Richard Eckert

Die Programmierbaren von HP

Mit 102 Beispielen und 65 Programmen

2. erweiterte Auflage

Oldenbourg

Die Programmierbaren von HP

Von Richard Eckert

Mit 102 Beispielen und 65 Programmen

2. erweiterte Auflage

R. Oldenbourg Verlag München Wien 1980

CIP-Kurztitelaufnahme der Deutschen Bibliothek

Eckert, Richard:
Die Programmierbaren von HP : mit 102 Beispielen u. 65 Programmen / von Richard Eckert. – 2., erw. Aufl. – München, Wien : Oldenbourg, 1980.
ISBN 3-486-23492-7

© 1980 R. Oldenbourg Verlag GmbH, München

Das Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, der Funksendung, der Wiedergabe auf photomechanischem oder ähnlichem Wege sowie der Speicherung und Auswertung in Datenverarbeitungsanlagen, bleiben auch bei auszugsweiser Verwertung vorbehalten. Werden mit schriftlicher Einwilligung des Verlages einzelne Vervielfältigungsstücke für gewerbliche Zwecke hergestellt, ist an den Verlag die nach § 54 Abs. 2 Urh.G. zu zahlende Vergütung zu entrichten, über deren Höhe der Verlag Auskunft gibt.

Gesamtherstellung: Hofmann-Druck KG, Augsburg

ISBN 3-486-23492-7

Inhalt

Vorwort 9

1. Das Arbeiten mit programmierbaren Rechnern 11

1.1 Einführung 11
1.2 Das Tastenfeld 12
1.3 Die Dateneingabe (Stack) 16
1.4 Der Arbeitsbereich 23
1.4.1 Unterlauf und Überlauf 23
1.4.2 Die verarbeitbare Stellenzahl 24
1.5 Der Umgang mit Zahlenmengen (Statistik) 26
1.5.1 Die statistischen Größen 26
1.5.2 Umgang mit Statistik 27
1.5.3 Verarbeiten von Zahlenmengen 28

2. Das Erstellen von Geradeausprogrammen 31

2.1 Grundsätzliches 31
2.2 Programmieren mit Labeln 31
2.3 Programmieren ohne Label 32
2.4 Der Programmspeicher 33
2.4.1 Der Speicherplatz 34
2.4.2 Kennzeichnung von Programmschritten 34
2.4.3 Erreichen von Programmzeilen 35
2.4.4 Ändern eines Programms beim HP 25 36
2.4.5 Ändern eines Programms beim HP 65, 67 und 97 39
2.5 Kurze Geradeausprogramme 41
2.6 Programme mit R/S 44
2.7 Die Speicherregister 46
2.7.1 Das Last-X-Register 47
2.7.2 Speichern und Zurückrufen von Zahlen 48
2.7.3 Speicherarithmetik 60
2.7.4 Die Sekundärregister 69
2.8 Überprüfen des Programms 75
2.9 Aufzeichnen von Programmen 78
2.9.1 Schutz der Programmkarten 80
2.9.2 Dokumentation 81
2.9.3 Aufzeichnen mit MERGE 81
2.10 Aufzeichnen von Daten 82
2.10.1 Programmkarte mit Daten 82
2.10.2 Datenübertragung mit W/DATA 90
2.10.3 Datenübertragung mit MERGE 91

3. Unterprogramme 92

3.1 Aufruf von Unterprogrammen mit GSB 92
3.2 Sprung zu Unterprogrammen mit GTO 101
3.2.1 Sprung GTO Zeilen-Nr. 101
3.2.2 Sprung GTO Programm-Marke 102

4. Programmverzweigungen 104

4.1 Veranschaulichen des Rechengangs 104
4.2 Vergleichsprüfungen 106
4.3 Flagprüfungen 118

5. Programmschleifen 126

5.1 Schleifen ohne Bedingung 126
5.2 Schleifen mit Vergleichsprüfungen 128
5.2.1 Vergleich gegenüber einer festen Zahl 129
5.2.2 Vergleich zweier Zwischenergebnisse 136
5.3 Schleifen mit fester Zahl von Durchläufen (DSZ und ISZ) . 146

6. Indirekte Adressierung 158

6.1 Speichern und Zurückrufen von Daten 158
6.2 Aufruf von Unterprogrammen 162

7. Rechenzeiten 165

8. Weitere Informationsmöglichkeiten 166

9. Programmschrittlisten 168

10. Rechnen mit dem HP-41 C 203

10.1 Das Tastenfeld 204
10.2 Das Alpha-Register 206
10.3 Die Standardfunktionen 209
10.4 Programmieren mit dem HP-41 C 215

Sachregister 233

Liste der Programme

Nr.	Inhalt	
2.1	Verdoppeln einer Zahl	31
2.2	Berechnen von Bruttopreisen	41
2.3	Funktionswerte einer logarithmischen Darstellung	42
2.4	Umrechnung Grad Celsius - Fahrenheit	42
2.5	Hyperbolische Funktionen sh x und ch x	43
2.6	Raumdiagonale eines Quaders	44
2.7	Gleichung mit einer Unbekannten	45
2.8	$f_1(x) = \sqrt{1 + x^2}$ und $f_2(x) = \frac{2}{\sqrt{1 + x^2}}$	46
2.9	$f(x) = \sin x + \sqrt{x}$	47
2.10	Relative Änderung	48
2.11	$f(x) = \frac{x + 1}{x - 1}$	49
2.12	Maßstabsänderung	50
2.13	$f(x) = \sqrt{1 + x^2}\ (1 + \sqrt{1 + x^2})$	51
2.14	Ellipsengleichung	52
2.15	Gleichungen mit zwei Unbekannten	52
2.16	Gleichungen mit drei Unbekannten	54
2.17	Zinseszins-Tabelle	55
2.18	Vermögensbildung durch Ratensparen	56
2.19	Wertpapier-Rendite	58
2.20	Verkehrszählung	61
2.21	Widerstand einer Parallelschaltung	62
2.22	Einwiegen zweier Stoffe	63
2.23	Bild einer Spirale	64
2.24	Cassinische Kurven	65
2.25	Spirale die eine Spindel umwickelt	66
2.26	$f(x) = \frac{\sin x}{x}$	67
2.27	Mondlandespiel	68
2.28	Optimale Gerade	70
2.29	Gleichungen aus der Plasmaphysik	83
2.30	Gleichungen aus der Strahlungsphysik	87
3.1	$f_1(x) = \sqrt{1 + x^2}$ und $f_2(x) = \sqrt{1 + x^2}\ (1 + \sqrt{1 + x^2})$ mit Unterprogramm	92
3.2	Gleichungen mit vier Unbekannten	93
3.3	Ableitungen $f'(x)$ bis $f^{IV}(x)$	96

3.4	$f_1(x) = \frac{x}{1 + x^2}$ und $f_2(x) = \frac{\sin x}{1 + \sin^2 x}$	102
3.5	Dreiecksrechnung	102
4.1	Umwandeln negativer Zahlen	106
4.2	Unterscheiden des Vorzeichens	107
4.3	Erraten einer Zahl	107
4.4	Sortieren von Zahlen in zwei Gruppen	108
4.5	Kreis durch drei Punkte	112
4.6	Preisstaffel	115
4.7	Einkommensteuer	116
4.8	$f_1(x) = \sqrt{1 + x^2}$ und $f_2(x) = \frac{2}{\sqrt{1 + x^2}}$ mit Flags	119
4.9	Seitenlängen im rechtwinkligen Dreieck	122
4.10	U-Boot-Jagd	123
5.1	Ansteigende Zahlenreihe	126
5.2	Mehrfache Wurzel einer Zahl	127
5.3	Glücksrad	128
5.4	Auf- und absteigende Zahlenreihe	129
5.5	e^x, cos x und Ei(x) als Reihen	130
5.6	Newtonsche Näherungslösung für f(x) = 0	132
5.7	Extremwertefinder	136
5.8	Primfaktorenzerleger	139
5.9	Primzahlenfinder	142
5.10	Zahlenumwandlung	143
5.11	$\sum n$, $\sum \frac{1}{n}$, $\sum \frac{1}{n^2}$ und n!	148
5.12	Numerische Integration nach Simpson	149
5.13	Numerische Lösung der Differentialgleichung $y' = f(x, y)$	152
5.14	Numerische Lösung der Differentialgleichung $y'' = f(x, y, y')$	155
6.1	Sortieren von Zahlen	160
6.2	Nachtgefecht	161
6.3	Direktes Anwählen von Unterprogrammen	163
10.1	Umrechnen von Tagen in Wochen	221
10.2	Überprüfen der Schleifenzahl	227
10.3	Abspeichern von Zahlen	229
10.4	Integer und Fractional	232

Vorwort

Mit einem programmierbaren Rechner verfügt der Benutzer über ein Hilfsmittel, das dem einfachen Taschenrechner weit überlegen ist. Auch dem teuren Tischrechner zeigt sich der programmierbare Taschenrechner durch seine Verfügbarkeit und Zeitersparnis beim Programmieren gewachsen. Vom Arbeiten mit schwerfälligen Datenverarbeitungsanlagen sei hier ganz abgesehen.

Leider muß jedoch mancher Anwender feststellen, daß auch die angeblich so leicht zu programmierenden Taschenrechner ihre Probleme haben. Rechnungen, die über ein Eintippen von Formeln hinausgehen, bereiten dem Anfänger oft unüberwindliche Schwierigkeiten.

Es ist festzuhalten, daß ein solcher programmierbarer Rechner tatsächlich ein ausgezeichnetes Werkzeug ist, mit dem sich auch komplexe Rechnungen überraschend einfach lösen lassen. Glücklicherweise sind die Schwierigkeiten meist nur die Folge von anfänglich ungenügender Kenntnis der Arbeitsweise des Rechners. Sie lassen sich also durch einen vertieften Einblick und eine gewisse Übung beheben.

Das Buch wendet sich nun an alle diejenigen, die über einen programmierbaren Taschenrechner verfügen, aber trotz Studiums des jeweiligen Bedienungshandbuches die praktische Anwendung weiterhin schwierig finden. Hier soll zum sicheren Bedienen sowie zum einfachen übersichtlichen Programmieren dieser Rechner Hilfestellung gegeben werden.

Angesprochen sind dem Buchtitel gemäß die Benutzer von HP-Rechnern. Bedingt brauchbar ist das Buch aber auch für Besitzer von Rechnern mit anderer Befehlsstruktur, sofern deren Bedienung beherrscht wird.

1. Das Arbeiten mit programmierbaren Rechnern

1.1 Einführung

Die Programmierbaren von HP - das sind im Augenblick die Hewlett-Packard-Rechner 33 E/C, 34 C, 38 E/C, 67/97 und der neue 41 C. Hierbei bilden die Rechner HP 33 E bis 38 C die Gruppe der einfacheren, dafür auch preiswerteren programmierbaren Taschenrechner. Mit 7 bis 20 Speicherregistern und 49 bis 99 Programmzeilen lassen sich mit ihnen auch kniffelige Aufgaben lösen. Da sie in vielem dem früheren HP 25 entsprechen, werden sie im Text unter diesem Stichwort behandelt.

Die Rechner HP 67 und HP 97 mit je 26 Speicherregistern und 224 Programmzeilen schieben die Grenzen der Anwendung weiter nach außen - weiter weg, als mancher Anwender vermuten würde. Entsprechend sind die Programme dieses Buches für den HP 67/97 gesetzt mit Hinweisen auf andere Rechner, u.a. den früheren HP 65.

Der neue HP-41 C bildet gleichsam eine Klasse für sich. Neben seinen bis zu 63 Speicherregistern bzw. 440 Programmzeilen sind es die steckbaren Zusätze wie Kartenleser oder Speichermodule, die seine Möglichkeiten erkennbar machen. Die Alphanumerik - der Dialog mit dem Rechner mit Worten - schließt die Lücke zu den Tischrechnern. Daher ist dem 41 C ein eigenes Kapitel gewidmet, von dem aus die Programme dieses Buches in den 41 C übertragen werden können.

Die Rechner HP 65, 67 und 97 können alle drei die Programme auf Magnetkarten speichern. Dies hebt sie merklich ab vom einfacheren und entsprechend billigeren HP 25. Darüber hinaus bieten die drei teureren HP-Rechner eine Reihe weiterer Möglichkeiten zum Programmieren auch schwieriger Rechenprobleme. Auf das Arbeiten mit diesen drei Rechnern wird daher besonders eingegangen. Für die Programm-Schrittfolgen wurde die Tastenbelegung des HP 67 gewählt. Dieser steht gleichsam in der Mitte

zwischen seinem Vorgänger HP 65 und dem HP 97 mit Druckwerk. Zum Glück ähneln sich die drei Rechner stark. HP 67 - Programme lassen sich z.B. direkt in den HP 97 übertragen. Als Schrittfolge sind sie meist ohne weiteres in den HP 65 einzutasten, häufig auch in den HP 25. Hier ist jedoch die begrenzte Schrittzahl und der Verzicht auf Label zu berücksichtigen.

1.2 Das Tastenfeld

Auf der Oberseite von programmierbaren Taschenrechnern befindet sich ganz oben das Anzeigefenster. Bei den HP-Rechnern 25, 65 und 67 sind darunter zwei Schiebeschalter angebracht. Sie sind beschriftet mit OFF - ON (Hauptschalter: aus - ein) sowie W/PRGM - RUN (write/program: auf Magnetkarten schreiben/programmieren - laufen von Programmen oder direkten Rechnungen). Zu Beginn wird (Stecker des Ladegerätes in einer Steckdose) der linke Schiebeschalter auf ON geschoben. Mit dem rechten Schiebeschalter in Stellung RUN zeigt nun die Anzeige 0.00. Das Tastenfeld ist damit zur Eingabe bereit.

Auf den ersten Blick erscheint das Tastenfeld etwas verwirrend. Auf und vor den Tasten sind jeweils mehrere Zeichen angebracht. Diese Mehrfachbelegung ist eine Folge der vielen Tastenfunktionen bei zugleich begrenzter Tastenzahl. Sie ist damit bereits ein Ausdruck der vielfältigen Möglichkeiten dieser Rechner.

Beim HP 67 ist neben dem Aufdruck auf den Tastenoberseiten vorn ein weiterer schwarzer Aufdruck angebracht, unterhalb links davon ein goldener sowie unterhalb rechts ein blauer.

Die drei zusätzlichen Funktionen werden mit drei entsprechenden Vortasten aufgerufen, nämlich der schwarzen h-Taste, der goldenen f- sowie der blauen g-Taste. Wird z.B. die Taste mit dem Aufdruck "4" gedrückt, so erscheint als Anzeige 4. Wird jetzt h und dann Taste 4 gedrückt (also h $\frac{1}{x}$), so bildet der Rechner $\frac{1}{4}$ (mit der 4 aus der Anzeige) und gibt dies als 0.25 an. Drückt man nun f und Taste 4 (also f SIN), so wird aus

den 0.25 der Sinus gebildet und angezeigt als $4 \cdot 10^{-3}$. Drücken von g und Taste 4 (also g SIN^{-1}) bewirkt den Arcus Sinus der Zahl im Anzeigefeld, also 0.25^{o}. Zu merken ist, daß die goldenen und blauen Beschriftungen beim HP 67 stets unterhalb der jeweiligen Taste stehen.

Vergleicht man die anderen programmierbaren HP-Rechner mit dem HP 67, so ist die Tastenzuordnung ganz ähnlich. Der HP 65 hat gleichfalls die f und die g-Taste, der h-Taste entspricht die Taste f^{-1}. Der HP 25 hat weniger eingebaute Funktionen, bei ihm fehlt die h-Taste ganz. Der in der Bauform größere HP 97 hat die gleichen Tastenfunktionen wie der HP 67. Jedoch ist beim HP 97 das Tastenfeld genügend groß, um fast jeder Funktion eine eigene Taste zuzuordnen.

Wie erwähnt wird im vorliegenden Buch die Tastenbelegung des HP 67 verwendet. Diese Belegung ist gleichsam eine Brücke zum HP 25 und HP 65 auf der einen sowie zum HP 97 auf der anderen Seite.

Die Beschriftungen an den Tasten sind Abkürzungen englischer Ausdrücke. Sie bedeuten im einzelnen:

Farbe	Aufdruck	Abkürzung von	Bedeutung
blau	SCI	scientific	wissenschaftliche Anzeige mit Zehnerexponent
	MERGE	merge	verbinde zwei Programmteile
	FRAC	fractional	nimm den nicht ganzzahligen Teil der Zahl
	STK	stack	Registerstapel
	NOP	no operation	Schritt ohne Programmwirkung
schwarz	RTN	return	kehre zum Programmanfang zurück
	ENG	engineering	technische Anzeige mit Zehnerexponent
	BST	back step	gehe im Programmspeicher einen Schritt zurück
	STI	store I	speichere im I-Register
	RCI	recall I	hole aus dem I-Register
	DEG	degree	Winkelgrad (360° = Vollkreis)
	RAD	radian	Bogenmaß (2π = Vollkreis)
	GRD	grade	Neugrad (400 = Vollkreis)
	DEL	delete	lösche den Programmschritt
	R↓	roll down	nach unten rollen des Stack
	R↑	roll up	nach oben rollen des Stack
	ABS	absolute	bilde den Betrag einer Zahl
	PAUSE	pause	Zwischenhalt
	REG	registers	zeige den Inhalt der Speicher-register
	LST X	last X	zeige den letzten x-Wert vor der Verarbeitung
	R/S	run/stop	laufe/halte an
	SPACE	space	Abstand (beim HP 97)
	SF	set flag	setze Fähnchen
	CF	clear flag	lösche Fähnchen
	F?	flag?	Fähnchen gesetzt?

Farbe	Aufdruck	Abkürzung von	Bedeutung
weiß	GTO	go to	gehe zu (Label 3, Schritt 25)
	DSP	display	Anzeige, Stellenzahl
	SST	single step	Einzelschritt
	STO	store	speichere in Register
	RCL	recall	hole aus Register
	ENTER	enter	gehe in den Registerstapel
	CHS	change sign	wechsle das Vorzeichen
	EEX	exponent	Zehnerexponent
	CLX	clear X	lösche das X-Register
gelb	GSB	go to subroutine	gehe zum Unterprogramm und kehre zurück
	FIX	fixed	Festkomma
	RND	round	runde die Zahl
	LBL	label	Programm-Marke
	DSZ	decrement and skip if zero	ziehe 1 ab und überspringe falls Null
	ISZ	increment and scip if zero	zähle 1 zu und überspringe falls Null
	W/DATA	write/data	schreibe/Daten
	P⇄S	primary versus secondary	vertausche die Inhalte der Primär- gegen Sekundärregister
	CL REG	clear registers	lösche die Speicherregister
	CL PRGM	clear program	lösche den Programmspeicher
	INT	integer	nimm den ganzzahligen Teil der Zahl

1.3 Die Dateneingabe (Stack)

Die HP-Taschenrechner verwenden zum Rechnen die Umgekehrte Polnische Notation (UPN). Das Verknüpfen zweier Zahlen durch eine Rechenoperation erfolgt dabei in der Reihenfolge

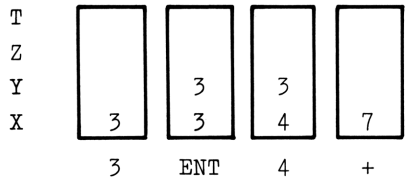
Eintasten der ersten Zahl
Drücken der Entertaste
Eintasten der zweiten Zahl
Drücken der entsprechenden Verknüpfungstaste

Beim Drücken der Entertaste (enter = gehe in den Registerstapel) kopiert der Rechner die eingegebene Zahl vom Aufnahmeregister X in das nächsthöhere Register Y. Nun kann das X-Register die nächste Zahl aufnehmen. Diese Art der Eingabe ist eine Folge des Rechneraufbaus aus 4 übereinander angeordneten Rechenregistern X, Y, Z und T. Diese bilden einen Registerstapel (englisch: stack). Am Beispiel sei die Belegung des Stack gezeigt:

3 + 4 (= 7)

Drücken der Taste	Wirkung
3	Die erste Zahl wird ins X-Register eingeschrieben. Dieses ist zugleich Anzeigefenster.
ENT	Durch Drücken der Entertaste wird der Inhalt des X-Registers ins Y-Register kopiert.
4	Die zweite eingegebene Zahl überschreibt die Zahl im X-Register.
+	Durch Drücken der Verknüpfungstaste + werden beide Zahlen zusammengezählt und das Ergebnis im X-Register angezeigt.

Im folgenden Bild sei die Belegung des Stack bei den einzelnen Schritten gezeigt. Das Rechteck stellt dabei die vier übereinander angeordneten Rechenregister X, Y, Z und T dar, die Zahlen die jeweiligen Registerinhalte.

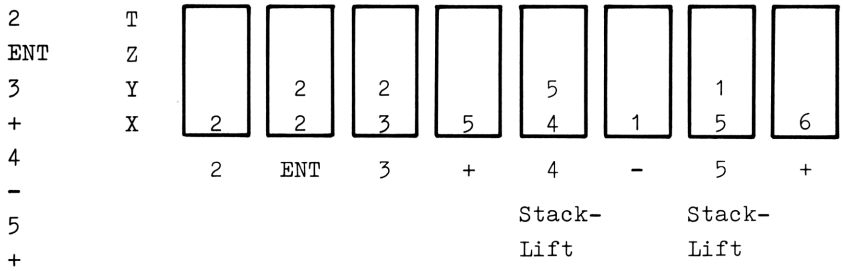


Entsprechend wird eingegeben

23 - 17 (= 6)	-7 + 3 (= -4)	5 x 6 (= 30)	23 ÷ 7 (= 3.29)
2	7	5	2
3	CHS	ENT	3
ENT	ENT	6	ENT
1	3	x	7
7	+		÷
-			

Das negative Vorzeichen einer Zahl wird dabei durch CHS (englisch: change sign = wechsle das Vorzeichen) ausgedrückt, nicht durch die Verknüpfungstaste - .

Sollen weitere Zahlen hinzugefügt werden, so ist nach dem Drücken der entsprechenden Zahlentaste lediglich die nächste Verknüpfungstaste zu drücken. Die bereits im Rechner befindlichen Zahlen werden bei der Eingabe einer weiteren Zahl selbsttätig im Stack angehoben (englisch: stack lift).



Entsprechend gilt, wenn als erstes eine der inneren Funktionen der HP-Rechner aufgerufen wird. Hier wird kein Enter gedrückt. Das Aufrufen der jeweiligen Funktion bewirkt bereits die Eingabe in den Rechner. Ein solcher Stack-Lift wird auch durch Drücken einer Labeltaste vorbereitet, vgl. Kapitel 2.2.

2 x sin 30° (= 1)

3	T					
0	Z					
f SIN	Y				0.5	
2	X	3	30	0.5	2	1
x		3	0	f SIN	2	x

5 π (= 15.71)

h π	T			
5	Z			
x	Y		3.14	
	X	3.14	5	15.71
		h π	5	x

auch: 5 π (= 15.71)

5	T			
h π	Z			
x	Y		5	
	X	5	3.14	15.71
		5	h π	x

Stack-Lift

sin 15° x cos 75° (= 0.07)	$\sqrt{3}$ x 5 (= 8.66)	1.4^2 x $\sqrt{7}$ (=5.19)
1	3	1
5	f $\sqrt{x}$	.
f SIN	5	4
7	x	g x^2
5		7
f COS		f $\sqrt{x}$
x		x

Die Entertaste muß dagegen jeweils gedrückt werden, wenn eine neue Serie von Rechenschritten eingegeben wird.

