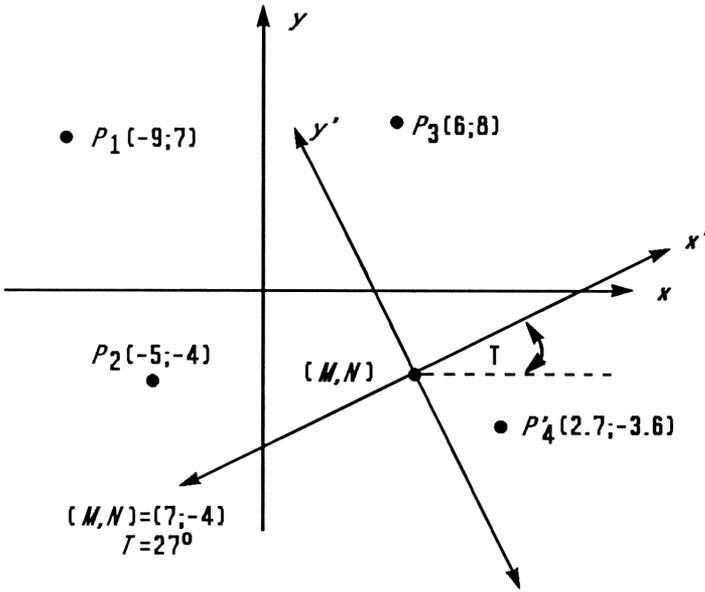
5. Tippen Sie den Rotationswinkel T ein und drücken Sie $\boxed{\text{R/S}}$.
6. Zur Ausführung der Transformation vom alten System in das neue System ist mit Schritt 7 fortzufahren. Um vom neuen System in das alte zu konvertieren, ist mit Schritt 12 fortzufahren.
7. Drücken Sie $\boxed{\text{XEQ}}$ N zum Starten der Alt-nach-Neu Transformationsroutine.
8. Tippen Sie X ein und drücken Sie $\boxed{\text{R/S}}$.
9. Tippen Sie Y ein und drücken Sie $\boxed{\text{R/S}}$ zur Berechnung der x -Koordinate U im neuen System.
10. Drücken Sie $\boxed{\text{R/S}}$, wonach Sie die y -Koordinate V im neuen System angezeigt erhalten.
11. Für eine erneute Alt-nach-Neu Transformation ist $\boxed{\text{R/S}}$ zu drücken und zu Schritt 8 zurückzugehen; Schritt 12 beginnt eine Neu-nach-Alt Transformation.
12. Drücken Sie $\boxed{\text{XEQ}}$ A zum Start einer Neu-nach-Alt Transformation.
13. Tippen Sie U ein (x -Koordinate im neuen System) und drücken Sie $\boxed{\text{R/S}}$.
14. Tippen Sie V ein (y -Koordinate im neuen System) und drücken Sie $\boxed{\text{R/S}}$ zur Anzeige von X .
15. Drücken Sie $\boxed{\text{R/S}}$ zur Anzeige von Y .
16. Für eine weitere Neu-nach-Alt Transformation ist $\boxed{\text{R/S}}$ zu drücken und mit Schritt 13 fortzufahren (Schritt 7 für eine Alt-nach-Neu Transformation).

Benutzte Variablen:

- M Die x -Koordinate des Ursprungs im neuen System.
 N Die y -Koordinate des Ursprungs im neuen System.
 T Der Rotationswinkel θ zwischen dem alten und neuen System.
 X Die x -Koordinate eines Punktes im alten System.
 Y Die y -Koordinate eines Punktes im alten System.
 U Die x -Koordinate eines Punktes im neuen System.
 V Die y -Koordinate eines Punktes im neuen System.

Anmerkungen: Bei einer reinen Translation ist Null für T einzugeben; bei einer reinen Rotation ist Null für M und N einzugeben.

Beispiel: Für das nachstehende Koordinatensystem sind die Punkte P_1 , P_2 und P_3 , welche aus dem (X,Y) System sind, in Punkte des (X',Y') Systems zu konvertieren. Rechnen Sie P'_4 , welcher aus dem (X',Y') System ist, in das (X,Y) System um.



Tastensequenz	Anzeige	Beschreibung
MODES {DG}		Stellt Grad-Modus ein, da T in Grad angegeben ist.
XEQ D	M?Wert	Startet Routine, welche Transformation definiert.
7 R/S	N?Wert	Speichert 7 in M .
4 +/- R/S	T?Wert	Speichert -4 in N .
27 R/S	M?7,0000	Speichert 27 in T .

$\boxed{\text{XEQ}}$ N	X?Wert	Startet Alt-nach-Neu Routine.
9 $\boxed{+/-}$ $\boxed{\text{R/S}}$	Y?Wert	Speichert -9 in X.
7 $\boxed{\text{R/S}}$	U=-9,2622	Speichert 7 in Y und berechnet U.
$\boxed{\text{R/S}}$	V=17,0649	Berechnet V.
$\boxed{\text{R/S}}$	X?-9,0000	Start von Alt-nach-Neu Routine für weitere Koordinaten.
5 $\boxed{+/-}$ $\boxed{\text{R/S}}$	Y?7,0000	Speichert -5 in X.
4 $\boxed{+/-}$ $\boxed{\text{R/S}}$	U=-10,6921	Speichert -4 in Y.
$\boxed{\text{R/S}}$	V=5,4479	Berechnet V.
$\boxed{\text{R/S}}$	X?-5,0000	Start von Alt-nach-Neu Routine für weitere Koordinaten.
6 $\boxed{\text{R/S}}$	Y?-4,0000	Speichert 6 in X.
8 $\boxed{\text{R/S}}$	U=4,5569	Speichert 8 in Y und berechnet U.
$\boxed{\text{R/S}}$	V=11,1461	Berechnet V.
$\boxed{\text{XEQ}}$ A	U?4,5569	Startet Neu-nach-Alt Routine.
2,7 $\boxed{\text{R/S}}$	V?11,1461	Speichert 2,7 in U.
3,6 $\boxed{+/-}$ $\boxed{\text{R/S}}$	X=11,0401	Speichert $-3,6$ in V und berechnet X.
$\boxed{\text{R/S}}$	Y=-5,9818	Berechnet Y.

13

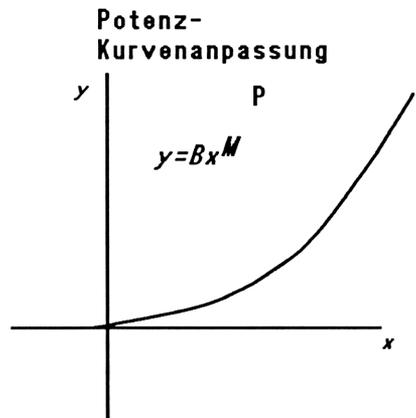
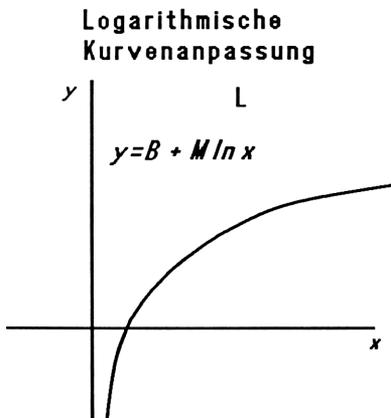
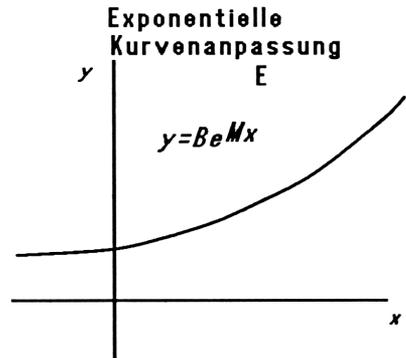
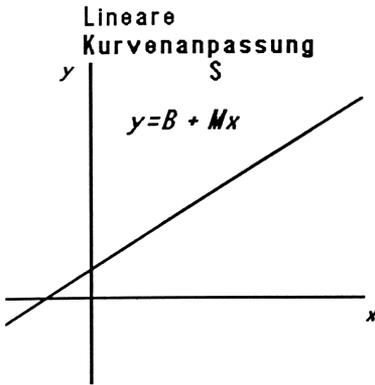
Statistikprogramme

Die Speicheranforderung und Prüfsumme für jedes Programm-Label kann über den Programmkatalog (Seite 85) überprüft werden.

Kurvenanpassung

Dieses Programm kann zur Anpassung Ihrer Statistikdaten an eines von vier Kurvenmodellen verwendet werden. Bei den vier Kurven handelt es sich um ein lineares, logarithmisches, exponentielles und Potenz-Kurvenmodell. Nach Eingabe von zwei oder mehr (x, y) Datenpaaren lassen sich der Korrelationskoeffizient r und die zwei Regressionskoeffizienten m und b berechnen. Das Programm enthält außerdem eine Routine, über welche die Berechnung der Näherungen \hat{x} und \hat{y} möglich ist. (Eine Definition dieser Werte finden Sie unter "Lineare Regression" in Kapitel 11.)

Beispiele der Kurven und die zugehörigen Gleichungen finden Sie nachstehend. Zur Berechnung der Regressionskoeffizienten werden die im HP-32S eingebauten Regressionsfunktionen verwendet.



Bei der Anpassung an die logarithmische Kurve sind nur positive Werte für x erlaubt; um eine Anpassung an eine exponentielle Kurve durchzuführen, müssen die Werte von y positiv sein; soll die Anpassung an eine Potenzkurve erfolgen, so müssen die Werte für x und y jeweils positiv sein. Werden diese Hinweise nicht beachtet und es wird ein negativer Wert eingegeben, dann wird die Fehlermeldung LOG<NEG> angezeigt.

Daten mit großen Beträgen aber relativ kleinen Differenzen können zu Problemen bei der Genauigkeit führen, ebenso Daten mit hohen betragsmäßigen Unterschieden. Beziehen Sie sich auf "Grenzen für die Genauigkeit von Daten" in Kapitel 11.

Programmliste:

Programmzeilen:

Beschreibung

S01 LBL S	Diese Routine führt die Voreinstellungen für das lineare Modell durch.
S02 1	Eingabe von Indexwert für spätere Speicherung in i (für indirekte Adressierung).
S03 CF 0	Löscht Flag 0, den Indikator für $\ln X$.
S04 CF 1	Löscht Flag 1, den Indikator für $\ln Y$.
S05 GTO Z	Verzweigt zum gemeinsamen Eingangspunkt Z.

Bytes und Prüfsumme: 007,5; 17CA

L01 LBL L	Diese Routine führt die Voreinstellungen für das logarithmische Modell aus.
L02 2	Eingabe von Indexwert für spätere Speicherung in i (für indirekte Adressierung).
L03 SF 0	Setzt Flag 0, den Indikator für $\ln X$.
L04 CF 1	Löscht Flag 1, den Indikator für $\ln Y$.
L05 GTO Z	Verzweigt zum gemeinsamen Eingangspunkt Z.

Bytes und Prüfsumme: 007,5; 6047

E01 LBL E	Diese Routine führt die Voreinstellungen für das exponentielle Modell aus.
E02 3	Eingabe von Indexwert für spätere Speicherung in i (für indirekte Adressierung).
E03 CF 0	Löscht Flag 0, den Indikator für $\ln X$.
E04 SF 1	Setzt Flag 1, den Indikator für $\ln Y$.
E05 GTO Z	Verzweigt zum gemeinsamen Eingangspunkt Z.

Bytes und Prüfsumme: 007,5; C0F1

P01 LBL P	Diese Routine führt die Voreinstellungen für das Potenz-Modell aus.
P02 4	Eingabe von Indexwert für spätere Speicherung in i (für indirekte Adressierung).
P03 SF 0	Setzt Flag 0, den Indikator für $\ln X$.
P04 SF 1	Setzt Flag 1, den Indikator für $\ln Y$.

Bytes und Prüfsumme: 006,0; A26B

Z01 LBL Z Definiert gemeinsamen Eingangspunkt für alle Modelle.
 Z02 CLΣ Löscht die Statistikregister.
 Z03 STO i Speichert Indexwert in i für indirekte Adressierung.
 Z04 0 Setzt für die erste Eingabe Schleifenzähler auf Null.

Bytes und Prüfsumme: 006,0; CC1B

W01 LBL W Definiert den Anfang der Eingabeschleife.
 W02 1 Paßt zwecks Eingabeaufforderung den Schleifenzähler um 1 an.
 W03 +
 W04 STO X Speichert Schleifenzähler in X , wodurch er während der Eingabeaufforderung für X angezeigt wird.
 W05 INPUT X Zeigt Zähler an und speichert Eingabe für X .
 W06 FS? 0 Wenn Flag 0 gesetzt ist...
 W07 LN ...berechne den natürlichen Logarithmus der Eingabe.
 W08 STO B Speichert diesen Wert für Korrekturroutine.
 W09 INPUT Y Eingeben und Speichern von Y .
 W10 FS? 1 Wenn Flag 1 gesetzt ist...
 W11 LN ...berechne natürlichen Logarithmus von Eingabewert.

W12 STO R
 W13 RCL B
 W14 Σ+ Akkumuliert B und R als x,y -Datenpaar in den Statistikregistern.
 W15 GTO W Zurück zum Schleifenanfang für weiteres X, Y Paar.

Bytes und Prüfsumme: 022,5; 1A43

U01 LBL U Definiert den Beginn der Korrekturroutine "Undo".
 U02 RCL R Ruft das letzte Datenpaar zurück.
 U03 RCL B
 U04 Σ- Löscht dieses Paar aus den Statistikregistern.
 U05 GTO W Zurück zum Schleifenanfang für weiteres X, Y Paar.

Bytes und Prüfsumme: 007,5; 5D02

R01 LBL R	Definiert den Beginn der Ausgaberoutine.
R02 r	Berechnet den Korrelationskoeffizienten.
R03 STO R	Speichert ihn in R .
R04 VIEW R	Zeigt den Korrelationskoeffizienten an.
R05 b	Berechnet Koeffizient b .
R06 FS? 1	Wenn Flag 1 gesetzt ist, Berechnung der natürlichen Exponentialfunktion von b .
R07 e ^x	
R08 STO B	Speichert b in B .
R09 VIEW B	Zeigt Wert an.
R10 m	Berechnet Koeffizient m .
R11 STO M	Speichert m in M .
R12 VIEW M	Zeigt Wert an.
Bytes und Prüfsumme: 018,0; 7492	
Y01 LBL Y	Definiert den Beginn der Schleife für die Näherung (Projektion).
Y02 INPUT X	
Y03 XEQ(i)	Ruft Unterprogramm zur Berechnung von \hat{y} auf.
Y04 STO Y	Speichert \hat{y} -Wert in Y .
Y05 INPUT Y	Anzeigen und Speichern (falls Eingabe/Änderung) des y -Wertes in Y .
Y06 6	
Y07 STO+ i	Paßt Indexwert zur Adressierung des entsprechenden Unterprogramms an.
Y08 XEQ(i)	Ruft Unterprogramm zur Berechnung von \hat{x} auf.
Y09 STO X	Speichert für nächste Schleife \hat{x} in X .
Y10 GTO Y	Rückkehr zum Schleifenanfang für weitere Näherung.
Bytes und Prüfsumme: 015,0; 9AEA	
A01 LBL A	Dieses Unterprogramm berechnet \hat{y} für das lineare Modell.
A02 RCL M	
A03 RCL× X	
A04 RCL+ B	Berechnet $\hat{y} = MX + B$.
A05 RTN	Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 007,5; 0E85	

G01 LBL G Dieses Unterprogramm berechnet \hat{x} für das lineare Modell.

G02 STO- i Speichert Originalwert für Index zurück.

G03 RCL Y

G04 RCL- B

G05 RCL÷ M Berechnet $\hat{x} = (Y - B) \div M$.

G06 RTN Rückkehr zur aufrufenden Routine.

Bytes und Prüfsumme: 009,0; FDF1

B01 LBL B Dieses Unterprogramm berechnet \hat{y} für das logarithmische Modell.

B02 RCL X

B03 LN

B04 RCL× M

B05 RCL+ B Berechnet $\hat{y} = M \ln X + B$.

B06 RTN Rückkehr zur aufrufenden Routine.

Bytes und Prüfsumme: 009,0; 1B06

H01 LBL H Dieses Unterprogramm berechnet \hat{x} für das logarithmische Modell.

H02 STO- i Speichert Originalwert für Index zurück.

H03 RCL Y

H04 RCL- B

H05 RCL÷ M

H06 e^x Berechnet $\hat{x} = e^{(Y - B) \div M}$.

H07 RTN Rückkehr zur aufrufenden Routine.

Bytes und Prüfsumme: 010,5; C783

C01 LBL C Dieses Unterprogramm berechnet \hat{y} für das exponentielle Modell.

C02 RCL M

C03 RCL× X

C04 e^x

C05 RCL× B Berechnet $\hat{y} = Be^{MX}$.

C06 RTN Rückkehr zur aufrufenden Routine.

Bytes und Prüfsumme: 009,0; B411

I01 LBL I Dieses Unterprogramm berechnet \hat{x} für das
exponentielle Modell.
I02 STO- i Speichert Originalwert für Index zurück.
I03 RCL Y
I04 RCL ÷ B
I05 LN
I06 RCL ÷ M Berechnet $\hat{x} = (\ln(Y \div B)) \div M$.
I07 RTN Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 010,5; 01D6

D01 LBL D Dieses Unterprogramm berechnet \hat{y} für das
Potenz-Modell.
D02 RCL X
D03 RCL M
D04 \succ^x
D05 RCL × B Berechnet $Y = B(X^M)$.
D06 RTN Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 009,0; B4D4

J01 LBL J Dieses Unterprogramm berechnet \hat{x} für das
Potenz-Modell.
J02 STO- i Speichert Originalwert für Index zurück.
J03 RCL Y
J04 RCL ÷ B
J05 RCL M
J06 $1/x$
J07 \succ^x Berechnet $\hat{x} = (Y/B)^{1/M}$.
J08 RTN Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 012,0; FAA4

Benutzte Flags: Flag 0 ist gesetzt, wenn der natürliche Logarithmus von der Eingabe für X berechnet wird; Flag 1 ist gesetzt, wenn der natürliche Logarithmus von der Eingabe für Y berechnet wird.

Erforderlicher Speicherplatz: 270 Bytes: 174 für Programm, 96 für Daten (Statistikregister 48).

Anwendung des Programms:

1. Tippen Sie die einzelnen Routinen ein; drücken Sie zum Abschluß \boxed{C} .
2. Drücken Sie \boxed{XEQ} und wählen Sie das Kurvenmodell durch Drücken von:
 - S für eine lineare Kurve
 - L für eine logarithmische Kurve
 - E für eine exponentielle Kurve
 - P für eine Potenzkurve
3. Tippen Sie den x -Wert ein und drücken Sie $\boxed{R/S}$.
4. Tippen Sie den y -Wert ein und drücken Sie $\boxed{R/S}$.
5. Wiederholen Sie Schritt 3 und 4 für jedes Datenpaar. Wenn Sie einen Tippfehler feststellen, nachdem Sie $\boxed{R/S}$ in Schritt 3 gedrückt haben (wobei $Y?$ Wert noch angezeigt wird), dann drücken Sie erneut $\boxed{R/S}$ (was zur Eingabeaufforderung $X?$ Wert führt) und anschließend \boxed{XEQ} U (Undo), um das letzte Datenpaar aus den Statistikregistern zu löschen. Bemerken Sie den Fehler erst nach Schritt 4, so drücken Sie \boxed{XEQ} U. In beiden Fällen ist mit Schritt 3 fortzufahren.
6. Drücken Sie am Ende der Dateneingabe \boxed{XEQ} R, um den Korrelationskoeffizienten R anzuzeigen.
7. Drücken Sie $\boxed{R/S}$ zur Anzeige des Regressionskoeffizienten B .
8. Drücken Sie $\boxed{R/S}$ zur Anzeige des Regressionskoeffizienten M .
9. Drücken Sie $\boxed{R/S}$ zur Anzeige der Eingabeaufforderung $X?$ Wert für die \hat{x} -, \hat{y} -Näherungsroutine.
10. Wenn Sie \hat{y} unter Vorgabe von x berechnen möchten, ist x bei der Aufforderung $X?$ Wert einzutippen und anschließend $\boxed{R/S}$ zur Anzeige von \hat{y} ($Y?$) zu drücken.
11. Wenn Sie \hat{x} unter Vorgabe von y berechnen möchten, ist $\boxed{R/S}$ bis zur Eingabeaufforderung $Y?$ Wert zu drücken; danach ist y einzugeben und $\boxed{R/S}$ zur Anzeige von \hat{x} ($X?$) zu drücken.
12. Wenn Sie weitere Näherungswerte berechnen möchten, ist mit Schritt 10 und 11 fortzufahren.
13. Gehen Sie zu Schritt 2 zurück, wenn Sie eine neue Aufgabenstellung berechnen möchten.

Benutzte Variablen:

- B** Regressionskoeffizient (y -Achsenabschnitt einer Geraden); auch als Hilfsvariable benutzt.
- M** Regressionskoeffizient (Steigung einer Geraden).
- R** Korrelationskoeffizient; auch als Hilfsvariable benutzt.
- X** Der x -Wert eines Datenpaares während der Eingabe; hypothetisches x bei der Projektion von \hat{y} ; oder \hat{x} (x -Näherung) bei einem hypothetischen y .
- Y** Der y -Wert eines Datenpaares während der Eingabe; hypothetisches y bei der Projektion von \hat{x} ; oder \hat{y} (y -Näherung) bei einem hypothetischen x .
- i** Indexvariable zur indirekten Adressierung der entsprechenden \hat{x} -, \hat{y} -Projektionsgleichung.

Statistikregister Statistische Summationen und Berechnungen.

Beispiel 1: Führen Sie für die nachstehenden Daten eine Kurvenanpassung über ein lineares Kurvenmodell durch. Machen Sie absichtlich einen Fehler beim Eingeben des dritten Datenpaares und führen Sie mit der "Undo"-Routine eine Korrektur durch. Weiterhin ist eine Näherung für y bei $x = 37$ zu bestimmen; schätzen Sie x für $y = 101$.

X	40,5	38,6	37,9	36,2	35,1	34,6
Y	104,5	102	100	97,5	95,5	94

Tastenfolge:	Anzeige	Beschreibung
<input type="button" value="XEQ"/> S	X?1,00000	Startet Routine für lineares Kurvenmodell.
40,5 <input type="button" value="R/S"/>	Y?Wert	Gibt x -Wert des Datenpaares ein.
104,5 <input type="button" value="R/S"/>	X?2,00000	Gibt y -Wert des Datenpaares ein.

38,6 Y?104,5000 Gibt x-Wert des Datenpaares ein.

102 X?3,0000 Gibt y-Wert des Datenpaares ein.

Geben Sie nun absichtlich 379 anstatt 37,9 ein, um die Anwendung der Korrekturroutine (*Undo*) kennenzulernen.

379 Y?102,0000 Gibt falschen x-Wert des Datenpaares ein.

X?4,0000 Bringt Eingabeaufforderung für X? zurück.

U X?3,0000 Löscht das letzte Datenpaar.

Fahren Sie nun mit der richtigen Dateneingabe fort.

37,9 Y?102,0000 Gibt korrekten x-Wert des Datenpaares ein.

100 X?4,0000 Gibt y-Wert des Datenpaares ein.

36,2 Y?100,0000 Gibt x-Wert des Datenpaares ein.

97,5 X?5,0000 Gibt y-Wert des Datenpaares ein.

35,1 Y?97,5000 Gibt x-Wert des Datenpaares ein.

95,5 X?6,0000 Gibt y-Wert des Datenpaares ein.

34,6 Y?95,5000 Gibt x-Wert des Datenpaares ein.

94 X?7,0000 Gibt y-Wert des Datenpaares ein.

R R=0,9955 Berechnet den Korrelationskoeffizienten.

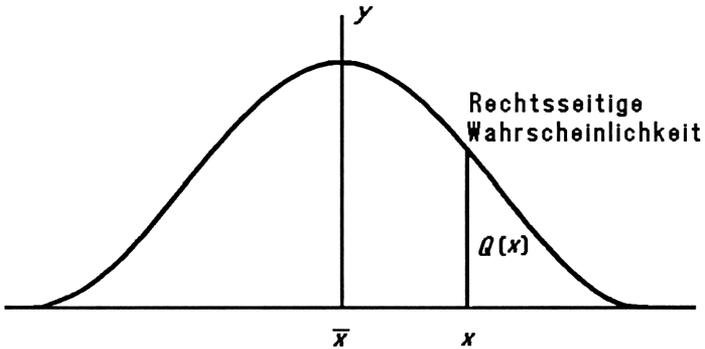
<input type="text" value="R/S"/>	B=33,5271	Berechnet Regressionskoeffizient B .
<input type="text" value="R/S"/>	M=1,7601	Berechnet Regressionskoeffizient M .
<input type="text" value="R/S"/>	X?7,0000	Eingabeaufforderung für hypothetischen x -Wert.
37 <input type="text" value="R/S"/>	Y?98,6526	Speichert 37 in X und berechnet \hat{y} .
101 <input type="text" value="R/S"/>	X?38,3336	Speichert 101 in Y und berechnet \hat{x} .

Beispiel 2: Wiederholen Sie Beispiel 1 (Verwendung der gleichen Daten) für eine Anpassung über ein logarithmisches, exponentielles und Potenz-Kurvenmodell. Die nachstehende Tabelle enthält die Labels und Ergebnisse (Korrelations- und Regressionskoeffizienten sowie x - und y -Näherungen) für jedes Kurvenmodell. Sie müssen die Datenpaare jeweils neu eingeben, wenn Sie das Programm für ein anderes Kurvenmodell starten.

	Logarithmisch	Exponentiell	Potenz
Programmstart durch:	<input type="text" value="XEQ"/> L	<input type="text" value="XEQ"/> E	<input type="text" value="XEQ"/> P
R	0,9965	0,9945	0,9959
B	-139,0088	51,1312	8,9730
M	65,8446	0,0177	0,6640
Y (\hat{y} für $X=37$)	98,7508	98,5870	98,6845
X (\hat{x} für $Y=101$)	38,2857	38,3628	38,3151

Normalverteilung und Verteilungsfunktion

Die Normalverteilung wird häufig als Modell für die Verteilung einer Zufallsvariablen um einen Mittelwert verwendet. Das Modell geht davon aus, daß die Stichprobenverteilung symmetrisch zum Mittelwert M ist und die Form einer Glockenkurve besitzt. Das Programm berechnet für ein gegebenes x die Wahrscheinlichkeit, daß ein zufällig gewählter Wert aus der Stichprobe einen höheren Wert besitzt. Dies wird auch als *rechtsseitige Wahrscheinlichkeit*, $Q(x)$, bezeichnet. Das Programm erlaubt auch die Umkehrung: für einen vorgegebenen Wert $Q(x)$ kann der korrespondierende x -Wert berechnet werden.



$$Q(x) = 0,5 - \frac{1}{\sigma \sqrt{2\pi}} \int_{\bar{x}}^x e^{-((x - \bar{x}) \div \sigma)^2 \div 2} dx.$$

Das Programm verwendet die im HP-32S eingebaute Integrationsroutine zur Berechnung des in der Verteilungsfunktion enthaltenen Integrals. Die Inverse hierzu ergibt sich durch die Anwendung des Newtonschen Iterationsverfahrens bei der Suche eines x -Wertes, der zu der gegebenen Wahrscheinlichkeit $Q(x)$ führt.

Programmliste:

Programmzeilen:

S01 LBL S

S02 Ø

S03 STO M

S04 INPUT M

Beschreibung

Diese Routine initialisiert das Programm für die Standardabweichung.

Speichert Vorgabewert für Mittelwert.

Eingeben und Speichern des Mittelwerts M .

S05 1 Speichert Vorgabewert für Standardabweichung.

S06 STO S

S07 INPUT S Eingeben und Speichern von S.

S08 RTN Beendet Anzeige von Standardabweichung.
Bytes und Prüfsumme: 012,0; 1F60

D01 LBL D Diese Routine berechnet $Q(X)$ bei gegebenem X.

D02 INPUT X Eingeben und Speichern von X.

D03 XEQ Q Berechnet rechtsseitige Wahrscheinlichkeit.

D04 STO Q Speichert Wert in Q, wodurch Anzeige durch VIEW Funktion möglich ist.

D05 VIEW Q Zeigt $Q(X)$ an.

D06 GTO D Rückkehr zum Schleifenanfang für weitere Berechnung von $Q(X)$.
Bytes und Prüfsumme: 009,0; 002C

I01 LBL I Berechnung von X bei gegebenem $Q(X)$.

I02 INPUT Q Eingeben und Speichern von $Q(X)$.

I03 RCL M Ruft den Mittelwert zurück.

I04 STO X Speichert den Mittelwert als Näherung/Schätzwert für X, als $X_{\text{Schätz}}$ bezeichnet.
Bytes und Prüfsumme: 006,0; ED6E

T01 LBL T Beginn der Iterationsschleife.

T02 XEQ Q Berechnet $(Q(X_{\text{Schätz}}) - Q(X))$.

T03 RCL- Q

T04 RCL X

T05 STO D

T06 R+

T07 XEQ F Berechnet die Ableitung bei $X_{\text{Schätz}}$.

T08 RCL÷ T

T09 ÷ Berechnet Korrekturwert für $X_{\text{Schätz}}$.

T10 STO+ X Addiert Korrekturwert, was zu neuem $X_{\text{Schätz}}$ führt.

T11 ABS

T12 0,0001

T13 <>? Test, ob der Korrekturwert signifikant ist.

T14 GTO T Rücksprung zur Korrekturschleife, falls signifikant, ansonsten nächste Anweisung.

T15 RCL X

T16 VIEW X	Zeigt berechneten Wert von X an.
T17 GTO I	Rückkehr zum Schleifenanfang zur Berechnung von weiterem X.
Bytes und Prüfsumme: 033,5; 4355	
Q01 LBL Q	Dieses Unterprogramm berechnet die rechtsseitige Wahrscheinlichkeit $Q(x)$.
Q02 RCL M	Ruft die Integrationsuntergrenze zurück.
Q03 RCL X	Ruft die Integrationsobergrenze zurück.
Q04 FN= F	Wählt die Funktion, welche durch LBL F für Integration definiert ist.
Q05 JFN d D	Integration über Hilfsvariable D.
Q06 2	
Q07 π	
Q08 ×	
Q09 SQRT	
Q10 RCL S	
Q11 ×	Berechnet $S \times \sqrt{2\pi}$.
Q12 STO T	Speichert Ergebnis für Umkehr-Routine.
Q13 ÷	
Q14 +/-	
Q15 0,5	
Q16 +	Addiert die Hälfte der Fläche unter der Kurve, da Mittelwert als Integrationsuntergrenze verwendet wurde.
Q17 RTN	Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 033,5; 4B20	
F01 LBL F	Dieses Unterprogramm berechnet den Integranden für die Funktion $e^{-((X-M) \div S)^2 \div 2}$.
F02 RCL D	
F03 RCL- M	
F04 RCL ÷ S	
F05 x ²	
F06 2	
F07 ÷	
F08 +/-	
F09 e ^x	
F10 RTN	Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 015,0; 034D	

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 157 Bytes: 109 für Programm, 48 für Variablen.

Anmerkungen: Die Genauigkeit dieses Programms hängt vom Anzeigeformat ab. Bei Eingaben im Bereich zwischen ± 3 Standardabweichungen ist eine Anzeige von vier oder mehr signifikanten Stellen für die meisten Anwendungen ausreichend. Bei voller Genauigkeit liegen die Eingabegrenzen zwischen ± 5 Standardabweichungen. Die Rechenzeit nimmt bei weniger angezeigten Stellen spürbar ab.

In Routine N kann "0,5" durch 2 und $\boxed{1/x}$ ersetzt werden; dies führt zur Einsparung von 6,5 Bytes auf Kosten der Klarheit.

Sie müssen die Umkehr-Routine (in den Routinen I und T) nicht eintippen, wenn Sie an dieser nicht interessiert sind.

Anwendung des Programms:

1. Tippen Sie die Routinen ein; drücken Sie zum Abschluß \boxed{C} .
2. Drücken Sie \boxed{XEQ} S.
3. Nach der Eingabeaufforderung für M ist der Mittelwert der Grundgesamtheit einzutippen und $\boxed{R/S}$ zu drücken. (Ist der Mittelwert gleich Null, drücken Sie einfach $\boxed{R/S}$.)
4. Nach der Eingabeaufforderung für S ist die Standardabweichung der Grundgesamtheit einzutippen und $\boxed{R/S}$ zu drücken. (Ist die Standardabweichung gleich 1, drücken Sie einfach $\boxed{R/S}$.)
5. Um X bei gegebenem $Q(X)$ zu berechnen, ist zu Schritt 9 überzugehen.
6. Drücken Sie \boxed{XEQ} D zur Berechnung von $Q(X)$ bei gegebenem X.
7. Nach der Eingabeaufforderung ist der Wert von X einzutippen und $\boxed{R/S}$ zu drücken. Es wird das Ergebnis, $Q(X)$, angezeigt.
8. Um $Q(X)$ für ein neues X mit gleichem Mittelwert und gleicher Standardabweichung zu berechnen, ist $\boxed{R/S}$ zu drücken und mit Schritt 7 fortzufahren.
9. Drücken Sie \boxed{XEQ} I zur Berechnung von X bei gegebenem $Q(X)$.
10. Nach der Eingabeaufforderung ist der Wert von $Q(X)$ einzutippen und $\boxed{R/S}$ zu drücken. Es wird das Ergebnis, X, angezeigt.
11. Um X für ein neues $Q(X)$ mit gleichem Mittelwert und gleicher Standardabweichung zu berechnen, ist $\boxed{R/S}$ zu drücken und mit Schritt 10 fortzufahren.

Benutzte Variablen:

- D* Hilfsvariable für Integration.
M Mittelwert der Grundgesamtheit, Vorgabewert ist Null.
Q Rechtsseitige Wahrscheinlichkeit.
S Standardabweichung der Grundgesamtheit, Vorgabewert ist 1.
T Temporär verwendete Variable zur Weitergabe von $S\sqrt{2\pi}$ an die Umkehr-Routine.
X Eingabewert für Zufallsvariable.

Beispiel 1: Ein Freund erzählt Ihnen, Ihre unbekannte Rendezvous-Partnerin hätte einen IQ von "3 σ ". Unter der Annahme, in der lokalen Umgebung gäbe es 10 000 mögliche unbekannte Rendezvous-Partnerinnen: Wieviel Personen fallen in die "3 σ " Bandbreite? Da dieses Problem in Bezug auf Standardabweichungen ausgedrückt ist, ist der Vorgabewert 0 für *M* und 1 für *S* zu verwenden.

Tastenfolge:	Anzeige	Beschreibung
<input type="text" value="XEQ"/> S	M?0,0000	Beginnt die Initialisierungsroutine.
<input type="text" value="R/S"/>	S?1,0000	Akzeptiert den Vorgabewert 0 für <i>M</i> .
<input type="text" value="R/S"/>	1,0000	Akzeptiert den Vorgabewert 1 für <i>S</i> .
<input type="text" value="XEQ"/> D	X?Wert	Beginnt das Programm für die Verteilungsfunktion und fordert zur Eingabe für <i>X</i> auf.
3 <input type="text" value="R/S"/>	Q=0,0014	Eingabe von 3 für <i>X</i> und Berechnung von $Q(X)$. Zeigt den Anteil der Bevölkerung mit einem IQ, der über dem "3 σ " Bereich liegt.

10000 <input type="checkbox"/> x	13,5049	Multiplikation mit der möglichen Anzahl von Rendezvous-Partnerinnen. Zeigt die ungefähre Anzahl von Personen der lokalen Bevölkerung, welche dem Kriterium genügen.
----------------------------------	---------	---

Da Sie wissen, daß Ihr Freund von Zeit zu Zeit zu Übertreibungen neigt, möchten Sie die Anzahl von Personen berechnen, welche dem Kriterium " 2σ " genügen. Beachten Sie, daß sich das Programm einfach durch Drücken von R/S wieder starten läßt.

Tastenfolge:	Anzeige	Beschreibung
<input type="checkbox"/> R/S	X?3,0000	Startet Programm erneut.
2 <input type="checkbox"/> R/S	Q=0,0227	Eingabe von X-Wert von 2 und Berechnung von $Q(X)$.
10000 <input type="checkbox"/> x	227,4937	Multiplikation mit der Anzahl aller möglichen Personen. Zeigt die ungefähre Anzahl der Personen an, welche dem revidierten Kriterium genügen.

Beispiel 2: Der Mittelwert einer Reihe von Testergebnissen liegt bei 55 Punkten; die Standardabweichung beträgt 15,3. Unter der Annahme, daß die Glockenkurve der Normalverteilung ein adäquates Modell der Verteilung darstellt: Wie hoch wäre die Wahrscheinlichkeit, daß ein zufällig ausgewählter Student 90 Punkte erzielt? Welches Ergebnis würde nur von 10% aller Studenten überschritten werden? Wie hoch wäre das Ergebnis, welches nur von 20% nicht erreicht werden könnte?

Tastensequenz:	Anzeige	Beschreibung
<input type="button" value="XEQ"/> S	M?0,0000	Startet die Initialisierungsroutine.
55 <input type="button" value="R/S"/>	S?1,0000	Speichert 55 für Mittelwert.
15,3 <input type="button" value="R/S"/>	15,3000	Speichert 15,3 für Standardabweichung.
<input type="button" value="XEQ"/> D	X?Wert	Startet das Programm für die Normalverteilung und fordert zur Eingabe für X auf.
90 <input type="button" value="R/S"/>	Q=0,0111	Eingabe von 90 für X und Berechnung von $Q(X)$.

Daraus kann geschlossen werden, daß nur etwa 1% aller Studenten eine höhere Punktezahl als 90 erreichen würden.

Tastensequenz:	Anzeige	Beschreibung
<input type="button" value="XEQ"/> I	Q?0,0111	Startet die Umkehr-Routine.
,1 <input type="button" value="R/S"/>	X=74,6078	Speichert 0,1 (10%) in $Q(X)$ und berechnet X.
<input type="button" value="R/S"/>	Q?0,1000	Startet erneut die Umkehr-Routine.
,8 <input type="button" value="R/S"/>	X=42,1232	Speichert 0,8 (100% – 20%) in $Q(X)$ und berechnet X.

Sonstige Programme

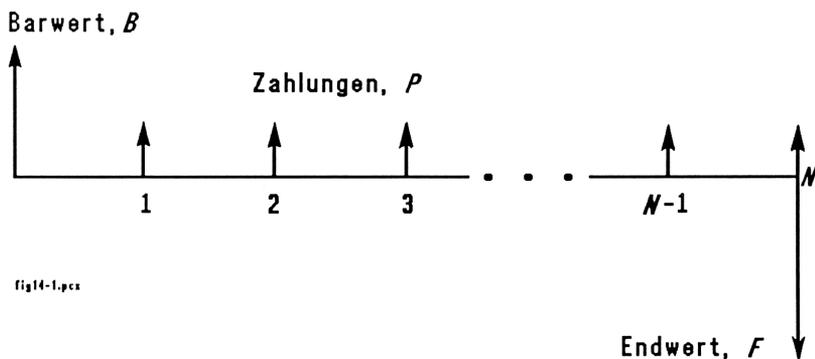
Die Speicherbelegung sowie die Prüfsumme können für jedes Programm-Label über den Programmkatalog (Seite 85) überprüft werden.

Annuitätenrechnung (TVM)

Sind die Werte von vier der fünf Variablen in der TVM-Gleichung (*Time Value of Money*) bekannt, so können Sie über dieses Programm die fünfte Variable berechnen. Das Programm dient zur Berechnung verschiedenartiger Problemstellungen im finanzmathematischen Bereich, wie z.B. Hypothekendarlehen, Sparplänen, Anschaffungskrediten, und so weiter.

Die Gleichung zur Lösung obiger Aufgabenstellungen ist:

$$P \left[\frac{1 - (1 + Z)^{-N}}{Z} \right] + F(1 + Z)^{-N} + B = 0$$



Cashflow Diagramm

Die Vorzeichen der Variablen (Barwert/*Balance B*; Zahlung/*Payment P*; Endwert/*Future value F*) entsprechen der Richtung der Cashflows. Zufließende Beträge haben ein positives Vorzeichen, abfließende Beträge haben ein negatives Vorzeichen. Beachten Sie, daß jedes Problem vom Standpunkt des Kreditgebers oder des Kreditnehmers betrachtet werden kann, wobei es jeweils mit dem umgekehrten Vorzeichen gesehen wird.

Programmliste:

Programmzeilen:	Beschreibung
N01 LBL N	Diese Routine berechnet die Anzahl von Zahlungen, <i>N</i> .
N02 14	Eingabe der zu <i>N</i> korrespondierenden Zahl für indirekte Adressierung.
N03 GTO L	Verzweigt zur gemeinsamen Steuerroutine L.
Bytes und Prüfsumme: 004,5; 61E5	
I01 LBL I	Diese Routine berechnet den Zinssatz <i>I</i> .
I02 9	Eingabe der zu <i>I</i> korrespondierenden Zahl für indirekte Adressierung.
I03 GTO L	Verzweigt zur gemeinsamen Steuerroutine L.
Bytes und Prüfsumme: 004,5; DA04	
B01 LBL B	Diese Routine berechnet den Barwert <i>B</i> .
B02 2	Eingabe der zu <i>B</i> korrespondierenden Zahl für indirekte Adressierung.
B03 GTO L	Verzweigt zur gemeinsamen Steuerroutine L.
Bytes und Prüfsumme: 004,5; 98EB	
P01 LBL P	Diese Routine berechnet die periodische Zahlung <i>P</i> .
P02 16	Eingabe der zu <i>P</i> korrespondierenden Zahl für indirekte Adressierung.
P03 GTO L	Verzweigt zur gemeinsamen Steuerroutine L.
Bytes und Prüfsumme: 004,5; A556	
F01 LBL F	Diese Routine berechnet den Endwert <i>F</i> .
F02 6	Eingabe der zu <i>F</i> korrespondierenden Zahl für indirekte Adressierung.
Bytes und Prüfsumme: 003,0; 6779	

L01	LBL L	Dieses Label steuert die Berechnung der gewählten Variablen.
L02	STO i	Speichert Indexwert (für indirekte Adressierung) in <i>i</i> .
L03	FN= T	Wählt für SOLVE die Routine T, welche die TVM-Gleichung enthält.
L04	SOLVE(i)	Löst die Variable, welche indirekt über <i>i</i> adressiert wurde.
L05	VIEW(i)	Zeigt das durch <i>i</i> adressierte Ergebnis an.
L06	GTO(i)	Rücksprung für weitere Berechnung.
Bytes und Prüfsumme: 009,0; 7878		
T01	LBL T	Diese Routine enthält TVM-Gleichung, welche die Beziehung zwischen den Variablen <i>Anzahl Verzinsungsperioden (N)</i> , <i>Zahlung (P)</i> , <i>Barwert (B)</i> , <i>Endwert (F)</i> und <i>Zinssatz I</i> definiert.
T02	INPUT N	Eingeben und Speichern von <i>N</i> .
T03	INPUT I	Eingeben und Speichern von <i>I</i> .
T04	INPUT B	Eingeben und Speichern von <i>B</i> .
T05	INPUT P	Eingeben und Speichern von <i>P</i> .
T06	INPUT F	Eingeben und Speichern von <i>F</i> .
T07	RCL I	Ruft den Zinssatz (in %) zurück.
T08	$x=0?$	Wenn $I=0 \dots$
T09	GTO K	\dots dann wird Gleichung in Routine K verwendet.
T10	100	
T11	\div	
T12	STO Z	Konvertiert <i>I</i> in Dezimalform und speichert es in <i>Z</i> .
T13	1	
T14	+	Berechnet $(1 + Z)$.
T15	RCL N	
T16	+/-	
T17	\times	Berechnet $(1 + Z)^{-N}$.
T18	ENTER	Dupliziert Zwischenergebnis für spätere Wiederverwendung.
T19	+/-	
T20	1	
T21	+	Berechnet $1 - (1 + Z)^{-N}$.
T22	RCL \div Z	Berechnet $(1 - (1 + Z)^{-N}) \div Z$.

T23	RCL× P	Berechnet $P \times (1 - (1 + Z)^{-N}) \div Z$.
T24	↔	Tauscht Kopie von $(1 + Z)^{-N}$ (aus Zeile T18) mit Inhalt von X-Register aus.
T25	RCL× F	Berechnet $F \times (1 + Z)^{-N}$.
T26	+	Berechnet $F \times (1 + Z)^{-N} + P \times (1 - (1 + Z)^{-N}) \div Z$.
T27	RCL+ B	Berechnet $F \times (1 + Z)^{-N} + P \times (1 - (1 + Z)^{-N}) \div Z + B$.
T28	RTN	Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 050,0; 429C		
K01	LBL K	Diese Routine wird aufgerufen, wenn $I = 0$.
K02	RCL P	
K03	RCL× N	
K04	RCL+ F	
K05	RCL+ B	Berechnet $P \times N + F + B$.
K06	RTN	Rückkehr zur aufrufenden Routine.
Bytes und Prüfsumme: 009,0; F2E0		

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 145 Bytes: 89 für Programm, 56 für Variablen.

Anmerkungen: Da alle Berechnungen des Programms in den Routinen T und K stattfinden, ist eine Verkürzung des Programms durch die Elimination der anderen Anwenderschnittstellen-Routinen möglich. Um das Programm in verkürzter Form ablaufen zu lassen, ist die durch LBL T definierte Funktion zu wählen (SOLVE/↓) {FN=} T) und danach die gesuchte Variable zu lösen (SOLVE/↓) {SOLVE} Variable).

Anwendung des Programms:

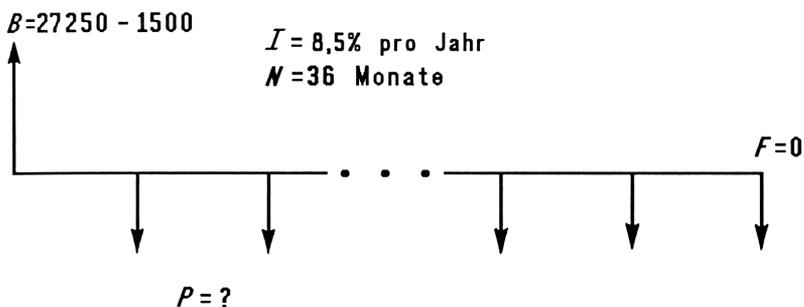
1. Tippen Sie die Routinen ein; drücken Sie zum Abschluß .
2. Wählen Sie gewünschte Routine:
 - XEQ N zum Berechnen der Anzahl Verzinsungsperioden.
 - XEQ I zum Berechnen des Zinssatzes je Periode.
 - XEQ B zum Berechnen des Barwerts eines Kredits oder Spareinlage.

- P zum Berechnen der periodisch auftretenden Zahlung.
 - F zum Berechnen des Endwerts bzw. der Restschuld eines Darlehens.
3. Tippen Sie die Werte von den anderen vier Variablen ein, wenn die Aufforderung dazu angezeigt wird; drücken Sie nach jedem Wert.
 4. Nach dem letzten wird das Ergebnis angezeigt.
 5. Um die gleiche Variable mit anderen Zahlenwerten zu berechnen, ist zu drücken und zu Schritt 3 zurückzukehren.
 6. Für eine vollständig neue Berechnung ist zu Schritt 2 zurückzukehren.

Benutzte Variablen:

- N* Die Anzahl der Verzinsungsperioden.
- I* Der *periodische* Zinssatz als Prozentsatz. (Ist beispielsweise der nominale *Jahreszinssatz* 15% und es gibt 12 Zahlungen pro Jahr, dann beträgt der *periodische* Zinssatz $15 \div 12 = 1,25\%$.)
- Z* Der *periodische* Zinssatz als Dezimalwert.
- B* Der Barwert des Darlehens oder der Spareinlage.
- P* Die periodische Zahlung.
- F* Der Endwert der Spareinlage oder die Restschuld eines Darlehens.
- i* Die Indexvariable, die in diesem Fall für indirekte Adressierung verwendet wird.