Beispiel:

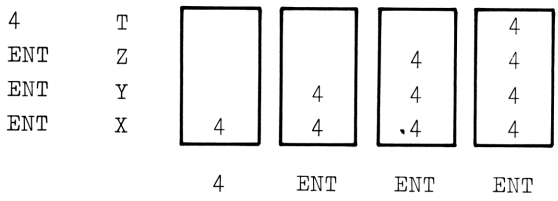
$\frac{2}{3+4}$ (= 0.29)

Eintasten	Wirkung
2 ENT	Eingabe des Zählers
3 ENT	Beginn der Eingabe des Nenners
4 +	Zwischenergebnis Nenner
÷	Teilen des Zählers durch den Nenner

$\frac{5+3}{4}$ (= 2)

Eintasten	Wirkung
5 ENT	Beginn der Eingabe des Zählers
3 +	Zwischenergebnis Zähler
4	Eingabe des Nenners
÷	Teilen des Zählers durch den Nenner

Wird die Entertaste mehrfach gedrückt, so wird der Stack entsprechend aufgefüllt:



Nach dreimaligem Drücken von Enter ist der Stack gefüllt. Vor einer weiteren Eingabe müssen erst die Stackinhalte verarbeitet werden.

```
6
ENT
5
ENT
4
ENT
3
```

T						6	6
Z				6	6	5	5
Y		6	6	5	5	4	4
X	6	6	5	5	4	4	3
	6	ENT	5	ENT	4	ENT	3

```
x
x
x
```

Werden nun die Inhalte des X- und Y-Registers miteinander verknüpft, so wird der Inhalt des Z-Registers nach Y kopiert, der Inhalt des T-Registers nach Z.

T	6	6	6	6
Z	5	6	6	6
Y	4	5	6	6
X	3	12	60	360
		x	x	x

Die Belegung des Stack läßt sich jederzeit mit g STK überprüfen. Die Stackregister werden daraufhin in der Reihenfolge T, Z, Y, X gezeigt.

Eine weitere Möglichkeit zur Überprüfung des Stack bilden die Funktionen h R↓ (roll down) und h R↑ (roll up). Mit ihnen werden die Registerinhalte schrittweise zyklisch heruntergezogen bzw. angehoben. Nach viermaligem Tastendruck ist der Ausgangszustand wieder erreicht.

T	13	2	5	7	13
Z	7	13	2	5	7
Y	5	7	13	2	5
X	2	5	7	13	2
		h R↓	h R↓	h R↓	h R↓

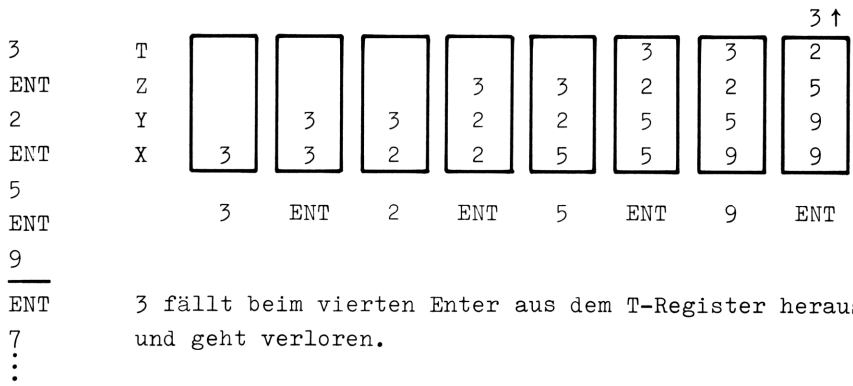
T	13	7	5	2	13
Z	7	5	2	13	7
Y	5	2	13	7	5
X	2	13	7	5	2
		h R↑	h R↑	h R↑	h R↑

Wie in Kapitel 2.3 gezeigt wird, lassen sich auch diese Funktionen beim Programmieren wirkungsvoll einsetzen.

Eine Kontrolle der jeweiligen Stackbelegung ist im allgemeinen nicht nötig, weder für Rechnungen auf dem Tastenfeld noch später bei einfachen Programmen. Andererseits lassen sich z.B. Programme mit Vergleichsprüfungen (Kapitel 4.2) nur erfolgreich aufstellen, wenn bekannt ist, was die zu vergleichenden Stackregister während der Prüfung enthalten.

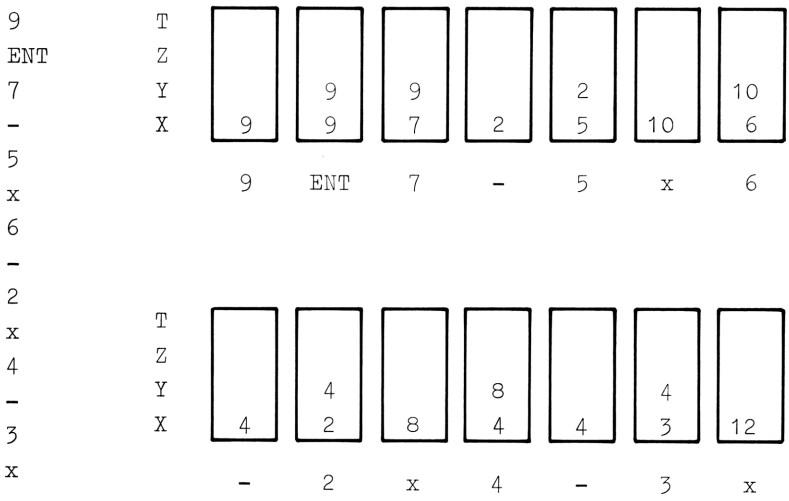
Die Beschränkung auf höchstens drei aufeinanderfolgende Eingaben mit Enter begrenzt theoretisch die Rechenmöglichkeiten. So ist bereits der folgende Ausdruck von links nach rechts gerechnet mit dem 4-Register-Stack nicht mehr zu lösen:

3{2[5 (9 - 7) - 6] - 4} (= 12)



3 fällt beim vierten Enter aus dem T-Register heraus und geht verloren.

Der Ausweg ist, solche Ausdrücke von innen nach außen zu lösen. Durch das wiederholte Bilden von Zwischenergebnissen bleibt der Stack entlastet. Die Eingabe für den beschriebenen Ausdruck würde also lauten:



Während des gesamten Rechenganges bleiben die Register Z und T frei. Durch das sofortige Ausrechnen von Zwischenergebnissen ist der Rechengang auch bei komplizierten Ausdrücken verhältnismäßig einfach. Eigentlich müßte bei jedem Klammerbeginn die Entertaste gedrückt werden, also Rechnungen auf höchstens vier Schachtelklammern beschränkt sein. Die Zwischenergebnisse bedeuten jedoch einen jeweiligen Neubeginn, so daß in der Regel mit halbleerem Stack gerechnet wird. Für den Fall eines dennoch überfüllten Stack können Zwischenergebnisse in den Speicherregistern (Kapitel 2.7) abgespeichert werden.

1.4 Der Arbeitsbereich

Mit der Kenntnis der Stackrechnung können innerhalb des Wertebereichs der HP-Rechner beliebig schwierige Ausdrücke berechnet werden. Die Grenzen sind gegeben durch die größten und kleinsten verarbeitbaren Zahlen sowie durch die endliche Stellenzahl, mit der Digitalrechner intern rechnen.

1.4.1 Unterlauf und Überlauf

Beispiel: Es sollen die Quadrate von 10^{49} und 10^{50} gebildet werden

$(10^{49})^2 = 10^{98}$ $\qquad$ $(10^{50})^2 = 10^{100}$

EEX		EEX	
4		5	
9		0	
g x^2	$= 10^{98}$	g x^2	$= 9.99 \cdot 10^{99}$

Im Fall $9.99 \cdot 10^{99}$ ist ein Überlauf eingetreten. Wenn die Zahl mit einer weiteren Zahl größer Eins malgenommen wird, so ändert sie sich nicht.

Beispiel: Es sollen die Kehrwerte von 10^{99} und $2 \cdot 10^{99}$ gebildet werden.

$\frac{1}{10^{99}} = 10^{-99}$ $\qquad$ $\frac{1}{2 \cdot 10^{99}} = 5 \cdot 10^{-100}$

EEX		2	
9		EEX	
9		9	
h $\frac{1}{x}$	$= 10^{-99}$	9	
		h $\frac{1}{x}$	$= 0$

Beim Ergebnis $5 \cdot 10^{-100}$ wird der Wertebereich unterschritten. Der Rechner setzt statt des zu kleinen Ergebnisses Null.

Der Wertebereich der HP-Rechner 25, 65, 67 und 97 erstreckt sich von 10^{-99} bis $9.99 \cdot 10^{99}$. Hierauf ist beim Rechnen allgemein zu achten. Dies gilt auch für das Rechnen mit Funktionen, bei denen ein Über- oder Unterschreiten des Wertebereichs evtl. nicht gleich bemerkt wird. So können z.B. Exponentialfunktionen nur mit einem Exponenten zwischen -227 und +230 gebildet werden. Es ist nämlich

$$e^{-228} = 0.96 \cdot 10^{-99}$$

und

$$e^{231} = 2.1 \cdot 10^{100}$$

Beide Ausdrücke liegen außerhalb des Wertebereichs.

1.4.2 Die verarbeitbare Stellenzahl

Die zweite Einschränkung ist durch die endliche verarbeitbare Stellenzahl gegeben. Diese beträgt bei den HP-Rechnern 25, 65, 67 und 97 intern 10 Stellen, auch wenn - wie beim HP 25 - nur 8 Stellen im Anzeigefenster sichtbar sind. Wird also einer Zahl eine andere Zahl, die um 10 oder mehr Stellen kleiner ist, zugezählt oder abgezogen, so wird als Ergebnis nur die ursprüngliche Zahl angezeigt.

Beispiel:

$10^9 + 1 - 10^9$ (= 1)

EEX	
9	
ENT	
1	
+	
EEX	
9	
-	= 1

$10^9 + 0.1 - 10^9$ (= 0.1)

EEX	
9	
ENT	
.	
1	
+	
EEX	
9	
-	= 0

Allgemein zeigen Digitalrechner, also auch die HP-Taschenrechner, dieses Verhalten. Auf diese einfache Art läßt sich ebenfalls bei anderen Rechnern z.B. eine Katalogangabe über die interne Stellenzahl (z.B. 12 oder 14 Stellen) überprüfen. Bei Rechnungen führt der Effekt zu unerwünschten Rundungen, die das Ergebnis verfälschen können.

Beispiel:

$10^9 + 4.4 - 10^9$ (= 4.4) Anzeige: 4.00
$10^9 + 4.5 - 10^9$ (= 4.5) Anzeige: 5.00.

Solche Rundungen treten auch beim Berechnen von Funktionen auf. So wird z.B. bei der Funktion $(e^x - 1)$ für x kleiner als $5 \cdot 10^{-10}$ vom Rechner gesetzt $(e^x - 1) = 0$ statt = x. Im Bereich $x = 5 \cdot 10^{-10}$ bis 0.69 (!) treten Rundungen auf, die sich als Stellenlöschungen bemerkbar machen. Diese Stellenlöschungen können mit dem passenden Anzeigeformat selbst beobachtet werden. Dazu wird gedrückt g SCI und DSP 9. Erst ab x = 0.7 sind für $(e^x - 1)$ auch die hinteren Stellen belegt. Wenn beim Rechnen diese Rundungen stören, so ist der Rechengang abzuändern.

In Beispiel 2.30 wird bei der Planckschen Strahlungsformel sowohl eine Wertebereichsunterschreitung wie auch eine Überschreitung der verarbeitbaren Stellenzahl behandelt.

1.5 Der Umgang mit Zahlenmengen (Statistik)

1.5.1 Die statistischen Größen

Häufig sind in einer Rechnung große Mengen gleichartiger Zahlen zu verarbeiten. Solche Zahlenmengen können mit den Methoden der Statistik aufgenommen und ausgewertet werden. Wegen der Bedeutung einer statistischen Auswertung haben heute auch einfachere Taschenrechner statistische Größen als Tastenfunktion. Dies sind in der Regel der Mittelwert und die Standardabweichung. Zusätzlich wird die Zahl der eingegebenen Werte angezeigt. Der Mittelwert ist gleich der Summe der eingegebenen Zahlen geteilt durch deren Anzahl, also

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

Die Standardabweichung ist gleich dem mittleren Abstand der Zahlenwerte vom Mittelwert, also

$$s = \sqrt{\frac{\Sigma x_i^2 - n \cdot \bar{x}^2}{n - 1}}$$

Die Tastenfunktionen sind beim HP 67 $\Sigma+$ für die Eingabe der Zahlen sowie h $\Sigma-$ für das nachträgliche Entfernen einer Zahl. Die Funktion f $\bar{x}$ bewirkt die Bildung des Mittelwertes und g s die Standardabweichung. Nach jedem Drücken von $\Sigma+$ wird die Zahl der eingegebenen Werte gezeigt, ebenso nach Entfernen mit h $\Sigma-$ die Zahl der verbleibenden. Die Eingabe kann auch nach Drücken von f $\bar{x}$ oder g s fortgesetzt werden. Ebenso lassen sich Fehleingaben noch nachträglich mit h $\Sigma-$ entfernen. Entsprechend zu dieser Eigenschaft sind vor Eingabe einer neuen Zahlenkolonne die bisherigen Daten zu löschen. Dies geschieht beim HP 25 durch f REG, beim HP 65 durch RTN R/S. Beim HP 67/97 wird gedrückt f CL REG, f P⇄S, f CL REG. Zum Löschen kann der Rechner auch kurzzeitig ausgeschaltet werden. Als Beispiel sei eine einfache statistische Rechnung gezeigt.

Beispiel:

Bei der Prüfung der Reißfestigkeit dünner Drähte wird als Reißlast in Millinewton gemessen

$$R = 46,\ 28,\ 35,\ 12,\ 31,\ 42,\ 28,\ 17\ mN.$$

Die überprüften 8 Drähte zeigen damit eine Reißfestigkeit von

$$R = 30 \pm 12\ mN.$$

1.5.2 Umgang mit Statistik

Die Größen n, $\bar{x}$ und s sind von vergleichbarer Wichtigkeit. Sie sind daher auch alle als Ergebnis einer Auswertung anzugeben. Häufig führt eine ungenügende Beachtung einer der drei Größen zu Mißdeutungen. So kann die überprüfte Zahl n zu klein, der Mittelwert $\bar{x}$ zu niedrig oder die Standardabweichung s zu groß sein.

Beispiel:

Bei einem gemeinsamen Einkaufsbummel haben 9 Teilnehmer kein Geld bei sich, 1 Teilnehmer hat 1000.- DM. Im Mittel verfügt also jeder über ca 100.- DM. Ein Rückschluß aus der mittleren Barschaft auf die Kaufmöglichkeiten wäre jedoch irrig: Bezahlen kann fast keiner. Liegen nicht die Originalzahlen, sondern bereits die statistische Auswertung vor, so ist dies auf einen Blick zu erkennen:

10 Teilnehmer, die mittlere Barschaft beträgt 100 ± 316 DM.

Die Streuung ist größer als der Mittelwert, die Auswertung muß daher überprüft werden. Ähnlich verhält es sich z.B. mit Aufstellungen über Einkommen oder Vermögensverteilungen. Sinnvolle Aussagen lassen sich auch hier nur durch genügende Einteilung in Gruppen erreichen.

Beispiel:

Bei einer Aufstellung über Klassenstärken an einer Schule werden als Schülerzahlen festgestellt: 40, 42, 39, 34, 36, 2, 1, 1. Daraus folgt, daß die Schule 8 Klassen hat mit jeweils 24 ± 19 Schülern. Die Rückfrage ergibt, daß es sich bei 2, 1, 1 um Privatschüler handelt, die ebenfalls als Klasseneinheit betrachtet wurden. Die Aufstellung erhält daraufhin die Form: "Die Schule hat 5 Klassen mit je 38 ± 3 Schülern. Weitere 4 Schüler werden privat unterrichtet". Die Angaben sind nun weniger irreführend und zugleich genauer.

Eine solche kritische Durchsicht der eingegebenen Werte empfiehlt sich, wenn die Standardabweichung überraschend groß ist. Werden nur positive Werte eingegeben, so sollte die Standardabweichung etwa $\frac{1}{4}$ des Mittelwertes nicht übersteigen.

Beispiel:

Bei einer Prüfung der Kontaktwiderstände von Relais wird als Spannungsabfall in Millivolt gemessen U = 14, 12, 46, 11 mV. Im Mittel beträgt also der Spannungsabfall U = 21 ± 17 mV. Nach Entfernen der 46 mit h Σ - haben 3 Relais U = 12 ± 2 mV sowie 1 Relais U = 46 mV. Dieses eine Relais kann z.B. als Ausfall weiter untersucht werden.

1.5.3 Verarbeitung von Zahlenmengen

Streuen die einzugebenden Werte stark, so genügt es, zur Auswertung nur die ersten Stellen zu berücksichtigen. Es sollen z.B. die vier Zahlen eingegeben werden:

431.754
585.071
399.482
413.817

Das Ergebnis lautet bei Eingabe der vollständigen Zahlen: 457.531 ± 86.046, bei Eingabe nur der Stellen vor dem Komma: 457.000 ± 86.333.

In beiden Fällen ergibt sich: 4 Zahlen mit 457 ± 86.

Allgemein sollte die Standardabweichung nicht mehr als etwa 3 unterschiedliche Stellen aufweisen. Der Mittelwert ist entsprechend zu runden.

Bei nur geringfügig unterschiedlichen Zahlen genügt andererseits die Eingabe derjenigen Stellen, in denen sich die Zahlen unterscheiden. Dies wird an einigen Beispielen gezeigt.

0.0011		11	
0.0009		9	
0.0012	Eingabe:	12,	also 10 ± 2. Ergebnis: 4 Werte mit
0.0008		8	$(10 \pm 2) \cdot 10^{-4}$

1274.81		481	
1273.98		398	
1275.03	Eingabe:	503,	also 459 ± 45. Ergebnis: 4 Werte mit
1274.55		455	1274.6 ± 0.5

3002.8		28	
3001.9		19	
2999.8		-2	
3001.2		12	
3002.5	Eingabe:	25,	also 14 ± 13. Ergebnis: 6 Werte mit
2999.9		-1	3001.4 ± 1.3

Gelegentlich sind bei Auswertungen auch Größen zu verarbeiten, die gesetzmäßig stark streuen. Dieses Verhalten zeigen z.B. Diodensperrströme bei Integrierten Schaltkreisen. Die Meßwerte in Nanoampere seien

0.4
132
0
8.7
34
211
0.3

Die gewohnte Eingabe ergibt 7 Werte mit 55 ± 84 nA. Die hohen

Werte werden offenbar zu stark gewichtet. Sollen sie für die Auswertung beibehalten werden, so empfiehlt sich der Übergang zum geometrischen Mittel $\bar{x} = \sqrt[n]{x_1 \cdot x_2 \cdot x_3 \cdots x_n}$. Dazu werden mit f LOG jeweils die Logarithmen eingegeben und der Mittelwert mit g 10^x gebildet. Werte "0" werden durch einen sinnvollen Wert ersetzt. Bei einer angenommenen Meßgrenze von 0.1 nA ist statt 0 z.B. 0.07 zu setzen. Dadurch wird ein "Error"-Stop infolge f LOG von 0 vermieden. Das geometrische Mittel ergibt sich im Beispiel zu 5 nA, auf die Standardabweichung wird verzichtet.

2. Das Erstellen von Geradeausprogrammen

2.1 Grundsätzliches

Ein Programm bildet das Ablaufschema für einen Rechengang, welches im Rechner gespeichert werden kann. Die Rechnung wird nach der Dateneingabe gemäß diesem Ablaufschema durchgeführt. Einfache Programme sind die Geradeausprogramme, bei denen auf jede Eingabe nach einem geradlinigen Rechengang das Ergebnis folgt. Etwas schwieriger sind Programme mit Verzweigungen. Bei diesen erfolgt je nach Eingabewert oder Zwischenergebnis im Rechengang die Entscheidung, welcher Zweig des Rechenschemas zum Weiterrechnen gewählt werden soll. Schließlich gibt es Programme mit Schleifen, in denen ein solcher Zweig mehrfach durchlaufen wird.

2.2 Programmieren mit Labeln

Die HP-Rechner 65, 67 und 97 besitzen eine Anzahl Labeltasten (label = Programm-Marke): A bis E beim HP 65, außerdem f a bis f e beim HP 67 und 97. Durch Drücken einer der Labeltasten wird ein Programm unter dieser Taste gestartet. Der Befehl RTN (return = umkehren) beendet die Ausführung eines Programms. Im Programmspeicher ist also eine Rechenvorschrift zwischen Programm-Marke und RTN zu speichern.

Beispiel 2.1: Verdoppeln einer Zahl

Die Rechenvorschrift hierfür lautet

```
Zahl
ENT
 2
 x
```

Soll das gleiche mit einer Reihe Zahlen erfolgen, so empfiehlt sich ein Programm. Die verdoppelte Zahl soll sich nach Drücken der Labeltaste A ergeben. In Stellung RUN ist dies also die Tastenfolge

```
Zahl
 A
```

Zum Eingeben des Programms wird der rechte Schiebeschalter in Stellung W/PRGM (write/program, schreibe/programmiere) geschoben. Beim HP 65 muß nun f PRGM gedrückt werden (zum Löschen der Hilfsfunktionen in der obersten Zeile), beim HP 67/97 wird direkt programmiert. Wie erwähnt, bereitet der Rechner beim Drücken einer Labeltaste den Stack-Lift vor. Dem Label im Programm folgende Zahlen heben die eingegebene Zahl im Stack an. Zur Zahleneingabe mit Label ist also kein Enter zu drücken.

```
f LBL A
  2
  x
h RTN
```

Nach Zurückschieben des rechten Schalters auf RUN wird nach jeder Eingabe A gedrückt:

Zahl	verdoppelte Zahl
3	6
7	14
-5	-10
⋮	⋮

2.3 Programmieren ohne Label

Grundsätzlich können Programme auch ohne Label aufgerufen werden. Dazu ist lediglich der Rechner vor dem Starten an den Beginn des Programmspeichers zu bringen. Dies geschieht in Stellung RUN durch Drücken der RTN-Taste. Die Programmausführung beginnt nach einem Drücken von R/S.

Beim HP 65, 67 und 97 wird in Stellung W/PRGM getastet
f CL PRGM, dann

```
2
x
```

In Stellung RUN wird eingegeben

```
Zahl
h RTN
 R/S
```

Erwartungsgemäß sind die eingegebenen Zahlen verdoppelt.

Zum Programmieren von Rechnern mit Labeltasten werden im allgemeinen die Label auch verwendet. Rechner jedoch wie der HP 25 haben keine Labeltasten. Hier ist das Programmieren ohne Label der normale Weg. Der Rücksprung an den Programmanfang ist von vornherein in den Rechner eingebaut: Alle freien Zeilen im Programmspeicher enthalten die Anweisung "Rücksprung zur Programmzeile Null". Hier ist ein Haltebefehl eingebaut.

Beim HP 25 ist in Stellung W/PRGM zu drücken f CL PRGM, dann

```
ENT
 2
 x
```

In Stellung RUN wird eingetastet

```
Zahl
 R/S
```

Auch hier verdoppeln sich die eingegebenen Zahlen.

2.4 Der Programmspeicher

Mehrfach wurde bereits vom Programmspeicher gesprochen. Nun soll näher auf ihn eingegangen werden.

2.4.1 Der Speicherplatz

Die einzelnen Rechenschritte belegen Speicherplatz im Programmspeicher. Im vorigen Beispiel sind dies 3 Schritte ohne bzw. 5 Schritte mit Label. Je nach Rechnertyp ist der Programmspeicher verschieden groß. So kann der HP 25 Programme bis zu einer Gesamtlänge von 49 Schritten aufnehmen, der HP 65 bis zu 100 Schritten. Beim HP 67 und HP 97 können bis zu 224 Schritte aufgenommen werden. Programme bis zu dieser Länge lassen sich als Ganzes im Rechner speichern. Bei längeren Programmen muß der Rechengang aufgeteilt werden. Z.B. enthält das erste Teilprogramm die Dateneingabe und den Beginn der Rechnung bis zu einem Zwischenergebnis. Das zweite Teilprogramm enthält den nächsten Teil der Rechnung bis zum zweiten Zwischenergebnis und so fort. Geradeausprogramme haben meist bis etwa 30 Rechenschritte, Programme mit Verzweigungen bis etwa 80.