Beispiel: Autokredit—Teil 1. Sie erhalten von Ihrem Autohändler einen Kredit mit 3 Jahren Laufzeit und einem jährlichen Zinssatz von 8,5% (bei monatlicher Zinsverrechnung) zum Kauf eines neuen Autos angeboten. Der Kaufpreis des Neuwagens beträgt DM 27 250, wobei Ihr betagtes Gefährt noch mit DM 3 500 in Zahlung genommen werden würde. Wie hoch sind Ihre monatlichen Zahlungen? (Unterstellen Sie, daß die Rückzahlung einen Monat nach dem Kauf beginnen soll—also am *Ende* der ersten Zahlungsperiode.)



Tastensequenz	Anzeige	Beschreibung
\blacksquare [DISP] {FX} 2		Stellt das Anzeigeformat auf FIX 2 ein.
[XEQ] P	N?Wert	Wählt Routine P, welche die Höhe der periodischen Zahlung berechnet.
36 [R/S]	I?Wert	Speichert 36 in N.
8,5 [ENTER] 12 [÷]	I?0,71	Konvertiert Jahreszinssatz in periodischen Zinssatz.
[R/S]	B?Wert	Speichert periodischen Zinssatz in I.
27250 [ENTER] 3500 [-]	B?23.750,00	Berechnet die Höhe des erforderlichen Kredits (Barwert).
[R/S]	F?Wert	Speichert Barwert in B.
0 [R/S]	P=-749,73	Speichert Null in F (Endwert), und berechnet die monatliche Zahlung.

Das Ergebnis hat ein negatives Vorzeichen, da das Problem aus der Sicht des Kreditnehmers behandelt wurde.

Teil 2: Welcher Zinssatz wäre erforderlich, um die monatlichen Zahlungen um DM 30 zu senken?

Tastenfolge:	Anzeige	Beschreibung
[XEQ] I	N?36,00	Wählt Routine I, welche den periodischen Zinssatz berechnet.
[R/S]	B?23.750,00	Akzeptiert 36 als Anzahl Zahlungen/Verzinsungsperioden.
[R/S]	P?-749,73	Akzeptiert DM 23 750,00 als Kreditsumme/Barwert.
[ENTER]	P?-749,73	Kopiert Höhe der Zahlung im Stack, um damit Berechnungen zu ermöglichen. (Das X-Register wird von der als nächstes eingegebenen Zahl überschrieben.)
30 [+]	P?-719,73	Reduziert die monatliche Zahlung um DM 30,00.
[R/S]	F?0,00	Speichert die modifizierte Zahlung.
[R/S]	I=0,48	Akzeptiert Null als Endwert und berechnet I, den monatlichen Zinssatz.
12 [x]	5,74	Berechnet den nominalen Jahreszinssatz.

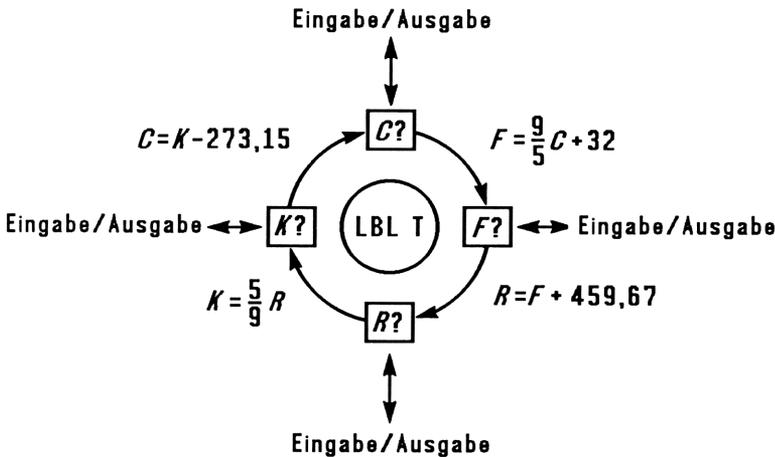
Teil 3: Unter Verwendung des Zinssatzes von 5,74% soll die Restschuld nach 2 Jahren berechnet werden.

Tastenfolge:	Anzeige	Beschreibung
<input type="button" value="XEQ"/> F	N?36,00	Wählt Routine F, welche den Endwert F berechnet.
24 <input type="button" value="R/S"/>	I?0,48	Ändert die Anzahl von Zahlungen/Verzinsungsperioden auf 24 Monate.
<input type="button" value="R/S"/>	B?23.750,00	Akzeptiert den Monatszinssatz.
<input type="button" value="R/S"/>	P?-719,73	Akzeptiert DM 23 750,00 als ursprüngliche Kreditsumme.
<input type="button" value="R/S"/>	F=-8.374,09	Akzeptiert den Zahlungsbetrag und berechnet Restschuld nach 2 Jahren. Das Ergebnis ist wieder mit einem negativen Vorzeichen versehen, was andeutet, daß dieser Betrag von Ihnen geschuldet wird (bzw. abfließen würde).

Einheitenkonvertierungen

Dieses Programm besteht aus zwei Routinen, welche eine physikalische Größe von einer Einheit in eine andere konvertieren. Eine der Routinen führt Umrechnungen zwischen Temperaturen (Celsius, Fahrenheit, Rankine und Kelvin) durch; die andere konvertiert Längen- und Flächenangaben (Inches, Feet und Meter bzw. deren Quadrate). Die Routinen lassen sich einfach modifizieren, womit Sie Ihre eigenen Konvertierungen programmieren können.

Beide Routinen basieren auf "Ferris Wheel Principle", weshalb das Programm eine kreisförmige Struktur besitzt. Nach dem Programmstart wird eine Schleife mit Eingabeaufforderungen durchlaufen. Sie ignorieren dabei alle Aufforderungen (durch Drücken von $\boxed{R/S}$), bis die Aufforderung für die zu konvertierende Einheit angezeigt wird. Möchten Sie z.B. Kelvin umrechnen, so ist die Temperaturroutine durch Drücken von \boxed{XEQ} T aufzurufen und danach $\boxed{R/S}$ zu drücken, bis $K?$ angezeigt wird. Nach der Eingabe des Temperaturwertes (Kelvin) ist $\boxed{R/S}$ zu drücken, bis die neue Temperatureinheit angezeigt wird (z.B. $F?$ für Fahrenheit). Der bei der Eingabeaufforderung angezeigte Wert stellt hierbei noch die Temperatur in °Fahrenheit dar. Drücken Sie \boxed{C} , wenn Sie das Programm abschließen möchten.



"Ferris Wheel" Struktur für Temperaturkonvertierungen

Das Programm ist so konzipiert, daß die Verwendung des Stacks auf ein Minimum reduziert wird. Wenn das Programm beendet wird, befinden sich die ursprünglichen X- und Y-Registerinhalte im Y- und Z-Register, und es wird der konvertierte Wert angezeigt. Befindet sich beim Programmstart der zu konvertierende Wert im X-Register, so drücken Sie bei der entsprechenden Eingabeaufforderung $\boxed{R\downarrow}$, um ihn zurückzuholen.

Die Längen- und Flächen-Konvertierungsroutinen repräsentieren gute Beispiele für die Anwendung von Flags. Beachten Sie, daß Flag 2 gewählt wurde, um durch den Indikator **2** anzuzeigen, daß die jeweilige Einheit *quadriert* wird.

Programmliste:

Programmzeilen:

	Beschreibung
T01 LBL T	Beginn der Temperatur-Konvertierungs-routine.
T02 INPUT C	Zeigt Temperatur in °C, oder erwartet und speichert Celsius-Eingabewert.
T03 9	
T04 ×	
T05 5	
T06 ÷	
T07 32	
T08 +	Konvertiert von Celsius nach Fahrenheit.
T09 STO F	Speichert Fahrenheit Temperatur in <i>F</i> .
T10 R+	Schiebt Stack nach unten, wodurch nur eine Ebene benutzt wird.
T11 INPUT F	Zeigt Temperatur in °F, oder erwartet und speichert Fahrenheit-Eingabewert.
T12 459,67	
T13 +	Konvertiert von Fahrenheit nach Rankine.
T14 STO R	Speichert Rankine Temperatur in <i>R</i> .
T15 R+	Schiebt Stack nach unten, wodurch nur eine Ebene benutzt wird.
T16 INPUT R	Zeigt Temperatur in °R, oder erwartet und speichert Rankine-Eingabewert.
T17 9	
T18 ÷	
T19 5	
T20 ×	Konvertiert von Rankine nach Kelvin.
T21 STO K	Speichert Kelvin Temperatur in <i>K</i> .
T22 R+	Schiebt Stack nach unten, wodurch nur eine Ebene benutzt wird.
T23 INPUT K	Zeigt Temperatur in Kelvin, oder erwartet und speichert Kelvin-Eingabewert.
T24 273,15	
T25 -	Konvertiert von Kelvin nach Celsius.
T26 STO C	Speichert Celsius Temperatur in <i>C</i> .
T27 R+	Schiebt Stack nach unten, wodurch nur eine Ebene benutzt wird.
T28 GT0 T	Rücksprung zum Schleifenanfang für weitere Konvertierungen.

Bytes und Prüfsumme: 058,0; 9EF4

A01	LBL	A	Beginn der Flächen-Konvertierungsroutine.
A02	SF	2	Setzt Flag 2 zum Kennzeichnen von Flächenumrechnungen (<i>Quadrat</i> der Länge).
A03	GTO	Q	Verzweigt zur Konvertierungsroutine.
Bytes und Prüfsumme: 004,5; 2A52			
L01	LBL	L	Beginn der Längen-Konvertierungsroutine.
L02	CF	2	Löscht den Flag für Flächenkonvertierung.
Bytes und Prüfsumme: 003,0; 2C3C			
Q01	LBL	Q	Beginn der kombinierten Längen- und Flächen-Konvertierungsroutine.
Q02	INPUT	I	Zeigt Inches oder Quadrat-Inches an, oder akzeptiert Eingabe.
Q03	12		Eingabe des Konvertierungsfaktors zwischen Inches/Feet.
Q04	FS?	2	Test, ob Flächenkonvertierung vorliegt.
Q05	\times^2		Falls ja, quadrierte Konvertierungsfaktor.
Q06	\div		Berechnet Ergebnis.
Q07	STO	F	Speichert Feet oder Quadrat-Feet.
Q08	R+		Schiebt Stack nach unten, wodurch nur eine Ebene benutzt wird.
Q09	INPUT	F	Zeigt Feet oder Quadrat-Feet an, oder akzeptiert Eingabe.
Q10	0,3048		Eingabe des Konvertierungsfaktors zwischen Feet/Meter.
Q11	FS?	2	Test, ob Flächenkonvertierung vorliegt.
Q12	\times^2		Falls ja, quadrierte Konvertierungsfaktor.
Q13	\times		Berechnet Ergebnis.
Q14	STO	M	Speichert Meter oder Quadratmeter.
Q15	R+		Schiebt Stack nach unten (nur 1 Ebene).
Q16	INPUT	M	Zeigt Meter oder Quadratmeter an, oder akzeptiert Eingabe.
Q17	0,0254		Eingabe des Konvertierungsfaktors zwischen Meter/Inches.
Q18	FS?	2	Test, ob Flächenkonvertierung vorliegt.
Q19	\times^2		Falls ja, quadrierte Konvertierungsfaktor.
Q20	\div		Berechnet Ergebnis.
Q21	STO	I	Speichert Inches oder Quadrat-Inches.
Q22	R+		Schiebt Stack nach unten (nur 1 Ebene).
Q23	GTO	Q	Rücksprung zum Schleifenanfang für weitere Konvertierungen.
Bytes und Prüfsumme: 050,5; 2A15			

Benutzte Flags: Flag 2.

Erforderlicher Speicherplatz: 164 Bytes: 116 für Programm, 48 für Variablen.

Anwendung des Programms:

1. Tippen Sie die Routinen ein; drücken Sie zum Abschluß **[C]**.
2. Drücken Sie **[XEQ]** und das gewünschte Label.
 - *A* für Flächenkonvertierungen (*Area*), oder
 - *L* für Längenkonvertierungen, oder
 - *T* für Temperaturkonvertierungen.
3. Drücken Sie **[R/S]**, bis die gewünschte Eingabeaufforderung angezeigt wird.
4. Tippen Sie einen Wert ein (oder drücken Sie **[R↓]** zum Zurückholen des Werts, falls er beim Programmstart angezeigt wurde).
5. Drücken Sie **[R/S]**, bis die Eingabeaufforderung (mit Ergebnis) der gewünschten Einheit angezeigt wird.
6. Gehen Sie zwecks weiterer Konvertierungen zu Schritt 3 zurück.
7. Drücken Sie **[C]** zum Abschluß des Programms.

Benutzte Variablen:

- C* Temperatur in °Celsius.
- F* Temperatur in °Fahrenheit; oder Länge in Feet.
- R* Temperatur in °Rankine.
- K* Temperatur in Kelvin.
- I* Inches.
- M* Meter.

Beispiel 1: Konvertieren Sie 212°F nach Kelvin.

Tastenfolge:	Anzeige	Beschreibung
[XEQ] T	C?Wert	Wählt die Temperatur-Konvertierungsroutine.
[R/S]	F?Wert	Suche nach der Eingabe für Fahrenheit.

212	<input type="button" value="R/S"/>	R?671,6700	Eingabe von °Fahrenheit und Konvertierung in °Rankine.
	<input type="button" value="R/S"/>	K?373,1500	Konvertiert nach Kelvin.

Beispiel 2: Ein Fußboden hat die Ausmaße 108×127 Inches. Wieviel Quadratmeter entspricht dies?

Tastensequenz	Anzeige	Beschreibung
108	<input type="button" value="ENTER"/>	
127	<input type="button" value="X"/>	K?13.716,0000
<input type="button" value="XEQ"/> A	I?Wert	Wählt Flächen- Konvertierungsroutine.
<input type="button" value="R↓"/>	I?13.716,0000	Ruft aus dem Y-Regi- ster den zuvor berechneten Wert für Quadrat-Inches zurück.
<input type="button" value="R/S"/>	F?95,2500	Berechnet ft ² .
<input type="button" value="R/S"/>	M?8,8490	Berechnet m ² .
<input type="button" value="C"/>	8,8490	Hebt die Eingabeauf- forderung auf und beendet das Programm.

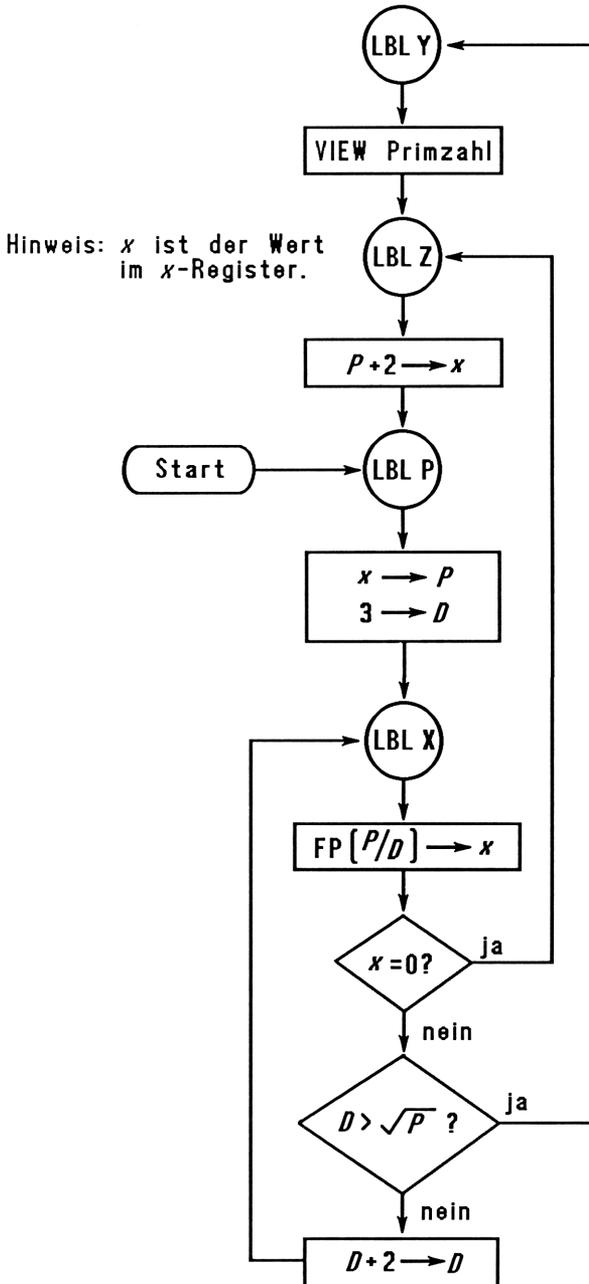
Beispiel 3: Nehmen Sie an, Ihr Rechner zeigt als Ergebnis einer längeren Berechnung den Wert 3,787 an. Weiterhin sei angenommen, daß dieser Wert durch eine in Meter spezifizierte Länge dividiert werden muß, um das Endergebnis zu erhalten. Sie wissen, daß die Länge, durch welche dividiert werden soll, 25,73 Feet beträgt. Berechnen Sie das Endergebnis.

Tastenfolge:	Anzeige	Beschreibung
3,787	3,787	Eingabe des hypothetischen Ergebnisses.
<input type="button" value="XEQ"/> L	I?Wert	Wählt die Routine für Längenkonvertierung.
<input type="button" value="R/S"/>	F?Wert	Geht zur Eingabeaufforderung für Feet weiter.
25,73 <input type="button" value="R/S"/>	M?7,8425	Eingabe des Divisors für Feet, anschließende Konvertierung in Meter.
<input type="button" value="C"/>	7,8425	Hebt die Eingabeaufforderung auf und schließt das Programm ab.
<input type="button" value="+"/>	0,4829	Berechnet das Endergebnis.

Primzahlengenerator

Das Programm erwartet als Eingabe eine ungerade ganze Zahl größer als 3. Wurde eine Primzahl eingegeben (nicht ohne Rest durch ganze Zahl teilbar, außer durch sich selbst und 1), so gibt das Programm den Eingabewert zurück. Handelt es sich um keine Primzahl, dann zeigt das Programm die erste Primzahl an, welche größer als der Eingabewert ist.

Das Programm erkennt "gewöhnliche" Zahlen, indem ausgiebig alle möglichen Faktoren getestet werden. Liegt keine Primzahl vor, dann wird zum eingegebenen Wert 2 addiert (unter der Annahme, es handelt sich noch immer um eine ungerade Zahl) und auf eine Primzahl getestet. Dieser Algorithmus wird so lange wiederholt, bis eine Primzahl gefunden wird.



Flußdiagramm für Primzahlen

Programmliste:

Programmzeilen:	Beschreibung
Y01 LBL Y	Diese Routine zeigt die Primzahl an.
Y02 VIEW P	Zeigt die Primzahl P an.
Bytes und Prüfsumme: 003,0; 9D08	
Z01 LBL Z	Diese Routine addiert 2 zu P , bevor P auf eine Primzahl getestet wird.
Z02 2	
Z03 RCL+ P	
Bytes und Prüfsumme: 004,5; E455	
P01 LBL P	Diese Routine speichert den Eingabewert für P .
P02 STO P	
P03 3	Speichert 3 in Testdivisor D .
P04 STO D	
Bytes und Prüfsumme: 006,0; 9E38	
X01 LBL X	Diese Routine testet P auf Eigenschaft einer Primzahl.
X02 RCL P	
X03 RCL÷ D	
X04 FP	Ermittelt den gebrochenen Teil von $P÷D$.
X05 $x=0?$	Testet, ob Rest gleich Null (Zahl keine Primzahl).
X06 GTO Z	Liegt keine Primzahl vor, so wird nächste Möglichkeit getestet.
X07 RCL P	
X08 SQRT	
X09 RCL D	
X10 $x>y?$	Überprüfung, ob alle möglichen Faktoren versucht wurden.
X11 GTO Y	Wurden alle Faktoren versucht, Verzweigung zur Anzeigeroutine.
X12 2	Berechnet den nächsten möglichen Faktor, $D + 2$.
X13 STO+ D	

X14 GTO X

Verzweigt zum Testen der eventuellen Primzahl mit dem neuen Faktor.

Bytes und Prüfsumme: 021,0; 43F2

Benutzte Flags: Keine.

Erforderlicher Speicherplatz: 50,5 Bytes: 34,5 für Programm, 16 für Variablen.

Anwendung des Programms:

1. Tippen Sie die Routinen ein; drücken Sie zum Abschluß .
2. Tippen Sie eine ungerade ganze Zahl ein.
3. Drücken Sie P zum Start des Programms. Es wird die Primzahl P angezeigt.
4. Wenn Sie die nächste Primzahl auffinden möchten, so drücken Sie .

Benutzte Variablen:

P Primzahl und potentielle Primzahlenwerte.

D Divisor, welcher zum Testen des momentanen Wertes von P benutzt wird.

Anmerkungen: Es erfolgt kein Testen auf das Vorliegen einer ungeraden, positiven ganzen Zahl größer als 3.

Beispiel: Welches ist die erste Primzahl nach 789? Welches ist die darauf folgende Primzahl?

Tastenfolge:	Anzeige	Beschreibung
789 <input type="button" value="XEQ"/> P	P=797,0000	Eintippen von 789 und Starten des Programms; Anzeige der ersten Primzahl nach 789.
<input type="button" value="R/S"/>	P=809,0000	Berechnet die nächste Primzahl nach 797.

Teil 5

Anhänge und Index

Seite 240	A: Kundenunterstützung, Batterien und Service
253	B: Benutzerspeicher und Stack
259	C: Näheres zum Lösen einer Gleichung
273	D: Näheres zur Integration
281	Meldungen
286	Funktionsindex
299	Sachindex

A

Kundenunterstützung, Batterien und Service

Unterstützung beim Anwenden des Rechners

Hewlett-Packard hat sich für eine kontinuierliche Unterstützung der Besitzer von HP-Taschenrechnern verpflichtet. Wenn Sie auf Schwierigkeiten bei der Anwendung des Rechners stoßen, können Sie sich über die Adresse/Telefonnummer auf der Innenseite des Rückumschlags mit Hewlett-Packard in Verbindung setzen.

Es ist jedoch empfehlenswert, daß Sie zuerst den Abschnitt "Antworten auf allgemeine Fragen" durchlesen, bevor Sie mit Hewlett-Packard Kontakt aufnehmen. Erfahrungen haben gezeigt, daß viele Kunden ähnliche Fragen haben und die nachstehende Auflistung enthält vielleicht bereits die Lösung für Ihr Problem.

Antworten auf allgemeine Fragen

F: Wie kann überprüft werden, ob der Rechner einwandfrei funktioniert?

A: Führen Sie den Selbsttest des Rechners durch, wie es auf Seite 246 beschrieben ist.

F: Wie kann ich die Anzahl der angezeigten Dezimalstellen verändern?

A: Verwenden Sie die Funktion  **DISP** (auf Seite 30 beschrieben).

F: Die angezeigten Zahlenwerte enthalten einen Punkt als Dezimalzeichen. Wie kann ich wieder ein Dezimalkomma spezifizieren?

A: Verwenden Sie die Funktion  **MODES** (auf Seite 29 beschrieben).

F: Wie lösche ich den gesamten Speicherbereich oder Teile davon?

A: Mit Hilfe von  **CLEAR** können Sie das CLEAR Menü aufrufen, welches das Löschen aller Variablen, aller Programme (nur im Programm-Modus), der Statistikregister oder des gesamten Speicherbereichs erlaubt (nicht während Programm-Modus).

F: Welche Bedeutung hat das "E" in einer Zahl (z.B. $2,51E-13$)?

A: Die Zahl wird in wissenschaftlicher oder technischer Notation angezeigt. "E" bedeutet dabei *Exponent* von 10, d.h. $2,51 \times 10^{-13}$.

F: Der Rechner zeigt die Meldung MEMORY FULL an. Was ist zu tun?

A: Sie müssen einen Teil des Speicherbereichs löschen, bevor Sie Ihre Operationen fortsetzen können (siehe Anhang B).

F: Warum ergibt die Berechnung des Sinus von π eine sehr kleine Zahl anstatt Null?

A: Der Rechner ist *nicht* funktionsgestört. π kann nicht *exakt* mit der 12-stelligen Genauigkeit des Rechners dargestellt werden.

F: Warum erhalte ich falsche Resultate beim Benutzen der trigonometrischen Funktionen?