2.4.2 Kennzeichnung von Programmschritten

Beim Eintasten von Programmschritten (Schiebeschalter auf W/PRGM) wird im Anzeigefeld ein Aufleuchten von Zahlengruppen bemerkt. Im vorigen Beispiel ist bei Eingabe in den HP 67:

Tastenfeld HP 67	Anzeigefeld HP 67	
	000	
f LBL A	001	31 25 11
2	002	02
x	003	71
h RTN	004	35 22

Die Zahlengruppen dienen der Orientierung im Programmspeicher. Die linke Zahlengruppe zeigt die jeweilige Zeilennummer im Programmspeicher. Die rechte Zahlengruppe gibt an, welcher Tastenbefehl gegeben wurde. So ist f LBL A als 31 25 11 dargestellt, entsprechend dem Ort der Tasten auf dem Rechner. Also

f als 3. Zeile 1. Taste
LBL 2. Zeile 5. Taste
A 1. Zeile 1. Taste

Dies bedeutet, daß ein Rechner mit einer anderen Belegung des Tastenfeldes auch andere Zahlengruppen zeigt. Der gleiche Programmbeginn lautet beim

```
HP 65:   LBL      23
          A       11   (zwei Programmschritte)
HP 97:   LBL A  21 11
```

Sind im Programm Zahlen enthalten, so werden diese in Stellung W/PRGM auch als Zahlen angezeigt. Z.B. wird die Zahl 3 als 03 angezeigt oder 7 als 07, unabhängig von ihrem Ort im Tastenfeld.

2.4.3 Erreichen von Programmzeilen

Zur Durchsicht oder zum Ändern eines Programms muß der Programmierer zu den einzelnen Programmzeilen gelangen können. Die HP-Rechner bieten dazu eine Anzahl Möglichkeiten.

a) Sprung zu den Labeln. Enthält ein Programm Label, so kann in Stellung RUN durch den Befehl GTO Label zu der gewünschten Programmzeile gesprungen werden. GTO E bedeutet Sprung zur Programm-Marke E, GTO 3 der Sprung zum Label 3. Nach Umschalten auf W/PRGM ist die gesuchte Programmzeile erreicht. Voraussetzung ist natürlich, daß das Programm diese Label auch enthält.

b) Durchlauf zum Programmende. In Stellung RUN wird die gewünschte Labeltaste gedrückt. Das Programm wird dadurch gestartet. Es wird erst durch ein R/S (run/stop, laufen/ anhalten) oder RTN gestoppt. Beim HP 65 wird nach Umschalten auf W/PRGM dieser Schritt angezeigt, beim HP 67 und HP 97 der folgende Schritt. Beim HP 25 erfolgt nach der Ausführung ein selbständiger Rücksprung zur Programmzeile Null und Anhalten. Ein Durchlauf ist beim HP 25 also nur bei R/S-Schritten im Programm möglich, vgl. Kapitel 2.6.

c) Sprung zu den Programmzeilen. In Stellung RUN und auch in Stellung W/PRGM kann gedrückt werden GTO Zeilennummer. Beim HP 67 und 97 erfolgt dadurch ein Sprung zur angegebenen

dreistelligen Zeilennummer. Es bedeutet GTO • 003 Sprung zur Zeile 3, GTO • 224 Sprung zur Zeile 224. Beim HP 25 ist der entsprechende Befehl in Stellung RUN die Anweisung GTO zweistellige Zeilennummer, also GTO 09 Sprung zur Zeile 9, GTO 49 Sprung zur Zeile 49. Beim HP 65 fehlt diese Möglichkeit.

d) Durchtasten vorwärts. Alle vier besprochenen Rechner haben die Taste SST (single step, also Einzelschritt). In Stellung W/PRGM kann sich der Benutzer damit schrittweise durchs Programm tasten. SST wirkt in Vorwärtsrichtung, also in Richtung höherer Programmzeilen. Darüber hinaus kann bei den vier Rechnern in Stellung RUN der Rechengang Schritt um Schritt beobachtet werden. Beim HP 25, 67 und 97 wird beim Drücken der SST-Taste der jeweilige Programmschritt gezeigt, beim Loslassen seine Auswirkung auf den Rechengang. Beim HP 65 ist dazu von RUN auf W/PRGM und umgekehrt zu schalten.

e) Durchtasten rückwärts. Mit der Taste BST (back step, also Rückwärtsschritt) beim HP 25 und 97 bzw. h BST beim HP 67 kann in Stellung W/PRGM schrittweise rückwärts durch das Programm getastet werden. Wird z.B. Zeile 4 gezeigt, so erscheint nach Drücken von BST Zeile 3, nach nochmaligem Drücken Zeile 2 und so fort.

In der folgenden tabellarischen Übersicht sind die Möglichkeiten zum Erreichen beliebiger Programmzeilen für die HP-Rechner zusammengestellt.

Befehl	HP 25	HP 65	HP 67	HP 97
GTO A, GTO 0	-	x	x	x
GTO 13 bzw. GTO • 007	x	-	x	x
A, f e	-	x	x	x
SST	x	x	x	x
BST	x	-	x	x

2.4.4 Ändern eines Programms beim HP 25

Ein Programm läßt sich beliebig ändern indem es neu eingegeben

wird. Außer dieser konsequenten Methode bietet der HP 25 mehrere Möglichkeiten zum Ändern eines Programms.

a) Ersetzen von Rechenschritten
Der Programmzeiger wird in Stellung RUN auf den letzten zu belassenden Schritt gestellt durch den Befehl GTO Zeilennummer (also z.B. GTO 13 oder GTO 27). In Stellung W/PRGM werden die neuen Schritte der Reihe nach eingegeben. Die ursprünglichen Schritte werden dadurch überschrieben.

Beispiel:

Das Programm "Verdoppeln einer Zahl" ist zu ändern. Die eingegebene Zahl ist nun durch 2 zu teilen. Das ursprüngliche Programm lautete

Tastenfeld HP 25	Anzeigefeld HP 25	
ENT	01	31
2	02	02
x	03	61

In Stellung RUN wird gedrückt GTO 02, dann in W/PRGM ÷ .

Die eingegebene Zahl soll jetzt durch 250 geteilt werden. In Stellung RUN wird gedrückt GTO 02, dann in W/PRGM

5
0
÷

Das Programm lautet nun

Tastenfeld HP 25	Anzeigefeld HP 25	
ENT	01	31
2	02	02
5	03	05
0	04	00
÷	05	71

b) Streichen von Programmschritten
Sollen Programmschritte ohne Ersatz gestrichen werden, so

kann jeweils g NOP (no operation, keine Programmwirkung) eingesetzt werden.

Im Beispiel soll nur durch 25 geteilt werden. In Stellung RUN wird gedrückt GTO 03, dann in W/PRGM g NOP. Nun heißt das Programm

Tastenfeld HP 25	Anzeigefeld HP 25
ENT	01 31
2	02 02
5	03 05
g NOP	04 15 74
÷	05 71

Im allgemeinen wird es bei einem so kurzen Programm einfacher sein, den Programmspeicher zu löschen (mit f CL PRGM) und die Schritte neu einzugeben.

c) Einfügen zusätzlicher Schritte
Hierzu wird der nicht belegte Teil des Programmspeichers benutzt. Zu diesem wird mit einem Sprungbefehl gesprungen. Zuerst wird in Stellung RUN mit GTO Zeilennummer der Programmzeiger auf den letzten zu belassenden Schritt gestellt. Dann ist in W/PRGM mit GTO Zeilennummer ein Sprung in den noch freien Teil des Programmspeichers zu programmieren. In Stellung RUN wird eingetastet GTO (Zeilennummer - 1). Die überschriebene Zeile und die neuen Schritte werden in W/PRGM eingegeben. Dann wird der Rücksprung zum ursprünglichen Programm eingegeben.

Das Beispiel soll die eingegebenen Zahlen mit 2/5 malnehmen. Die bisherige Tastenfolge sei

ENT
2
5
÷

Tastenfeld HP 25	Anzeigefeld HP 25	
	00	
ENT	01	31
2	02	02
GTO 06	03	13 06
÷	04	71
GTO 00	05	13 00
x	06	61
5	07	05
GTO 04	08	13 04

Wie zu erkennen, enthalten die Zeile 5 und die Zeilen 9 bis 49 den Befehl GTO 00. Dadurch springt der Programmzeiger nach einer Programmausführung zurück zur Zeile 0. Dort ist ein Haltebefehl eingebaut, der durch ein R/S aufgehoben wird.

2.4.5 Ändern eines Programms beim HP 65, 67 und 97

Zum Ändern eines Programms können beliebig viele Programmschritte herausgenommen, ausgetauscht oder neu eingefügt werden.

a) Einsetzen zusätzlicher Rechenschritte
   Der Programmzeiger wird auf die letzte zu belassende Zeile gestellt. In W/PRGM werden die weiteren Rechenschritte eingetastet.

Beispiel:

Im vorigen Programm sind die eingegebenen Zahlen statt mit 2 mit 250 malzunehmen. Die Tastenfolge am HP 67:

In Stellung RUN	GTO • 003
In Stellung W/PRGM	5
	0

Das Programm lautet jetzt also

Tastenfeld HP 67	Anzeigefeld HP 67	
f LBL A	001	31 25 11
2	002	02
5	003	05
0	004	00
x	005	71
h RTN	006	35 22

Nach Zurückschalten auf RUN wird ermittelt

Zahl	250 x Zahl
0.1	25
2	500
-1	-250
:	:

Wie erinnerlich, ist bei -1 die Tastenfolge 1 CHS, nicht die Verknüpfungstaste - zu verwenden.

b) Löschen von Rechenschritten
Der Programmzeiger wird auf den zu löschenden Programmschritt gesetzt und in W/PRGM h DEL (delete, streichen) gedrückt. Dieser Schritt wird gelöscht, die vorausgehende Programmzeile wird sichtbar. Sollen mehrere aufeinanderfolgende Schritte gelöscht werden, so wird der Programmzeiger zweckmäßig auf den letzten zu löschenden Schritt gestellt und mehrmals h DEL gedrückt. Falls in einem Programm z.B. die Schritte 37 bis 41 gelöscht werden sollen: In Stellung RUN GTO • 041, dann in W/PRGM 5 mal h DEL.

Beispiel:

Im vorigen Programm sollen die eingegebenen Zahlen nicht mit 250 malgenommen, sondern es soll jeweils 2 hinzugezählt werden. Beim HP 67 ist einzugeben in Stellung RUN GTO • 006

dann in Stellung W/PRGM h DEL
h DEL
h DEL
+

Das Programm enthält jetzt die Schritte

```
f LBL A
   2
   +
 h RTN
```

In Stellung RUN erhält man:

Zahl	(Zahl + 2)
2	4
10	12
-2	0
⋮	⋮

2.5 Kurze Geradeausprogramme

Einfache Programme dieser Art sind bei der täglichen Arbeit von großem Nutzen. Daher hier noch einige kurze Geradeausprogramme.

Beispiel 2.2: Berechnen von Bruttopreisen

Zu Nettopreisen wird 13 % Mehrwertsteuer erhoben. Gesucht sind die Bruttopreise, aufzurufen durch Taste B. Lösung: Die Nettopreise werden mit 1.13 malgenommen. Das Programm:

```
f LBL B
   1
   .
   1
   3
   x
 h RTN
```

Nettopreise	Bruttopreise
100.--	113.00
21.--	23.73
44.25	50.00
⋮	⋮

Sind aus den Bruttopreisen die entsprechenden Nettopreise zu berechnen, so wird durch 1.13 geteilt.

Ist der Programmspeicher bereits voll belegt, so ist bei Pro-

grammänderungen Vorsicht geboten. Grundsätzlich sollten zuerst die zu entfernenden Schritte gelöscht werden, ehe weitere Schritte eingefügt werden. Andernfalls würden die letzten Programmzeilen (beim HP 65 Nr. 100, 99, 98 ... bzw. beim HP 67 oder 97 Nr. 224, 223, 222 ...) beim Einfügen weiterer Schritte aus dem Programmspeicher hinausgedrängt. Sie wären damit verloren. Nach dem anschließenden Löschen sind diese Programmzeilen beim HP 65 mit g NOP (no operation, also keine Programmwirkung) bzw. beim HP 67 oder 97 mit R/S besetzt. Während durch R/S wenigstens ein Programmstop bewirkt wird, läuft beim HP 65 dann das Programm ohne Halt durch. Der Durchlauf kann durch Drücken einer beliebigen Taste abgebrochen werden.

Beispiel 2.3: Funktionswerte einer logarithmischen Darstellung

Aus einer logarithmischen Darstellung sollen die Funktionswerte möglichst genau entnommen werden. In der Zeichnung seien als Bezugslinien lediglich die Zehnerpotenzen mit einem Abstand a = 47.3 mm eingetragen. Ein Funktionswert b hat in der Darstellung den in mm gemessenen Abstand von der Linie mit dem Wert Eins $c = a \cdot \lg b$. Damit ist $b = 10^{\frac{c}{a}}$.

		Gemessene Strecke in mm	Funktionswerte
f LBL C	c wird eingegeben		
4		0	1.00
7		47.3	10.00
•		21.5	2.85
3	a = 47.3	41.8	7.65
÷		68.3	27.80
g 10^x	$b = 10^{\frac{c}{a}}$	90.1	80.33
h RTN		⋮	⋮

Beispiel 2.4: Umrechnung von Temperaturangaben ^{o}C in F und F in ^{o}C

Lösung: Mit 0 ^{o}C = 32 F und 100 ^{o}C = 212 F folgt

$$t_{o_F} = 32 + 1.8 \cdot t_{o_C} \quad \text{oder} \quad t_{o_C} = \frac{t_{o_F} - 32}{1.8}$$

f LBL D Eingabe in Celsius	f LBL E Eingabe in Fahrenheit
1	3
.	2
8	-
x	1
3	.
2	8
+ Temperatur in Fahrenheit	÷ Temperatur in Celsius
h RTN	h RTN

Beispiel 2.5: Hyperbolische Funktionen sh x und ch x

Die Funktionen Hyperbelsinus $\text{sh } x = \frac{e^x - e^{-x}}{2}$ und Hyperbelkosinus $\text{ch } x = \frac{e^x + e^{-x}}{2}$ sollen auf Label A bzw. Label B aufgerufen werden.

1	f LBL A	sh x		f LBL B	ch x
	g e^x		10	g e^x	
	ENT			ENT	
	h $\frac{1}{x}$			h $\frac{1}{x}$	
5	-			+	
	2			2	
	÷		15	÷	
	h RTN			h RTN	

In den Programmen wird ein Enter notwendig, da g e^x nur im X-Register steht und durch h $\frac{1}{x}$ in e^{-x} umgewandelt wird. Durch Enter wird der jeweilige Wert von g e^x ins Y-Register kopiert. Dort steht er dann zum Abziehen oder Zuzählen zur Verfügung.

Wie in den Beispielen gezeigt werden Programme zweckmäßig von oben nach unten geschrieben. Jeder Programmschritt entspricht dabei einer Programmzeile. Die Tasten-Codes brauchen nicht aufgeführt zu werden. Bei längeren Programmen sollte jedoch alle 5 oder 10 Zeilen die jeweilige Zeilen-Nr. vermerkt sein.

Die Programme bleiben dadurch übersichtlich.

2.6 Programme mit R/S

Verschiedentlich ist schon die Tastenfunktion R/S erwähnt worden. R/S ist dabei die Abkürzung von run/stop, also Programmlauf/Halt. Enthält ein Programm den Schritt R/S, so wird die Ausführung des Programms an dieser Stelle angehalten. Wird nun in Stellung RUN die Taste R/S gedrückt, so wird die Programmausführung fortgesetzt. Der Programmstop durch ein R/S dient zur Dateneingabe wie auch zur Ausgabe von Teilergebnissen. R/S ist daher eine wichtige Hilfe beim Erstellen übersichtlicher Programme.

Beispiel 2.6: Raumdiagonale eines Quaders

Gesucht ist die Raumdiagonale eines Quaders mit den Seiten a, b, c. Die Seite a soll beim Programmstart eingegeben werden, b und c jeweils mit R/S. Lösung: $d = \sqrt{a^2 + b^2 + c^2}$. Frühere Programme werden gelöscht durch Aus- und wieder Einschalten des Rechners oder in W/PRGM durch f CL PRGM.

```
1    f LBL A     Eingabe von a
       g x²
       R/S       Eingabe von b
       g x²
5       +
       R/S       Eingabe von c
       g x²
        +
       f √x      d = √(a² + b² + c²)
10     h RTN
```

Seiten			Diagonale
a	b	c	d
1	1	1	1.73
$\sqrt{3}$	$\sqrt{3}$	$\sqrt{3}$	3 00
3	4	5	7.07

Natürlich kann diese Aufgabe auch ohne R/S gelöst werden, zum Beispiel durch die kompakte Dateneingabe a ↑ b ↑ c (↑ = Enter), gefolgt vom Drücken der Labeltaste. Dieses Verfahren wird bei den offiziellen HP-Programmen bevorzugt. Allerdings hat das Auffüllen des Stack zur Dateneingabe gewisse Nachteile: Zum einen können höchstens 4 verschiedene Daten eingegeben werden, dann ist der Stack voll. Zum anderen muß sich der Programmierer bei jedem Rechenschritt die Belegung des Stack vergegenwärtigen. Ein mehrfaches Vertauschen der Stackinhalte durch h x⇄y, h R↑ und h R↓ ist dabei die Regel. Zum Vergleich sei ein Programm für die Raumdiagonale mit kompakter Dateneingabe angegeben. In RUN wird eingegeben a ↑ b ↑ c, dann Drücken von Labeltaste B.

```
11      f LBL B
          g x²
          h R↓
          g x²
15        h R↓
          g x²
          h R↓
          h R↓
           +
20         +
        f √x
         h RTN
```

Zum Erstellen eines solchen Programms sollte für jeden Rechenschritt die Belegung des Stack aufgezeichnet werden.

Beispiel 2.7: Gleichung mit einer Unbekannten

Aus der Beziehung $a_0 x = a_1$ kann die Unbekannte $x = \frac{a_1}{a_0}$ ermittelt werden.

```
1       f LBL A      a0
          R/S        a1
           ÷
          h 1/x      x = a1/a0
5         h RTN
```

Die Run-Stop-Taste R/S kann auch zur Ausgabe von Teilergebnissen verwendet werden. Nach Ablesen des Ergebnisses läuft das Programm bei Drücken von R/S weiter.

Beispiel 2.8: $f_1(x) = \sqrt{1 + x^2}$ und $f_2(x) = \frac{2}{\sqrt{1 + x^2}}$

Für die Veränderliche x sollen die Funktionen $\sqrt{1 + x^2}$ sowie $\frac{2}{\sqrt{1 + x^2}}$ gebildet werden. Der zweite Ausdruck soll nach Drücken von R/S berechnet werden.

```
 6      f LBL C        Eingabe von x
          g x²
10          1
            +
          f √x
          R/S          Ausgabe von √(1 + x²)
          h 1/x
15          2
            x
        h RTN          Ausgabe von 2/√(1 + x²)
```

Mit dem Programm läßt sich nun eine Wertetabelle aufstellen:

x	0	0.5	1	1.5	2	...
$\sqrt{1 + x^2}$	1.00	1.12	1.41	1.80	2.24	...
$\frac{2}{\sqrt{1 + x^2}}$	2.00	1.79	1.41	1.11	0.89	...

2.7 Die Speicherregister

Zum Abspeichern von Eingabewerten oder Zwischenergebnissen stehen in den HP-Rechnern eine Anzahl Speicherregister zur Verfügung. Außer dem Last-X-Register sind es 8 Speicher beim HP 25, 9 beim HP 65 und 26 beim HP 67 und 97.

2.7.1 Das Last-X-Register

Das einfachste dieser Register ist das Last-X-Register. Es ist an den Stack angeschlossen und wird mit h LST X (last X, also letzter x-Wert vor der Verarbeitung) aufgerufen. Wie erinnerlich, kommen eingegebene Zahlen in das X-Register des Stack. Werden sie mit anderen Zahlen verknüpft oder wird eine Funktion gebildet, so kommen die eingegebenen Zahlen ins Last-X-Register.

Der Rechner werde jeweils in Stellung RUN eingeschaltet. Dann enthält das Last-X-Register:

3 + 4	(= 7)	Last X	3 ENT 4 +
sin 30°	(= 0.50)	Last X	30 f SIN
$\sqrt{7}$	(= 2.65)	Last X	7 f $\sqrt{x}$
3^5	(= 243)	Last X	3 ENT 5 h y^x

Beim Verwenden des Last-X-Registers wird beim Rechnen auf dem Tastenfeld die Eingabe häufig einfacher. Insbesondere entfällt eine sonst erforderliche Mehrfacheingabe von Zahlen.

Beispiel 2.9: $f(x) = \sin x + \sqrt{x}$

Es soll das Last-X-Register verwendet werden.

```
f LBL D
 f SIN
h LST X
 f √x
   +
 h RTN
```

x	0	5	10	15	20	···
f(x)	0	2.32	3.34	4.13	4.81	···

Auch das nächste Beispiel soll mit LST X gerechnet werden.

Beispiel 2.10: Relative Änderung

Die relative Änderung in Prozent ist gegeben durch

$$\frac{\text{(neuer Wert - alter Wert)}}{\text{alter Wert}} \cdot 100\ \%$$

```
f LBL E      neuer Wert
  R/S        alter Wert
   -
h LST X      alter Wert
   ÷
  EEX
   2
   x
 h RTN
```

In Stellung RUN also

```
neuer Wert
    E
alter Wert
   R/S
```

neuer Wert	alter Wert	rel. Änderung in %
88	55	60.0
224	219	2.3
118	134	-11.9
⋮	⋮	⋮

2.7.2 Speichern und Zurückrufen von Zahlen

Unabhängiger in der Speicherung als das Last-X-Register sind die eigentlichen Speicherregister. Dies sind beim HP 25 die Register 0 bis 7 und beim HP 65 die Register 1 bis 9. Beim HP 67 und HP 97 sind es die Primärregister 0 bis 9, A bis E und I, außerdem die Sekundärregister S 0 bis S 9, also insgesamt 26 Datenspeicher. In den Registern 0 bis 9 und A bis E wird eine Zahl gespeichert durch die Tastenfolge

Abzuspeichernde Zahl
STO Name des Registers.

Dabei bewirkt der Rechner selbsttätig den Stack-Lift. Der Registerinhalt wird ins X-Register des Stack geholt durch

RCL Name des Registers.

Daten im I-Register speichert der Rechner bzw. ruft sie zurück mit den Anweisungen h STI und h RCI. Hierauf wird bei der Besprechung der Anwendungen des I-Registers in Kapitel 5.3 weiter eingegangen.

Bei Speicheroperationen zeigen das X-Register des Stack und das jeweilige Speicherregister (hier Register 1) folgende Belegung:

Speicherregister 1		27	27	27	0	27
X-Register des Stack	27	27	0	27	27	27
jeweilige Tastenfolge	27	STO 1	CLX	RCL 1	f CL REG	STO 1

Hierbei bedeutet CLX (clear X) Löschen des X-Registers und damit der Anzeige. Der Befehl f CL REG (clear registers) löscht die Inhalte aller zugänglichen Speicherregister. Dies sind die 8 bzw. 9 Register beim HP 25 oder HP 65 sowie die 16 Primärregister beim HP 67 und HP 97.

Beispiel 2.11: $f(x) = \frac{x + 1}{x - 1}$

Für die Funktion $f(x) = \frac{x + 1}{x - 1}$ soll eine Funktionstabelle aufgestellt werden.

```
f LBL A
 STO 1          x für Zähler
   1
   +
 RCL 1          x für Nenner
   1
   -
   ÷
 h RTN
```

x	0	0.5	0.9	1.1	1.5	...
f(x)	-1	-3	-19	21	5	...

Beispiel 2.12: Maßstabsänderung

Die Beziehung $g = a \cdot x$ wird durch die Schrittfolge gegeben

```
f LBL A        Eingabewert x
 RCL 1         Faktor a
   x           g = ax
 h RTN
```

Der Faktor a wird in Stellung RUN vor Beginn der x-Eingabe in Register 1 gespeichert.

Beispiel:

Umrechnung von Inch in Millimeter

In Stellung RUN wird gedrückt

```
2
5
.
4
STO 1
```

inch	mm
1	25.4
15	381.0
37.5	952.5
⋮	⋮

Beispiel:

Umrechnung von Gramm in Unzen

```
2
8
.
3
5
h 1/x
STO 1
```

g	ounce
10	0.35
28.35	1.00
115	4.06
⋮	⋮

Beispiel:

Verkleinerung im Maßstab 1 : 25

```
2
5
h 1/x
STO 1
```

Original in mm	Verkleinerung in mm
1000	40.0
485	19.4
211	8.4
⋮	⋮

Weitere Anwendungen sind Wechselkurse, Rabatte, Normierungen aller Art.

Beispiel 2.13: $f(x) = \sqrt{1 + x^2}\ (1 + \sqrt{1 + x^2})$

Für die Funktion soll eine Wertetabelle aufgestellt werden.

```
f LBL B      Eingabe von x
  g x²
   1
   +
  f √x
 STO 8       √(1 + x²)
   1
   +
 RCL 8
   x         √(1 + x²) (1 + √(1 + x²))
 h RTN
```

x	0	1	2	3	...
f(x)	2	3.41	7.24	13.16	...

Die Speicherregister eignen sich also ebenso zum Abspeichern von Konstanten wie auch von veränderlichen Größen.

Beispiel 2.14: Ellipsengleichung

Die Ellipsen $y = b\sqrt{1 - (\frac{x}{a})^2}$ sollen für verschiedene a und b gezeichnet werden. a und b sind also je Ellipse konstant, sollen aber im Programm beliebige Werte annehmen können. Sie werden zweckmäßig in den Speicherregistern abgespeichert, z.B. a in Register 2 und b in Register 3.

```
f LBL C          x
  RCL 2          a
    ÷
  g x²
   CHS
    1
    +
  f √x
  RCL 3          b
    x            y = b √(1 - ( x/a )²)
  h RTN
```

Die erste Ellipse habe a = 5 und b = 3. In Stellung RUN wird daher eingegeben

```
  5
STO 2
  3
STO 3
```

Es ergibt sich

x	0	1	2	3	...
y	± 3	± 2.94	± 2.75	± 2.40	...

Beispiel 2.15: Gleichungen mit zwei Unbekannten

Die Gleichungen

$$a_0\, x + a_1\, y = a_2$$
$$b_0\, x + b_1\, y = b_2$$

haben die Lösungen

$$x = \frac{a_2\, b_1 - a_1\, b_2}{a_o\, b_1 - a_1\, b_o}$$

und

$$y = \frac{a_o\, b_2 - a_2\, b_o}{a_o\, b_1 - a_1\, b_o}$$

In die Speicherregister sind also die sechs Größen a_o, a_1, a_2 und b_o, b_1, b_2 einzugeben. Während der Berechnung von x wird zur leichteren Berechnung von y der Nenner gespeichert.

1	f LBL B		21	RCL 4	
	STO 0	a_o		x	
	R/S			RCL 1	
	STO 1	a_1		RCL 3	
5	R/S		25	x	
	STO 2	a_2		-	
	R/S			STO 6	Nenner (= $a_o\, b_1 - a_1\, b_o$)
	STO 3	b_o		÷	$x = \frac{a_2\, b_1 - a_1\, b_2}{a_o\, b_1 - a_1\, b_o}$
	R/S			R/S	
10	STO 4	b_1	30	RCL 0	
	R/S			RCL 5	
	STO 5	b_2		x	
	CHS			RCL 2	
	RCL 1			RCL 3	
15	x		35	x	
	RCL 2			-	
	RCL 4			RCL 6	$y = \frac{a_o\, b_2 - a_2\, b_o}{a_o\, b_1 - a_1\, b_o}$
	x			÷	
	+		39	h RTN	
20	RCL 0				

Die Schrittliste ist im Anhang nochmals aufgeführt.
Zur Eingabe wird nach der ersten Größe a_o die Labeltaste B gedrückt, dann nach jeder weiteren Größe R/S. Nach der Eingabe von b_2 erfolgt die Berechnung von x. Nach erneutem Drücken von R/S wird y berechnet und angezeigt.

Bei den Gleichungen $3\,x + 4\,y = 10$
$2\,x - 3\,y = 1$

ist in Stellung RUN zu drücken

3	2
B	R/S
4	3
R/S	CHS
1	R/S
0	1
R/S	R/S

Hierauf wird x berechnet und angezeigt, dann nach Drücken von R/S y. Ergebnis: x = 2
y = 1.

Entsprechendes ergibt sich z.B. für $x + y = 2$
$2x - y = 13$

$x + y = -1$
$x - y = 3$

Beispiel 2.16: Gleichungen mit drei Unbekannten

Die Gleichungen $a_0 x + a_1 y + a_2 z = a_3$
$b_0 x + b_1 y + b_2 z = b_3$
$c_0 x + c_1 y + c_2 z = c_3$
enthalten die drei Unbekannten x, y, z sowie die zwölf bekannten Größen a_0 bis a_3, b_0 bis b_3 und c_0 bis c_3. Lineare Gleichungen dieser Art lassen sich besonders übersichtlich mit Determinanten lösen als $x = \frac{D_1}{D}$, $y = \frac{D_2}{D}$ und $z = \frac{D_3}{D}$. Hierbei ist mit D das Rechenschema gemeint

$$D = \begin{vmatrix} a_0 & a_1 & a_2 \\ b_0 & b_1 & b_2 \\ c_0 & c_1 & c_2 \end{vmatrix} = a_0 (b_1 c_2 - c_1 b_2) - a_1 (b_0 c_2 - c_0 b_2) + a_2 (b_0 c_1 - c_0 b_1)$$

und entsprechend

$$D_1 = \begin{vmatrix} a_3 & a_1 & a_2 \\ b_3 & b_1 & b_2 \\ c_3 & c_1 & c_2 \end{vmatrix} = a_3 (b_1 c_2 - c_1 b_2) - a_1 (b_3 c_2 - c_3 b_2) + a_2 (b_3 c_1 - c_3 b_1)$$

$$D_2 = \begin{vmatrix} a_0 & a_3 & a_2 \\ b_0 & b_3 & b_2 \\ c_0 & c_3 & c_2 \end{vmatrix} = a_0 (b_3 c_2 - c_3 b_2) - a_3 (b_0 c_2 - c_0 b_2) + a_2 (b_0 c_3 - c_0 b_3)$$

$$D_3 = \begin{vmatrix} a_0 & a_1 & a_3 \\ b_0 & b_1 & b_3 \\ c_0 & c_1 & c_3 \end{vmatrix} = a_0 (b_1 c_3 - c_1 b_3) - a_1 (b_0 c_3 - c_0 b_3) + a_3 (b_0 c_1 - c_0 b_1).$$

Eine Programmliste ist im Anhang wiedergegeben. Wie in Beispiel 2.15 werden die bekannten Größen zeilenweise eingegeben, a_0 auf Label C, a_1 und die weiteren als R/S. Nach Eingabe von c_3 wird x berechnet und angezeigt, nach Drücken von R/S y und nach weiterem R/S z. Im Programm sind drei Speicher doppelt belegt: Wird der Inhalt eines Speichers nicht mehr benötigt, so kann der Speicher statt dessen ein Zwischenergebnis aufnehmen.

Es sind auf diese Weise die folgenden Gleichungen zu lösen:

x + y + z	=	6	5 x + y + z	=	3
x - y - z	=	-4	9 x - 2 y + z	=	0
- 3 x + 2 y + z	=	4	x + 3 y + z	=	5
x + y + z	=	1	3 x + 4 y + 3 z	=	9
- x + y + 3 z	=	-1	7 x - 2 y + 0	=	0
- 2 x - 3 y + 4 z	=	6	2 x + 5 y - 2 z	=	-6

Beispiel 2.17: Zinseszinstabelle

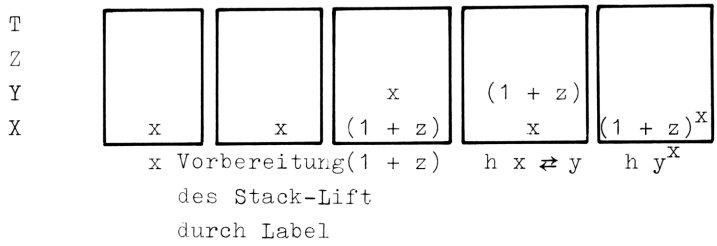
In einer Tabelle soll das Wachstum von Kapital mit der Anlagedauer dargestellt werden. Es gilt für das Kapitalwachstum

$$\frac{\text{Endkapital}}{\text{Anfangskapital}} = (1 + \text{Zinssatz})^{\text{Anlagedauer}}$$

Der Wachstumsfaktor (1 + Zinssatz) wird zu Beginn abgespeichert, im Beispiel in Register 1. Ein Zinssatz von 5 % wird als Wachstumsfaktor 1.05 abgespeichert, einer von 8.5 % als 1.085. Als Eingabe werde die Anlagedauer verwendet. Das Programm lautet also:

```
1       f LBL A     Anlagedauer  x
         RCL 1      (1 + Zinssatz)
        h x⇄y       jetzt ist die Anlagedauer x im X-Register
         h y^x
5        h RTN
```

Im Programm werden die Inhalte der Stackregister X und Y vertauscht. Dadurch kann bei der gewählten Eingabe die Funktion y^x verwendet werden.



Die Tabelle des Kapitalwachstums sieht wie folgt aus:

Anlagedauer in Jahren	Zinssatz in %					
	2	4	6	8	10	...
2	1.0404	1.0816	1.1236	1.1664	1.2100	
4	1.0824	1.1699	1.2625	1.3605	1.4641	
6	1.1262	1.2653	1.4185	1.5869	1.7716	
8	1.1717	1.3686	1.5938	1.8509	2.1436	
⋮			⋮			

Beispiel 2.18: Vermögensbildung durch Ratensparen

In gleichen Abständen werden Raten auf ein Sparkonto eingezahlt und dort verzinst. Das gebildete Kapital ist

$$K = a \frac{(1 + b)^n - 1}{b}$$

a = Ratenhöhe
b = Zinsfaktor
n = Zahl der Raten

Das entsprechende Programm:

```
 6      f LBL B
         STO 1        Ratenhöhe a
          R/S         Zinssatz in %
          EEX
10          2
            ÷
         STO 2        Zinsfaktor b
            1
            +
15        R/S        Zahl der Raten n
         h y^x
            1
            -
         RCL 1
20          x
         RCL 2
            ÷
         h RTN
```

Die Größen a, b und n beziehen sich auf die gleichen Zeiten. Z.B. betragen die monatlichen Raten 450.- DM, der jährliche Zinssatz 5 1/2 % und die Laufzeit 5 Jahre. Der Monatszins errechnet sich zu $\frac{5\ 1/2\ \%}{12}$ = 0.4583 %, die Zahl der Monatsraten zu 5 · 12 = 60.

Es wird in Stellung RUN eingegeben:

4	5
5	ENT
0	1
B	2
5	x
.	R/S
5	
ENT	
1	
2	
÷	
R/S	

Das Gesamtkapital beträgt nach diesen fünf Jahren 30996.- DM, als Sparraten eingezahlt wurden davon 27000.- DM. Im Prinzip muß ein solches Ratensparen nicht auf einem Sparkonto stattfinden, wichtig ist nur eine möglichst hohe Verzinsung des eingesetzten Kapitals sowie eine Wiederanlage der Erträge.

Ist es demnach möglich, durch energisches Sparen während des Berufslebens eine Million DM zu erwerben? Ohne Zins im allgemeinen sicher nicht, denn 1 Million in 40 Jahren bedeutet 2083.- DM monatliche Sparrate. Mit Zinsen sieht es jedoch günstiger aus:

Jemand spart mit aller Kraft (Vorschlag: 1000.- DM im Monat) 20 Jahre lang. Die Verzinsung betrage 5 %. Anschließend wird nicht mehr gespart. Das gebildete Kapital wird mit 5 % für weitere 20 Jahre wieder angelegt. Nach den 20 Jahren Ratensparen beträgt das Kapital 411034.- DM, davon sind nur 240000.- eingezahlt. In den nächsten 20 Jahren wächst das Kapital bei Wiederanlage auf 1.09 Millionen DM.

Beispiel 2.19: Wertpapier-Rendite

Kurslisten von festverzinslichen Wertpapieren enthalten die Nominalverzinsung in Prozent, den Tageskurs für 100.- DM-Stückelung und die restliche Laufzeit. Das Programm errechnet die gegenwärtige Rendite (d.h. die laufende Verzinsung des eingesetzten Kapitals) sowie die effektive Rendite bezogen auf die gesamte Laufzeit. Es ist

$$\text{Gegenwärtige Rendite} = \text{Nominalzins} \times \frac{100}{\text{Tageskurs}}$$

Die effektive Rendite enthält den zusätzlichen Zins aus dem Unterschied zwischen Rückzahlungskurs (= 100.- DM) und Tageskurs, bezogen auf die restliche Laufzeit. Dieser Zusatzzins ist annähernd

$$\text{Zusatzzins} = \frac{\text{Rückzahlungskurs} - \text{Tageskurs}}{\text{Restlaufzeit}} \text{ in Prozent}$$

Für eine genaue Rechnung folgt aus

$$\text{Endkapital} = \text{Anfangskapital} \times (\text{Verzinsungsfaktor})^{\text{Laufzeit}}$$

sinngemäß

$$100 = \text{Tageskurs} \times \left(1 + \frac{\text{Zusatzzins}}{100}\right)^{\text{Restlaufzeit}}.$$

Damit ist

$$\text{Zusatzzins} = 100 \times \left[10^{\frac{\lg\left(\frac{100}{\text{Tageskurs}}\right)}{\text{Restlaufzeit}}} - 1 \right].$$

Es wird eingegeben auf Label A der Nominalzins in Prozent, auf B der Tageskurs und auf C die Restlaufzeit. Errechnet wird in B die gegenwärtige und in C die effektive Rendite.

```
1   f LBL A          f LBL B              f LBL C
     STO 0       5    STO 1          15    h 1/x
     h RTN             h 1/x                EEX
                      RCL 0                  2
                        x                  RCL 1
                       EEX                    ÷
                10      2             20    f LOG
                        x                     x
                      STO 2 gegenwärtige    g 10^x
                      h RTN Rendite           1
                                              -
                                      25     EEX
                                              2
                                              x     Zusatzzins
                                            RCL 2
                                              +     effektive Ren-
                                      30    h RTN   dite
```

Nominal-zins %	Tageskurs DM	restliche Laufzeit Jahre	gegenwärtige Rendite %	effektive Rendite %
A	B	C	(B)	(C)
5.5	92	11	5.98	6.74
6	95	13	6.32	6.71
7	101	22	6.93	6.89
⋮			⋮	

Für eine kurzfristige Geldanlage interessiert vorwiegend die gegenwärtige Rendite, die Restlaufzeit braucht nicht angegeben zu werden.

2.7.3 Speicherarithmetik

Soll eine Zahl dem Inhalt eines Registers zugezählt werden, so kann dies in jedem Fall über den Stack erfolgen:

```
RCL  Name des Registers
Zähle  Zahl zu
STO  Name des Registers.
```

Das Register 1 enthalte die Zahl 27. Sollen hierzu 4 hinzugezählt werden, so sind dies die Rechenschritte

```
RCL 1      (= 27)
  4
  +        (= 31)
STO 1
```

Entsprechend gilt für die Rechenoperationen −, x und ÷.

Die HP-Rechner haben außerdem eine erweiterte Speicherarithmetik. Diese besitzen sämtliche Speicher des HP 25 und HP 65. Beim HP 67 und HP 97 sind es die Speicherregister 0 bis 9 und die Sekundärregister S 0 bis S 9. Hier wird durch die Schrittfolge

```
Zahl
STO  Rechenoperation  Registername
```

in einem Rechenschritt die angegebene Zahl dem bisherigen Speicherinhalt zugeordnet.

Es kann also wahlweise mit dem Stack oder der Speicherarithmetik gerechnet werden:

Vorgang	Stack	Speicherarithmetik
1 zum Inhalt von Speicher 0 zuzählen	RCL 0 1 + STO 0	1 STO + 0
7 vom Inhalt von Speicher 3 abziehen	RCL 3 7 - STO 3	7 STO - 3
Inhalt von Speicher 5 mit 3 malnehmen	RCL 5 3 x STO 5	3 STO x 5
Inhalt von Speicher 9 durch 2 teilen	RCL 9 2 ÷ STO 9	2 STO ÷ 9

In Programmen wird häufig die Speicherarithmetik gewählt, u.a. um die Programme kurz und übersichtlich zu halten. Bei Programmen mit Speicherarithmetik sollte vor der Dateneingabe f CL REG stehen oder das betreffende Speicherregister zuerst mit STO Registername belegt werden. So wird vermieden, daß Speicherinhalte einer vorhergehenden Rechnung zu Fehlern führen.

Beispiel 2.20: Verkehrszählung

Bei einer Verkehrszählung soll zwischen innerstädtischem und dem Stadt-Umlandverkehr unterschieden werden. Bei Autos aus der eigenen Stadt soll Taste A gedrückt werden, bei auswärtigen Autos Taste E.

```
1     f LBL A          In Stellung RUN wird gedrückt DSP 0
        1              (display 0, Anzeige ohne Stellen nach
      STO + 0          dem Komma) sowie vor Beginn jeder Zähl-
       RCL 0           reihe f CL REG.
5      h RTN
-------------------
      f LBL E
        1
      STO + 1
       RCL 1
10     h RTN
```

Beispiel 2.21: Widerstand einer Parallelschaltung

In der Elektrotechnik ist der Gesamtwiderstand von n parallel geschalteten Teilwiderständen

$$\frac{1}{R_{ges}} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3} + \cdots + \frac{1}{R_n}$$

Das entsprechende Programm lautet

```
1     f LBL A
        h 1/x          Eingabe der R_i
      STO + 1
       h RTN
-------------------------------------
5     f LBL B
       RCL 1
        h 1/x          Anzeige von R_ges
      f CL REG         Löschen der Speicher
       h RTN
```

Die Teilwiderstände werden auf Label A eingegeben. Drücken von B bewirkt das Berechnen des Gesamtwiderstandes. Zugleich wird Speicher 1 gelöscht. Die nächste Datenreihe kann eingegeben werden. Für $R_1 = R_2 = 10$ Ohm folgt $R_{ges} = 5$ Ohm. $R_1 = 10$, $R_2 = 8$ und $R_3 = 6$ Ohm ergeben $R_{ges} = 2.55$ Ohm.

Ersichtlich kann das Programm allgemein für Ausdrücke der Form $\frac{1}{A_{ges}} = \frac{1}{A_1} + \frac{1}{A_2} + \frac{1}{A_3} + \cdots + \frac{1}{A_n}$ verwendet werden. Dies wäre

z.B. die Kapazität von hintereinander geschalteten Kondensatoren, aber auch die Berechnung von Leitwerten in der Vakuumtechnik oder der Brennweite von Linsensystemen.

Beispiel 2.22: Einwiegen zweier Stoffe

Zwei Stoffe sollen in gegebenem Verhältnis gemischt werden. Wieviel der Stoffmenge 2 ist zum Leergewicht + Stoffmenge 1 hinzuzufügen?

1	f LBL A	Leergewicht (Tara)
	R/S	
	STO 2	Leergewicht + Stoffmenge 1
	-	
5	CHS	Stoffmenge 1 allein
	RCL 1	Mischverhältnis
	x	
	RCL 2	
	+	Leergewicht + Stoff 1 + Stoff 2
10	h RTN	

Zu Beginn wird das Mischverhältnis $\frac{\text{Stoff 1}}{\text{Stoff 2}}$ in Register 1 eingegeben. Das Leergewicht wird auf Label A eingegeben, nach der Zugabe des ersten Stoffes Leergewicht + Stoff 1 auf R/S. Das zu erreichende Gesamtgewicht wird angezeigt.

Es sei das vorgegebene Mischungsverhältnis $\frac{\text{Stoff 1}}{\text{Stoff 2}} = 15$, das Leergewicht 3.1 g. Vom ersten Stoff werde eine kleinere Menge eingewogen, so daß (Leergewicht + Stoff 1) zusammen 7.2 g wiegen. Wie hoch ist das Gesamtgewicht nach Zugabe von Stoff 2?

In Stellung RUN wird gedrückt

1	, dann 3
5	.
STO 1	1
	A
	7
	.
	2
	R/S.

Ergebnis: Der Wägebehälter muß mit Stoff 2 zum Gesamtgewicht 68.7 g aufgefüllt werden.

Häufig ist beim Aufstellen von Wertetabellen oder beim Zeichnen von Kurven eine Veränderliche schrittweise zu erhöhen. Dies läßt sich einfach mit den Programmzeilen durchführen

```
schrittweise Änderung
STO + Registername.
```

In den folgenden Beispielen wird dies gezeigt.

Beispiel 2.23: Bild einer Spirale

Die Spirale $r = \frac{\sqrt{\varphi}}{4}$ soll für vier Umläufe gezeichnet werden, die schrittweise Änderung betrage $\Delta\varphi = 30^{\circ}$. Dazu werden vom Nullpunkt aus strahlenförmig alle 30° dünne Hilfslinien gezeichnet. Auf diesen wird der jeweilige Abstand r vom Nullpunkt vermerkt.

```
 1      f LBL A
          3
          0
        STO + 1
 5      RCL 1
        h PAUSE
          f √x
          4
          ÷
10       h RTN
```

φ	r
0	0
15	0.97
30	1.37
60	1.94
90	2.37
⋮	⋮

Um Werte für r zu erhalten, wird ohne Zahleneingabe Taste A gedrückt. Während des Zwischenhalts zeigt die Anzeige das jeweilige φ, dann das zugehörige r. Nach dem Einschalten oder nach f CL REG sind dies φ = 30, 60, 90°, ... Den Wert für $\varphi = 0^{\circ}$ erhält man durch

```
3
0
CHS
STO 1
```

den Wert für $\varphi = 15^{\circ}$ durch

```
1
5
CHS
STO 1
```

beides gefolgt vom Tastendruck auf A. Bei Rechnern ohne den h PAUSE - Befehl kann Schritt 6 durch R/S ersetzt werden.

Beispiel 2.24: Cassinische Kurven

Die Funktion $f(x) = \pm\sqrt{\sqrt{4\,a^2\,x^2 + c^2} - a^2 - x^2}$ beschreibt die Feldlinienverteilung in der Ebene senkrecht zu zwei stromführenden Drähten (Cassinische Kurven). Die Kurven sind für einen Abstand der beiden Drahtachsen von der Mittellinie a = 10 zu zeichnen. Der Kurvenparameter c soll die Werte annehmen c = 20, 40, 60, ... Die Veränderliche x soll betragen x = 0, 1, 2, 3, ...

Das Programm lautet

```
 1    f LBL B
        1
      STO + 2    x
       RCL 2
 5    h PAUSE
       g x²
        4
        0
        0        4 a² = 400
10      x
      RCL 3      c
      g x²
        +
      f √x
15     EEX
16        2
          -
       RCL 2
       g x²
20        -
       f √x
       h RTN
      f LBL C
        2
25      0
      STO + 3
       RCL 3          c
       h RTN
```

Drücken der Labeltaste C bewirkt ein Weiterschalten des Kurvenparameters c um 20. Das neue c wird jeweils angezeigt. Je Kurvenzug wird C nur einmal gedrückt. Ein Drücken von B erhöht x um 1. Das neue x wird in der Programmpause angezeigt, anschließend wird f(x) berechnet. Da es sich bei den Cassinischen Kurven um geschlossene Kurvenzüge handelt, gibt es nicht zu jedem willkürlichen c und x ein entsprechendes f(x). Vorschlag für eine Ausgangskurve: c = 120, x = -14, also (x - 1) = -15.

In Stellung RUN wird gedrückt

1	1
2	5
0	CHS
STO 3	STO 2

Besonders bei Aufgaben aus der Darstellenden Geometrie kann das einfache Weiterschalten eines Kurvenparameters von Nutzen sein. Hier sollen häufig Gegenstände, aber auch Hüllkurven, perspektivisch dargestellt werden. Eine solche Aufgabe wird im folgenden Beispiel vorgestellt.