A: Sie müssen den richtigen Winkelmodus eingestellt haben—verwenden Sie dazu  **MODES**.

F: Was bedeutet das kleine Symbol in der Anzeige?

A: Das Symbol wird als *Indikator* bezeichnet und kennzeichnet einen besonderen Status des Rechners. Beziehen Sie sich auf "Indikatoren" in Kapitel 1.

Stromversorgung und Batterien

Der Rechner wird mit drei Alkali-Batterien ausgeliefert. Ein neuer Batteriesatz reicht bei normaler Betriebsweise etwa 1 Jahr, wobei die tatsächliche Zeit natürlich von der jeweiligen Anwendungsweise abhängt. Häufige längere Berechnungen verbrauchen mehr Strom als kurze Berechnungen. Lange Programme oder Berechnungen mit SOLVE oder \int FN beanspruchen die Batterien am meisten. Unabhängig von der Anwendungsweise ist festzustellen, daß Quecksilber- und Silberoxid-Batterien etwa zweimal so lange halten wie Alkali-Batterien.

Verwenden Sie nur neue Batterien (Knopfzellen)—keine wiederaufladbaren. Nachfolgende Batterien werden empfohlen:

Alkali	Quecksilber	Silberoxid
Panasonic LR44	Panasonic NP675	Panasonic SR44W oder SP357
Eveready A76	Eveready EP675E	Eveready 357
Varta V13GA	Toshiba NR44 oder MR44	Ray-O-Vac 357
Duracell LR44	Radio Shack NR44 oder MR44 Duracell MP675H	Varta V357

“Schwache Batterien” Indikator

Wenn der Rechner eine abfallende Batteriespannung erkennt () wird angezeigt), dann sollten Sie die Batterien so bald wie möglich ersetzen.

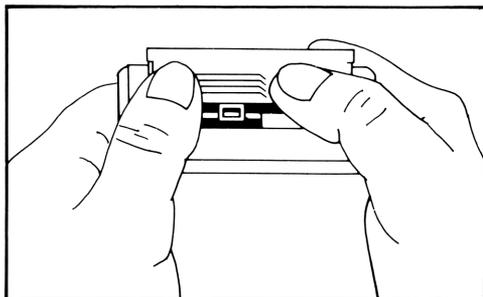
Wenn Sie den Rechner weiterhin benutzen, kann unter Umständen die Batteriespannung unter einen Mindestpegel fallen, welcher die Anzeigequalität sowie die sichere Speicherung Ihrer Daten beeinträchtigt. Tritt dieser Fall ein, dann müssen Sie die Batterien zuerst ersetzen, bevor der Rechner wieder zuverlässig arbeitet. Sollte die Spannung bereits soweit abgefallen sein, daß ein Datenverlust eingetreten ist, so erscheint die Meldung MEMORY CLEAR.

Einsetzen der Batterien

Sind die Batterien entfernt worden, so muß innerhalb einer Minute der neue Batteriesatz eingesetzt werden, wenn kein Datenverlust erfolgen soll. Die neuen Batterien sollten deshalb direkt greifbar sein, bevor Sie die alten entnehmen. Außerdem muß der Rechner während des gesamten Vorgangs ausgeschaltet sein.

Um die Batterien einzusetzen:

1. Halten Sie die drei neuen Batterien griffbereit.
2. Versichern Sie sich, daß der Rechner ausgeschaltet ist. **Drücken Sie nicht [C], bevor das Austauschen der Batterien abgeschlossen ist. Wird der Rechner vorher eingeschaltet, so kann dies die Löschung des Speicherbereichs zur Folge haben.**
3. Halten Sie den Rechner wie abgebildet. Um die Abdeckung des Batteriefachs abzunehmen, drücken Sie diese nach unten und schieben Sie sie nach außen, bis sie abgenommen werden kann.



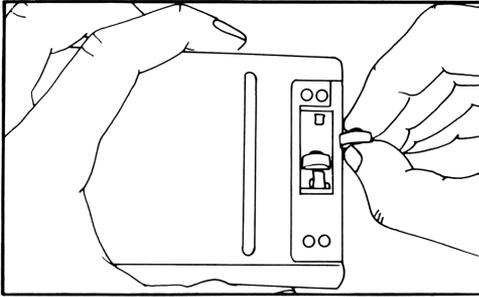
4. Drehen Sie den Rechner um, damit die Batterien herausfallen.



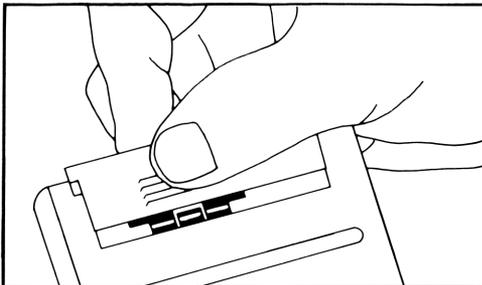
Warnung

Beschädigen Sie nicht die Batterien und werfen Sie diese nicht ins Feuer. Die Batterien könnten dabei gefährliche Chemikalien freisetzen.

5. Setzen Sie die drei neuen Batterien wie abgebildet ein. Die erforderliche Polarität ist auf der Innenseite des Batteriefachs abgebildet. Stellen Sie sicher, daß die Polarität mit der Abbildung übereinstimmt.



6. Schieben Sie die Abdeckung des Batteriefachs in die vorgesehene Führung des Rechnergehäuses (siehe Abbildung).



Schalten Sie nun den Rechner wieder ein. Wenn der Rechner nach dem Einsetzen der neuen Batterien nicht richtig funktioniert, hat es eventuell zu lange gedauert oder Sie haben versehentlich den Rechner eingeschaltet, während die Batterien ausgebaut waren. *Entnehmen Sie die Batterien nochmals und schließen Sie die beiden Batteriekontakte für einige Sekunden kurz* (z.B. mit einer Münze). Setzen Sie die Batterien wieder ein und schalten Sie den Rechner ein; er sollte die Meldung MEMORY CLEAR anzeigen.

Umgebungsbedingungen

Im Hinblick auf die Produktzuverlässigkeit sollten Sie folgende Temperatur- und Luftfeuchtigkeitsgrenzen für den HP-32S einhalten:

- Betriebstemperatur: 0° bis 45°C
- Lagerungstemperatur: –20° bis 65°C
- Luftfeuchtigkeit für Betrieb und Lagerung: 90% relative Luftfeuchtigkeit bei max. 40°C

Feststellen der Reparaturbedürftigkeit

Verwenden Sie nachstehende Richtlinien, um die zuverlässige Funktionsweise des Rechners zu überprüfen. Wenn der Rechner repariert werden muß, beachten Sie bitte den Abschnitt "Im Reparaturfall" auf Seite 249.

■ **Wenn nach dem Einschalten die Anzeige leer bleibt:**

1. Versuchen Sie, den Rechner zurückzusetzen. (Halten Sie gedrückt, während Sie drücken.)
2. Führt Schritt 1 zu keiner Anzeige, so ersetzen Sie die Batterien (siehe Seite 242).

Führen die Schritte 1 und 2 keine Abhilfe herbei, so ist eine Reparatur des Rechners erforderlich.

■ **Wenn das Drücken von Tasten keine Auswirkung auf die Betriebsweise des Rechners hat:**

1. Versuchen Sie, den Rechner zurückzusetzen. (Halten Sie \boxed{C} gedrückt, während Sie \boxed{LN} drücken.)
2. Führt Schritt 1 zu keiner Reaktion, so versuchen Sie das Löschen des Speicherbereichs. (Halten Sie \boxed{C} , $\boxed{\sqrt{x}}$ und $\boxed{\Sigma+}$ gedrückt, wie auf Seite 255 beschrieben). Damit löschen Sie alle von Ihnen gespeicherten Daten.
3. Ist die Funktionsweise nach den Schritten 1 und 2 noch nicht wieder hergestellt, dann entnehmen Sie die Batterien (Seite 243) und schließen Sie die Batteriekontakte kurz (z.B. mit einer Münze). Setzen Sie anschließend die Batterien wieder ein und schalten Sie den Rechner ein. Es sollte MEMORY CLEAR angezeigt werden.

Führen die Schritte 1 bis 3 keine Abhilfe herbei, so ist eine Reparatur erforderlich.

■ **Wenn der Rechner auf das Drücken von Tasten reagiert, Sie aber eine Funktionsstörung vermuten:**

1. Starten Sie den Selbsttest (nachstehend beschrieben). Endet der Test mit einer Fehlermeldung, so ist eine Reparatur erforderlich.
2. Wird der Selbsttest fehlerfrei abgeschlossen, dann liegt unter Umständen eine unkorrekte Bedienungsweise vor. Versuchen Sie nochmals, über den Abschnitt "Antworten auf allgemeine Fragen" auf Seite 240 eine Lösung für Ihr Problem zu finden.
3. Sie können bei Hewlett-Packard zwecks weiterer Unterstützung anfragen. Anschrift und Telefonnummer finden Sie auf der Innenseite des Rückumschlags.

Funktionsprüfung des Rechners—der Selbsttest

Läßt sich die Anzeige einschalten, während jedoch der Rechner anscheinend Probleme bei der Funktionsweise aufweist, so können Sie zur Diagnose einen Selbsttest starten. Gehen Sie wie folgt vor:

1. Um den Selbsttest zu starten, ist \boxed{C} gedrückt zu halten, während $\boxed{y^x}$ gedrückt wird.*
2. Drücken Sie jede Taste achtmal und beachten Sie dabei das Muster in der Anzeige. Nachdem die Taste zum 8. Mal gedrückt wurde, erscheint temporär die Meldung `COPR, HP 1987` und danach `KBD 01`. Dies bedeutet, daß der Rechner zum Testen des Tastenfelds bereit ist.
3. Beginnen Sie in der linken oberen Ecke ($\boxed{\sqrt{x}}$), von links nach rechts vorgehend, und drücken Sie jede Taste in der jeweiligen Reihe. Gehen Sie danach zur nächstunteren Zeile vor, usw., bis Sie jede Taste gedrückt haben.
 - Wurden die Tasten in der richtigen Reihenfolge gedrückt und liegt keine Funktionsstörung vor, so zeigt der Rechner `KBD` an, gefolgt von einer zweistelligen hexadezimalen Zahl.
 - Wurde die Reihenfolge nicht eingehalten oder funktioniert eine Taste nicht einwandfrei, so wird nach dem nächsten Tastendruck eine Fehlermeldung angezeigt (siehe Schritt 4).
4. Am Ende zeigt der Selbsttest eine der zwei nachstehenden Meldungen an:
 - Anzeige von `32S - OK`, wenn der Test erfolgreich abgeschlossen wurde. Gehen Sie zu Schritt 5 über.
 - Anzeige von `32S - FAIL`, gefolgt von einer einstelligen Zahl, wenn ein Fehler vorliegt. Wurde die Meldung aufgrund der falschen Tastenreihenfolge erzeugt, so sollten Sie den Rechner zurücksetzen (\boxed{C} gedrückt halten, während Sie \boxed{LN} drücken) und den Selbsttest erneut starten. Wurde die Meldung nicht durch einen Bedienungsfehler verursacht, dann ist der Selbsttest zu wiederholen, um das Ergebnis zu verifizieren. Bestätigt sich die Fehlerbedingung, so ist eine Reparatur erforderlich (siehe Seite 249). Legen Sie eine Kopie der Fehlermeldung bei, wenn Sie den Rechner zur Reparatur einsenden.
5. Um den Selbsttest abzubrechen, ist der Rechner zurückzusetzen (\boxed{C} gedrückt halten, während \boxed{LN} gedrückt wird).

* Wenn Sie $\boxed{1/x}$ drücken, während Sie \boxed{C} niederhalten, starten Sie einen fortlaufenden Selbsttest, welcher werksseitig verwendet wird. Sie können den Test durch Drücken von \boxed{C} abbrechen.

Einjährige Gewährleistungsfrist

Gewährleistungsumfang

Hewlett-Packard gewährleistet, daß der Rechner frei von Material- und Verarbeitungsfehlern ist. Die Garantiezeit beginnt ab dem Kaufdatum und beträgt ein Jahr. Während dieser Zeit verpflichtet sich Hewlett-Packard, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn der Rechner direkt oder über einen autorisierten Vertragshändler an Hewlett-Packard eingeschickt wird. (Ein Ersatzrechner kann einem neueren Modell mit gleichwertiger oder besserer Funktionalität entsprechen.) Versandkosten bis zur Auslieferung bei einem Hewlett-Packard Service-Zentrum gehen zu Ihren Lasten, unabhängig davon, ob sich das Gerät noch in der Garantiezeit befindet oder nicht. Wenn Sie den Rechner verkaufen oder verschenken, so wird die Gewährleistung automatisch auf den neuen Eigentümer übertragen und bezieht sich weiterhin auf das ursprüngliche Kaufdatum.

Gewährleistungsausschluß

Batterien sowie durch Batterien verursachte Schäden sind von der Gewährleistung durch Hewlett-Packard nicht erfaßt. Setzen Sie sich mit dem Hersteller der Batterien zwecks einer diesbezüglichen Gewährleistung in Verbindung.

Die von Hewlett-Packard angebotene Gewährleistung gilt nicht für Schäden, die durch unsachgemäße Betriebsweise entstanden sind. Der Ausschluß gilt ebenso, wenn Modifikationen oder Servicearbeiten durch nicht von Hewlett-Packard autorisierten Reparaturzentren durchgeführt wurden.

Es gibt keinen weiteren Gewährleistungsumfang. Die Einleitung der erforderlichen Reparatur- oder Ersatzleistungen ist ausschließlich dem Kunden überlassen. **Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden.** Dies gilt nicht, soweit gesetzlich zwingend gehaftet wird.

Im Reparaturfall

Hewlett-Packard unterhält in den meisten Ländern der Welt Reparaturzentren. Diese Zentren reparieren Ihren Rechner oder ersetzen ihn durch ein gleich- oder höherwertigeres Modell, unabhängig vom Garantiefall. Nach der Garantiezeit von einem Jahr werden Reparaturkosten berechnet. Der Service wird normalerweise innerhalb von 5 Arbeitstagen ausgeführt.

Service-Adressen

- **In Europa:** Sofern Sie sich in der BRD aufhalten, können Sie sich auf die Anschriften auf der Innenseite des Rückumschlags beziehen. Die Anschrift der europäischen Zentrale finden Sie nachstehend. *Nehmen Sie zuerst Kontakt mit Hewlett-Packard auf, bevor Sie Ihren Rechner zur Reparatur einschicken.*

Hewlett-Packard S.A.
150, Route du Nant-d'Avril
1217 Meyrin 2
Schweiz
Tel: (022) 82 81 11

■ In den USA:

Hewlett-Packard
Calculator Service Center
1030 N.E. Circle Blvd
Corvallis, OR 97330, USA
Tel: (503) 757 2002

- **In anderen Ländern:** Nehmen Sie Kontakt mit der nächstgelegenen Hewlett-Packard-Geschäftsstelle auf, um die korrekte Anschrift eines Reparaturzentrums zu erfahren.

Reparaturkosten

Für Reparaturen nach der Garantiezeit wird eine Reparaturkostenpauschale erhoben. Diese schließt sämtliche Arbeits- und Materialkosten mit ein. In der BRD unterliegt die Pauschale der Mehrwertsteuer. Sämtliche Steuern werden auf der Rechnung getrennt ausgewiesen.

Die Reparaturkostenpauschale deckt nicht die Reparatur von Rechnern, welche durch Gewalteinwirkung oder Fehlbedienung zerstört wurden. In diesem Fall werden die Reparaturkosten individuell nach Arbeits- und Materialaufwand festgesetzt.

Versandanweisungen

Wenn Ihr Rechner repariert werden muß, senden Sie ihn bitte mit folgenden Unterlagen ein:

- Vollständige Absenderangabe und eine Beschreibung des Fehlers. Wenn der Verpackung Ihres Rechners eine Servicekarte beigelegt war, können Sie diese für die Angabe der entsprechenden Informationen verwenden.
- Rechnung oder anderer Kaufbeleg, wenn die einjährige Garantiezeit noch nicht abgelaufen ist.

Der Rechner und die erforderlichen Begleitinformationen sollten in der Originalverpackung oder einer adäquaten Schutzverpackung versandt werden, um Transportschäden zu vermeiden. Solche Transportschäden werden durch die einjährige Garantiezeit nicht abgedeckt; der Versand zum Reparaturzentrum erfolgt auf Ihre Gefahr, wobei Hewlett-Packard Ihnen zu einer Transportversicherung rät.

Alle Versand- und Zollkosten unterliegen der Verantwortlichkeit des Kunden.

Gewährleistung bei Reparaturen

Für Reparaturen außerhalb der Garantiezeit leistet Hewlett-Packard eine Garantie von 90 Tagen ab Reparaturdatum bezüglich Material- und Bearbeitungsfehlern.

Servicevereinbarungen

Für Ihren Rechner gibt es eine Vereinbarung über Serviceunterstützung. Beziehen Sie sich auf die Dokumentation, welche der Versandpackung beigelegt ist. Für zusätzliche Informationen sollten Sie sich mit Ihrem HP Vertragshändler oder einer Hewlett-Packard-Geschäftsstelle in Verbindung setzen.

Sicherheitsbestimmungen

Funkschutz

Der HP-32S wurde von Hewlett-Packard geprüft und entspricht den Bestimmungen der Allgemeinen Verfügung FTZ 1046/84. Als Nachweis ist der Rechner mit dem VDE-Funkschutzzeichen mit Index 0871B gekennzeichnet.

Benutzerspeicher und Stack

Dieser Anhang behandelt:

- Zuweisung und Anforderungen des Benutzerspeichers
- Zurücksetzen des Rechners ohne Beeinflussung des Speicherbereichs
- Löschen des gesamten Benutzerspeichers und Zurücksichern der Voreinstellungswerte
- Operationen, welche sich auf den Stack Lift auswirken.

Verwalten des Speicherbereichs

Der HP-32S verfügt über einen Benutzerspeicher von etwa 390 Bytes (Speichereinheiten). SOLVE, fFN und statistische Berechnungen erfordern neben Variablen und Programmen ebenfalls einen Bereich im Benutzerspeicher. (fFN ist besonders "speicherextensiv" bei der Ausführung.)

Alle von Ihnen gespeicherten Daten bleiben so lange erhalten, bis sie von Ihnen explizit gelöscht werden. Ist für die von Ihnen beabsichtigte Operation nicht mehr genügend freier Speicherplatz vorhanden, so zeigt der Rechner MEMORY FULL an. In dieser Situation müssen Sie zuerst bereits belegten Speicherplatz wieder freigeben; Sie haben dabei folgende Möglichkeiten:

- Löschen aller nicht mehr benötigten Variablen (Seite 50).
- Löschen aller nicht mehr benötigten Programme (Seite 85).
- Löschen der Statistikregister (drücken Sie CLEAR {Σ}).
- Löschen des gesamten Benutzerspeichers (drücken Sie CLEAR {ALL}).

Speicherplatz-Anforderungen

Daten oder Operation	Erforderlicher Speicherplatz
Variablen	8 Bytes je Variable (mit Inhalt ungleich 0). Kein Byte, wenn Inhalt gleich Null.
Anweisungen in Programmzeilen	1,5 Bytes.
Zahlen in Programmzeilen	Ganze Zahlen 0 bis 99: 1,5 Bytes. Alle anderen Zahlen: 9,5 Bytes.
Statistikdaten	48 Bytes.
SOLVE Berechnungen	33,5 Bytes.
\int FN (Integration) Berechnungen	140 Bytes.

Um die gesamte Speicheranforderung von bestimmten Programmen angezeigt zu erhalten, drücken Sie **MEM**{PGM}. Drücken Sie **▼** oder **▲** zum Ansehen der Einträge. (Ein Beispiel finden Sie auf Seite 86.)

Zur manuellen Freigabe des Speicherbereichs, welcher von einer unterbrochenen SOLVE oder \int FN Berechnung belegt ist, müssen Sie **LBL/RTN** {RTN} drücken. Diese Freigabe erfolgt automatisch bei jeder Programmausführung bzw. dem Beginn einer SOLVE oder \int FN Berechnung.

Zurücksetzen des Rechners

Sollte der Rechner nicht mehr auf einen Tastendruck reagieren oder ist die Betriebsweise anderweitig gestört, dann sollten Sie das Zurücksetzen des Rechners versuchen. Dadurch wird die momentane Berechnung abgebrochen und Programmeingabe, Zifferneingabe, Programmausführung, SOLVE und \int FN Berechnungen, VIEW oder INPUT Anzeigen werden aufgehoben. Gespeicherte Daten bleiben normalerweise erhalten.

Drücken Sie zum Zurücksetzen des Rechners \boxed{LN} , während Sie \boxed{C} gedrückt halten. Läßt sich der Rechner nicht zurücksetzen, sollten Sie neue Batterien einsetzen. Führt der Austausch der Batterien zu keiner Abhilfe bzw. ist der Rechner noch immer funktionsgestört, so sollten Sie versuchen, unter der nachfolgenden Anleitung den Speicherbereich zu löschen.

Es kann ein Zurücksetzen des Rechners erfolgen, wenn er fallen gelassen oder wenn die Stromversorgung unterbrochen wird.

Löschen des Speicherbereichs

Der normale Weg zum Löschen des Benutzerspeichers besteht im Drücken von $\blacksquare \boxed{CLEAR} \{ALL\}$. Allerdings gibt es noch eine intensivere Löschoption, welche weitere Informationen zurücksetzt und auch in den Fällen hilfreich ist, wenn das Tastenfeld nicht mehr einwandfrei zu funktionieren scheint.

Wenn der Rechner nicht mehr auf Tastendruck reagiert und das Zurücksetzen des Rechners oder das Austauschen der Batterien nicht die gewünschte Abhilfe bringt, sollten Sie über die nachstehende Tastenfolge den Speicherbereich löschen. Dadurch wird der gesamte Speicherbereich gelöscht, der Rechner zurückgesetzt *und* alle Formate und Modi wieder in ihre Voreinstellung gesetzt (siehe unten).

1. Halten Sie \boxed{C} gedrückt.
2. Halten Sie $\boxed{\sqrt{x}}$ gedrückt.
3. Drücken Sie $\boxed{\Sigma+}$. (Sie halten nun 3 Tasten gleichzeitig gedrückt.) Wenn Sie alle Tasten wieder freigeben, wird bei der erfolgreichen Operation die Meldung MEMORY CLEAR angezeigt.

Voreinstellungen

Kategorie	Einstellung
Winkelmodus	Grad (Altgrad)
Basis-Modus	Dezimal
Kontrast	Mittel
Dezimalzeichen	“.”
Anzeigeformat	FIX 4
Flags	Gelöscht bzw. Null
FN= (momentane Funktion)	Null
Programmzeiger (momentane Zeile)	PRGM TOP
Programmspeicher	Gelöscht
Startwert für Zufallszahl	Null
Stack Lift	Aktiviert
Stackregister	Null
Variablen	Null

Der Speicher kann unbeabsichtigt gelöscht werden, wenn der Rechner fallen gelassen oder die Stromversorgung unterbrochen wird.

Status des Stack Lifts

Die vier Stackregister sind immer vorhanden und der Stack besitzt immer einen *Stack Lift Status*. Damit ist gemeint, daß in Bezug auf die als nächstes in das X-Register gebrachte Zahl der Stack Lift immer entweder *aktiviert* (Lift freigegeben) oder *desaktiviert* (Lift gesperrt) ist. (Beziehen Sie sich auf Kapitel 2, “Der automatische Speicherstack“.)

Jede in den nachstehenden zwei Listen *nicht* aufgeführte Funktion aktiviert den Stack Lift.

Desaktivierende Operationen

Es gibt vier Operationen, welche den Stack Lift sperren. Eine Zahl, welche nach einer dieser sperrenden Operationen eingetippt wird, überschreibt die momentan im X-Register gespeicherte Zahl. Der Y-, Z- und T-Registerinhalt bleibt dabei unverändert.

ENTER $\Sigma+$ $\Sigma-$ CLx

Weiterhin sperren die Tasten **C** und **↵**, sofern sie wie CLx wirken, ebenfalls den Stack Lift.

Die INPUT Funktion *desaktiviert* den Stack Lift, wenn ein Programm zur Eingabe von Daten angehalten wird (womit jede dabei eingetippte Zahl das X-Register überschreibt), *aktiviert* jedoch wieder den Stack Lift nach der Fortsetzung des Programms.

Neutrale Operationen

Die nachstehenden Operationen haben keine Auswirkung auf den seitherigen Status des Stack Lifts:

DEG,RAD, GRAD	FIX,SCI, ENG,ALL	DEC,HEX, OCT,BIN	CLVARS
PSE	SHOW	RADIX.;RADIX,	CL Σ
OFF	R/S und STOP	▲ , ▼	C *, ↵ *
MEM {VAR}†	MEM {PGM}†	GTO ◦ ◦	GTO ◦ <i>Label</i> <i>nn</i>
Umschalten von Binär-Fenstern	Zifferneingabe	Fehler	PRGM und Pro- grammeingabe

* Außer bei Anwendung als CLx.

† Einschließlich aller während der Kataloganzeige ausgeführten Operationen, außer {VAR} **ENTER** und {PGM} **XEQ**, was den Stack Lift aktiviert.

Status des LAST X Registers

Die folgenden Operationen sichern x im LAST X Register:

$+$, $-$, \times , \div	SQRT, x^2	e^x , 10^x
LN, LOG	y^x	$1/x$
%, %CHG	$\Sigma+$, $\Sigma-$	RCL+, $-$, \times , \div *
\hat{x} , \hat{y}	SIN, COS, TAN	ASIN, ACOS, ATAN
SINH, COSH, TANH	ASINH, ACOSH, ATANH	IP, FP, RND, ABS
$y, x \rightarrow \theta, r$; $\theta, r \rightarrow y, x$	\rightarrow HR, \rightarrow HMS	\rightarrow DEG, \rightarrow RAD
Cn, r; Polarnotation, r	$x!$	CMPLX+/-
CMPLX+, $-$, \times , \div	CMPLX e^x , LN, y^x , $1/x$	CMPLX SIN, COS, TAN

* Beachten Sie, daß alle Folgen x $\boxed{\text{RCL}} \boxed{+}$ Variable der Rückruf-Arithmetik einen anderen Wert im LAST X Register als die Folge x $\boxed{\text{RCL}} \boxed{\text{Variable}} \boxed{+}$ speichert. Die erste speichert x in LAST X, die letzte speichert die zurückgerufene Zahl in LAST X.

Näheres zum Lösen einer Gleichung

Dieser Anhang enthält Informationen über die Anwendung des SOLVE Algorithmus, welche über die grundlegende Beschreibung in Kapitel 7 hinausgehen.

Arbeitsweise von SOLVE

SOLVE stellt eine *iterative* Operation dar, d.h. sie führt wiederholt die spezifizierte Funktion aus. Sie beginnt mit einer Anfangsnäherung für die unbekannte Variable x und verbessert diese Näherung mit jeder nachfolgenden Ausführung von $f(x)$.*

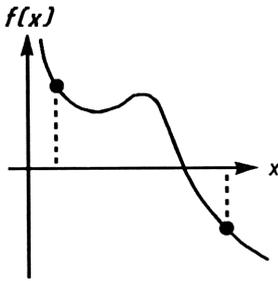
Wenn zwei aufeinanderfolgende Näherungen der Funktion $f(x)$ unterschiedliche Vorzeichen besitzen, nimmt der Algorithmus an, daß der Graph von $f(x)$ im Intervall zwischen diesen Näherungen die x -Achse mindestens einmal schneidet. Dieses Intervall wird dann systematisch verkleinert, bis eine Nullstelle gefunden wird.

Damit SOLVE eine Nullstelle auffinden kann, muß die Nullstelle innerhalb des Zahlenbereichs des Rechners liegen und $f(x)$ muß für den Wertebereich definiert sein, in welchem die iterative Suche stattfindet. SOLVE findet immer eine Lösung, wenn eine oder mehrere der nachstehenden Bedingungen erfüllt sind:

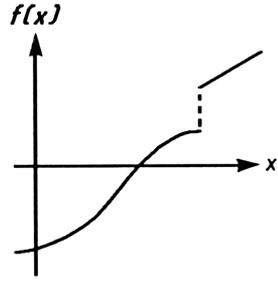
- Zwei Anfangsnäherungen führen zu $f(x)$ Werten mit entgegengesetzten Vorzeichen, und der Graph der Funktion schneidet die x -Achse wenigstens an einer Stelle zwischen diesen Anfangsnäherungen (Abbildung a, nächste Seite).

* $f(x)$ ist das mathematische Kürzel für eine Funktion, die die Definition der unbekanntenen Variablen x enthält.

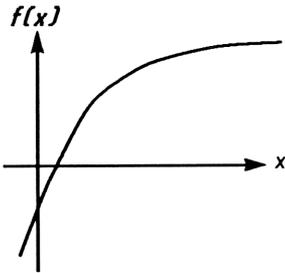
- $f(x)$ nimmt immer zu oder ab, wenn x erhöht wird (Abbildung b, unten).
- Der Graph von $f(x)$ hat überall entweder eine konkave oder konvexe Form (Abbildung c).
- $f(x)$ besitzt ein oder mehrere lokale Minima und Maxima und jedes tritt einzeln zwischen benachbarten Nullstellen von $f(x)$ auf (Abbildung d).



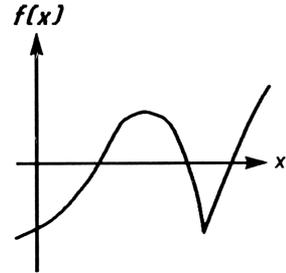
a



b



c



d

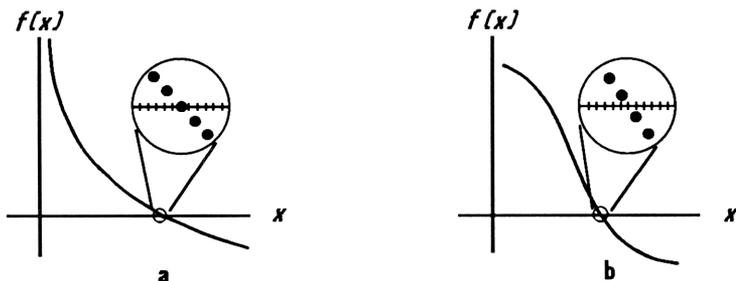
Funktionen mit lösbaren Nullstellen

In den meisten Situationen stellt die berechnete Nullstelle eine genaue Schätzung der theoretischen, unendlich genauen Nullstelle der Gleichung dar. Eine "ideale" Lösung wäre vorhanden, wenn $f(x) = 0$. Allerdings ist auch ein Wert ungleich Null für $f(x)$ ebenso oft akzeptabel, da dieser aus angenäherten Werten mit begrenzter (12-stelliger) Genauigkeit resultiert.

Interpretieren von Ergebnissen

Der SOLVE Algorithmus kann unter einer der beiden Bedingungen eine Lösung ermitteln:

- Er findet eine Näherung, bei welcher sich für $f(x)$ Null ergibt (Abbildung a, unten).
- Er findet eine Näherung, bei welcher sich für $f(x)$ nicht genau Null ergibt. Die berechnete Lösung stellt jedoch einen 12-stelligen Wert dar, welcher in unmittelbarer Nachbarschaft zum Schnittpunkt des Graphen mit der x -Achse liegt (Abbildung b, unten). Dies ist dann der Fall, wenn die zwei Endnäherungen Nachbarn sind (sie unterscheiden sich nur um 1 in der 12. Stelle) und der Funktionswert ist für eine Näherung negativ, aber positiv für die andere.* In den meisten Fällen liegt $f(x)$ sehr nahe bei Null.



Fälle einer aufgefundenen Nullstelle

Um zusätzliche Informationen über das gefundene Ergebnis zu erhalten, drücken Sie $\boxed{R\downarrow}$ zur Anzeige der vorangehenden Näherung der Nullstelle (x), welche im Y-Register gespeichert ist. Drücken Sie erneut $\boxed{R\downarrow}$, um den Wert von $f(x)$ anzusehen, welcher im Z-Register geblieben ist. Wenn $f(x)$ gleich Null oder sehr klein ist, so ist es sehr wahrscheinlich, daß eine Lösung gefunden wurde. Ist $f(x)$ jedoch relativ groß, so sollten Sie die Ergebnisse sorgfältig interpretieren.

* Oder sie sind $(0, 10^{-499})$ oder $(0, -10^{-499})$.

Beispiel: Gleichung mit einer Nullstelle. Ermitteln Sie die Lösung der Gleichung:

$$-2x^3 + 4x^2 - 6x + 8 = 0,$$

welche sich unter Verwendung des Hornerschen Schemas (Kapitel 5) wie folgt vereinfachen läßt:

$$x(x(-2x + 4) - 6) + 8 = 0.$$

Geben Sie die Funktion als Programm ein:

```
A01 LBL A
A02 -2
A03 RCL X
A04 4
A05 +
A06 RCL X
A07 6
A08 -
A09 RCL X
A10 8
A11 +
A12 RTN
```

Tastenfolge:	Anzeige	Beschreibung
■ SOLVE/ {FN} A 0 STO X 10 ■ SOLVE/ {SOLVE} X	X=1,6506	Berechnet x unter Verwendung der Anfangsnäherungen 0 und 10.
R ↓	1,6506	Die zwei Endnäherungen sind auf 4 Dezimalstellen gleich.
R ↓	-1,0000E-11	$f(x)$ ist <i>sehr</i> klein, womit die Approximation eine gute Nullstelle darstellt.

Beispiel: Gleichung mit zwei Nullstellen. Berechnen Sie die zwei Nullstellen der Parabelgleichung:

$$x^2 + x - 6 = 0.$$

Geben Sie die Funktion als Programm ein:

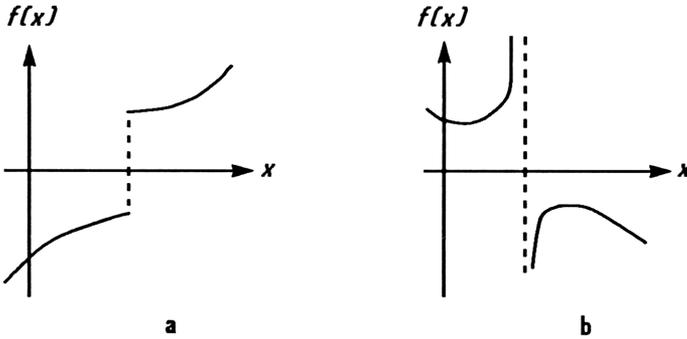
```
D01 LBL D
D02 RCL X
D03 x2
D04 RCL+ X
D05 6
D06 -
D07 RTN
```

Tastenfolge:	Anzeige	Beschreibung
■ SOLVE/f {FN} D 0 STO X 10 ■ SOLVE/f {SOLVE} X	x=2,0000	Berechnet positive Nullstelle über Anfangsnäherungen 0 und 10.
R↵	2,0000	Die beiden Endnäherungen sind gleich.
R↵ ■ SHOW	0,000000000000	$f(x) = 0$.
0 STO X 10 +/- ■ SOLVE/f {SOLVE} X	x=-3,0000	Berechnet negative Nullstelle über Anfangsnäherungen 0 und -10.
R↵ R↵ ■ SHOW	0,000000000000	$f(x) = 0$.

Bestimmte Fälle erfordern zusätzliche Überlegungen:

- Besitzt der Graph der Funktion eine Unstetigkeitsstelle, die die x -Achse überspringt, wird von SOLVE ein an die Sprungstelle angrenzender x -Wert als Nullstelle angegeben (siehe Abbildung a, nächste Seite). In diesem Fall kann $f(x)$ relativ groß sein.

- Werte von $f(x)$ können an der Stelle, wo der Graph sein Vorzeichen wechselt, auch gegen unendlich streben (siehe Abbildung b, unten). Diese Situation wird als *Pol* bezeichnet. Wenn der Funktionswert im Pol sein Vorzeichen wechselt, wird der zugehörige x -Wert als Nullstelle der Funktion aufgefaßt. Der Wert von $f(x)$ wird allerdings relativ groß sein. Ist der Pol ein x -Wert, der mit 12 Stellen *genau* dargestellt werden kann, dann benutzt SOLVE diesen Wert und bricht die Berechnung mit einer Fehlermeldung vorzeitig ab.



Spezielle Fälle: Unstetigkeitsstelle und Pol

Beispiel: Unstetige Funktion. Berechnen Sie die Nullstelle der Gleichung:

$$IP(x) - 1,5 = 0$$

Geben Sie die Funktion als Programm ein:

```
E01 LBL E
E02 RCL X
E03 IP
E04 1,5
E05 -
R06 RTN
```

Tastenfolge:	Anzeige	Beschreibung
■ SOLVE/ {FN}E 0 STO X 5 ■ SOLVE/ {SOLVE}X	X=2,0000	Berechnet eine Nullstelle über Anfangsnäherungen 0 und 5.
■ SHOW	1,999999999999	Zeigt Nullstelle mit 11 Dezimalstellen.
R↓ ■ SHOW	2,000000000000	Die nun angezeigte zweitletzte Näherung ist etwas größer.
R↓	-0,5000	$f(x)$ ist relativ groß.

Beachten Sie den Unterschied zwischen den letzten zwei Näherungen sowie den relativ großen Funktionswert von $f(x)$. Das Problem liegt darin, daß kein x -Wert existiert, für welchen $f(x)$ gleich Null ist. Es gibt jedoch bei $x = 1,9999999999$ einen benachbarten Wert von x , welcher zu einem Vorzeichenwechsel für $f(x)$ führt.

Beispiel: Ein Pol. Bestimmen Sie die Nullstelle für die Gleichung:

$$\frac{x}{x^2 - 6} - 1 = 0$$

Bei der Näherung von x an $\sqrt{6}$ entwickelt sich $f(x)$ zu einem sehr großen positiven oder negativen Wert.

Geben Sie die Funktion als Programm ein:

```
F01 LBL F
F02 RCL X
F03 x2
F04 6
F05 -
F06 RCL X
F07 x<>y
F08 ÷
F09 1
F10 -
F11 RTN
```

Beachten Sie, daß sich das Programm verkürzen läßt, indem die Zeilen F06–F07 gelöscht werden und eine zweite RCL X Anweisung nach Zeile F02 hinzugefügt wird.

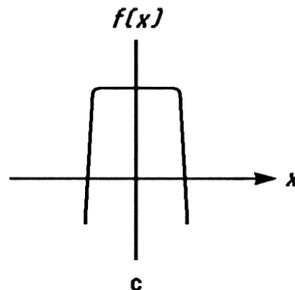
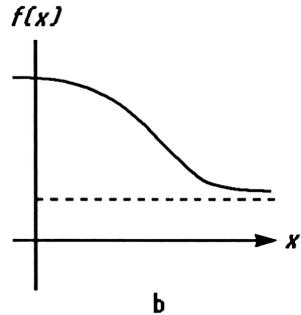
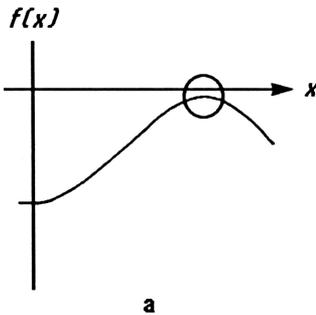
Tastenfolge:	Anzeige	Beschreibung
<div style="border: 1px solid black; padding: 2px;">SOLVE/↓</div> {FN}F		Berechnet die Nullstelle über die Anfangsnäherungen, die $\sqrt{6}$ ein- grenzen.
2,3 <div style="border: 1px solid black; padding: 2px;">STO</div> X 2,7		
<div style="border: 1px solid black; padding: 2px;">SOLVE/↓</div> {SOLVE}X	X=2,4495	
<div style="border: 1px solid black; padding: 2px;">R↓</div> <div style="border: 1px solid black; padding: 2px;">R↓</div>	81.649.658.092,0	$f(x)$ ist relativ groß.

Es gibt einen Pol zwischen den beiden Endnäherungen. Die Anfangsnäherungen führten zu entgegengesetzten Vorzeichen für $f(x)$. SOLVE hat das Intervall zwischen den aufeinanderfolgenden Näherungen ständig verkleinert, bis zwei benachbarte Werte gefunden wurden. Unglücklicherweise verursachten diese benachbarten Werte die Annäherung von $f(x)$ an einen Pol anstatt der x -Achse. Die Funktion *besitzt tatsächlich* Nullstellen bei -2 und 3 , welche durch Vorgabe guter Anfangsnäherungen gefunden werden können.

Wenn SOLVE keine Nullstelle finden kann

Es kann vorkommen, daß SOLVE keine Nullstellen auffinden kann. Nachstehende Bedingungen führen zur Meldung `NO ROOT FND`:

- Die Suche kann bei einem lokalen Minimum/Maximum abgebrochen werden (siehe Abbildung a, unten). Wenn der letzte Wert von $f(x)$ (im Z-Register gespeichert) nahe 0 ist, liegt möglicherweise eine Nullstelle vor; der in der Unbekannten gespeicherte Wert kann eine 12-stellige Zahl sehr nahe der theoretischen Nullstelle sein.
- Die Suche kann abgebrochen werden, wenn SOLVE sich einer horizontalen Asymptoten nähert, wobei $f(x)$ für einen großen Bereich von x im wesentlichen konstant ist (siehe Abbildung b, unten). Der letzte Wert von $f(x)$ ist der Wert der potentiellen Asymptoten.
- Die Suche kann sich auf einen "flachen" Bereich der Funktion konzentrieren (siehe Abbildung c, unten). Der letzte Wert von $f(x)$ ist der Funktionswert in diesem Bereich.



Fälle, wo keine Nullstelle gefunden wurde

SOLVE zeigt einen mathematischen Fehler an, wenn eine Näherung zu einer unzulässigen Operation führt (z.B. Division durch 0, Quadratwurzel einer negativen Zahl, usw.). Sie sollten daran denken, daß SOLVE Näherungen über einen weiten Bereich erzeugen kann. Sie können manchmal Situationen, die zu mathematischen Fehlern führen, vermeiden, indem Sie sehr gute Anfangsnäherungen vorgeben. Zeigt SOLVE eine mathematisch bedingte Fehlermeldung an, so drücken Sie **RCL** *unbekannte Variable* (oder **VIEW** *Variable*) zur Anzeige des fehlerverursachenden Wertes.

Beispiel: Ein relatives Minimum. Berechnen Sie die Nullstelle der Parabelgleichung:

$$x^2 - 6x + 13 = 0$$

Sie besitzt ein Minimum bei $x = 3$.

Geben Sie die Funktion als Programm ein:

```
G01 LBL G
G02 RCL X
G03 x²
G04 6
G05 RCL× X
G06 -
G07 13
G08 +
G09 RTN
```

Tastenfolge:	Anzeige	Beschreibung
SOLVE/ {FN}G 0 STO X 10 SOLVE/ {SOLVE}X	NO ROOT FND	Suche erfolglos mit den Näherungen 0 und 10.
↵ SHOW	3,00000010001	Zeigt die Endnäherung von x an.
R↵ SHOW	3,00000468443	Vorherige Näherung ist nicht gleich.
R↵	4,0000	Endwert für $f(x)$ ist relativ groß.

Beispiel: Asymptote. Berechnen Sie die Nullstelle der Gleichung:

$$10 - \frac{1}{x} = 0$$

Geben Sie die Funktion als Programm ein:

```
H01 LBL H
H02 10
H03 RCL X
H04 1/x
H05 -
H06 RTN
```

Tastenfolge:	Anzeige	Beschreibung
■ SOLVE/∫ {FN}H ,005 STO X 5 ■ SOLVE/∫ {SOLVE}X	X=0,1000	Löst x über die Anfangsnäherungen 0,005 und 5.
R↵	0,1000	Vorherige Näherung ist gleich.
R↵ ■ SHOW	999999999999	$f(x) = 0$.

Beachten Sie die Auswirkung bei negativen Anfangswerten:

Tastenfolge:	Anzeige	Beschreibung
1 +/- STO X 2 +/- ■ SOLVE/∫ {SOLVE}X	NO ROOT FND	Es konnte keine Nullstelle für $f(x)$ gefunden werden.
⏴	-46.666.666.692,1	Zeigt letzte Näherung von x an.
R↵	-5,7750E15	Vorherige Näherung hat viel größeren Betrag.
R↵	10,0000	$f(x)$ für die letzte Näherung ist ziemlich groß.

Bei einer Betrachtung der Gleichung wird deutlich, daß bei negativem x der kleinste mögliche Funktionswert 10 ist. $f(x)$ nähert sich 10 mit zunehmenden negativen Werten für x .

Beispiel: Mathematisch bedingter Fehler. Ermitteln Sie die Nullstelle von:

$$\sqrt{[x \div (x + 0,3)]} - 0,5 = 0$$

Geben Sie die Funktion als Programm ein:

```

I01 LBL I
I02 RCL X
I03 0,3
I04 RCL + X
I05 ÷
I06 SQRT
I07 0,5
I08 -
I09 RTN

```

Versuchen Sie zuerst, eine positive Nullstelle zu finden.

Tastenfolge:	Anzeige	Beschreibung
■ SOLVE/ {FN} I 0 STO X 10 ■ SOLVE/ {SOLVE} X	X=0,1000	Berechnet Nullstelle über Anfangsnäherungen 0 und 10.

Versuchen Sie nun, eine negative Nullstelle über die Näherungen 0 und -10 zu finden. Beachten Sie, daß die Funktion für x -Werte zwischen 0 und $-0,3$ nicht definiert ist, da diese Werte zu einer negativen Wurzel führen.

0 STO X 10 +/- ■ SOLVE/ {SOLVE} X	SQRT(NEG)	Mathematisch bedingter Fehler.
■ VIEW X	X=-0,1308	Zeigt Endnäherung für x an.

Beispiel: Lokaler "flacher" Bereich. Ermitteln Sie die Nullstelle von:

$$f(x) = \begin{cases} x + 2 & \text{für } x < -1 \\ 1 & \text{für } -1 \leq x \leq 1 \quad (\text{lokaler flacher Bereich}) \\ -x + 2 & \text{für } x > 1 \end{cases}$$

Geben Sie die Funktion als Programm ein:

```
J01 LBL J
J02 1
J03 ENTER *
J04 2
J05 RCL+ X
J06 x<y?
J07 RTN
J08 4
J09 -
J10 +/-
J11 x>y?
J12 R+
J13 RTN
```

Lösen Sie X über die Anfangsnäherungen 10^{-8} und -10^{-8} .

Tastenfolge:	Anzeige	Beschreibung
■ [SOLVE/] {FN} J [E] 8 [+/-] [STO] X 1 [+/-] [E] 8 [+/-] ■ [SOLVE/] {SOLVE}	X	Über die sehr kleinen Näherungswerte nahe Null (eingrenzen des Suchbereichs) konnte keine Nullstelle gefunden werden.
◀ [R+]	1,0000E-8	Die letzten zwei Näherungen liegen weit auseinander und der Endwert von $f(x)$ ist relativ groß.
[R+]	0,0025	
[R+]	1,0000	

Bei der Verwendung größerer Anfangsnäherungen findet SOLVE die Nullstellen außerhalb des flachen Bereichs auf (bei $x=2$ und $x=-2$).

*Sie können anschließend Zeile J03 löschen, um Speicherplatz zu sparen.

Rundungsfehler und “Underflow”

Rundungsfehler. Die begrenzte (12-stellige) Genauigkeit des Rechners kann “Rundungsfehler” verursachen, welche nachteilig eine iterative Lösung beeinflussen. Die Gleichung:

$$[(|x| + 1) + 10^{15}]^2 - 10^{30} = 0$$

hat z.B. keine Nullstelle, da die linke Seite immer positiv ist. Allerdings gibt SOLVE als Ergebnis 2,0000 für die Anfangsnäherungen 1 und 2 zurück—was auf Rundungsfehler zurückzuführen ist.

Rundungsfehler können SOLVE auch dazu veranlassen, keine Nullstelle zu finden. Die Gleichung:

$$|x^2 - 7| = 0$$

hat eine Nullstelle bei $\sqrt{7}$. Allerdings kann $\sqrt{7}$ nicht genau mit 12-stelliger Genauigkeit dargestellt werden. Außerdem wechselt die Funktion nie das Vorzeichen. SOLVE gibt daher die Meldung NO ROOT FND zurück. Die Endnäherung von x (drücken Sie \blacksquare zur Anzeige) stellt jedoch die bestmögliche 12-stellige Approximation der Nullstelle dar, wenn die Routine den Lösungsprozeß abbricht.