Beispiel 2.25: Spirale, die eine Spindel umwickelt

Die Figur gehört zu den perspektivischen Darstellungen. Der Kurvenparameter φ ist aufgeteilt in eine x-Komponente $a \cdot \cos \varphi$ und eine y-Komponente $b \cdot \sin \varphi$. Für $a = b$ ergibt sich ein Kreis, für $a \neq b$ eine Ellipse. Diese ist das perspektivische Bild eines Kreises, der schräg von oben gesehen wird. y enthält einen Summanden $d \cdot \varphi$, der mit wachsendem φ ein gleichmäßiges Ansteigen entlang der y-Achse bewirkt. Die Ellipse läuft also nicht in sich selbst zurück, sondern wird zu einer Spirale der Ganghöhe $h = \frac{d}{360}$ auseinandergezogen. Schließlich wird die x- und die y-Komponente multipliziert mit $\sin^2 c\varphi$. Dadurch ändert sich der Spiralendurchmesser von Null bis zu einem Höchstwert und zurück. Die Gesamtfigur hat daher das Aussehen einer Spirale, die eine Spindel umwickelt. Die vollständige Gleichung lautet

$$x = a \cdot \sin^2 c\varphi \cdot \cos \varphi$$
$$y = b \cdot \sin^2 c\varphi \cdot \sin \varphi + d\varphi \quad .$$

Es sei $a = 80$, $b = 20$, $c = \frac{1}{40}$ und $d = \frac{1}{30}$. Es ist daher

$$x = 80 \sin^2 \frac{\varphi}{40} \cdot \cos \varphi$$
$$y = 20 \sin^2 \frac{\varphi}{40} \cdot \sin \varphi + \frac{\varphi}{30} \quad .$$

Wiederum wächst φ in Schritten von 30°.

1	f LBL D		16	RCL 4	
	3			f COS	
	0			x	
	STO + 4			R/S	$x = 80 \sin^2 \frac{\varphi}{40} \cos\varphi$
5	RCL 4		20	RCL 5	
	h PAUSE			4	
	4			÷	
	0			RCL 4	
	÷			f SIN	
10	f SIN		25	x	
	g x^2			RCL 4	
	8			3	
	0			0	
	x			÷	
15	STO 5	$80 \sin^2 \frac{\varphi}{40}$	30	+	$y = 20 \sin^2 \frac{\varphi}{40} \sin\varphi + \frac{\varphi}{30}$
				h RTN	

Drücken von Taste D erhöht φ um 30^0, das neue φ wird in der Programmpause angezeigt. Es wird x berechnet und angezeigt, dann nach Drücken von R/S y. Die Figur läßt sich in 20 Umläufen zeichnen. Werden die x- und y-Werte in mm gezählt, so kann die Figur (Abmessungen 16 x 24 cm) auf einem DIN-A-4-Blatt aufgetragen werden.

Beispiel 2.26: $f(x) = \frac{\sin x}{x}$

Das Schaubild der Funktion ist zu zeichnen. Lösung: Für x in Bogenmaß mit $\sin \frac{\pi}{2} = 1$ empfiehlt sich ein Vorprogramm.

```
1     g LBL e
       h RAD      Umschalten
         0        auf Bogenmaß
       STO 7
5       h π
         1
         2
         ÷
       STO 6      π/12 = 15°
10     h RTN
```

```
       f LBL E
        RCL 6
       STO + 7    x
        RCL 7
15     h PAUSE
        f SIN
        RCL 7
          ÷
        h RTN
```

Nach Umschalten auf RUN wird zum Starten des Vorprogramms f e gedrückt. Dann kann durch wiederholtes Drücken von E die Funktion aufgerufen werden. In der Programmpause wird das jeweilige x angezeigt, anschließend $\frac{\sin x}{x}$ berechnet. Für $x = \pi$ sollte $f(x) = 0$ erscheinen. Der Rechner zeigt $-4 \cdot 10^{-10}$, also annähernd Null. Der Inhalt von Register 7 beträgt $\pi + 10^{-9}$. Die Abweichung erklärt sich durch Rundungsfehler bei der fortgesetzten Summierung.

Beispiel 2.27: Mondlandespiel

Die Mondfähre befindet sich im geradlinigen Anflug auf den Mond. Durch Bremsstöße von je 10 Sekunden Dauer soll die Geschwindigkeit der Fähre so verringert werden, daß ein Aufsetzen mit einer Sinkgeschwindigkeit von weniger als 10 m/s erfolgt (weiche Landung).

Die anzuwendenden Formeln sind:

Treibstoffvorrat $m_T = m_{To} - \Sigma m_S$

Geschwindigkeit $v = v_o - v_S \cdot \sum \frac{m_S}{m_o + m_T} + b_o \cdot t_o \sum \frac{1}{(1 + \frac{a}{r_o})^2}$

Abstand zur Mondoberfläche $a = a_o - t \cdot \Sigma v$

Die Konstanten haben folgende Werte:

Gewicht der leeren Fähre	m_o =	1500 kg
Treibstoffgesamtmenge	m_{To} =	12000 kg
Anfangsgeschwindigkeit	v_o =	3000 m/s
Anfangsabstand	a_o =	500 km
Strahlgeschwindigkeit	v_S =	1500 m/s
Dauer der Bremsstöße	t_o =	10 s
Mondradius	r_o =	1738 km
Beschleunigung an der Mondoberfläche	b_o =	1.62 m/s^2

Vor Spielbeginn ist zur Spielvorbereitung E zu drücken. Im Spiel wird jeweils die Treibstoffmenge für den 10-Sekunden-Schub auf Label A eingegeben. Der neue Abstand nach den 10 Sekunden Schubdauer wird angezeigt. Durch Drücken von B kann die Geschwindigkeit der Fähre festgestellt werden, durch Drücken von C der noch vorhandene Treibstoffvorrat. C braucht also nicht jedesmal gedrückt zu werden. Wird aber mehr Treibstoff für den Schub angefordert als vorrätig, so wird "Error" angezeigt. "Error" zeigt ebenfalls nach erfolgter Landung das Spielende an. Das Spiel wird zweckmäßig schriftlich gespielt mit Notieren von Schub, Abstand und Geschwindigkeit. Sinkt der Abstand zur Mondoberfläche auf weniger als 2 km, so sollte durch DSP 3 auf Meteranzeige geschaltet werden. Die Programmschrittliste ist im Anhang aufgeführt.

2.7.4 Die Sekundärregister

Die bisher behandelten Speicheroperationen wurden mit dem Last-X- und den Primärregistern 0 bis 9 durchgeführt. Daneben verfügen der HP 67 und HP 97 noch über die zehn Sekundärregister S 0 bis S 9. Diese lassen sich nach Drücken von f P ⇄ S (primär gegen sekundär) aufrufen. Dabei werden die Inhalte der Register 0 bis 9 nach S 0 bis S 9 übertragen, zugleich die von S 0 bis S 9 nach Register 0 bis 9. Hier können sie als RCL 0 bis RCL 9 aufgerufen werden und an der Speicherarithmetik teilnehmen.

Beispiel:

Das Produkt von Register 4 und Sekundärregister 5 soll in Register 6 übertragen werden.

```
RCL 4
f P⇄S
RCL 5     (S 5)
  x
f P⇄S
STO 6
```

Die Sekundärregister S 0 bis S 3 sind frei verfügbar. Die Register S 4 bis S 9 werden dagegen bei Statistikeingabe (Kapitel 1.5) belegt. Bei Eingaben der Form

```
 x_i
ENT
 y_i
Σ+
```

werden die Sekundärregister S 4 bis S 9 wie folgt besetzt:

Reg Nr.	Inhalt
S 4	Σy_i
S 5	Σy_i^2
S 6	Σx_i
S 7	Σx_i^2
S 8	$\Sigma x_i y_i$
S 9	n

Beispiel 2.28: Optimale Gerade

Durch eine Anzahl Wertepaare x_i , y_i soll eine Gerade $y = a\,x + b$ möglichst genau hindurchgelegt werden. Steigung a und Achsenabschnitt b der Gerade sind gesucht. Lösung: Die Summe der Abstandsquadrate zwischen den Punkten x_i , y_i und der Geraden muß ein Minimum sein. Es ist dann

$$a = \frac{n \cdot \Sigma x_i\, y_i - \Sigma x_i \Sigma y_i}{n \cdot \Sigma x_i^2 - (\Sigma x_i)^2} \qquad b = \frac{1}{n}\,(\Sigma y_i - a\,\Sigma x_i)$$

Der mittlere Abstand m_s in y-Richtung zwischen den Punkten

und der Geraden ist

$$m_s = \sqrt{\frac{\Sigma v_i^2}{n-2}} \quad \text{mit} \quad \Sigma v_i^2 = \Sigma (y_i - ax_i - b)^2$$
$$= \Sigma y_i^2 - nb^2 + a\,(a\,\Sigma x_i^2 - 2\,\Sigma x_i y_i) .$$

Die Geradengleichung kann also in der Form $y = a\,x + b \pm m_s$ wiedergegeben werden. Der mittlere Abstand zwischen den Punkten und der Geraden, also die Standardabweichung, ist

$$s = \frac{m_s}{\sqrt{1 + a^2}}$$

Die Programmliste ist im Anhang wiedergegeben. Die Eingabe der n Wertepaare erfolgt als

x_i
ENT
y_i
$\Sigma+$

Wie bei der einfachen Statistik zeigt der Rechner jeweils die Anzahl der bereits eingegebenen Wertepaare an. Sind alle Wertepaare über $\Sigma+$ eingegeben, so wird Taste A gedrückt. Der Rechner ermittelt die Geradensteigung a. Nach Drücken von R/S wird der Achsenabschnitt b berechnet, nach erneutem R/S der mittlere Abstand m_s der Wertepaare in y-Richtung von der optimalen Geraden. Aus der Formel für m_s ist zu ersehen, daß m_s nur für mehr als zwei Wertepaare angegeben werden kann: Durch zwei Wertepaare kann eine Gerade hindurchgehen, eine Streuung besteht nicht. Sind Wertepaare versehentlich falsch eingegeben oder sollen bereits eingegebene infolge zu hohen m_s nicht berücksichtigt werden, so sind sie nachträglich zu entfernen:

x_i
ENT
y_i
h $\Sigma-$

Anschließend kann wieder A gedrückt werden. Unter Label B ist die gefundene optimale Gerade als $y = a\,x + b$ gespeichert: Zu jedem auf B eingegebenen x wird das zugehörige y berechnet.

Soll eine weitere Anzahl Wertepaare für die nächste optimale Gerade eingegeben werden, so ist zuvor E zu drücken. Dadurch werden die Inhalte der Primär- und Sekundärregister gelöscht.

Beispiel:

Durch die Wertepaare 3,5; 7,9 und 10,14 soll die optimale Gerade hindurchgelegt werden. Die Geradengleichung und die zu den x_i berechneten y-Werte sind anzugeben.

Wertepaare		Berechnete
x_i	y_i	y_i-Werte
3	5	4.7
7	9	9.8
10	14	13.6

$y = 1.27\ x + 0.86 \pm 0.93$

Das Programm "Optimale Gerade" kann auch für nichtlineare Zusammenhänge verwendet werden. In diesem Fall wird der Maßstab so gewählt, daß in einer graphischen Darstellung wiederum eine Gerade eingetragen werden kann.

Beispiel:

Bei der Diodenkennlinie $I = A \cdot e^{B \cdot U}$ ergibt sich wegen

$$\ln I = \ln A + B \cdot U$$

für den Logarithmus des Stromes I eine Gerade, wenn die Spannung U linear aufgetragen wird.

Dem bisherigen	$y = a \cdot x + b$	$\pm m_s$
entspricht nun	$\ln I = B \cdot U + \ln A$	$\pm \ln A_1$

Als Wertepaare x_i , y_i werden eingegeben U_i, $\ln I_i$. Für größere Datenmengen kann das Programm entsprechend geändert werden. Es wird gedrückt GTO · 000 (bzw. RTN beim HP 65)

und in Stellung W/PRGM

```
f LBL C
f LN
Σ+
h RTN
```

Die Eingabe erfolgt in Stellung RUN als

U_i
ENT
I_i
C .

Gemäß der Unterbringung von I im X-Register wird im Rechner ln I verarbeitet. In der folgenden Tabelle sind den Meßwerten U_i, I_i die berechneten Werte I_i für die jeweiligen U_i gegenübergestellt.

Wertepaare		Berechnete Werte
U_i/V	I_i/mA	I_i/mA
1.1	2.54	2.53
2.1	28.0	28.2
3.2	400	399

Nach der Eingabe wird durch Drücken von Taste A berechnet

$$a = B = 2.4093$$
$$b = \ln A = -1.7213$$
$$m_s = \ln A_1 = 0.0075$$

Es ist $\ln A + \ln A_1 = -1.7138$, also $A \cdot A_1 = 0.1802$
$\ln A - \ln A_1 = -1.7288$, also $\frac{A}{A_1} = 0.1775$.

Der Übergang auf die gewohnte $\pm \Delta A$-Schreibweise gelingt durch

$$A + \Delta A = 0.1802$$
$$A - \Delta A = 0.1775.$$

Damit ist $A \pm \Delta A = 0.179 \pm 0.001$

Es folgt für I in mA, U in V:

$$I = (179 \pm 1) \cdot 10^{-3} \cdot e^{2.409 \cdot U}$$

Beispiel:

Temperaturabhängigkeit eines Halbleiterwiderstandes R. Dieser ist gegeben durch

$R = C \cdot e^{\frac{D}{T}}$ mit T als absoluter Temperatur, C und D als Konstanten.

Dem bisherigen $y = a\,x + b \pm m_s$

entspricht nun $\ln R = \frac{D}{T} + \ln C \pm \ln C_1$.

Als Wertepaare x_i , y_i werden eingegeben $\frac{1}{T_i}$, $\ln R_i$. Für eine nur gelegentliche Rechnung wird direkt gedrückt h $\frac{1}{x}$ T_i, f LN R_i. Das Programm läßt sich ergänzen durch GTO · 000 (bzw. durch RTN beim HP 65)

sowie in Stellung W/PRGM

f LBL D
f LN
h x⇄y
h $\frac{1}{x}$
h x⇄y
Σ+
h RTN .

In Stellung RUN wird gedrückt

T_i
ENT
R_i
D .

Die Programmänderung wird verständlich, wenn zur Veranschaulichung die jeweilige Stackbelegung bei der Eingabe aufgezeichnet wird.

Auch hier sind den gemessenen Wertepaaren T_i, R_i die berechneten R_i gegenübergestellt.

Wertepaare		Berechnete Werte
T_i /K	R_i /Ohm	R_i /Ohm
200	500	565
300	100	80.0
400	30	30.1
800	6.3	6.94

Nach der Eingabe folgt durch Drücken von Taste A

$$a = D = 1173$$
$$b = \ln C = 0.4710$$
$$m_s = \ln C_1 = 0.1928.$$

Mit $\ln C + \ln C_1 = 0.6638$ und $\ln C - \ln C_1 = 0.2782$ ist
$C \cdot C_1 = 1.942$, $\frac{C}{C_1} = 1.3208$. Damit ist $C \pm \Delta C = 1.63 \pm 0.31$.

Die Gleichung lautet daher $R = (1.63 \pm 0.31) \cdot e^{\frac{1173}{T}}$.

Zum Berechnen der R_i wird eingetastet

T_i
h $\frac{1}{x}$
B
g e^x .

Zur Programmänderung wird in RUN gedrückt GTO B, dann in W/PRGM h $\frac{1}{x}$. Anschließend wird auf RUN zurückgeschaltet und eine Zahl auf B eingegeben. Das Programm durchläuft jetzt B und hält einen Schritt nach h RTN an. Nun wird auf W/PRGM geschaltet und gedrückt

h BST
h BST
g e^x.

Nach Zurückschalten auf RUN können die T_i direkt auf B eingegeben werden, die R_i werden berechnet.

2.8 Überprüfen des Programms

Nach der Fertigstellung ist ein Programm mit mehreren einfachen Eingabewerten zu überprüfen. Beim Programmbeispiel 2.6, Raumdiagonale eines Quaders mit

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

wären solche Eingabewerte

Eingabewerte	Raumdiagonale d
x = y = z = 0	0
= 1	1.73 $(=\sqrt{3})$
= -1	1.73 $(=\sqrt{3})$
$= 10^3$	1732 $(= 10^3 \cdot \sqrt{3})$
x = 2, y = -3, z = 5	6.16

Auf jeden Fall sind als Abschluß Eingabewerte ungleich Null, ungleich Eins und ungleich untereinander zu wählen. Die Lösung sollte auch für diese Werte bekannt sein.

Beim Überprüfen werden manchmal Fehler festgestellt.

a) Das Programm hält nicht an
Stoppen der Programmausführung durch Drücken einer beliebigen Taste. In der Schrittliste fehlt offensichtlich ein GTO 00 beim HP 25 bzw. ein h RTN bei den anderen Rechnern.

b) Unterschiedliche Ergebnisse bei gleicher Eingabe
Beim Verwenden der Speicherarithmetik fehlt ein anfängliches STO Registername oder ein f CL REG.
Außerdem haben evtl. zwei Programme den gleichen Label. Hierzu ist nach der Ausführung in W/PRGM die Schrittnummer zu prüfen.

c) Programmabbruch
Der Rechner zeigt "Error" oder blinkende Nullen. Hier wurde ein unerlaubter Rechenschritt versucht. Ein solcher Schritt ist z.B. Teilen durch Null oder - bei Zahlen kleiner als Null - ein Ziehen der Quadratwurzel sowie ein Logarithmieren der Zahl.

Beispiel:

Beim HP 67 sei unter Label C ein Programm eingegeben. Nach dem Starten des Programms in Stellung RUN bricht der Rechner die Programmausführung ab. Der Schiebeschalter wird auf W/PRGM gesetzt. Schritt 37 zeigt "81", also teile. Drücken von h BST gibt als Schritt 36 "34 08", d.h. hole aus Register 8. Dieses enthält also entgegen der Erwartung Null. In RUN sind nun auch die übrigen Register zu prüfen. Dies geschieht durch

RCL Registername oder durch h REG. Meist wird hierbei bereits der Fehler gefunden. Ist dies nicht der Fall, so muß wie unter d) verfahren werden.

d) Unrichtige Ergebnisse
Bei kurzen Programmen wird zunächst in W/PRGM die einprogrammierte Schrittfolge durchgesehen.

Beispiel:

Das Programm sei weiter unter Label C. In RUN wird gedrückt

```
GTO C,
```

dann in W/PRGM

```
SST
SST
SST
 ⋮
```

Ist alles in Ordnung, so prüft der Programmierer als nächstes schrittweise die Programmausführung. Bei nur einem Eingabewert wird in Stellung RUN gedrückt

```
Eingabewert
   GTO C
    SST
    SST
     ⋮
```

Der Rechner führt nun das Programm schrittweise aus. Beim HP 25, 67 und 97 gibt das Anzeigefenster beim Niederdrücken der SST-Taste den jeweiligen Programmschritt wieder. Nach dem Loslassen zeigt der Rechner die Auswirkung auf den Rechengang. Beim HP 65 kann gleichfalls mit SST durchs Programm getastet werden. Scheint ein Zwischenergebnis bedenklich, so wird in W/PRGM die entsprechende Programmzeile überprüft. Besonders ist auf Umordnungen des Stack zu achten wie h $x \rightleftarrows y$, h R↑ oder h R↓. Außerdem sind auch die in Kapitel 4 besprochenen Programmverzweigungen zu überprüfen.

2.9 Aufzeichnen von Programmen

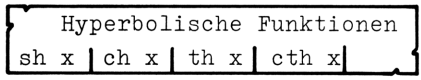
Ein wichtiger Vorzug der HP-Rechner 65, 67 und 97 ist ihre Fähigkeit, Programme auf Magnetkarten aufzuzeichnen. Bekanntlich wird bei den meisten programmierbaren Rechnern beim Abschalten des Gerätes auch der Programmspeicher gelöscht. Die ganze sorgfältig eingetastete und überprüfte Programmschrittfolge ist damit verloren. Wird dagegen der Programmspeicher vor dem Abschalten auf eine Magnetkarte kopiert, so bleibt auf dieser das Programm erhalten. Bei Bedarf kann der Benutzer das Programm wieder in den Rechner einlesen. So lassen sich beliebig viele Programme auf Magnetkarten speichern. Der Benutzer kann sich damit eine eigene Sammlung von Programmen zusammenstellen. Die in diesem Buch aufgeführten Programmbeispiele bilden bereits den Grundstock einer solchen Sammlung. Eine eigene Programmbibliothek ist für die Arbeit wie fürs Hobby ein scharf geschliffenes Werkzeug. Mit der Zeit verfügt der Eigentümer dieser Programmbibliothek über eine Lösung für fast jedes praktisch vorkommende Problem.

Der Vorgang einer Übertragung ist denkbar einfach. Zur Übernahme eines Programms auf Magnetkarte wird der Schiebeschalter des Rechners auf W/PRGM gestellt (write: auf Magnetkarte schreiben). Dann wird eine leere Magnetkarte in den Aufnahmeschlitz des Kartenlesers gesteckt. Dieser Schlitz befindet sich beim HP 65 und 67 rechts an der Schmalseite des Rechners, beim HP 97 vorn neben dem Anzeigefenster. Die Lage der Magnetkarte ist weiße Kartenseite nach oben, den Pfeil mit Nr. 1 Richtung Schlitz. Beim Einschieben schaltet sich der Kartenlesemotor ein und zieht die Magnetkarte durch den Rechner. Als Anzeige erscheint die gleiche Programmzeile wie vor dem Einlesen. Das Programm ist jetzt sowohl im Programmspeicher des Rechners wie auf der Magnetkarte enthalten.

Werden weitere Magnetkarten in den Rechner geschoben, so wird das Programm auch auf diese übertragen. Es können also beliebig viele Kopien hergestellt werden. Eine fehlerhafte Übertragung (z.B. bei kaltem Rechner: Motor zu langsam) zeigt der HP 65 durch blinkende Nullen, der HP 67/97 durch "Error" (Fehler) an. In diesem Fall sollte die Übertragung nochmals, ggf.

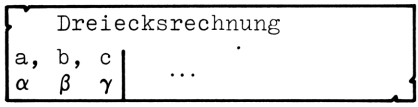
mit einer anderen Magnetkarte, versucht werden. Beim HP 67 und HP 97 bewirken Programme mit mehr als 112 Programmschritten nach dem Durchgang die Anzeige "Crd" (card: auch die zweite Seite der Magnetkarte beschreiben). Zum Übertragen des Restprogramms wird die Karte nochmals, diesmal die Seite Nr. 2 nach vorn, durch den Leseschlitz geschickt.

Eine Magnetkarte mit Programm sollte hinreichend beschriftet werden. Dies kann mit Bleistift, Tuschefeder oder dünnem Faserschreiber geschehen. Weniger geeignet sind Kugelschreiber oder Schreibmaschine. Hier wird durch die entstandene wellige Kartenoberfläche ein späterer Lesevorgang gestört. Die Beschriftung sollte sowohl den Titel des Programms wie auch die Belegung der Label enthalten. Eine beschriftete Programmkarte kann z.B. wie folgt aussehen:



Nach Eintasten eines x-Wertes gibt bei diesem Programm ein Drücken der Labeltasten A bis D den Wert der jeweils gewünschten Funktion.

Umfaßt die Eingabe mehrere Werte, so sollte auch dies aus der Beschriftung der Magnetkarte zu erkennen sein. Bei einer Eingabe mehrerer Größen mit R/S sowie einer Ausgabe ebenfalls mit R/S kann die Karte z.B. so beschriftet werden:

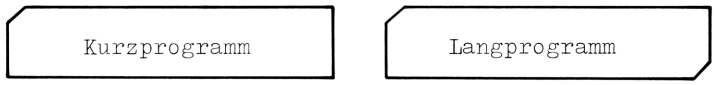


Bei einer Eingabe mit mehrfachem Enter würde als Eingabehinweis stehen

a ↑ b ↑ c .

2.9.1 Schutz der Programmkarten

Wird eine Magnetkarte mit gespeichertem Programm versehentlich in Stellung W/PRGM durch den Rechner geschickt, so überschreibt der Programmspeicher das Kartenprogramm. Das auf der Karte vorher gespeicherte Programm ist damit verloren, es muß aus der Programmliste wieder in den Rechner eingetastet werden. Dieses versehentliche Löschen einer Kartenseite wird durch Entfernen einer Kartenecke nach dem Kopieren verhindert. Die Kartenecke ist dazu entsprechend der Einkerbung mit einer Schere abzuschneiden. Achtung: wird mehr abgeschnitten, so fehlt ein Stück des Programms.



Schutz einer Kartenseite Schutz beider Kartenseiten

Das Programm auf einer solchen Karte kann durch den Rechner nicht mehr gelöscht werden.