“Underflow.” Ein *Bereichsunterlauf* kann eintreten, wenn der Betrag einer Zahl kleiner als die kleinste im Rechner darstellbare Zahl ist, wodurch eine Substitution mit Null erfolgt. Dieser Umstand kann ebenfalls die Ergebnisse von SOLVE beeinflussen. Betrachten Sie z.B. die Gleichung:

$$\frac{1}{x^2} = 0$$

Die Nullstelle dieser Gleichung ist nicht definiert. Aufgrund eines Bereichsunterlaufs gibt SOLVE einen sehr großen Wert als Nullstelle zurück.

Näheres zur Integration

Dieser Anhang enthält Informationen über die Integrationsroutine, welche über die in Kapitel 8 enthaltene Beschreibung hinausgehen.

Auswertung des Integrals

Der von der Integrationsroutine $\int_{FN} dx$ benutzte 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: 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 ungenau berechneten Funktionswerte—eine exakte Lösung finden.

Die Auswertung der Funktionswerte an unendlich vielen Stellen würde unendlich lange dauern. Dies ist jedoch nicht erforderlich, da die Genauigkeit 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 Ungenauigkeit von $f(x)$ zuläßt.

Der Integrationsalgorithmus betrachtet zuerst nur einige wenige Stützstellen und liefert ein entsprechend ungenaues Ergebnis. Sind diese Approximationen noch nicht so exakt, wie es die Genauigkeit von $f(x)$ zulassen würde, so wird der Algorithmus mit einer größeren Anzahl von Stützstellen wiederholt. Diese Iterationen werden fortgesetzt, jedesmal mit verdoppelter Stützstellenzahl, bis die resultierende Approximation die maximale Genauigkeit erreicht hat, welche die Ungenauigkeit von $f(x)$ zuläßt.

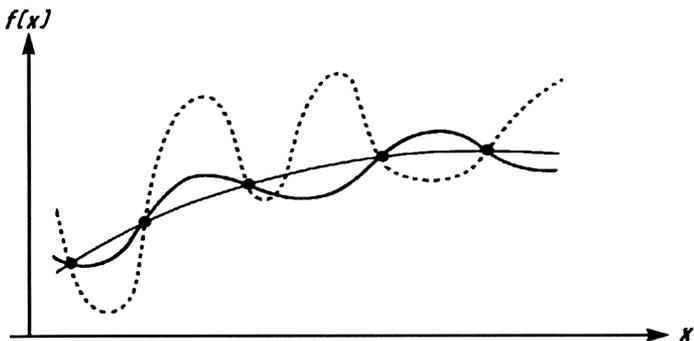
Die Ungenauigkeit der berechneten Approximation ist eine vom Anzeigeformat abgeleitete Zahl, die von der Ungenauigkeit der Funktionswerte bestimmt wird. Am Ende jeder Iteration vergleicht der Algorithmus die erhaltene Approximation mit den zwei vorhergehenden Approximationen. Ist der Unterschied zwischen einer dieser 3 Approximationen und den anderen beiden kleiner als die in der letzten Approximation tolerierte Ungenauigkeit, so wird der Algorithmus beendet; die letzte Approximation steht dabei 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 dem tatsächlichen Integral und der Approximationen—immer größer als der Unterschied zwischen den Approximationen selbst ist. Deshalb ist der Fehler der letzten Approximation kleiner als deren Fehlerabschätzung (vorausgesetzt, daß $f(x)$ nicht zu rasch variiert). 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 stellt mit ziemlicher Sicherheit die maximale Differenz dar.

Mögliche Ursachen für unkorrekte Ergebnisse

Obwohl der Integrationsalgorithmus im HP-32S einen der besten verfügbaren Algorithmen ist, kann er Ihnen in bestimmten Situationen—wie fast alle Algorithmen für numerische Integration—ein unkorrektes Ergebnis liefern. *Die Wahrscheinlichkeit dafür ist jedoch sehr gering.* Der 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.

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 Abbildung unten verdeutlicht diese Situation anhand dreier Funktionen, deren Graphen auf einem Abschnitt des Integrationsintervalls an endlich vielen Stützstellen übereinstimmen.



Mit diesen Stützstellen berechnet der Algorithmus die gleichen Approximationen für das Integral aller abgebildeten Funktionen. Die tatsächlichen Integrale der beiden mit durchgezogenen Linien dargestellten 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 anderen beiden Integralen ziemlich unterschiedlichen 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, so 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 z.B. die Approximation von

$$\int_0^{\infty} x e^{-x} dx$$

Da Sie dieses Integral numerisch auswerten, ist es naheliegend (aber dennoch irreführend, wie Sie sehen werden), die obere Integrationsgrenze als 10^{499} zu wählen—die größtmögliche Zahl, die Sie in den Rechner eingeben können. Versuchen Sie es und sehen Sie, was passiert. Geben Sie das Programm zur Auswertung der Funktion $f(x) = x e^{-x}$ ein.

F01 LBL F
 F02 RCL X
 F03 +/-
 F04 e^x
 F05 RCL× X
 F06 RTN

Stellen Sie als Anzeigeformat SCI 3 ein und spezifizieren Sie Null als untere und 10⁴⁹⁹ als obere Integrationsgrenze; starten Sie danach die Integrationsroutine.

Tastenfolge:	Anzeige	Beschreibung
■ [DISP] {SC} 3 0 [ENTER] [E] 499	1E499_	Spezifiziert die Genauigkeit und die Integrationsgrenzen.
■ [SOLVE/∫] {FN} F ■ [SOLVE/∫] {∫FN} X	∫=0,000E0	Approximation des Integrals.

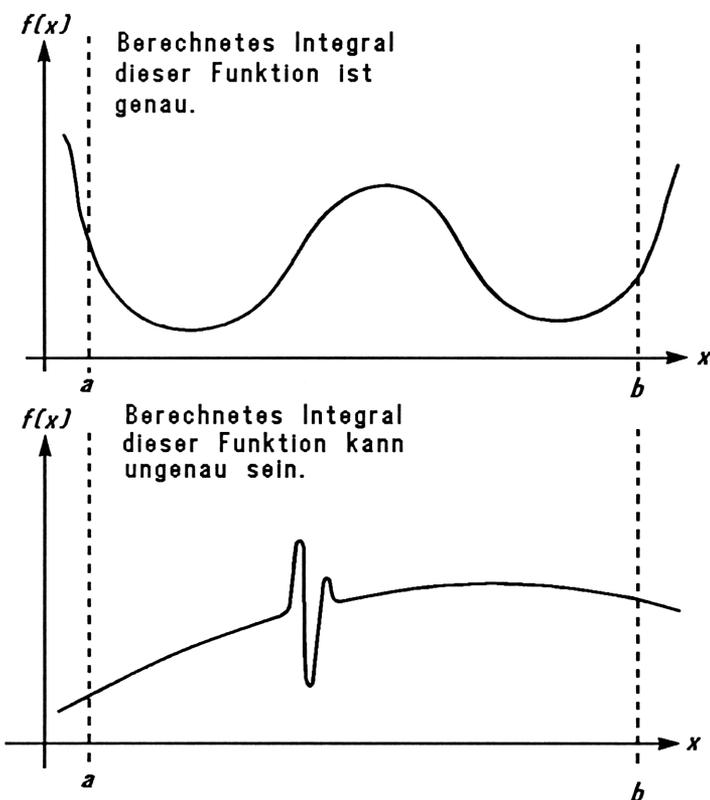
Die vom Rechner ermittelte Lösung ist mit Sicherheit unkorrekt; das Integral von $f(x) = xe^{-x}$ von 0 bis ∞ hat exakt den Wert 1. Das Problem ist aber *nicht*, daß Sie ∞ durch 10⁴⁹⁹ dargestellt haben, da das Integral dieser Funktion von 0 bis 10⁴⁹⁹ beinahe den Wert 1 hat. Der Grund für das unkorrekte Ergebnis wird offensichtlich, wenn Sie den Graphen von $f(x)$ über das Integrationsintervall betrachten:



Der Graph hat einen Spike nahe des Ursprungs. (Um $f(x)$ darzustellen, wurde die Breite des Spikes beträchtlich vergrößert. In maßstabsge- treuer Darstellung des Integrationsintervalls wäre der Spike nicht von der vertikalen Achse des Graphen unterscheidbar.) Da keine Stützstel- le den Spike des Graphen entdeckte, nahm der Algorithmus an, daß die Funktion über das ganze Integrationsintervall gleich Null sei. So- gar bei erhöhter Anzahl von Stützstellen im SCI 11 oder ALL Format würde für diese spezielle Funktion und Integrationsintervall keine der zusätzlichen Stützstellen den Spike entdecken. (Bessere Methoden zur Bewältigung derartiger Probleme werden am Ende des nächsten Abschnitts, "Bedingungen für verlängerte Rechenzeiten", behandelt.)

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 unkorrekten Ergebnissen führen könnten, werden durch die rasche Variation ihrer Werte und der Werte ihrer ersten Ableitungen charak- terisiert. Grundsätzlich gilt, je stärker die Variation der Werte der Funktion oder ihrer Ableitung, 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 Stützstel- lenwerte besser dargestellt werden, wenn diese über das ganze Inter- vall verteilt sind, als wenn sie auf einem kleinen Teil des Intervalls konzentriert sind. (Diese beiden Situationen werden in den folgenden zwei Abbildungen 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 zuver- lässiger wird dieses sein.



In vielen Fällen werden Sie die zu integrierende Funktion gut genug kennen, um zu wissen, ob die Funktion irgendwo große Schwankungen (im Vergleich zum Integrationsintervall) aufweist. Wenn Sie die Funktion nicht kennen und Sie Schwierigkeiten erwarten, so können Sie mit dem zur Berechnung der Funktionswerte geschriebenen Unterprogramm schnell ein paar Punkte berechnen und über diese eine Skizze des Funktionsverlaufs anfertigen.

Wenn Sie aus irgendeinem Grund die Zulässigkeit eines erhaltenen Näherungswertes Ihrer Integration bezweifeln, können Sie mit einer sehr einfachen Methode Ihren Verdacht überprüfen: Teilen Sie Ihr Integrationsintervall in zwei oder mehrere Abschnitte und addieren Sie die Näherungen für die Integrale über diese Abschnitte. (Die Funktion wird in diesem Fall an vollständig neuen Stützstellen berechnet.) Falls die anfängliche Approximation richtig ist, stimmt sie mit der Summe der Approximationen über die Teilintegrale überein.

Bedingungen für verlängerte Rechenzeiten

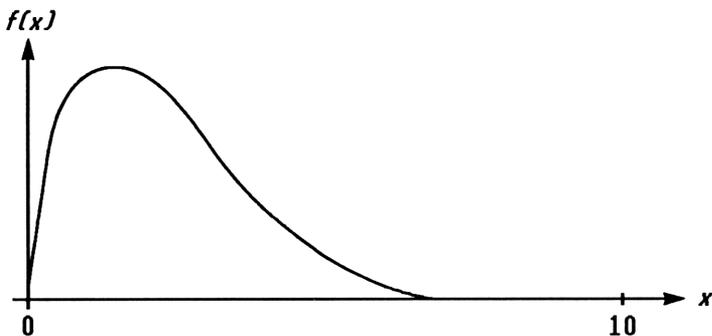
Im vorherigen Beispiel lieferte der Algorithmus ein unkorrektes Ergebnis, weil er den Spike der Funktion gar nicht entdeckte. Dies geschah deshalb, weil die Varianten der Funktion im Vergleich zur Intervallbreite zu schnell war. Bei kleinerer Intervallbreite könnten Sie die richtige Lösung erhalten, aber das würde sehr lange dauern, falls die Intervallbreite noch immer zu groß ist.

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 $f(x)$ im Fall $f(x) = xe^{-x}$ 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^{499} ersetzen—zum Beispiel 10^3 .

Starten Sie nochmals die vorherige Integration mit den neuen Integrationsgrenzen. Wenn Sie zwischenzeitlich keine andere Integration ausgeführt haben, brauchen Sie $FN=$ F nicht erneut spezifizieren.

Tastenfolge:	Anzeige	Beschreibung
0 ENTER E 3	1E3_	Neue Obergrenze.
SOLVE/∫ {∫FN} X	∫=1,000E0	Integral. (Die Berechnung dauert eine kleine Weile.)
xzy	1,824E-4	Fehlerabschätzung der Approximation.

Dies ist das richtige Ergebnis, aber die Berechnung dauert sehr lange. Um dies zu verstehen, vergleichen Sie den Graphen der Funktion über das Integrationsintervall von $x = 0$ und $x = 10^3$, welcher dem auf Seite 276 abgebildeten sehr ähnelt, mit dem Graphen der Funktion zwischen $x = 0$ und $x = 10$:



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 stetig und glatt in einer vorhersehbaren Weise abfällt.

Der Algorithmus erhöht die Dichte der Stützpunkte, bis der Unterschied zwischen aufeinanderfolgenden Approximationen genügend klein wird. Für ein kleines Intervall im interessanten Funktionsbereich wird weniger Zeit benötigt, um diese kritische Dichte zu erreichen.

Um die gleiche Dichte von Stützstellen zu erhalten, ist die gesamte Anzahl der erforderlichen Stützstellen über das große Intervall viel größer als die erforderliche Anzahl für das kleine Intervall. Demzufolge sind im 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 vorwiegend 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 Verfahren können Sie die zu integrierende Funktion oder die Integrationsgrenzen so verändern, daß die Funktion über das/die Integrationsintervall(e) leichter integrierbar wird.

Meldungen

Der Rechner gibt unter bestimmten Bedingungen eine Meldung aus. Außerdem erscheint der ▲ Indikator in der Anzeige, um Sie auf die besondere Situation hinzuweisen. Bei wichtigen Meldungen bleibt die Meldung angezeigt, bis Sie von Ihnen explizit gelöscht wird. Durch Drücken von [C] oder [◀] können Sie eine Meldung löschen—das Drücken einer anderen Taste bewirkt das Löschen *und* die Ausführung der Tastenfunktion.

JFN ACTIVE

Ein gestartetes Programm versuchte, ein Programm-Label (FN=Label) zu wählen, während die Berechnung eines Integrals abläuft.

J<JFN>

Ein gestartetes Programm versuchte die Berechnung eines Integrals (JFN ≠ Variable), während eine andere Berechnung eines Integrals abläuft.

J<SOLVE>

Ein gestartetes Programm versuchte die Ausführung einer SOLVE Operation, während die Berechnung eines Integrals abläuft.

ALL VARS=0

Der Variablenkatalog (■ [MEM] {VAR}) enthält keine gespeicherten Daten.

CALCULATING

Der Rechner führt eine Funktion aus, welche längere Zeit in Anspruch nehmen kann.

DIVIDE BY 0

Versuch, durch Null zu dividieren. (Einschließlich [%CHG], falls das Y-Register Null enthält.)

DUPLICAT. LBL

Versuch, ein Programm-Label zu speichern, welches bereits für eine andere Routine verwendet wurde.

INTEGRATING

Der Rechner berechnet ein Integral. Dies *kann* etwas länger dauern.

INVALID DATA

Fehlerhafte Daten:

- Versuch, Kombinationen oder Permutationen mit $r > n$, mit gebrochenem r oder n oder mit $n \geq 10^{12}$ zu berechnen.
- Versuch, eine trigonometrische oder hyperbolische Funktion mit einem unzulässigen Argument zu berechnen: $\boxed{\text{TAN}}$ mit x als ungerades Mehrfaches von 90° ; $\boxed{\text{ACOS}}$ oder $\boxed{\text{ASIN}}$ mit $x < -1$ oder $x > 1$; $\boxed{\text{HYP}} \boxed{\text{ATAN}}$ mit $x \leq -1$ oder $x \geq 1$; $\boxed{\text{HYP}} \boxed{\text{ACOS}}$ mit $x < 1$.

INVALID $\times!$

Versuch, Fakultät oder Gammafunktion mit x als negativer ganzer Zahl zu berechnen.

INVALID \succ^{\times}

Unzulässige Anwendung der Potenzfunktion:

- Versuch, 0 zur 0. oder einer negativen Potenz zu erheben.
- Versuch, eine negative Zahl mit einer gebrochenen Zahl zur Potenz zu erheben.
- Versuch, die komplexe Zahl $(0 + i0)$ zu einer Zahl mit einem negativen Realteil zu erheben.

INVALID $\langle i \rangle$

Versuch, eine Operation über indirekte Adressierung auszuführen, wobei die Zahl im Indexregister unzulässig ist ($|i| \geq 27$ oder $0 \leq |i| < 1$).

LOG $\langle \emptyset \rangle$

Versuch, den Logarithmus von Null oder $(0 + i0)$ zu berechnen.

LOG $\langle \text{NEG} \rangle$

Versuch, den Logarithmus von einer negativen Zahl zu berechnen.

MEMORY CLEAR

Der gesamte Speicherbereich wurde gelöscht (siehe Seite 255).

RUNNING

Der Rechner führt gerade ein Programm aus (anderes als SOLVE oder \int FN Routine).

SELECT FN

Versuch, SOLVE *Variable* oder \int FN \square *Variable* ohne spezifiziertes Programm-Label auszuführen. Dies kann nur bei der ersten Verwendung von SOLVE oder \int FN eintreten, nachdem MEMORY CLEAR angezeigt wurde, oder wenn das momentane Label nicht mehr existiert.

SOLVE ACTIVE

Ein gestartetes Programm versuchte, ein Programm-Label (FN=*Label*) zu wählen, während eine SOLVE Operation ausgeführt wurde.

SOLVE(SOLVE)

Ein gestartetes Programm versuchte SOLVE aufzurufen, während eine andere SOLVE Operation ausgeführt wurde.

SOLVE(\int FN)

Ein gestartetes Programm versuchte die Berechnung eines Integrals, während eine SOLVE Operation ausgeführt wurde.

SOLVING

Der Rechner berechnet die Nullstelle/Lösung einer Gleichung. Dies *kann* etwas länger dauern.

SQRT(NEG)

Versuch, die Quadratwurzel einer negativen Zahl zu berechnen.

STAT ERROR

Statistikfehler:

- Versuch, s_x , s_y , \hat{x} , \hat{y} , m , r oder b mit $n = 1$ zu berechnen.
- Versuch, r , \hat{x} oder $\bar{x}w$ nur mit x -Daten zu berechnen (alle y -Werte gleich Null).
- Versuch, \hat{x} , \hat{y} , r , m oder b nur mit gleichen x -Werten zu berechnen.
- Versuch, eine Statistikberechnung auszuführen, nachdem n durch $\boxed{\Sigma-}$ auf 0 reduziert wurde.

TOO BIG

Der Betrag einer Zahl ist zu groß, um in HEX, OCT oder BIN Werte konvertiert werden zu können. Die Zahl muß im Bereich $-34\ 359\ 738\ 368 \leq n \leq 34\ 359\ 738\ 367$ liegen.

XEQ OVERFLOW

Ein gestartetes Programm versuchte, ein achttes verschachteltes XEQ *Label* auszuführen. (Es können bis zu sieben verschachtelte Unterprogramme definiert werden.) Da SOLVE und JFN jeweils eine Ebene verwenden, kann dieser Fehler auch von diesen Operationen verursacht werden.

Funktionsindex

Dieser Abschnitt stellt eine Kurzanleitung für alle Funktionen und Operationen dar. Die Auflistung erfolgt in alphabetischer Reihenfolge (sofern ein eindeutiger Funktionsname existiert). Dieser Name entspricht demjenigen, welcher in Programmzeilen verwendet wird. Beispielsweise wird die Funktion $\text{FIX } n$ als $\blacksquare \text{ [DISP] } \{F\}n$ ausgeführt.

Funktionen, die nicht programmierbar sind, haben ihren Namen in Form einer Taste, wie z.B. \blacktriangleleft .

Funktionen, die nicht mit einem Alphazeichen beginnen, erscheinen am Anfang der Auflistung; ist den Funktionsnamen ein Pfeil vorangestellt (z.B. $\rightarrow\text{DEG}$), so wird dieser für die Sortierung ignoriert.

Funktion	Tasten und Beschreibung	Seite
+/-	$\boxed{+/-}$ Vorzeichenwechsel einer Zahl.	21
+	$\boxed{+}$ <i>Addition.</i> Gibt $y + x$ zurück.	25
-	$\boxed{-}$ <i>Subtraktion.</i> Gibt $y - x$ zurück.	25
×	$\boxed{\times}$ <i>Multiplikation.</i> Gibt $y \times x$ zurück.	25
÷	$\boxed{\div}$ <i>Division.</i> Gibt $y \div x$ zurück.	25
\blacktriangleleft	Löscht das zuletzt eingetippte Zeichen; löscht x ; löscht ein Menü; löscht einen Programmschritt.	16, 19, 32, 83
$\blacksquare \blacktriangleup$	Zeigt vorherigen Katalogeintrag an; verschiebt den Programmzeiger an vorhergehende Zeile.	33, 76

Funktion	Tasten und Beschreibung	Seite
▼	Zeigt den nächsten Katalogeintrag an; verschiebt den Programmzeiger an die nächste Programmzeile (bei Programmeingabe); führt die momentane Programmzeile aus (außerhalb von Programmeingabe-Modus).	33, 76
1/x	<input type="button" value="1/x"/> <i>Kehrwert.</i>	24
10 ^x	<input type="button" value="10<sup>x</sup"/> <i>Dekadische Exponentialfunktion.</i> Gibt 10, potenziert mit x , als Ergebnis zurück.	55
%	<input type="button" value="%"/> <i>Prozent.</i> Gibt $(y \times x) \div 100$ zurück.	59
%CHG	<input type="button" value="%CHG"/> <i>Prozentuale Änderung.</i> Gibt $(x - y)(100 \div y)$ zurück.	59
π	<input type="button" value="π"/> Gibt die Approximation 3,14159265359 zurück.	56
$\Sigma +$	<input type="button" value="Σ+"/> Akkumuliert (y, x) in die Statistikregister.	154
$\Sigma -$	<input type="button" value="Σ-"/> Löscht (y, x) aus den Statistikregistern.	155
Σx	<input type="button" value="STAT"/> {Σ} {x} Gibt die Summe der x -Werte zurück.	161
Σx^2	<input type="button" value="STAT"/> {Σ} {x ² } Gibt die Summe der quadrierten x -Werte zurück.	162
Σxy	<input type="button" value="STAT"/> {Σ} {xy} Gibt die Summe der Produkte der x - und y -Werte zurück.	162
Σy	<input type="button" value="STAT"/> {Σ} {y} Gibt die Summe der y -Werte zurück.	161
Σy^2	<input type="button" value="STAT"/> {Σ} {y ² } Gibt die Summe der quadrierten y -Werte zurück.	162

Funktion	Tasten und Beschreibung	Seite
$\theta, r \rightarrow y, x$	[P\leftrightarrowRECT] { $\theta, r \rightarrow y, x$ } <i>Polar- nach Rechtecksnotation.</i> Konvertiert (r, θ) in (x, y) .	61
\int FN d Variable	[SOLVE/∫] { \int FN} Variable Integriert die momentane Funktion über die Variable, wobei die untere Integrationsgrenze im Y-Register und die obere Grenze im X-Register gespeichert ist.	127
ABS	[PARTS] {ABS} <i>Absolutbetrag.</i> Gibt $ x $ zurück.	67
ACOS	[ACOS] <i>Arcuscosinus.</i> Gibt $\cos^{-1} x$ zurück.	57
ACOSH	[HYP] [ACOS] <i>Arcuscosinus hypberbolicus.</i> Gibt $\cosh^{-1} x$ zurück.	59
ALL	[DISP] {ALL} Spezifiziert die Anzeige aller signifikanten Stellen als Anzeigeformat.	30
ASIN	[ASIN] <i>Arcussinus.</i> Gibt $\sin^{-1} x$ zurück.	57
ASINH	[HYP] [ASIN] <i>Arcussinus hypberbolicus.</i> Gibt $\sinh^{-1} x$ zurück.	59
ATAN	[ATAN] <i>Arcustangens.</i> Gibt $\tan^{-1} x$ zurück.	57
ATANH	[HYP] [ATAN] <i>Arcustangens hypberbolicus.</i> Gibt $\tanh^{-1} x$ zurück.	59
b	[STAT] {L.R.} {b} Gibt den <i>y-Achsen Schnittpunkt</i> der Regressionsgeraden zurück: $\bar{y} - m\bar{x}$.	159
[BASE]	Zeigt Menü für Konvertierung in andere Zahlenbasis an.	144
BIN	[BASE] {BN} Wählt Binär-Modus (Basis 2).	144
[C]	Schaltet Rechner ein; löscht x ; löscht Meldungen und Eingabeaufforderungen; hebt Menüs, Katalog und Programmeingabemodus auf; unterbricht Programmablauf.	14, 16, 19, 32, 36, 73