Zum Einlesen eines Kartenprogramms wird der Rechner eingeschaltet und der rechte Schiebeschalter in Stellung RUN gesetzt. Die Programmkarte wird mit Seite 1 nach vorn in den Leseschlitz gesteckt und vom Lesemotor durchgezogen. Die Anzeige "Crd" zeigt, daß auch die andere Kartenseite Programm-Information enthält. In diesem Fall wird die Karte nochmals, mit Seite 2 nach vorn, durchgeschoben. Das Anzeigefenster zeigt nach gelungener Übertragung den Inhalt des X-Registers. Das Anzeigeformat ist beim HP 67/97 dasjenige, welches im Rechner beim Kopieren auf die Magnetkarte eingestellt war, also z.B. 0.0000 bei f FIX DSP 4 oder 0. 00 bei g SCI DSP 0. Entsprechendes gilt für die eingestellte Winkelart. Wurde nichts derartiges eingetastet, so wird die Grundstellung f FIX DSP 2 sowie h DEG übertragen.

2.9.2 Dokumentation

Außer der Magnetkarte sollten zu jedem Programm weitere Angaben verfügbar sein. Dies sind vor allem die verwendeten Gleichungen und die Programmschrittliste. Außerdem ist die Belegung der Label und Speicherregister anzugeben, sowie - falls verwendet - die Flags (Kapitel 4.3). Das Anzeigeformat und die Winkelart sind nur zu nennen, wenn sie von der Grundstellung f FIX DSP 2 und h DEG abweichen. Auf jeden Fall sollte jedoch die Eingabe und Ausgabe von Werten angeführt sein, ggf. mit einem Beispiel. Ist das Programm in den Eingabewerten beschränkt (z.B. Eingabe nur von Werten größer als Null), so ist auch dies anzugeben. Mit einer solchen schriftlichen Dokumentation können Programme auch nach längerer Zeit wieder verwendet, abgeändert oder weitergegeben werden.

2.9.3 Aufzeichnen mit MERGE

Zum Ordnen der Programmbibliothek bietet der HP 67/97 eine nützliche Hilfe: die Tastenfunktion g MERGE. Damit können Programmteile an vorhandene Programme angehängt werden. Außerdem lassen sich auf mehrere Karten verstreute Programme auf einer Karte unterbringen. In Stellung RUN wird gedrückt

```
GTO   Zeilennummer
   g MERGE
```

Dann wird die gewünschte Programmkarte durch den Rechner geschoben. Programme werden dadurch erst ab der vorgegebenen Zeilennummer des Programmspeichers aufgezeichnet.

Beispiel:

Drei Programme der Länge 31, 45 und 34 Schritte sind auf einer gemeinsamen Karte unterzubringen. In Stellung RUN wird die erste Karte durch den Rechner geschoben. Damit sind die Programmzeilen 1 bis 31 belegt. Es wird gedrückt

```
GTO · 031
  g MERGE .
```

Nun kommt die zweite Karte durch den Rechner. Die Programmzeilen 32 bis 76 sind jetzt auch besetzt. Es wird gedrückt

```
GTO · 076
 g MERGE  ,
```

dann kommt die dritte Karte durch den Rechner. Das dritte Programm belegt die Zeilen 77 bis 110. Bei bisher gleichen Labeln muß nun jedes Programm seinen eigenen Label erhalten (z.B. A, B und C). Falls Unterprogramme (vgl. Kapitel 3) aufzurufen sind, müssen auch die sich unterscheiden. Dann kommt in Stellung W/PRGM eine ungeschützte Karte in den Rechner. Auf dieser werden alle drei Programme aufgezeichnet.

2.10 Aufzeichnen von Daten

2.10.1 Programmkarte mit Daten

Daten in Programmen können im Prinzip jedesmal vor Beginn der Rechnung in die zugehörigen Speicher eingetastet werden. Sind es jeweils die selben Daten, also Konstanten, so werden sie zweckmäßig als Teil des Programms in die Speicher geladen. Konstanten können auch mit einer eigenen Karte die Speicher laden, z.B. mit

```
f LBL E
   ⋮
 STO 1          Konstante 1
   ⋮
 STO 2          Konstante 2
   ⋮                 ⋮
 h RTN
```

Bei Kurzprogrammen läßt sich zum Datenspeichern die unbenutzte Kartenseite verwenden. Solche Kurzprogramme mit nur einer Kartenseite umfassen bis zu 100 Programmschritte beim HP 65 bzw. bis zu 112 Schritte beim HP 67 oder HP 97. In Stellung RUN wird zuerst das Konstantenprogramm eingelesen und über die zugehörige Labeltaste gestartet. Dadurch füllen sich die Spei-

cher mit den benötigten Konstanten. Anschließend wird die Kartenseite mit dem Rechenprogramm eingelesen. Das Konstantenprogramm im Programmspeicher wird überschrieben, die Speicherinhalte bleiben unverändert.

Beispiel 2.29: Gleichungen aus der Plasmaphysik

Die benötigten Konstanten e_o und m_e werden zusammen mit h, k, c und σ auf einer Programmkarte gespeichert. Eine solche Programmkarte ist im Anhang wiedergegeben. Die Programmkarte enthält:

Register-Nr.	Inhalt
1	Plancksches Wirkungsquantum h
2	Boltzmann-Konstante k
3	Lichtgeschwindigkeit im Vakuum c
4	Stefan-Boltzmannsche Konstante σ
5	Ladung des Elektrons (Elementarladung) e
6	Ruhemasse des Elektrons m_e

Die zugehörigen Zahlenwerte sind:

$h = 6.625 \cdot 10^{-34}$ J s
$k = 1.381 \cdot 10^{-23}$ J/K
$c = 2.998 \cdot 10^{8}$ m/s
$\sigma = 5.670 \cdot 10^{-8}$ J/K^4 m^2 s
$e = 1.602 \cdot 10^{-19}$ C
$m_e = 9.109 \cdot 10^{-31}$ kg

Das Konstantenprogramm wird nach dem Einlesen durch Drücken von E gestartet. Beim folgenden Einlesen der Hauptprogrammkarte in Stellung RUN bleiben die Speicher 1 bis 6 belegt, während das Konstantenprogramm überschrieben wird.

Larmor-Frequenz:

Im Magnetfeld der Stärke B beschreiben geladene Teilchen Spiralbahnen mit einer Frequenz

$$\nu = \frac{e}{m} \cdot \frac{B}{2\pi}$$

Das Programm gibt auf Eingabe von B in Tesla die Larmor-Frequenz in Hertz von Elektronen, dann nach R/S die von Protonen ($m_p = 1836 \cdot m_e$) an.

Format: h ENG DSP 2.

1	f LBL A	B in Tesla	10	R/S	
	h π			1	
	÷			8	
	2			3	
5	÷		15	6	
	RCL 5			÷	ν_p
	x			h RTN	
	RCL 6				
	÷	ν_e			

Für B = 1 Tesla (= 1 $\frac{Vs}{m^2}$) folgt $\nu_e = 2.8 \cdot 10^{10}$ Hz,

$\nu_p = 1.5 \cdot 10^7$ Hz .

Larmor-Radius:

Der Radius der Spiralbahnen beträgt

$$r = v \cdot \frac{m}{e\ B}$$

B in Tesla wird in Register 7 eingespeichert, die Geschwindigkeit v in $\frac{m}{s}$ wird auf Label B eingegeben. Der Larmor-Radius von Elektronen wird in Metern angegeben, nach R/S der von Protonen.

17	f LBL B	v	25	R/S	
	RCL 6			1	
	x			8	
20	RCL 5			3	
	÷			6	
	RCL 7			x	r_p
	÷	r_e	30	h RTN	
	R/S				

Mit B = 1 T wird für $v = 10^5 \frac{m}{s}$ $r_e = 600$ nm und $r_p = 1$ mm.

Mittlere thermische Geschwindigkeit:

Wegen $\frac{m}{2}\overline{v^2} = \frac{3}{2}kT$ wird für Elektronen und Protonen

$$\overline{v} = \sqrt{\frac{3\,kT}{m}}$$

Die absolute Temperatur T in Kelvin wird auf Label C eingegeben und die mittlere Geschwindigkeit für Elektronen v_e in $\frac{m}{s}$ berechnet. Nach Drücken von R/S wird die mittlere Geschwindigkeit von Protonen angezeigt.

31	f LBL C	T in Kelvin	41	f $\sqrt{x}$	v_e
	RCL 2			R/S	
	x		40	1	
	3			8	
35	x			3	
	RCL 6			6	
	÷			f $\sqrt{x}$	
	f $\sqrt{x}$	v_e	45	÷	v_p
				h RTN	

Bei Zimmertemperatur (T = 300 K) ist $v_e = 1.2 \cdot 10^5 \frac{m}{s}$, $v_p = 2.7 \cdot 10^3 \frac{m}{s}$. Bei 10^4 K wird die mittlere thermische Geschwindigkeit $v_e = 6.7 \cdot 10^5 \frac{m}{s}$, $v_p = 1.6 \cdot 10^4 \frac{m}{s}$.

Beschleunigung durch ein elektrisches Feld:

Beim Durchfliegen einer Potentialdifferenz wird die kinetische Energie eines geladenen Teilchens erhöht um $\frac{mv^2}{2} = e_o U$.

$$v = \sqrt{\frac{2\,e_o\,U}{m}}$$

Die Spannung U in Volt wird auf Label D eingegeben. Die Geschwindigkeitszunahme wird für Elektronen angezeigt, nach Drücken von R/S die für Protonen.

```
47    f LBL D      U in Volt
       RCL 5
         x
50       2
         x
       RCL 6
         ÷
       f √x        v_e
55     R/S
         1
         8
         3
         6
60     f √x
         ÷          v_p
       h RTN
```

Wird ein Gebiet mit einer Potentialdifferenz von 1 V durchflogen, so erhöht sich die mittlere Geschwindigkeit um $v_e = 5.9 \cdot 10^5 \frac{m}{s}$ bzw. $v_p = 1.4 \cdot 10^4 \frac{m}{s}$. Ein Vergleich mit der thermischen Geschwindigkeit zeigt Übereinstimmung für 10^4 K, d.h. 10^4 K entspricht 1 eV.

Freie Weglänge:

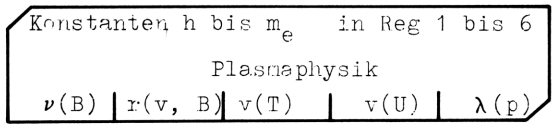
Im Neutralgas der Dichte n durchfliegen die Teilchen zwischen zwei Zusammenstößen im Mittel die Strecke $\lambda = \frac{1}{n \cdot F\sqrt{2}}$ mit F = Querschnittsfläche der Teilchen für einen Stoß. Beträgt die Teilchendichte bei Normalbedingungen $n_o = 2.7 \cdot 10^{25} \frac{\text{Teilchen}}{m^3}$ und ist annähernd $F = 10^{-19}\ m^2$, so folgt für die freie Weglänge beim Druck p

$$\lambda = \frac{2.63 \cdot 10^{-7}}{p}$$

mit p in Bar und λ in Metern.

```
63      f LBL E
          h 1/x          p in Bar
65          2
            6
            3
          EEX
          CHS
70          9
            x
        h RTN
```

Für Normalbedingung ist mit p = 1 bar λ = 300 nm, bei p = 10^{-7} bar λ = 3 m. Die vollständige Programmkarte hat also folgendes Aussehen:



Beispiel 2.30: Gleichungen aus der Strahlungsphysik

Die Konstantenkarte von Beispiel 2.29 wird auch hier als zweite Kartenseite verwendet.

Beziehung zwischen Wellenlänge und Frequenz:

Wellenlänge und Frequenz hängen zusammen über $\nu = \frac{c}{\lambda}$ oder $\lambda = \frac{c}{\nu}$ mit c als Ausbreitungsgeschwindigkeit, hier die des Lichts.

Format: g SCI DSP 2

```
1   f LBL C
      h 1/x        ν oder λ
    RCL 3
      x
5   h RTN
```

Wahlweise kann λ in Metern oder ν in Hertz auf Label C eingegeben werden. Z.B. folgt für λ = 3 m $\nu = 10^8$ Hz oder für $\nu = 6 \cdot 10^{14}$ Hz λ = 500 nm, also die Wellenlänge gelben Lichtes.

Photonenenergie:

Mit $E = h \cdot \nu$ wird ν in Hertz eingegeben und die Energie E in Joule berechnet. Nach Drücken von R/S wird die Energie in Elektronvolt angegeben.

```
 6    f LBL D
      RCL 1
        x          E in Joule
      R/S
10    RCL 5
        ÷          E in Elektronvolt
      h RTN
```

So ist für $\nu = 1$ MHz $E = 6{,}6 \cdot 10^{-28}$ J oder $4{,}1 \cdot 10^{-9}$ eV, für $\nu = 6 \cdot 10^{14}$ Hz wird $E = 4{,}0 \cdot 10^{-19}$ J bzw. 2,5 eV.

Stefan-Boltzmannsches Gesetz:

Die Gesamtstrahlung des schwarzen Körpers ist gegeben zu

$$S = \sigma \cdot T^4$$

mit der absoluten Temperatur T und der Stefan-Boltzmannschen Konstante $\sigma = \frac{2\ \pi^5 \cdot k^4}{15\ c^2 \cdot h^3}$. Wird T in Kelvin eingegeben so folgt die Strahlungsleistung S in $\frac{W}{m^2}$.

```
13    f LBL E
        g x²       T in Kelvin
15      g x²
      RCL 4
        x
      h RTN
```

Die gesamte Strahlungsleistung ist bei T = 300 K S = 459 $\frac{W}{m^2}$, für 1000 K aber bereits S = 56700 $\frac{W}{m^2}$.

Plancksche Strahlungsformel:

Die Strahlungsleistung des schwarzen Körpers im Frequenzbebereich zwischen ν und $\nu + d\nu$ ist $K_\nu \cdot d\nu$, wobei die Gesamtstrahlung durch $S = \int_0^\infty K_\nu \cdot d\nu$ gegeben ist. Es ist

$$K_\nu = \frac{2\,h\nu^3}{c^2\left(e^{\frac{h\nu}{kT}} - 1\right)}$$

```
19    f LBL A                    30    g x²
       STO 8     ν in Hertz            ÷
        ENT                           RCL 8
        ENT                           RCL 1
         x                              x
         x        ν³               35  RCL 2      k
25     RCL 1      h                     ÷
         x                             RCL 7
         2                               ÷
         x                             g eˣ
       RCL 3      c                40    1
                                         -
                                         ÷        Kν
                                       h RTN
```

In Register 7 wird die absolute Temperatur eingegeben, dann kann für die Frequenz ν das zugehörige Emissionsvermögen K_ν berechnet werden. Es ist z.B. das Emissionsvermögen für $\nu = 10^{11}$ bis 10^{15} Hz bei den Temperaturen T = 300 K (etwa Zimmertemperatur), 1000 und 10000 K anzugeben.

T in K	K_ν in W s / m²				
	10^{11} Hz	10^{12} Hz	10^{13} Hz	10^{14} HZ	10^{15} Hz
$3 \cdot 10^2$	$9.1 \cdot 10^{-16}$	$8.5 \cdot 10^{-14}$	$3.7 \cdot 10^{-12}$	$1.7 \cdot 10^{-15}$	$5.3 \cdot 10^{-75}$
10^3	$3.1 \cdot 10^{-15}$	$3.0 \cdot 10^{-13}$	$2.4 \cdot 10^{-11}$	$1.2 \cdot 10^{-10}$	$2.2 \cdot 10^{-26}$
10^4	$3.1 \cdot 10^{-14}$	$3.1 \cdot 10^{-12}$	$3.0 \cdot 10^{-10}$	$2.4 \cdot 10^{-8}$	$1.2 \cdot 10^{-7}$

Der Wertebereich des Programms erstreckt sich von $\frac{h\nu}{kT} = 5 \cdot 10^{-10}$ bis 230. Bei Exponenten größer als 230 ist e^x größer als $9.9 \cdot 10^{99}$, also außerhalb des Arbeitsbereichs von Taschenrechnern. Bei Exponenten kleiner als $x = 5 \cdot 10^{-10}$ setzt der Rechner $e^x = 1$. Dadurch verschwindet in der Formel für K_ν der Nenner. In Kapitel 4.2 wird ein Weg zum Berechnen von Funktionswerten aus diesem Bereich gezeigt.

2.10.2 Datenübertragung mit W/DATA

Außer der allgemeinen Möglichkeit eines Konstantenprogramms haben der HP 67 und HP 97 noch eine besondere Tastenfunktion zur Datenspeicherung. Die Tastenfunktion f W/DATA (write/data, d.h. schreibe/Daten) ermöglicht es, die Inhalte sämtlicher Speicher auf eine Magnetkarte zu übertragen. Die gespeicherten Daten können später von der Magnetkarte wieder in den Rechner eingelesen werden.

Zum Übertragen der Speicherinhalte auf eine Magnetkarte bleibt der Rechner in Stellung RUN. Es wird gedrückt f W/DATA, der Rechner zeigt daraufhin "Crd". Nun wird eine leere ungeschützte Magnetkarte durch den Rechner geschickt. Dabei überträgt der Rechner die Inhalte der Speicher 0 bis 9, A bis E und I auf eine Kartenseite. Sind die Sekundärspeicher S 0 bis S 9 leer, so wird dies ebenfalls vermerkt, die Übertragung ist damit abgeschlossen. Sind dagegen ein oder mehrere Speicherinhalte von S 0 bis S 9 ungleich Null, so bleibt die Anzeige "Crd". Die Karte wird in diesem Fall nochmals durch den Rechner geschoben, die andere Seite nach vorn. Diesmal werden lediglich die Speicherinhalte von S 0 bis S 9 auf die Karte übertragen.

Zum Einlesen einer Datenkarte in Stellung RUN wird die Karte wie eine Programmkarte in den Kartenleser geschoben. Dabei überschreibt die Karteninformation alle 26 Speicherinhalte.

2.10.3 Datenübertragung mit MERGE

Mit der Funktion f W/DATA lassen sich neben Konstanten auch Zwischenergebnisse aufnehmen, die erst zu einem späteren Zeitpunkt ausgewertet werden sollen. Dazu ist es wichtig, nur einen Teil der auf der Datenkarte gespeicherten Registerinhalte in den Rechner zu übernehmen. Dies ist beim Auswerten von Daten nötig, damit die eingelesene Information nicht ständig die vom Programm beanspruchten Arbeitsregister überschreibt. Hierzu dient die Tastenfunktion g MERGE (zusammenbringen). Mit g MERGE bestimmt der Programmierer, bis einschließlich welchen Registers von der Karte Information übernommen wird, mit Register 0 beginnend. Zu einer einfachen Kenntlichmachung sind die Speicherregister durchnumeriert. Es bedeutet

Speichername	Nummer
Primärregister 0 bis 9	0 bis 9
Sekundärregister S 0 bis S 9	10 bis 19
A bis E	20 bis 24
I-Register	25

Die Nummer des höchsten zu füllenden Registers wird im I-Register mit h STI gespeichert. Sollen z.B. nur für die Register 0 bis 7 Daten übernommen werden, so ist 7 im I-Register zu speichern. Steht im I-Register eine 23, so werden in die Register 0 bis 9, S 0 bis S 9 und A bis D Daten übertragen. Ein Wert von 25 oder mehr im I-Register bewirkt, daß alle Registerinhalte überschrieben werden. Der Ablauf ist in Stellung RUN:

1. Speichern der höchsten Register-Nummer im I-Register
2. Drücken von g MERGE
3. Einführen der Datenkarte in den Kartenleser.

3. Unterprogramme

Als Unterprogramme werden selbständige Programmteile bezeichnet, die sich als Teil eines Hauptprogramms aufrufen lassen oder zu denen vom Hauptprogramm gesprungen werden kann.

3.1 Aufruf von Unterprogrammen mit GSB

Unterprogramme können grundsätzlich durch den Befehl f GSB Programm-Marke aufgerufen werden (GSB = go to subroutine, gehe zum Unterprogramm). Das Programm mit Label 0 wird als f GSB 0 aufgerufen, das Programm mit Label A als f GSB A. Bei den Buchstaben-Labeln A bis E sowie f a bis f e genügt zum Aufruf die Label-Nennung, also A oder f c. Während der Programmausführung durchläuft der Rechner schrittweise das Hauptprogramm bis zum f GSB-Befehl bzw. bis zur Labelnennung. Nun erfolgt der Sprung zum genannten Unterprogramm. Dieses wird schrittweise abgearbeitet. Beim Ende des Unterprogramms (h RTN) angekommen erfolgt ein Rücksprung zum Hauptprogramm zu der auf den GSB-Befehl folgenden Programmzeile. Auch im Unterprogramm können GSB-Befehle enthalten sein zum Aufruf eines zweiten Unterprogramms sowie im zweiten Unterprogramm Befehle zum Aufruf eines dritten.

Solche mit f GSB aufzurufenden Unterprogramme werden verwendet, wenn Teile des Hauptprogramms unter einem eigenen Label erreichbar sein sollen. Dies ist öfters bei komplizierten Funktionen der Fall. Das gleiche gilt, wenn Funktionswerte f(x) mit unterschiedlichem x im Hauptprogramm benötigt werden.

Beispiel 3.1: $f_1(x) = \sqrt{1 + x^2}$ und $f_2(x) = \sqrt{1 + x^2}\ (1 + \sqrt{1 + x^2})$ mit Unterprogrammen

Die Funktionen sollen getrennt aufzurufen sein. Auch ohne Unterprogramm lassen sich beide Funktionen einfach programmieren, vgl. Beispiel 2.13. Mit Unterprogramm würde die Lösung folgendermaßen aussehen:

```
1         f LBL A
            g x²
             1
             +
5           f √x
           STO 8        √(1 + x²)
            h RTN
-------------------------------------
           f LBL B
             A          √(1 + x²)
10           1
             +
           RCL 8
             x
            h RTN
```

In Schritt 9 wird $\sqrt{1 + x^2}$ als Unterprogramm aufgerufen. Da hierauf der Funktionswert bekannt ist, kann er in Schritt 12 als RCL 8 entnommen werden. Dies ist bei komplizierten Ausdrücken günstiger, als den gleichen Funktionswert wiederholt zu berechnen.

Beispiel 3.2: Gleichungen mit vier Unbekannten

Die Gleichungen
$$\begin{aligned} a_o\,x + a_1y + a_2z + a_3u &= a_4 \\ b_o\,x + b_1y + b_2z + b_3u &= b_4 \\ c_o\,x + c_1y + c_2z + c_3u &= c_4 \\ d_o\,x + d_1y + d_2z + d_3u &= d_4 \end{aligned}$$

enthalten die vier Unbekannten x, y, z, u sowie die zwanzig bekannten Größen a_o bis a_4, b_o bis b_4, c_o bis c_4 und d_o bis d_4. Wie in Beispiel 2.16 ergeben sich die Lösungen aus den Determinanten D und D_1 bis D_4 zu

$$x = \frac{D_1}{D},\quad y = \frac{D_2}{D},\quad z = \frac{D_3}{D} \text{ und } u = \frac{D_4}{D}$$

Hierbei ist

$$D = \begin{vmatrix} a_0 & a_1 & a_2 & a_3 \\ b_0 & b_1 & b_2 & b_3 \\ c_0 & c_1 & c_2 & c_3 \\ d_0 & d_1 & d_2 & d_3 \end{vmatrix}$$

$$D_1 = \begin{vmatrix} a_4 & a_1 & a_2 & a_3 \\ b_4 & b_1 & b_2 & b_3 \\ c_4 & c_1 & c_2 & c_3 \\ d_4 & d_1 & d_2 & d_3 \end{vmatrix} \qquad D_2 = \begin{vmatrix} a_0 & a_4 & a_2 & a_3 \\ b_0 & b_4 & b_2 & b_3 \\ c_0 & c_4 & c_2 & c_3 \\ d_0 & d_4 & d_2 & d_3 \end{vmatrix}$$

$$D_3 = \begin{vmatrix} a_0 & a_1 & a_4 & a_3 \\ b_0 & b_1 & b_4 & b_3 \\ c_0 & c_1 & c_4 & c_3 \\ d_0 & d_1 & d_4 & d_3 \end{vmatrix} \qquad D_4 = \begin{vmatrix} a_0 & a_1 & a_2 & a_4 \\ b_0 & b_1 & b_2 & b_4 \\ c_0 & c_1 & c_2 & c_4 \\ d_0 & d_1 & d_2 & d_4 \end{vmatrix}$$

Ersichtlich unterscheiden sich die Determinanten D_1 bis D_4 von der Determinante D nur darin, daß jeweils eine Spalte durch a_4 bis d_4 ersetzt ist. Statt nun wie in Beispiel 2.16 alle Determinanten auszurechnen und die Werte einzusetzen, sollen D und D_1 bis D_4 durch ein Unterprogramm bestimmt werden. Als weiteres Unterprogramm wird jeweils der Platztausch der Spalte a_4 bis d_4 mit den anderen Spalten durchgeführt.