Funktion	Tasten und Beschreibung	Seite
CF n	<p>■ FLAGS {CF} n <i>Löscht (Clears) Flag n ($0 \leq n \leq 6$).</i></p>	98
■ CLEAR	<p>Zeigt das Menü zum Löschen von Zahlen oder Speicherbereichsteilen an; oder löscht Variable oder Programm aus dem MEM Katalog.</p>	16, 33
■ CLEAR {ALL}	<p>Löscht alle gespeicherten Daten und Programme.</p>	34
■ CLEAR {PGM}	<p>Löscht alle Programme.</p>	86
CL Σ	<p>■ CLEAR {Σ} <i>Löscht Statistikregister.</i></p>	154
CLVARS	<p>■ CLEAR {VARS} <i>Setzt alle Variablen auf Null.</i></p>	50
CLx	<p>■ CLEAR {∞} <i>Setzt x auf Null.</i></p>	36, 40, 73
■ CMPLX	<p>Zeigt den CMPLX_ Vorsatz für komplexe Funktionen.</p>	139
CMPLX + / -	<p>■ CMPLX +/- <i>Komplexer Vorzeichenwechsel. Gibt $-(z_x + iz_y)$ zurück.</i></p>	139
CMPLX +	<p>■ CMPLX + <i>Komplexe Addition. Gibt $(z_{1x} + iz_{1y}) + (z_{2x} + iz_{2y})$ zurück.</i></p>	140
CMPLX -	<p>■ CMPLX - <i>Komplexe Subtraktion. Gibt $(z_{1x} + iz_{1y}) - (z_{2x} + iz_{2y})$ zurück.</i></p>	140
CMPLX \times	<p>■ CMPLX x <i>Komplexe Multiplikation. Gibt $(z_{1x} + iz_{1y}) \times (z_{2x} + iz_{2y})$ zurück.</i></p>	140

Funktion	Tasten und Beschreibung	Seite
CMPLX ÷	<p>■ CMPLX ÷</p> <p><i>Komplexe Division.</i> Gibt $(z_{1x} + iz_{1y}) \div (z_{2x} + iz_{2y})$ zurück.</p>	140
CMPLX1/x	<p>■ CMPLX 1/x</p> <p><i>Komplexer Kehrwert.</i> Gibt $1/(z_x + iz_y)$ zurück.</p>	139
CMPLXCOS	<p>■ CMPLX COS</p> <p><i>Komplexer Cosinus.</i> Gibt $\cos(z_x + iz_y)$ zurück.</p>	139
CMPLX e ^x	<p>■ CMPLX e^x</p> <p><i>Komplexe natürliche Exponentialfunktion.</i> Gibt $e^{z_x + iz_y}$ zurück.</p>	139
CMPLXLN	<p>■ CMPLX LN</p> <p><i>Komplexer natürlicher Logarithmus.</i> Gibt $\log_e(z_x + iz_y)$ zurück.</p>	139
CMPLXSIN	<p>■ CMPLX SIN</p> <p><i>Komplexer Sinus.</i> Gibt $\sin(z_x + iz_y)$ zurück.</p>	139
CMPLXTAN	<p>■ CMPLX TAN</p> <p><i>Komplexer Tangens.</i> Gibt $\tan(z_x + iz_y)$ zurück.</p>	139
CMPLX y ^x	<p>■ CMPLX y^x</p> <p><i>Komplexe Potenzfunktion.</i> Berechnet $(z_{1x} + iz_{1y})^{(z_{2x} + iz_{2y})}$</p>	140
Cn,r	<p>■ PROB {C,n,r }</p> <p><i>Kombinationen von n Elementen, aus welchen jeweils r Elemente gewählt werden.</i> Gibt $n! \div (r!(n - r)!)$ zurück.</p>	65
COS	<p>COS</p> <p><i>Cosinus.</i> Gibt $\cos x$ zurück.</p>	57
COSH	<p>■ HYP COS</p> <p><i>Cosinus hypberbolicus.</i> Gibt $\cosh x$ zurück.</p>	59
DEC	<p>■ BASE {DEC }</p> <p>Wählt Dezimal-Modus (Basis 10).</p>	144
→DEG	<p>■ D↔RAD {→DEG }</p> <p><i>Radiant nach Grad (Altgrad).</i> Gibt $(360/2\pi)x$ zurück.</p>	64

Funktion	Tasten und Beschreibung	Seite
DEG	■ MODE {DG} Spezifiziert Grad (Altgrad) als Winkelmodus.	57
■ DISP	Zeigt Menü zum Einstellen des Anzeigeformats an.	30
■ D↔RAD	Zeigt Menü zum Konvertieren zwischen Grad und Radianten/Bogenmaß an.	64
DSE Variable	■ LOOP {DSE} Variable <i>Decrement; Skip if Equal or less.</i> Für die in einer Variablen gespeicherte Schleifensteuerzahl <i>cccccc.ffff</i> . Subtrahiert <i>ii</i> (Schrittweite) von <i>cccccc</i> (Schleifenzähler), und, falls Ergebnis \leq <i>fff</i> (Endwert), überspringt nächste Programmzeile.	101
[E]	Beginnt Eingabe von Exponent und fügt "E" vor dem Exponenten ein. Kennzeichnet, daß 10-er Potenz folgt.	22
e^x	[e ^x] <i>Natürliche Exponentialfunktion.</i> Gibt e , potenziert mit x , zurück.	55
ENG n	■ DISP {EN} n Spezifiziert technisches Anzeigeformat mit n Stellen, die der ersten Ziffer folgen. $0 \leq n \leq 11$.	30
ENTER	[ENTER] Trennt zwei nacheinander eingetippte Zahlen; kopiert x in Y-Register, schiebt y in Z-Register, schiebt z in T-Register, wobei dessen alter Inhalt verloren geht.	23, 39
FIX n	■ DISP {FX} n Spezifiziert Festkommaformat mit n Dezimalstellen. $0 \leq n \leq 11$.	30
■ FLAGS	Zeigt Menü zum Setzen, Löschen und Abfragen von Flags an.	98
FN = Label	■ SOLVE/ {FN} Label Wählt das mit Label benannte Programm als momentane Funktion (von SOLVE und FN benutzt).	111, 127
FP	■ PARTS {FP} Gebrochener Teil von x .	67

Funktion	Tasten und Beschreibung	Seite
FS? <i>n</i>	<p>■ [FLAGS] {FS?} <i>n</i> Wenn Flag <i>n</i> ($0 \leq n \leq 6$) gesetzt ist, Ausführen der nächsten Programmzeile; wenn <i>n</i> gelöscht ist, Überspringen der nächsten Programmzeile.</p>	98
GRAD	<p>■ [MODES] {GR} Spezifiziert Neugrad als Winkelmodus.</p>	57
GTO <i>Label</i>	<p>■ [GTO] <i>Label</i> Stellt Programmzeiger im Programmspeicher auf Programm-Label.</p>	93, 100
■ [GTO] [.] <i>Label nn</i>	<p>Stellt Programmzeiger auf Programmzeile <i>Label nn</i>.</p>	94
■ [GTO] [.] [.]	<p>Stellt Programmzeiger auf PRGM TOP.</p>	94
HEX	<p>■ [BASE] {HX} Wählt Hexadezimal-Modus (Basis 16).</p>	144
■ [HYP]	<p>Zeigt HYP_ Vorsatz für hyperbolische Funktionen an.</p>	59
■ [H↔HMS]	<p>Zeigt Menü zur Konvertierung zwischen Dezimalgrad/-stunden und Stunden-Minuten-Sekunden an.</p>	63
→HMS	<p>■ [H↔HMS] {→HMS} Dezimalstunden nach Stunden-Minuten-Sekunden. Konvertiert <i>x</i> von Dezimalbruch nach Minuten-Sekunden Format.</p>	64
→HR	<p>■ [H↔HMS] {→HR} Stunden-Minuten-Sekunden nach Stunden. Konvertiert <i>x</i> von Minuten-Sekunden Format in Dezimalbruch.</p>	64
(i)	<p>Parameter für indirekte Adressierung. Adressiert Variable oder Label, dessen Buchstabe zu dem numerischen Wert in Variable <i>i</i> korrespondiert.</p>	103
INPUT <i>Variable</i>	<p>■ [INPUT] <i>Variable</i> Ruft Variableninhalt in X-Register zurück, zeigt Variablenname zusammen mit Inhalt von X-Register an und unterbricht Programmausführung; Drücken von [R/S] (oder [▼]) speichert Zahl in Variable. (Nur in Programmen verwendet.)</p>	77
IP	<p>■ [PARTS] {IP} Ganzzahliger Teil von <i>x</i>.</p>	67

Funktion	Tasten und Beschreibung	Seite
ISG Variable	<p>■ LOOP { I S G } Variable <i>Increment, Skip if Greater.</i> Für die in Variable gespeicherte Schleifensteuerzahl <i>ccccccc,ffii</i>. Addiert <i>ii</i> (Schrittweite) von <i>ccccccc</i> (Schleifenzähler). und, falls Ergebnis > <i>fff</i> (Endwert), überspringt die nächste Programmzeile.</p>	101
LASTx	<p>■ LASTx Ruft in LAST X gespeicherte Zahl zurück.</p>	41
LBL Buchstabe	<p>■ LBL/RTN { L B L } Label Benennt ein Programm mit einzelmem Buchstaben zur Verwendung mit XEQ, GTO oder FN Operationen. (Nur in Programmen verwendet.)</p>	71
■ LBL/RTN	<p>Zeigt Menü für LBL, RTN und PSE an.</p>	71
LN	<p>■ LN <i>Natürlicher Logarithmus.</i> Gibt $\ln x$ zurück.</p>	55
LOG	<p>■ LOG <i>Dekadischer Logarithmus.</i> Gibt $\log x$ zurück.</p>	55
■ LOOP	<p>Zeigt Menü für DSE und ISG an.</p>	99
{L.R.}	<p>■ STAT { L.R. } Zeigt Menü für lineare Regression an.</p>	158
m	<p>■ STAT { L.R. } { m } Gibt Steigung der Regressionsgeraden zurück: $[\sum(x_i - \bar{x})(y_i - \bar{y})] \div \sum(x_i - \bar{x})^2$.</p>	159
■ MEM	<p>Zeigt die Größe des freien Speicherbereichs und das Katalog-Menü an.</p>	33
■ MEM { PGM }	<p>Startet Auflistung von Programmkatalog.</p>	85
■ MEM { VAR }	<p>Startet Auflistung von Variablenkatalog.</p>	49
■ MODES	<p>Zeigt Menü zum Spezifizieren des Winkelmodus und des Dezimalzeichens (". " oder ",").</p>	29
n	<p>■ STAT { Σ } { n } Gibt die Anzahl der Sätze von Datenpunkten zurück.</p>	161
OCT	<p>■ BASE { OC } Wählt Oktal-Modus (Basis 8).</p>	144
■ OFF	<p>Schaltet Rechner aus.</p>	14

Funktion	Tasten und Beschreibung	Seite
P↔RECT	Zeigt Menü zum Konvertieren zwischen Polar- und Rechteckskordinaten an.	60
PARTS	Zeigt Menü zum Wählen von Teilen von Zahlen.	67
Pn,r	PROB {Pn,r} Permutationen von n Elementen, aus welchen jeweils r Elemente gewählt werden. Gibt $n! \div (n - r)!$ zurück.	65
PRGM	Aktiviert oder hebt Programmeingabe auf (Umschaltfunktion).	72
PROB	Zeigt Menü für Wahrscheinlichkeitsfunktionen an.	65
PSE	LBL/RTN {PSE} Pause. Hält Programm kurz zur Anzeige von x an, setzt danach Ausführung fort. (Nur in Programmen verwendet.)	82
r	STAT {L.R.} {r} Gibt den Korrelationskoeffizienten für x- und y-Werte zurück: $\frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \times \sum(y_i - \bar{y})^2}}$	159
→RAD	D↔RAD {→RAD} Grad nach Radianen/Bogenmaß. Gibt $(2\pi/360)x$ zurück.	64
{R}	PROB {R} Zeigt Menü für Zufallszahl an.	65
RAD	MODES {RD} Spezifiziert Radianen als Winkelmodus.	57
RADIX,	MODES { , } Spezifiziert Komma als Dezimalzeichen.	29
RADIX.	MODES { . } Spezifiziert Punkt als Dezimalzeichen.	29
RANDOM	PROB {R} {RANDOM} Gibt eine Zufallszahl im Bereich $0 \leq x < 1$ zurück.	65
RCL Variable	RCL Variable ReCall. Kopiert Variable in X-Register.	48
RCL+ Variable	RCL + Variable Gibt $x + Variable$ zurück.	51
RCL- Variable.	RCL - Variable. Gibt $x - Variable$ zurück.	51

Funktion	Tasten und Beschreibung	Seite
RCL \times Variable.	<input type="button" value="RCL"/> <input type="button" value="x"/> Variable. Gibt $x \times$ Variable zurück.	51
RCL \div Variable.	<input type="button" value="RCL"/> <input type="button" value="÷"/> Variable. Gibt $x \div$ Variable zurück.	51
RND	<input type="button" value="PARTS"/> {R/N} <i>Rundet x auf n Dezimalstellen (im FIX n Anzeigeformat) oder auf $n + 1$ signifikante Stellen (im SCI n oder ENG n Anzeigeformat).</i>	67
RTN	<input type="button" value="LBL/RTN"/> {RTN} <i>Return. Kennzeichnet das Ende eines Programms; der Programmzeiger kehrt an den Anfang des Programmspeichers oder zur aufrufenden Routine zurück.</i>	71, 91
<input type="button" value="R/S"/>	<i>Run/Stop. Beginnt Programmausführung bei momentaner Programmzeile oder hält Programm an.</i>	82
R↓	<input type="button" value="R↓"/> <i>Rollen nach unten. Verschiebt t in Z-Register, z in Y-Register, y in X-Register und x in T-Register.</i>	36
SCI n	<input type="button" value="DISP"/> {SC} n <i>Spezifiziert wissenschaftliches Anzeigeformat mit n Dezimalstellen, $0 \leq n \leq 11$.</i>	30
SEED	<input type="button" value="PROB"/> {R} {SEED} <i>Neustart der Zufallszahlenfolge mit Startwert x.</i>	65
SF n	<input type="button" value="FLAGS"/> {SF} n <i>Setzt Flag n ($0 \leq n \leq 6$), kennzeichnend für "wahr".</i>	98
<input type="button" value="SHOW"/>	<i>Zeigt ganze Mantisse (alle 12 Stellen) von x (oder der Zahl in momentaner Programmzeile).</i>	31
SIN	<input type="button" value="SIN"/> <i>Sinus. Gibt $\sin x$ zurück.</i>	57
SINH	<input type="button" value="HYP"/> <input type="button" value="SIN"/> <i>Sinus hyperbolicus. Gibt $\sinh x$ zurück.</i>	59
<input type="button" value="SOLVE/f"/>	<i>Zeigt Menü zum Lösen einer Unbekannten und für Integration an.</i>	111, 127

Funktion	Tasten und Beschreibung	Seite
SOLVE Variable	[SOLVE/] {SOLVE} Variable Löst momentane Funktion für Variable unter Verwendung der Anfangsnäherungen in Variable und x.	111
SQRT	[√x] Quadratwurzel von x.	24
[STAT]	Zeigt Menü für Statistikfunktionen an.	156
STO Variable	[STO] Variable STOre. Kopiert (bzw. speichert) x in Variable.	48
STO+ Variable	[STO] [+] Variable Speichert Variable + x in Variable.	50
STO- Variable	[STO] [-] Variable Speichert Variable - x in Variable.	50
STO× Variable	[STO] [x] Variable Speichert Variable × x in Variable.	50
STO÷ Variable	[STO] [÷] Variable Speichert Variable ÷ x in Variable.	50
STOP	[R/S] Hält Programmausführung an und zeigt X-Register an.	82
sx	[STAT] {s} {sx} Gibt die Standardabweichung der x-Werte zurück: $\sqrt{\sum(x_i - \bar{x})^2 \div (n - 1)}$	157
sy	[STAT] {s} {sy} Gibt die Standardabweichung der y-Werte zurück: $\sqrt{\sum(y_i - \bar{y})^2 \div (n - 1)}$	157
TAN	[TAN] Tangens. Gibt tan x zurück.	57
TANH	[HYP] [TAN] Tangens hyperbolicus. Gibt tanh x zurück.	59
[TESTS]	Zeigt Menü zum Testen von Bedingungen an.	96
VIEW Variable	[VIEW] Variable Zeigt benannten Inhalt von Variable an, ohne Wert in den Stack zurückzurufen.	79

Funktion	Tasten und Beschreibung	Seite
XEQ <i>Label</i>	<p>XEQ <i>Label</i> Führt das durch <i>Label</i> gekennzeichnete Programm aus.</p>	75, 91
x^2	<p>x² Quadrat von x.</p>	24
\bar{x}	<p>STAT {\bar{x}, \bar{y}} {\bar{x}} Gibt Mittelwert der x Werte zurück: $\Sigma x_i \div n$.</p>	156
\hat{x}	<p>STAT {L,R,} {\hat{x}} Bei gegebenem y-Wert im X-Register: Gibt x-Schätzung/Näherung, basierend auf Regressionsgerade $\hat{x} = (y - b) \div m$, zurück.</p>	159
$x!$	<p>PROB {$x!$} Fakultät (oder Gammafunktion). Gibt $(x)(x - 1) \dots (2)(1)$, oder $\Gamma(x+1)$ zurück.</p>	65
\bar{x}_w	<p>STAT {\bar{x}, \bar{y}} {\bar{x}_w} Gibt <i>gewogenes Mittel</i> der x Werte zurück: $(\Sigma y_i x_i) \div \Sigma y_i$.</p>	157
$x \leftrightarrow y$	<p>x\leftrightarrowy x vertauschen mit y. Verschiebt x in Y-Register und y in X-Register.</p>	37
$x < 0?$	<p>TESTS {$x?0$} {<0} Wenn $x < 0$, Ausführen der nächsten Programmzeile; wenn $x \geq 0$, Überspringen der nächsten Programmzeile.</p>	96
$x < y?$	<p>TESTS {$x?y$} {$<y$} Wenn $x < y$, Ausführen der nächsten Programmzeile; wenn $x \geq y$, Überspringen der nächsten Programmzeile.</p>	96
$x = 0?$	<p>TESTS {$x?0$} {$=0$} Wenn $x = 0$, Ausführen der nächsten Programmzeile; wenn $x \neq 0$, Überspringen der nächsten Programmzeile.</p>	96
$x = y?$	<p>TESTS {$x?y$} {$=y$} Wenn $x = y$, Ausführen der nächsten Programmzeile; wenn $x \neq y$, Überspringen der nächsten Programmzeile.</p>	96
$x > 0?$	<p>TESTS {$x?0$} {>0} Wenn $x > 0$, Ausführen der nächsten Programmzeile; wenn $x \leq 0$, Überspringen der nächsten Programmzeile.</p>	96

Funktion	Tasten und Beschreibung	Seite
$x > y?$	<p>■ TESTS {$x?y$} {$>y$}</p> <p>Wenn $x > y$, Ausführen der nächsten Programmzeile; wenn $x \leq y$, Überspringen der nächsten Programmzeile.</p>	96
$x \neq 0?$	<p>■ TESTS {$x?0$} {$\neq 0$}</p> <p>Wenn $x \neq 0$, Ausführen der nächsten Programmzeile; wenn $x = 0$, Überspringen der nächsten Programmzeile.</p>	96
$x \neq y?$	<p>■ TESTS {$x?y$} {$\neq y$}</p> <p>Wenn $x \neq y$, Ausführen der nächsten Programmzeile; wenn $x = y$, Überspringen der nächsten Programmzeile.</p>	96
\bar{y}	<p>■ STAT {\bar{x}, \bar{y}} {\bar{y}}</p> <p>Gibt den <i>Mittelwert</i> der y-Werte zurück: $\Sigma y_i \div n$.</p>	156
\hat{y}	<p>■ STAT {L.R.} {\hat{y}}</p> <p>Bei gegebenem x-Wert im X-Register: Gibt den y-<i>Schätzwert/Näherung</i>, basierend auf der Regressionsgeraden $\hat{y} = mx + b$, zurück.</p>	159
$y, x \rightarrow \theta, r$	<p>■ P↔RECT {$y, x \rightarrow \theta, r$}</p> <p><i>Rechteck- nach Polarnotation</i>. Konvertiert (x, y) in (r, θ).</p>	61
y^x	<p>y^x</p> <p>Potenzfunktion. Gibt y, potenziert mit x, zurück.</p>	56

Index

Fettgedruckte Seitenangaben kennzeichnen einen primären Eintrag, Angaben in normaler Druckweise kennzeichnen einen sekundären Eintrag. *Eine alphabetische Auflistung der Funktionen finden Sie im Funktionsindex, der diesem Index vorangeht.*

Sonderzeichen

, 15

, 32

, 242

, 17

, 150

, 49

←. *Siehe* Rückschritt-Taste

π , 56, 57

0 1 2 3, 98

A..Z, 15, 48

(i), 103–106

Funktionen, Verwendung in, **104**
zur Programmsteuerung, **105**

A

Abbilden von SOLVE Funktionen,
123

Abhängige Variable, **154**

Absolutbetrag, **67**

Adresse, indirekt, **103–106**

ALL Format, **31**

Andauernde Datenspeicherung, **14**,
243

Anfangsnäherungen, (SOLVE), **111**,
118, **120**

Auswählen, **123**

Vorkommen von, **120**

Anzeige

des Stacks, **36**, **40**

Format für Integration, **127**

Format, **29–31**

Kontrast, einstellen, **14**

temporäre, **31**

Anzeigen von Zahlen in Programm,
79–80

Arcuscosinus, **57**

Arcussinus, **57**

Arcustangens, **57**

Arithmetik, **24–29**, **38–46**

im Stack, **38**

komplexe, **139–140**

mit gespeicherten Variablen, **50–52**
nichtdezimal. *Siehe* Basis-

Vektor, **164–175**

Aufforderung für Variable, **77**, **79**

Aus, **14**

Ausgabe, Programm, **78**

Austauschen von Zahlen (X- und Y-
Register), **25**, **37**

B

Barwerte, positive und negative, **223**

Basis

Arithmetik, **146–148**

Konvertierungen, **144–145**

Modi, Programmierung, **151–152**

Batterien,
 Beschädigung durch, **248**
 Einsetzen, **243-245**
 Typen, **242**
Bedingte Abfragen, **95-99**
Bedingte Funktionen, **95-99**, 100
 SOLVE, **124**
 fFN, **134**
Beschädigung, **250**
Besselsche Funktion, **128-130**
Bestimmtes Integral, **126**
Betrag, **24**
Binärzahlen, **144-150**
 große, **49**
 lange, **149**
 positive, **148**
Bit, hochwertigstes, **148**
Blockieren der Rechnerfunktion,
 245-246
Bogenmaß-Modus, **57**
Bruchteil, **67**
 nichtdezimale Arithmetik, **146**
Bruchteile, konvertieren, **63-64**
Buchstabentasten, **15**, **48**, **71**
Bytes in Programmen, **85**

C

C. *Siehe* Löschtaste
Celsius Konvertierungen, **229-235**
Checksum, **85**, **86-87**
Clear x , **36**, **40-41**, **73**
Cosinus, **57**
Cramersche Regel, **175**
Cursor, **15-16**, **23**

D

Daten, anzeigen, **79-80**
Dateneingabe, in Programm, **78**
DEG, **57**
Determinantenverfahren, **175-182**
Dezimalpunkt, **29**