Die Determinante ergibt

$$\begin{aligned} D = \; & (a_0\, b_1 - b_0\, a_1)\,(c_2\, d_3 - d_2\, c_3) \\ & + (a_1\, b_2 - b_1\, a_2)\,(c_0\, d_3 - d_0\, c_3) \\ & + (a_2\, b_3 - b_2\, a_3)\,(c_0\, d_1 - d_0\, c_1) \\ & + (a_3\, b_1 - b_3\, a_1)\,(c_0\, d_2 - d_0\, c_2) \\ & + (a_2\, b_0 - b_2\, a_0)\,(c_1\, d_3 - d_1\, c_3) \\ & + (a_0\, b_3 - b_0\, a_3)\,(c_1\, d_2 - d_1\, c_2). \end{aligned}$$

Zu Bildung von D_1 wird die erste Determinantenspalte a_0 bis d_0 abgespeichert und durch a_4 bis d_4 ersetzt. Bei der gezeigten Speicherbelegung wird in S 4 bis S 7 abgespeichert und die Inhalte von S 0 bis S 3 (d.h. a_4 bis d_4) in die Speicher 0, 4, 8 und C übertragen. Zur Bildung von D_2 werden zuerst die Speicherinhalte von S 4 bis S 7 in die erste Spalte zurückübertragen. Dann werden die Inhalte von 1 bis D (also a_1 bis d_1) in S 4

bis S 7 abgespeichert und die Inhalte von S 0 bis S 3 in die zweite Spalte übertragen. Entsprechend werden D_3 und D_4 berechnet.

$$D = \begin{vmatrix} a_0 & a_1 & a_2 & a_3 & a_4 & a_0 \\ b_0 & b_1 & b_2 & b_3 & b_4 & b_0 \\ c_0 & c_1 & c_2 & c_3 & c_4 & c_0 \\ d_0 & d_1 & d_2 & d_3 & d_4 & d_0 \end{vmatrix}$$

Speicherbelegung

0	1	2	3	S 0	S 4
4	5	6	7	S 1	S 5
8	9	A	B	S 2	S 6
C	D	E	S 8	S 3	S 7

Im Anhang ist die Programmliste für ein Programm auf zwei Karten aufgeführt. Nach Einlesen des ersten Teils werden die Größen a_0 bis a_4, b_0 bis b_4, c_0 bis c_4 und d_0 bis d_4 zeilenweise eingegeben. Dabei wird a_0 auf Taste A eingegeben, die weiteren Werte a_1 bis d_4 mit R/S. Zur Kontrolle erscheint nach Eingabe jeder vollständigen Zeile die Zeilen-Nr. 1 bis 4 als 1111111111 bis 4444444444.

Nach dem Eingeben der Daten bleibt der Rechner eingeschaltet. Nun wird das eigentliche Rechenprogramm eingegeben. Gestartet wird das Rechenprogramm durch Drücken von Taste E. Der Rechner berechnet zuerst die Determinante D. Ist sie ungleich Null, so hat das Gleichungssystem eindeutige Lösungen. Durch Drücken von R/S wird x berechnet und angezeigt. Durch weitere R/S folgen der Reihe nach y, z und u. Ein versehentliches viertes Drücken von R/S schadet nichts: Die Programmausführung kehrt nach Label E zurück. Welche der Größen x, y, z, u der Rechner gerade anzeigt läßt sich auch durch Umschalten auf W/PRGM und Prüfen der Schritt-Nr. feststellen.

Folgende Gleichungen sind zu lösen:

```
x + y - z + u                 = -2
-x - y + 3 z - 4 u            =  6
2 x + 3 y - 4 z + 8 u         = -9
0.5 x + 1.5 y + 0.5 z - 2 u   =  0

x + y + z + u                 =  5
x - y + z - u                 =  1
2 x + 3 y - 4 z + 5 u         =  4
3 x - 2 y + 5 z + 5 u         =  0
```

Die Eingabe erfolgt nach Einschieben der Eingabekarte in Stellung RUN beim linken Gleichungssystem als

1. Zeile	2. Zeile	3. Zeile	4. Zeile
1	1	2	•
A	CHS	R/S	5
1	R/S	3	R/S
R/S	1	R/S	1
1	CHS	4	•
CHS	R/S	CHS	5
R/S	3	R/S	R/S
1	R/S	8	•
R/S	4	R/S	5
2	CHS	9	R/S
CHS	R/S	CHS	2
R/S	6	R/S	CHS
	R/S		R/S
			0
			R/S
Anzeige: 1111111111	2222222222	3333333333	4444444444

Anschließend werden beide Seiten der Rechenkarte eingelesen und E gedrückt. x wird angezeigt, dann nach wiederholtem R/S y, z und u.

Beispiel 3.3: Ableitungen $f'(x)$ bis $f^{IV}(x)$

In der Differentialrechnung bezeichnet man als die Ableitung $f'(x)$ einer Funktion $f(x)$ ihre Steigung im Punkt x. Die Steigung ist gegeben durch den Unterschied zweier eng benachbarter

Funktionswerte geteilt durch ihren Abstand. Der Abstand 2 h wird dabei als beliebig klein angenommen.

$$f'(x) = \frac{f(x + h) - f(x - h)}{2\ h}, \quad 2\ h \to 0$$

Für eine zahlenmäßige Berechnung von $f'(x)$ auf einem Digitalrechner kann 2 h nicht beliebig klein gewählt werden. Wegen der endlichen Stellenzahl mit der Rechner rechnen (vgl. Kapitel 1.4.2) unterscheiden sich $f(x + h)$ und $f(x - h)$ bei kleinem h nur um wenige Stellen. Das berechnete $f'(x)$ wird dann ungenau. Wird andererseits 2 h zu groß gewählt, so gibt die Berechnung nicht die Steigung $f'(x)$ im Punkte x, sondern an einer benachbarten Stelle wieder. Das Ergebnis ist dann ebenfalls ungenau. Der Abstand 2 h muß also dem Kurvenverlauf angepaßt sein. Z.B. ist bei $f'(x) = 1$ und $2\ h = 10^{-4}$ auch

$$f(x + h) - f(x - h) = 10^{-4}.$$

Dies bedeutet bei einem 10-stelligen Rechner Rundung in der 6. Stelle, also ein noch recht genaues Ergebnis.

$f'(x)$, auch erste Ableitung genannt, gibt die Steigung der Funktion $f(x)$ wieder. Die zweite Ableitung berechnet die Änderung der Steigung. Sie bildet damit ein Maß für die Krümmung. Es ist entsprechend

$$f''(x) = \frac{f'(x + h) - f'(x - h)}{2\ h}, \quad 2\ h \to 0.$$

Die Beziehung läßt sich durch die vorige Gleichung für $f'(x)$ ausdrücken. So wird auch die zweite Ableitung durch $f(x)$ dargestellt. Entsprechend kann mit der dritten und vierten Ableitung verfahren werden. Es ist damit

Ausdruck für eine zahlenmäßige (numerische) Lösung
$f'(x) = \frac{1}{2h}[f(x+h) - f(x-h)]$
$f''(x) = \frac{1}{4h^2}[f(x+2h) - 2f(x) + f(x-2h)]$
$f'''(x) = \frac{1}{8h^3}[f(x+3h) - 3f(x+h) + 3f(x-h) - f(x-3h)]$
$f^{IV}(x) = \frac{1}{16h^4}[f(x+4h) - 4f(x+2h) + 6f(x) - 4f(x-2h) + f(x-4h)]$

Nach L. Collatz+ werden die berechneten Ableitungen genauer, wenn zu den vorhandenen Stützstellen noch weitere hinzugenommen werden durch Bilden einer Taylorreihe mit anschließendem Koeffizientenvergleich. Mit zwei zusätzlichen Stützstellen ergibt sich nach einer Rechnung von J. Gruss++:

Ausdruck für eine zahlenmäßige (numerische) Lösung mit erhöhter Genauigkeit
$f'(x) = \frac{1}{12h}[-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)]$
$f''(x) = \frac{1}{12h^2}[-f(x+2h) + 16f(x+h) - 30f(x) + 16f(x-h) - f(x-2h)]$
$f'''(x) = \frac{1}{8h^3}[-f(x+3h) + 8f(x+2h) - 13f(x+h) + 13f(x-h) - 8f(x-2h) + f(x-3h)]$
$f^{IV}(x) = \frac{1}{6h^4}[-f(x+3h) + 12f(x+2h) - 39f(x+h) + 56f(x) - 39f(x-h) + 12f(x-2h) - f(x-3h)]$

+ Zitiert von Becker, Numerische Mathematik für Ingenieure, Teubner Verlag 1977

++ J. Gruss, Display 1978, V4, N1/2

Eine Programmschrittliste für den erweiterten Ausdruck ist im Anhang wiedergegeben. Die Funktion f(x) ist unter Label f e untergebracht, Funktionswerte werden durch Drücken von E abgerufen. Die Funktion f(x) wird also nur einmal eingegeben. Sie wird dann für x, x + 2 h, ... jeweils als Unterprogramm aufgerufen. Das Programm bleibt dadurch übersichtlich. Vor Beginn der Rechnung ist h als Zahlenwert im Register 2 zu speichern.

Beispiel:

$f(x) = e^{-x^2}$. In Stellung W/PRGM wird gedrückt GTO · 179 (Beginn von f(x)), dann wird eingegeben

g x^2
CHS
g e^x.

Anschließend wird in RUN gedrückt EEX
CHS
2
STO 2 (h = 10^{-2}).

Nun kann für beliebige x-Werte die erste bis vierte Ableitung aufgerufen werden. Es wird der Reihe nach eingegeben
$x = 0, \frac{1}{\sqrt{2}}, 1, \sqrt{\frac{3}{2}}, \ldots$

Ein gewisser Nachteil des Programms ist, daß h bei jeder neuen Funktion neu bestimmt werden muß, gelegentlich sogar innerhalb einer Funktion. Grundsätzlich sollte h nicht zu klein sein und 10^{-4} des x-Wertes nicht unterschreiten. Wie groß manchmal h gewählt werden muß, zeigt die folgende Aufgabe.

Beispiel:

$f(x) = x^4$, gesucht sind die Ableitungen für $x = 10^4$. In Stellung W/PRGM wird gedrückt GTO · 179, dann

g x^2
g x^2.

In Stellung RUN wird gedrückt EEX
2
STO 2 (h = 100).

Die Werte der Ableitungen stimmen voll überein mit den nach den Ableitungsregeln berechneten Werten. Auch für h = 1000 ergibt sich Übereinstimmung, nicht dagegen für h = 10, 50 oder 70.

Beispiel:

f(x) = cos x, x soll in Winkelgrad eingegeben werden. Bei Winkelfunktionen ist zu beachten, daß h in den Ableitungen an zwei verschiedenen Stellen auftaucht: in der Funktion (Eingabe meist in Winkelgrad) sowie im Nenner (Eingabe in Radian). Da 180° dem Winkel π entsprechen, können die Winkelgrad im Nenner umgewandelt werden gemäß

$$\hat{\alpha} = \alpha^{\circ} \cdot \frac{\pi}{180}.$$

Zur Eingabe der Funktion wird in Stellung W/PRGM gedrückt GTO · 179, dann f COS.

Danach wird gedrückt GTO · 169 und eingesetzt

h π
x
1
8
0
÷

Dies wird an den weiteren Programmstellen wiederholt. In W/PRGM wird gedrückt GTO · 114 und eingesetzt
GTO · 105 "
GTO · 067 "
GTO · 028 "

Durch das Einsetzen zuerst in der Nähe des Programmendes ist die Schrittnumerierung der vorderen Änderungsstellen zunächst unverändert geblieben. Dadurch wird das Auffinden dieser Stellen merklich erleichtert.

Ein h = 1.5^{o} bringt die gewünschte Genauigkeit, in RUN wird gedrückt

```
  1
  .
  5
STO 2
```

Nun können die Werte der Ableitungen aufgerufen werden. Es wird z.B. gedrückt:

x = 0°, 45°, 90°, ...

3.2 Sprung zu Unterprogrammen mit GTO

Mit dem Befehl GTO (go to, gehe zu) wird ein Sprung zum angegebenen Programmteil angeordnet. Ein solcher Sprung wird besonders bei den noch zu besprechenden Programmverzweigungen eingesetzt (vgl. Kapitel 4). GTO-Befehle werden auch verwendet, wenn eine längere Schrittfolge mehrfach im Programm auftritt.

3.2.1 Sprung GTO Zeilen-Nr.

Dieser Sprung ist die Betriebsart im HP 25. Mit dem Befehl GTO Zeilen-Nr. lassen sich beim HP 25 mühelos sämtliche 50 Programmzeilen erreichen. Der Befehl GTO 08 bewirkt einen Sprung zur Zeile 8, GTO 49 den Sprung zur Zeile 49. Insbesondere bewirkt der Befehl GTO 00 den Rücksprung zum Anfang des Programmspeichers. Wie in Kapitel 2.4.3 beschrieben, ist in Zeile 0 ein Haltebefehl eingebaut. GTO 00 hat daher beim HP 25 die gleiche Wirkung wie ein RTN bei den anderen HP-Rechnern.

3.2.2 Sprung GTO Programm-Marke

Der Sprung GTO Programm-Marke wird bei den HP-Rechnern 65, 67 und 97 verwendet. Ein Sprung zu Label 7 erfolgt durch GTO 7, ein Sprung zu Label E durch GTO E. Endet ein Teilprogramm mit GTO, so ist kein zusätzliches RTN erforderlich: Mit GTO erfolgt der Sprung zum angegebenen Label, die nächste Programmzeile wird nicht mehr aufgerufen.

Beispiel 3.4: $f_1(x) = \frac{x}{1 + x^2}$ und $f_2(x) = \frac{\sin x}{1 + \sin^2 x}$

Wie zu sehen ist, entspricht die zweite Funktion der ersten, wenn x durch sin x ersetzt wird. Die zweite Funktion kann also über einen GTO-Befehl an die erste angeschlossen werden. $f_1(x)$ sei durch Drücken von A aufzurufen, $f_2(x)$ durch B.

```
1     f LBL A                                    f LBL B
        ENT                                       f SIN
        g x²                                10    GTO A
         1
5        +
         ÷          f₁(x) = x / (1 + x²)
       h RTN
```

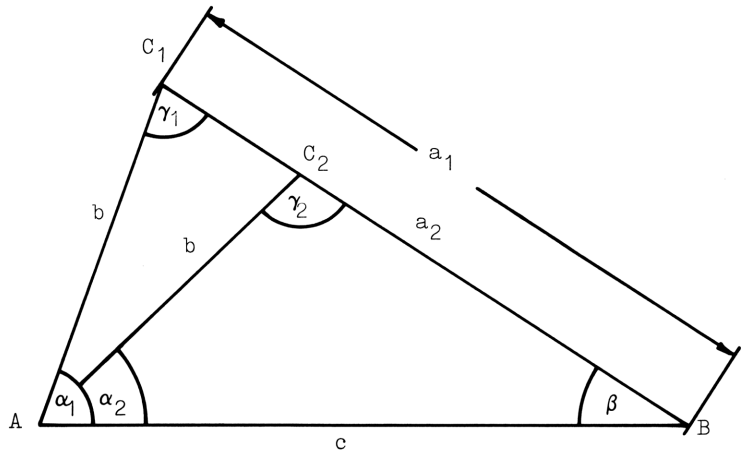
Beispiel 3.5: Dreiecksrechnung

Sind von den sechs Bestimmungsgrößen eines Dreiecks (3 Seiten, 3 Winkel) drei bekannt, so können die drei unbekannten ausgerechnet werden. Die Bestimmungsgleichungen für die Winkel α, β, γ und die den Winkeln gegenüberliegenden Seiten a, b, c sind:

Sinussatz, z.B. $\frac{\sin \alpha}{\sin \beta} = \frac{a}{b}$

Kosinussatz, z.B. $a^2 = b^2 + c^2 - 2\,b\,c \cdot \cos \alpha$

Winkelsumme $\alpha + \beta + \gamma = 180^\circ$



Das Programm gibt für

Gegebene Stücke	Label	Eingabe	Ausgabe
Drei Seiten	A	a ↑ b ↑ c	α, ß, γ
Zwei Seiten, eingeschlossener Winkel	B	b ↑ c ↑ α	a, ß, γ
Zwei Seiten, anliegender Winkel rechts	C	b ↑ c ↑ ß	a_1, α_1, γ_1/ a_2, α_2, γ_2
Zwei Seiten, anliegender Winkel links	D	b ↑ c ↑ γ	a_1, α_1, $ß_1$/ a_2, α_2, $ß_2$
Eine Seite, beide anliegenden Winkel	E	c ↑ α ↑ ß	a, b, γ

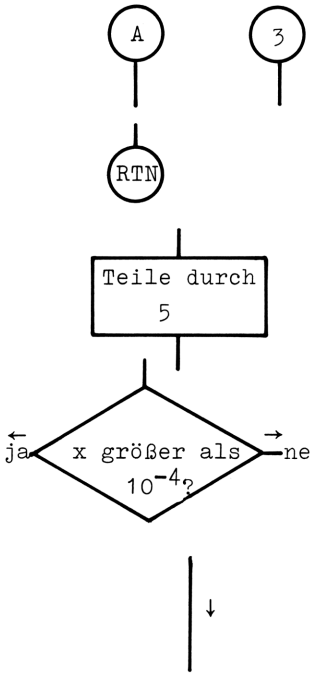
Die Eingabe erfolgt als zweimaliges Enter mit anschließender Labeltaste, also bei A durch a ↑ b ↑ c. Die Ausgabe des ersten Wertes erfolgt selbständig, die nächsten folgen mit R/S. Bei C und D sind als Ausgabe je zwei Gruppen angegeben. Hier bestehen zwei Lösungen, die durch insgesamt fünfmaliges Drücken von R/S abgerufen werden. Die erste Lösung ergibt das spitzwinkelige Dreieck, die zweite das kleinere stumpfwinkelige. Die Programmliste ist im Anhang aufgeführt.

4. Programmverzweigungen

Programmverzweigungen sind die Stellen im Programm, von denen aus wahlweise ein geänderter Rechengang durchgeführt oder aber die Programmausführung unverändert fortgesetzt wird. Welcher Zweig gewählt wird, hängt vom Ergebnis einer Vergleichsprüfung oder einer Flagprüfung ab. Sowohl Vergleichs- wie auch Flagprüfer lassen beim Prüfergebnis "ja" die Programmausführung ungeändert, bewirken jedoch beim Ergebnis "nein" bei den Rechnern HP 25, 67 und 97 ein Überspringen der auf die Prüfung folgenden Programmzeile. Beim HP 65 werden hierbei zwei Zeilen übersprungen. Die übersprungene Zeile enthält z.B. einen GTO-Befehl, eine GSB-Routine, Speicherarithmetik oder eine CHS-Anweisung. Das bedeutet, daß im allgemeinen bei "nein" die ungeänderte Programmausführung mit einer übersprungenen Programmzeile erfolgt. Bei "ja" wird dagegen mit einem GTO-Befehl die weitere Programmausführung in einem ganz anderen Programmteil durchgeführt.

4.1 Veranschaulichen des Rechengangs

Um bei einer Programmerstellung den Überblick zu behalten, sollte der Rechengang aufgezeichnet werden. Dazu werden verschiedene Zeichen verwendet.



Ein Kreis mit einem Buchstaben oder einer Zahl kennzeichnet den einzelnen Label

Ein Kreis mit RTN zeigt das jeweilige Programmende

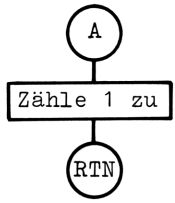
Ein Rechteck mit Text beschreibt den durchzuführenden Rechenschritt

Eine Raute kennzeichnet die Vergleichs- oder Flagprüfung

Eine gerade Linie verbindet die Zeichen. Falls erforderlich kann ein Pfeil die Richtung des Rechenganges angeben.

Beispiel:

Zur eingegebenen Zahl soll jeweils 1 dazugezählt werden.




```
f LBL A
   1
   +
 h RTN
```

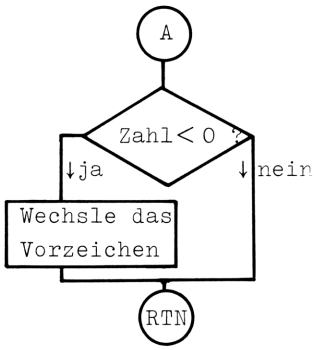
4.2 Vergleichsprüfungen

Der HP 65 verfügt über vier Vergleichsprüfungen für den Vergleich der Inhalte des X- gegenüber dem Y-Register. Die Rechner HP 25, 67 und 97 haben zusätzlich vier Prüfungen zum Vergleich des X-Registers gegenüber Null. Es bedeuten:

Prüfung	Bedeutung
f x = 0	Ist der Inhalt des X-Registers gleich Null?
f x ≠ 0	" ungleich " ?
f x < 0	" kleiner " ?
f x > 0	" größer " ?
g x = y	Ist der Inhalt des X-Registers gleich dem Inhalt des Y-Registers?
g x ≠ y	" ungleich "
g x ≤ y	" kleiner oder gleich "
g x > y	" größer "

Beispiel 4.1: Umwandeln negativer Zahlen

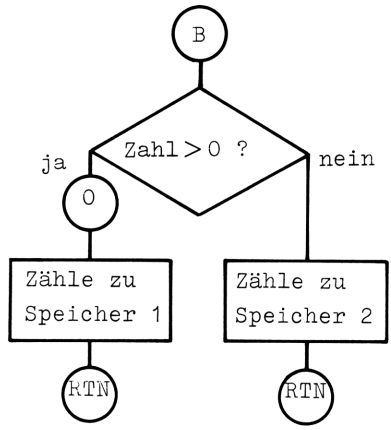
Eine Folge von Zahlen mit unterschiedlichem Vorzeichen kann z.B. durch h ABS oder g x^2 mit nachfolgendem f $\sqrt{x}$ in positive Zahlen umgewandelt werden. Dies gelingt auch durch jeweilige Vorzeichenprüfung mit anschließendem CHS für die negativen Zahlen:




```
      f LBL A
      f x<0
ja ↓   CHS      nein
      h RTN
```

Beispiel 4.2: Unterscheiden des Vorzeichens

Eingetastete Zahlen sollen aufsummiert werden, die positiven Zahlen in Speicher 1, die negativen in Speicher 2.

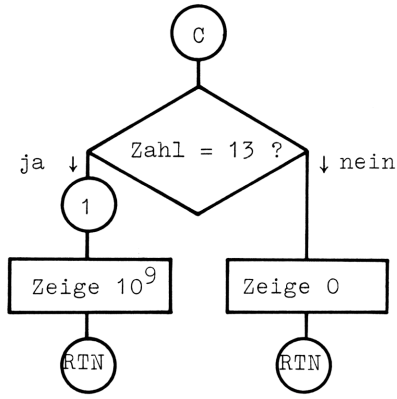



```
          f LBL B
           f x>0
ja ↓       GTO 0      nein
          STO + 2
           h RTN
-------------------------
          f LBL 0
          STO + 1
           h RTN
```

Beispiel 4.3: Erraten einer Zahl

Im Programmspeicher ist eine Zahl verborgen. Werden Zahlen eingegeben, so ist die Anzeige 0 außer wenn die Zahl erraten wurde.