Dezimalstellen, **30**
Dezimalzeichen, **29**
DSE, **101-102**
Durchschnitt. *Siehe* Mittelwert

E

E, **22**
 e , **55**
Ein, **14**
Einfügen von Programmzeilen, **82**
Einheitenkonvertierungen, **229-235**
Einvariablige Daten, **154**
Ellipse in der Anzeige, **150**
Endwert, **226**
ENG Format, **30**
ENTER, **23**, **25-27**, **39**
Erhöhung des Schleifenzählers, **101**
Exponent, **22-23**, **30**
 Eintippen von, **22**
 Ziffern in, **21**
Exponentialfunktion
 dekadische, **55**
 natürliche, **55**
 Kurve von, **204-205**, **211**

F

$f(x)$, **126**
 in Integration, **273**
 in SOLVE, **259**
Fahrenheit Konvertierung, **229-235**
Fakultät, **19**, **65**
Falsche Funktion, korrigieren, **42**
Falsche Zahlen, korrigieren, **42**
Feet Konvertierung, **229-235**
Fehler
 Anhalten eines Programms, **82**
 einer Funktion, Korrektur, **42**
 Meldung, **32**, **82**, **281-285**
 Fehler bei Integration, **274**

Fehler, numerische
 in quadratischer Gleichung, 191, 197
 in SOLVE, 272
 in Statistik, 161, 205
 in Trigonometrie, 57
 Fehlerkorrektur über LAST X, 41, 42–43
 Fehlersuche/-behebung, 245–247
 Fenster, 149–150
 Ferris Wheel Prinzip, 230
 Festkommaformat, 30
 Feuchtigkeitsgrenzen, 245
 Finanzielle Berechnungen. *Siehe* Finanzmathematik
 Finanzmathematik, 222–229
 FIX Format, 30
 Flag
 Abfragen, 95, 97, 98–99
 Löschen, 98
 Nummern, 97
 Setzen, 98
 Status, 98–99
 Flags,
 Overflow, 97–98
 Typen von, 97–98
 Flächenkonvertierungen, 229–235
 FOR-NEXT Schleife, 101
 Fragen, 240–241
 Funktion
 Auswerten, (SOLVE), 112–113
 Auswerten, (JFN), 128
 einwertige, 24–25
 Namen, 67
 Namen in Programmen, 74
 Taste, 24
 zweiwertige, 25
 Funktionen
 Index von, 286–298
 lösbare, 259–260
 Löser, 112
 numerische, 54–69
 SOLVE, 112

G

Ganzzahliger Teil, 67
 bei nichtdezimaler Arithmetik, 146
 Genauigkeit
 Integration, 132
 numerische, 30–31
 SOLVE, 272
 trigonometrische, 57
 volle, 31
 von Statistikdaten, 160–161
 Genauigkeit für Integration, 132
 Gespeicherte Daten, 253
 Gewogenes Mittel. *Siehe* Mittelwert, gewogen
 Gewährleistung, 248–249
 Service, 251
 Go To. *Siehe* GTO
 GRAD, 57
 Grad, Konvertierung, 64
 Grad-Modus, 57
 Größte Zahl für Basis-Konvertierung, 149
 GTO, 76, 84, 93–94, 100

H

Helligkeit der Anzeige, 14
 Hexadezimalzahlen, 144–149
 Horner Schema, 262
 Programmieren, 87–88
 Hyperbolische Funktionen, 25, 59
 Höchstes Bit, 148
i, 103–106
 Funktionen, Verwendung in, 103
 Variable, 53

I

Imaginäre Zahlen, 137
 Inaktive Taste, 32
 Inch Konvertierung, 229–235

Indexwert, 105
Indikatoren, 20–21
 Flag, 98
Indirekte Adressierung, 103–106
INPUT, 77–79
 Aufheben, 79
 Auswirkung auf Stack von, 257
 bei Integration, 128
 mit nichtdezimalen Zahlen, 150
 mit SOLVE, 112
Input, Programm, 78
Integral, Approximation, 131
Integrand, 127, 131
Integration, 126–136
 Algorithmus, 130, 272–274
 Anhalten, 127
 Anwendung, 127
 Approximationen, 273–274
 Ausgabe, 134
 Bedingungen, 134
 Besonderheiten, 275–277
 Einschränkungen für, 135
 Ergebnisse überprüfen, 278
 Ergebnisse, 127, 134, 274–278
 Fehler, 274
 Fehlerabschätzung von, 127, 132, 274
 Funktion für, 128
 Funktionsweise, 273–280
 Genauigkeit von, 127, 131–134
 Grenzen, 127, 130, 134
 in Programmen, 134–135
 Iterationen, 274
 Mehrfache Variableneingabe bei, 128
 Rechenzeit, 279–280
 Schreiben von Programmen für, 128
 Stützstellen, 274, 277
 Verfahren, 274
 verschachtelt, 135
Interne Genauigkeit, 30–31

Inverse Trigonometrie, 57, 58
Inverse, Matrix. *Siehe* Matrix-
 Invertierung
ISG, 101–102

K

Kartesische Koordinaten. *Siehe*
 Rechteckskoordinaten
Katalog
 von Programmen, 85
 von Variablen, 49
Kelvin Konvertierung, 229–235
Kettenrechnungen, 26, 44–46
Klammern, 26, 28, 45
Kombinationen, 65–66
Kommas in Zahlen, 29
Komplexe Arithmetik, 139–140
komplexe Wurzeln, quadratische, 191
Komplexe Zahlen, 137–143
 bei Integration, 126
 bei SOLVE, 112
 Eingabe, 137, 138
Konstante, anwenden, 39–40, 43
Konstantes Wachstum, 40
Kontrast, 14
Konvertierungen,
 Bruchteile, 63–64
 Koordinaten, 60–62
 Winkel, 64
Koordinaten, konvertieren, 60–62
Koordinatentransformationen,
 198–203
Kopieren von Variablen aus Katalog,
 49
Kopieren von Zahlen. *Siehe*
 Speichern von Zahlen
Korrelationskoeffizient, 159, 204,
 211–212
Kraftvektor, 174
Kreditberechnungen. *Siehe*
 Finanzmathematik

Kreisfläche, 70, 74, 78
Kreuzprodukt, Vektor, 164, 171
Kubische Gleichung, 194
Kubische Wurzel, 56
Kundenunterstützung, 240
Kurven, Grenzen für, 205
Kurvenanpassung, 158-160
 nichtlinear, 204-214
Kurvenmodelle, 204, 211
Kürzung bei nichtdezimaler
 Arithmetik, 146, 147

L

Łukasiewicz, 35
Labels. *Siehe* Programm-Labels
LAST X Register, 41-44
 betroffene Operationen, 258
LBL, 71-72, 73. *Siehe auch* Pro-
 gramm-Labels
Lineare
 Bewegung, Lösen für, 115
 Näherung. *Siehe* Lineare
 Regression
 Regression, 156, 158-160
Lineares Gleichungssystem
 Determinantenverfahren, 175-182
 Matrix-Inversionsverfahren,
 183-190
Logarithmische
 Funktionen, 25, 55, 139
 Kurve, 204-205, 211
Logarithmus,
 dekadischer, 55
 komplexer, 139
 natürlicher, 55
LOOP, 99, 101
Längenkonvertierungen, 229-235
Lösche *x*, 36, 40-41, 73
Löschen des Anzeigeinhalts, 36
Löschen von Programmzeilen, 82

Löschen, 15-16
 Programme, 85-86
 Speicherbereich, 34, 253, 255-256
 Statistikdaten, 154
 Variablen, aus Katalog, 49
Löschtaste, 16, 19, 32, 36, 40, 73
Lösen einer Gleichung, 110-125
Lösen nach unbekannter Variable,
 110-125
Lösungen. *Siehe* SOLVE Ergebnisse

M

Mantisse, 22, 30-31, 49
Maschenströme, 181
Matrix
 Ergebnis-, 183
 Formeln, 175-176, 183
 Inverse, 183-190
 Inversion, 183-190
 Koeffizienten-, 183
Matrizen, lösen. *Siehe* Lineares Glei-
 chungssystem
Meldungen, 32, 281-286
MEM, 33, 49, 85
MEMORY CLEAR, 243, 245, 255
MEMORY FULL, 85, 162, 253
Menü, 17
 Aufheben, 19
 Tasten, 16-19
 Typen von, 18
 Verlassen, 19-20
 Verwenden eines, 16-19
Meter Konvertierung, 229-235
Mittelwert, 156-157
 gewogener, 157-158
MODES Menü, 29
Moment, 174

N

Negative

- ganze Zahl, größte, **149**
- Newtonsches Verfahren, **215**
- Nicht programmierbare Funktionen, **87**
- nichtdezimale Zahlen, **148**
- Nichtkommutative Funktionen, **25, 37, 45**
- NO ROOT FND, **119, 267**
- Normalverteilung, **215–221**
- Zahlen, **21**

Zahl

- angezeigt, **30**
 - benannt, **41**
 - Bereich **24, 272**
 - Betrag von, **24, 272**
 - gerundet, **30**
 - zweimalige Verwendung, **39**
- Normalverteilung, **215**
- Null in Variable, **50**
- Null, **40**
- Nullstelle
- Auffinden einer, **259–260**
 - keine, **267–271**
 - Maximum, **267**
 - Minimum, **267–268**
 - Näherung für, **119, 197, 261**
 - quadratische, **191–197**
 - von Gleichung, **110, 116–117, 119**

O

- Oberfläche eines Zylinders, **80–81**
- Oktalzahl, **144–149**
- Operation
- Hilfe für, **240**
 - Überprüfung, **245–247**
- Operationen, Verzeichnis von, **286–298**
- Overflow, **24**
- bei nichtdezimaler Arithmetik, **146**
 - gekennzeichnet, **97–98**
 - in Programmen, **98**

P

- PARTS Menü, **67**
- Pause, programmiert, **82**
- Permutationen, **65**
- Phasenform, komplex, **142**
- Pol, SOLVE Funktion, **264–265**
- Polar
- Form, komplex, **142**
 - Koordinaten, konvertieren, **60–62**
 - Vektorkoordinaten, **170**
- Polynom
- Ausdrücke, programmieren, **87–88**
 - zweiten Grades, **191–197**
- Positive größte ganze Zahl, **149**
- Potenz
- Funktion, **56**
 - Funktion, komplex, **140**
 - Kurve, **204–205, 211**
- Potenzfunktion. *Siehe* y^x
- PRGM TOP, **72, 73, 84**
- Verschiebung zu, **84, 94**
- PRGM**, **72, 73, 75**
- Primzahlengenerator, **235–238**
- PROB Menü, **65**
- Programm
- Anhalten von, **82**
 - Anzeigen, über Katalog, **85**
 - Ausführung, **75**
 - Ausführung, schrittweise, **76**
 - Ausführung, über Katalog, **85**
 - Durchsehen von, **76**
 - Eingabe, **72–73**
 - Fortsetzen, **78, 82**
 - Grenzen, **71–72**
 - Katalog, **85**
 - Löschen, über Katalog, **85**
 - Modifizieren, **82**
 - Namen. *Siehe* Programm-Labels
 - Rücksprung zu, **72, 73**
 - Schreiben eines, **71–74**
 - Speicher, **72, 84–87**
 - Starten eines, **75, 76, 85**
 - Testen, **75–76**
 - Überprüfen eines, **86**
 - Zeiger, **76, 84, 94**

Programmzeilen, 72
 Einfügen, 82
 in nichtdezimalem Modus, 151
 Löschen, 73, 82
 neu numerieren, 82
 Programmzeilennummern
 in nichtdezimalem Modus, 151
 Verschieben zu, 84, 94
 Programm-Labels, 71-72, 73, 77, 85,
 86, 94, 95
 doppelte, 72
 in Katalog, 85
 indirekte, 103, 104
 Verzweigen zu, 94
 Programme, löschen, 85-86
 Programmierung mit Basis-Modi,
 151-152
 Programmierung, 70-89
 Prozent, 59-60
 Prozentuale Änderung, 59-60
 Prüfsumme, 85, 86-87
 Punkte in der Anzeige, 150
 Punkte in Zahlen, 29
 P↔RECT, 60-62

Q-R

Quadratische Gleichung, 191-197
 RAD, 57
 Radiant-Modus, 57
 Radianten, konvertieren, 64
 Radiusvektor, 174
 Rankine Konvertierung, 229-235
 RCL, 48
 Rechner, Funktionsstörung des
 245-247, 249-250
 Rechtecks
 Form, komplex, 142
 Koordinaten, konvertieren, 58,
 60-62
 Vektorkoordinaten, 170
 Reduzieren des Schleifenzählers, 101
 Reelle Zahlen, 54
 Referenzfunktion, 286-298

Register,
 Austauschen, 37
 Speicher. *Siehe* Variablen
 Stack, 35-41, 78
 Register, LAST X, 41-44
 Regression. *Siehe auch* lineare
 Regression
 Koeffizienten. *Siehe* Steigung und
 y-Achsenschnittpunkt
 nichtlinear, 204-214
 Reihenfolge
 bei Berechnungen, 26, 45-46
 bei Eingabe, 25
 von Zahlen, 37
 Reparatur, 248. *Siehe auch* Service
 Rollen, nach unten, 36-37
 Rotation, Koordinate, 198-203
 Routinen, Programm, 90
 RTN, 72, 73
 RTN, Unterprogramm, 91
 Run/Stop, 78
 Runden, 24, 30, 49, 67
 Rundungsfehler bei Integration, 131
 Rückruf-Arithmetik, 51-52
 Rückrufen von Zahlen, 48
 in einem Programm, 78
 Rückschritt-Taste, 16, 19, 23, 32, 40,
 73
 R+, 36-37

S

Saldo, 226
 Schachtel, Lösen der Dimensionen,
 113, 121
 Schleife,
 bedingte, 100
 mit Zähler, 95, 101-102
 Steuerzahl, 101
 unendliche, 100
 Schleifenbildung, 99-102
 mit (i), 106
 Schleifenzähler, 101
 Schwache Batterien, 242-243

- SCI Format, **30**
 Selbsttest, für Rechner, **246–247**
 Service, **249–251**
 internationaler, **250**
 Kosten, **250**
 Verträge, **251**
 Zentren, **250**
 SHOW, **31, 49, 79**
 nichtdezimale Zahlen, **150**
 Signifikante Ziffern, **22, 31, 49**
 Sinus, **57**
 Integral, **130–131**
 Skalarprodukt, Vektor, **164, 171**
 SOLVE, **259–272**
 Asymptote, **267, 269**
 Ausgabe, **124**
 Bedingungen, **124**
 Berechnung, unterbrechen, **119**
 bei unstetigen Funktionen, **263–264**
 Definition von Funktionen für,
 112–113
 Einschränkungen für, **125**
 Ergebnisse, **111, 119, 120, 124,**
 268, 272
 Ergebnisse, Interpretation, **261**
 flacher Bereich, **267, 271**
 Funktionsweise, **259–260**
 Genauigkeit, **272**
 in Programmen, **124**
 Iterationen, **118, 259**
 keine Ergebnisse, **267–271**
 mathematischer Fehler, **270**
 Maximum, **267**
 mehrvariablige Eingabe für, **112**
 Minimum, **267–268**
 Näherungen, **261**
 Programme (Funktionen), **112–113**
 Restriktionen, **259–260**
 Suche, **120, 267–268**
 Underflow, **272**
 Verfahren, **259–260**
 verschachtelt, **125**
 Verwendung, **111–113**
 Spaltenvektor, **189**
 Sparplanberechnungen. *Siehe*
 Finanzmathematik
 Speicherbereich,
 alles löschen, **255–256**
 Anforderungen, **254**
 Anwendung, **254**
 Anwendung für Statistik, **162**
 Anwender, **47**
 für Variablen, **50**
 Löschen, **34, 50, 253**
 Neuzuweisen, **254**
 Programm, **72, 78, 84–87**
 Sicherung, **51**
 Überprüfen, **33**
 verfügbarer, **33, 49**
 Verlust nach Batteriewechsel, **245**
 Verlust, schwache Batterien, **243**
 Verwaltung, **253–254**
 Verwendung für Programme,
 84–85
 Speicherarithmetik, **50–51**
 Speichern von Zahlen, **48**
 Sphärische Koordinaten. *Siehe*
 Polarnotation
 Stack Lift, **38, 39**
 Aktivieren, **257**
 betroffene Operationen, **256–257**
 neutral, **257**
 Sperren, **257**
 Stack,
 Auffüllen mit Konstante, **39–40**
 automatischer Speicher, **35–46**
 Drop, **38, 39**
 komplexer, **138**
 Stack,
 Durchsehen, **36**
 Durchsehen ohne Veränderung
 von, **49**
 Unterprogramm, **92, 125, 135**
 Standardabweichung, **156–157**
 Grundgesamtheit, **219**
 Stichprobe, **157**
 wahre, **157**
 STAT Menü, **156**

Statistik, 153–162
 Statistikberechnungen, 156–162
 Einschränkungen für, 160–161
 Statistikdaten
 akkumulierte, 161
 Eingabe von, 153–154
 Genauigkeit von, 160–161
 Korrektur, 155
 Löschen von, 154, 162
 Normalisieren, 161
 Sätze, Anzahl von, 154, 161
 Voraussage, 158–160
 Statistikregister, 161–162
 Einrichten von, 162
 Löschen, 162
 Steigung, 159, 204, 211–212
 STO, 48
 Störungen, 248
 Stromverbrauch, 242
 Summationswerte, statistische, 156,
 161–162
 Summe
 von Produkten, 162
 von Quadraten, 162
 von x -Werten, 161
 von y -Werten, 161

T

T-Register, 35–36, 38–40, 47
 Tangens, 57
 Technisches Anzeigeformat, 30
 Teile-von-Zahlen Funktionen, 25, 67
 Temperaturen,
 Betriebs-, 245
 Konvertieren, 229–235
 Lager-, 245
 TESTS, 96
 Translation, Koordinate, 198–203
 Trigonometrie 25, 56
 komplexe, 139

U

Überlauf, 24
 Umgekehrte polnische Notation.
 Siehe UPN
 Umschaltfunktion, aufheben, 15
 Umschalttaste, 15
 Unabhängige Variable, 154
 Underflow, 24
 SOLVE, 272
 Unstetigkeitsstelle, SOLVE Funktion,
 263–264
 Unterlauf, 24
 Unterprogramme, 91
 verschachtelt, 92
 Unterstützung, 240
 UPN (umgekehrte polnische
 Notation), 25–26, 28, 35, 44–46

V

Variable, ansehen einer, 49
 Variablen, 47–53
 Anzeigen von, 49
 Auflisten von, 49
 in Programmen, 77
 in Programmen, anzeigen, 79–80
 in Programmen, kopieren, 79–80
 indirekte, 103
 Integrations-, 128
 Katalog von, 49
 Kopieren, 49
 Löschen, 49–50
 momentaner Wert von, 77
 Namen von, 47–48, 77
 SOLVE, 112
 unbekannte, 110–112, 120
 Vektor
 Addition, 142
 Komponenten, 171
 Konvertieren in Rechteckskoordinaten, 62
 Operationen, 164–175

Vergleichsabfragen, 95–97
Versand, 251
Verteilungen, statistische, 157
Verzinsungsperioden, 226
Verzweigen, 93–94, 95
 rückwärts, 99–102
 unbedingt, 94
VIEW, 49, 79–80
 für nichtdezimale Zahlen, 150
Volumen eines Zylinders, 80–81
Voreinstellungswerte, zurücksichern,
 255–256
Vorzeichenbit, 148
Vorzeichenwechsel, 21

W

Wahrscheinlichkeit, 65–66
 normale, 215–221, 219
Wahr/Falsch Test, 95–99
Wertebereich von Zahlen, 24, 149
Wiederverwenden von Zahlen mit
 LAST X, 41, 43–44
Winkel, Konvertierung zwischen
 Grad und Bruchteilen, 64
Winkelmodus, 56–57
Wissenschaftliches Format, 30
Wortlänge, 149

X–Z

x -Näherung, 158–159, 212
X-Register, 35–40, 47
 Austauschen mit Y-Register, 37
 bei Programmierung, 70
 für Statistikdaten, 154
 Löschen, 40–41
 Löschen in einem Programm, 73
 mit Null vergleichen, 96
 mit SOLVE, 113, 120
 mit Y-Register vergleichen, 95–96
 Testen, 95–96
 und Integration, 128
XEQ, 75
 Unterprogramm, 91
 y^x , 56

y -Achsen Schnittpunkt, 159, 204,
 211–212
 y -Näherung, 158–159, 212
Y-Register, 35–37, 47
 für Statistikdaten, 154
 und Integration, 132
Z-Register, 35–36, 47
Zahl zur Potenz erheben, 56
Zahlen
 Eintippen von, 21
 Größe von, 21
 in Programmzeilen, 73, 151
 interne Darstellung von, 147–148
 komplexe, 137–143
 Korrigieren, 15–16, 41
 negative, 21
 nichtdezimale, 144–150
 nichtdezimale, interne Darstellung
 von, 147–148
 Prim-, 235–238
 rechts ausgerichtet, 148
 reelle, 54
 teilweise verborgen, 150
 Trennung von, 23, 27, 39
 zu groß, 21, 22, 49
 zu klein, 22
Zahlung, 226
Zahlungsbetrag, Vorzeichen von, 223
Zeilennummern, Programm, 72
Zeit, Konvertieren zwischen Minuten
 und Bruchteilen, 63–64
Ziffern
 Eingabe, 23
 Eingabe abschließen, 23
 Trennzeichen, 29
Ziffern, maximale Anzahl von, 21
Zinssatz, 226
Zufallszahl
 Generator, 65
 Startwert, 65
Zurücksetzen des Speicherbereichs,
 254–255
Zurücksichern von Zahlen. *Siehe*
 Rückrufen von Zahlen
Zweierkomplement, 146, 148
Zweivariablige Daten, 154
Zwischenergebnisse, 26, 28, 35

Unterstützung von Hewlett-Packard

Bezüglich Antworten auf die Anwendungsweise des Rechners: Wenn Sie Fragen zur Anwendung des Rechners haben, sollten Sie sich zuerst auf das Inhaltsverzeichnis, den Sachindex und den Abschnitt "Antworten auf allgemeine Fragen" in Anhang A beziehen. Sollten Sie in diesem Handbuch keine ausreichende Auskunft für Ihre Problemstellung finden, so können Sie sich über die nachstehende Adresse mit Hewlett-Packard in Verbindung setzen:

Hewlett-Packard GmbH
Support Zentrum Ratingen
Berliner Straße 111
D-4030 Ratingen
Telefon: (02102) 47504-0

Im Fall einer erforderlichen Reparatur: Falls die Hinweise in Anhang A auf eine notwendige Reparatur hindeuten, dann können Sie den Rechner an das nachstehende Reparaturzentrum schicken:

Hewlett-Packard GmbH
Reparaturzentrum Frankfurt
Berner Straße 117
D-6000 Frankfurt 56
Telefon: (069) 500001-0

Informationen über Hewlett-Packard Fachhändler, Produkte und Preise: Setzen Sie sich diesbezüglich mit der Hewlett-Packard Vertriebszentrale in Verbindung:

Hewlett-Packard Vertriebszentrale
Hewlett-Packard-Straße
D-6380 Bad Homburg
Telefon: (06172) 400-0

Inhaltsverzeichnis

Seite	13	Teil 1: Allgemeine Anwendungsweise Bedienungsgrundlagen • Der automatische Speicherstack • Datenspeicherung in Variablen • Reellwertige Funktionen
	69	Teil 2: Programmierung Einfache Programme • Programmierungstechniken
	109	Teil 3: Fortgeschrittene Operationen Lösen einer Unbekannten in Gleichungen • Numerische Integration • Operationen mit komplexen Zahlen • Rechnen in verschiedenen Zahlensystemen • Statistikberechnungen
	163	Teil 4: Applikationen Mathematikprogramme • Statistikprogramme • Sonstige Programme
	239	Teil 5: Anhänge und Index Kundenunterstützung, Batterien und Service • Benutzerspeicher und Stack • Näheres zum Lösen einer Gleichung • Näheres zur Integration • Meldungen • Funktionsindex • Sachindex



Bestellnummer
00032-90041

00032-90042 German
Printed in West Germany 12/88