Format: DSP 0

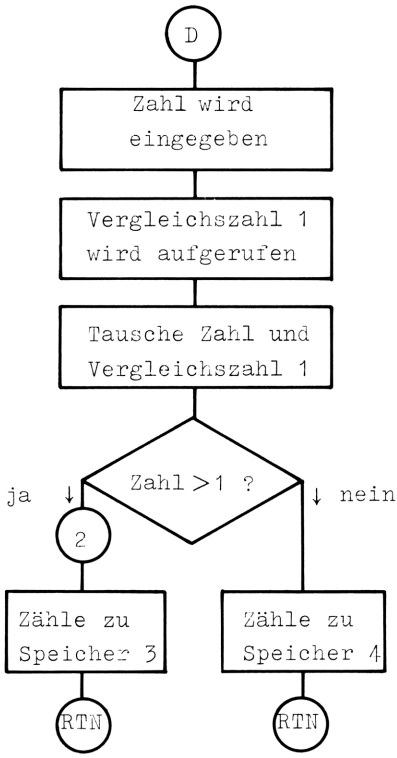



```
          f LBL C          Y-Register
            1
            3              X-Register
          g x = y
ja ↓      GTO 1     nein
            0
           h RTN
-------------------------
          f LBL 1
            EEX
            9
           h RTN
```

Bei der Vergleichsprüfung befindet sich die eingegebene Zahl im Y-Register, die Prüfzahl 13 im X-Register. Im allgemeinen soll nach dem Vergleich die geprüfte Zahl weiter verarbeitet werden. Dazu muß sie zuerst vom Y- ins X-Register gebracht werden, z.B. mit h x ⇄ y oder mit h R↓ . Häufig ist es vorteilhaft, den Austausch bereits vor der Vergleichsprüfung durchzuführen.

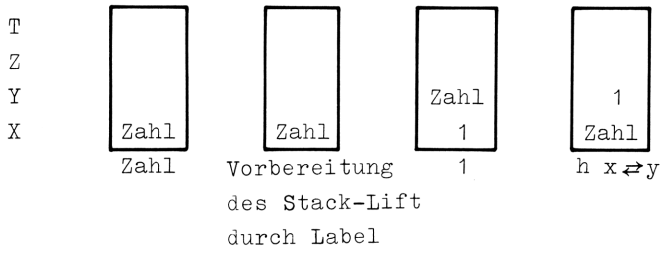
Beispiel 4.4: Sortieren von Zahlen in zwei Gruppen

Alle Zahlen größer als eine gegebene Vergleichszahl sollen in Speicher 3 aufsummiert werden. Alle Zahlen kleiner oder gleich der Vergleichszahl sollen in Speicher 4. Die Vergleichszahl sei 1.




```
f LBL D
   1
 h x⇄y
 g x>y
ja↓ GTO 2   nein
 STO + 4
 h RTN
f LBL 2
STO + 3
 h RTN
```

Zur Veranschaulichung ist zusätzlich die Belegung des Stack gezeigt:



Jetzt steht bei der Vergleichsprüfung g x>y die zu prüfende Zahl im X-Register. Dadurch kann der gewünschte Vergleich Zahl>1 ? durchgeführt werden.

Solche Vergleichsprüfungen werden manchmal im Rechengang unvermutet notwendig. So wird beim Rechnen mit Brüchen gelegentlich ein Nenner zu Null. Der Rechner bricht daraufhin die Ausführung mit einer Fehlermeldung ab. Die Nachprüfung ergibt, daß in diesem Fall (Entartung) ein abgeänderter Rechengang durchzuführen ist. So hat die quadratische Gleichung

$$a\,x^2 + b\,x + c = 0$$

die Lösung

$$x_{1,2} = \frac{1}{a}\left(-\frac{b}{2} \pm \sqrt{\frac{b^2}{4} - a\,c}\right).$$

Ist a = 0, so ist für $x_{1,2}$ keine Lösung ersichtlich. Die ursprüngliche Gleichung jedoch lautet jetzt

$$b\,x + c = 0.$$

Als Lösung folgt damit

$$x = -\frac{c}{b}.$$

Für a = 0 ist also ein anderer Rechengang zu wählen.

Ein ähnliches Verschwinden des Nenners tritt in Beispiel 2.30 auf. In diesem Beispiel wurde ein Programm für die Plancksche Strahlungsformel aufgestellt mit

$$K_\nu = \frac{2\,h\,\nu^3}{c^2\left(e^{\frac{h\nu}{kT}} - 1\right)}$$

Für $\frac{h\nu}{kT}$ kleiner als $5 \cdot 10^{-10}$ setzt der Rechner

$$e^{\frac{h\nu}{kT}} = 1$$

Dadurch wird der Nenner von K_ν Null, die Programmausführung wird abgebrochen. Außerdem treten im Bereich $5 \cdot 10^{-10} < \frac{h\nu}{kT} < 5 \cdot 10^{-5}$ unerwünschte Rundungen auf (vgl. Kapitel 1.4.2).

Bei kleinen Exponenten $\frac{h\nu}{kT}$ ist daher für K_ν eine andere Beziehung zu wählen. Die Funktion wird dazu als Reihe dargestellt. Es ist

$$\left(e^{\frac{h\nu}{kT}} - 1\right) = \left[1 + \frac{h\nu}{kT} + \frac{1}{1\cdot 2}\left(\frac{h\nu}{kT}\right)^2 + \frac{1}{1\cdot 2\cdot 3}\left(\frac{h\nu}{kT}\right)^3 + \ldots\right]^{-1}$$

$$= \frac{h\nu}{kT}\left[1 + \frac{1}{2}\frac{h\nu}{kT} + \frac{1}{6}\left(\frac{h\nu}{kT}\right)^2 + \cdots\right]$$

Im Bereich $\frac{h\nu}{kT} < 5 \cdot 10^{-5}$, also $\frac{\nu}{T} < 10^6$, ist das zweite Glied ebenso wie die folgenden Reihenglieder klein gegen die Eins. Sie können daher ohne merkliche Einbuße an Genauigkeit fortgelassen werden. Es wird damit

$$\left(e^{\frac{h\nu}{kT}} - 1\right) \approx \frac{h\nu}{kT}$$

Für K_ν folgt

$$K_\nu = \frac{2\,h\nu^3}{c^2 \cdot \frac{h\nu}{kT}}$$

oder

$$K_\nu = 2\,kT\left(\frac{\nu}{c}\right)^2.$$

Dies ist die bekannte Rayleigh-Jeanssche Näherung für niedrige Frequenzen. Damit kann K_ν für $\frac{\nu}{T}$ kleiner als 10^6 berechnet werden. Für $\frac{\nu}{T}$ größer als 10^6 wird weiterhin die Plancksche Formel verwendet. Im Programm wird also $\frac{\nu}{T}$ mit 10^6 verglichen. Wie in Beispiel 2.29 wird zur Berechnung von K_ν die Konstantenkarte verwendet. Die Temperatur in Kelvin wird in Register 7 gespeichert. Das vollständige Programm lautet daher:

```
20        f LBL A
           STO 8      ν
           RCL 7      T
             ÷
            EEX
25            6
          g x > y ┐
     ja↓  GTO 1   │ nein
           RCL 8 ┘
          g x²
30        RCL 8
             x       ν³
          RCL 1      h
             x
             2
35           x
          RCL 3
          g x²
             ÷
          RCL 8
40        RCL 1
```

```
41           x
          RCL 2      k
             ÷
          RCL 7
45           ÷
          g eˣ
             1
             -
             ÷       Kν als
50        h RTN      Plancksche Form
```

```
          f LBL 1
          RCL 8      ν
          RCL 3
             ÷
55        g x²
          RCL 7
             x
          RCL 2
             x
60           2
             x       Kν als Rayleigh-
          h RTN      Jeans-Näherung
```

Zusätzlich läßt sich K als Funktion der Wellenlänge λ ausdrücken. Das Emissionsvermögen zwischen λ und $\lambda + d\lambda$ ist

$$K_\lambda \cdot d\lambda = K_\nu \cdot d\nu \, .$$

Wegen $\lambda = \frac{c}{\nu}$ und $d\lambda = -\frac{c}{\nu^2} \cdot d\nu$

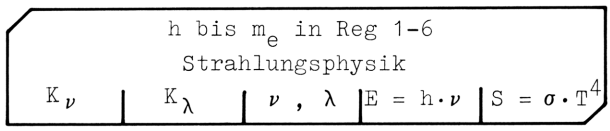
ist

$$K_\lambda = \frac{\nu^2}{c} \cdot K_\nu \, .$$

Auf Label B wird die Wellenlänge λ in Metern eingegeben, Register 7 enthalte weiterhin die Temperatur.

```
                f LBL B
65              h 1/x
                RCL 3       c
                  x         ν = c/λ
                  A         K_ν
                RCL 8       ν
70              g x^2
                  x
                RCL 3
                  ÷         K_λ
                h RTN
```

Die vollständige Programmkarte hat also folgendes Aussehen:



Bei T = 300 K, λ = 3 mm folgt $\nu = 10^{11}$ Hz, $K_\nu = 9.1 \cdot 10^{-16} \frac{Ws}{m^2}$

und $K_\lambda = 3 \cdot 10^{-2} \frac{W}{m^3}$

Für T = 6000 K, λ = 500 nm folgt $\nu = 6 \cdot 10^{14}$ Hz,

$K_\nu = 2.7 \cdot 10^{-8} \frac{Ws}{m^2}$

und $K_\lambda = 3.2 \cdot 10^{13} \frac{W}{m^3}$.

Beispiel 4.5: Kreis durch drei Punkte

Ein Kreis mit den Mittelpunktskoordinaten a und b und dem Radius R wird durch die Gleichung beschrieben

$$(x - a)^2 + (y - b)^2 = R^2.$$

Sind umgekehrt 3 Punkte x_1, y_1; x_2, y_2 sowie x_3, y_3 gegeben, so lassen sich die Achsabstände a und b des Mittelpunktes sowie der Radius R des Kreises berechnen.

Es ist

$$2\,a = \frac{A}{B}$$

mit

$$\begin{aligned} A = \quad & y_1\,(x_3^2 - x_2^2 + y_3^2 - y_2^2) \\ & + y_2\,(x_1^2 - x_3^2 + y_1^2 - y_3^2) \\ & + y_3\,(x_2^2 - x_1^2 + y_2^2 - y_1^2) \end{aligned}$$

und

$$B = y_1\,(x_3 - x_2) + y_2\,(x_1 - x_3) + y_3\,(x_2 - x_1)$$

Entsprechend ist

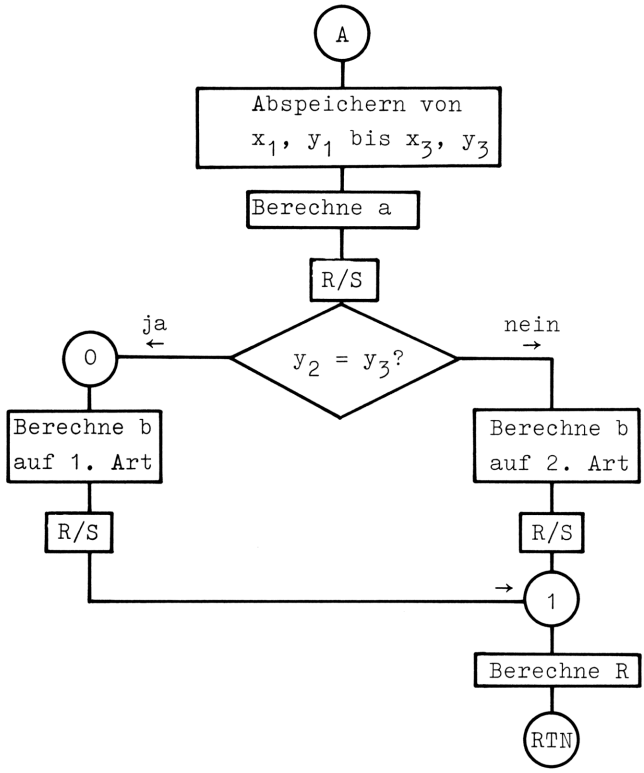
$$2\,b = \frac{x_1^2 - x_3^2 + y_1^2 - y_3^2 + 2\,a\,(x_3 - x_1)}{y_1 - y_3}$$

oder

$$2\,b = \frac{x_2^2 - x_3^2 + y_2^2 - y_3^2 + 2\,a\,(x_3 - x_2)}{y_2 - y_3}$$

$$R = \sqrt{(x_1 - a)^2 + (y_1 - b)^2}$$

Für $x_1 = x_2 = x_3$ oder $y_1 = y_2 = y_3$ liegen die Punkte auf einer Geraden, können also nicht durch einen Kreis mit endlichem a, b und R beschrieben werden. Falls aber lediglich $y_1 = y_3$ oder $y_2 = y_3$ ist, muß nur die passende Gleichung für b gewählt werden mit einem Nenner ungleich Null. Der Ablaufplan hat also folgendes Aussehen:



Eine Programmliste ist im Anhang aufgeführt. Die Eingabe erfolgt in der gewohnten Weise, also x_1 auf Label A, dann y_1, x_2, y_2 und x_3, y_3 nacheinander auf R/S. Der Mittelpunktsabstand a wird berechnet und angezeigt, dann nach Drücken von R/S b, schließlich R. Nach der Berechnung können auf Label B x-Werte des Kreises eingegeben werden. Das Programm berechnet die zugehörigen Funktionswerte des Kreises

$$y_1 = b + \sqrt{R^2 - (x - a)^2}$$

sowie nach R/S

$$y_2 = b - \sqrt{R^2 - (x - a)^2}.$$

Es werden die drei Wertepaare eingegeben

$x_1 = 3,\ y_1 = 1$	als	3	x_1
$x_2 = 1,\ y_2 = 3$		A	
$x_3 = 5,\ y_3 = 3$		1	y_1
		R/S	
		1	x_2
		R/S	
		3	y_2
		R/S	
		5	x_3
		R/S	
		3	y_3

Nach der Berechnung von a, b, R wird auf Taste B eingegeben

$x = 1$ $\quad y_1 = \cdots\ ,\ y_2 = \cdots$
$\quad\ 3$
$\quad\ 5$

Entsprechend werden weitere drei Wertepaare als Punkte auf einem anderen Kreis als A, R/S eingegeben:

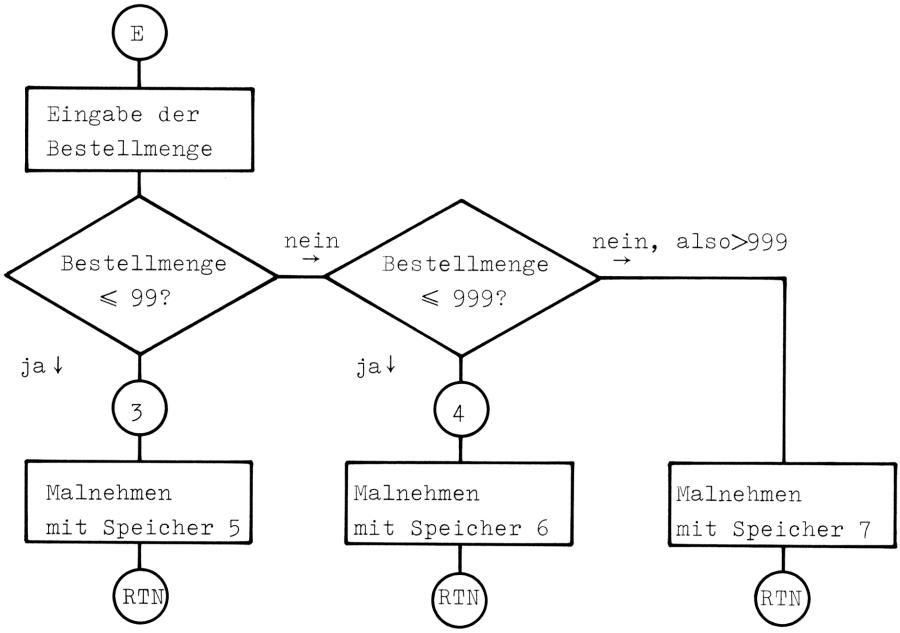
$x_1 = 2,\ y_1 = 0$
$x_2 = 6,\ y_2 = -2$
$x_3 = 10,\ y_3 = 0$

Beispiel 4.6: Preisstaffel

Es soll der Gesamtpreis von Produkten bei beliebigen Bestellungen berechnet werden, wobei eine Preisstaffelung vorliegt.

Gruppierung	Preis	Belegung
1 bis 99 Stück	3.20 DM/Stück	Speicher Nr. 5
100 bis 999	2.50	6
1000 und mehr	2.10	7

Der Ablaufplan und das Programm sehen dann folgendermaßen aus:




```
 1        f LBL E
            9
            9
          h x⇄y
 5        g x⩽y
   ja ↓    GTO 3      nein
            9
            9
            9
10        h x⇄y
          g x⩽y
   ja ↓    GTO 4      nein
          RCL 7
            x
15        h RTN
```

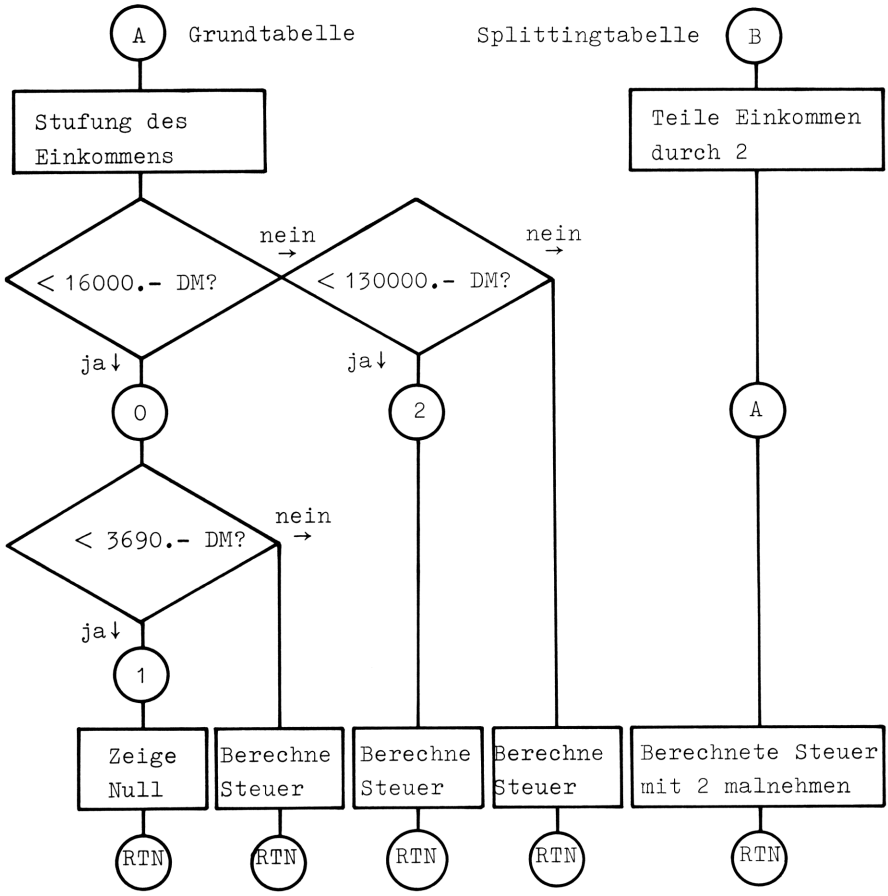
```
          f LBL 3
          RCL 5
            x
          h RTN
20        f LBL 4
          RCL 6
            x
          h RTN
```

Beispiel 4.7: Einkommensteuer

Ähnlich wie bei der Preisstaffel Beispiel 4.6 lassen sich bei der Einkommensteuer ab Januar 1979 verschiedene Bereiche unterscheiden.

Jahreseinkommen	Besteuerung für Ledige (Grundtabelle)
bis 3690.- DM	keine
3690.- bis 16000.- DM	22 % des Einkommens höher als 3690.- DM
16000.- bis 130000.- DM	2712.- DM + (22 bis 49,5)% des Einkommens höher als 16000.- DM
über 130000.- DM	59133.- DM + 56 % des Einkommens höher als 130000.- DM

Die Steuer für Verheiratete nach der Splittingtabelle berechnet sich durch Halbierung des Einkommens, Ermittlung der Steuer und Verdoppelung dieses Betrages. Der Ablaufplan zur Ermittlung der Einkommensteuer hat damit folgendes Aussehen:



Die unter Label A aufgeführte Stufung des Jahreseinkommens in 30.- DM-Stufen läßt sich bequem mit der Integer-Funktion durchführen:

```
Einkommen
   ENT
    3
    0
    ÷
  f INT
    3
    0
    x
```

Der gleichmäßig ansteigende Steuersatz im Bereich 16000.- bis 130000.- DM folgt der Hyperbel

$$\text{Prozentsatz} = 58.28 - \frac{1329400}{(\text{Einkommen höher } 16000.-) + 37530}$$

Die Programmschrittliste ist im Anhang aufgeführt. Die Abweichungen zu den Tabellenwerten bleiben unter 50.- DM.

4.3 Flagprüfungen

Flags (flags, Fähnchen) bilden zusätzliche Kennzeichen von Zahlenwerten. Mit Flags können diese Werte bei einer Flagprüfung zu anderen Programmteilen verzweigen. Die Beherrschung der Flags ist an sich nicht schwierig. Als "Leckerbissen" erfreuen sie jedoch eher den fortgeschrittenen Programmierer. Im allgemeinen lassen sich auch schwierige Rechenprobleme ohne Flags sicherer und mit weniger Programmierzeit lösen.

Flags werden gesetzt durch den Befehl h SF Flagnummer (set flag, setze das Flag). Die Flagprüfung erfolgt als h F? Flagnummer. Ist das überprüfte Flag gesetzt, so wird das Programm weiter ausgeführt. Ist das Flag nicht gesetzt, so wird ein

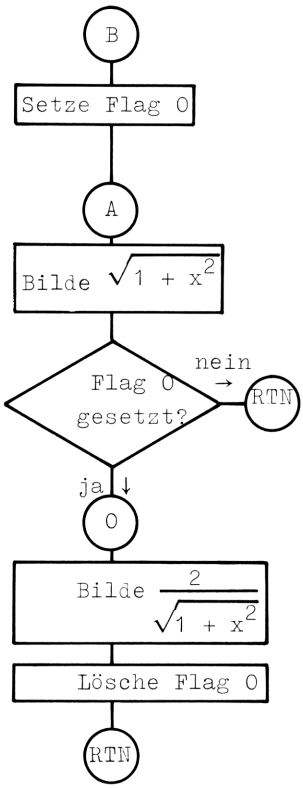
Programmschritt übersprungen beim HP 67 und 97 bzw. zwei Schritte beim HP 65. Ein Flag wird gelöscht durch den Befehl h CF Flagnummer (clear flag, lösche das Flag). Der HP 65 besitzt zwei gleiche Flags. Der HP 67 und der HP 97 verfügen dagegen über vier Flags mit unterschiedlichen Eigenschaften:

	Besonderheiten	
Flag Nr.	Eingabe	Abfrage
0	-	-
1	-	-
2	-	bei Abfrage gelöscht
3	wird bei Dateneingabe über Tastenfeld selbsttätig gesetzt	bei Abfrage gelöscht

Beispiel 4.8: $f_1(x) = \sqrt{1 + x^2}$ und $f_2(x) = \frac{2}{\sqrt{1 + x^2}}$ mit Flags

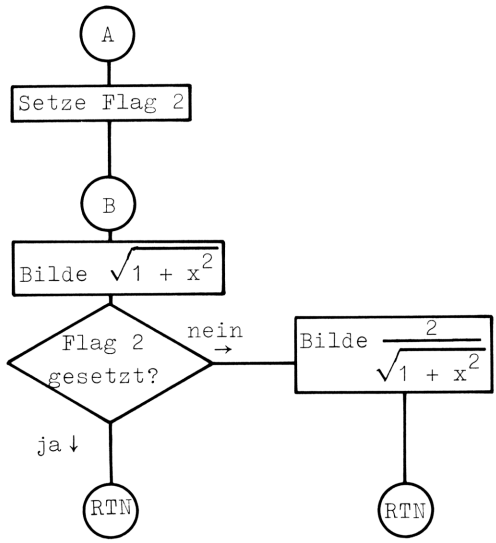
Die Funktionen (vgl. Beispiel 2.8) sollen mit Flags über die Label A und B eingegeben werden. Je nach verwendetem Flag sind verschiedene Programmabläufe möglich. Jeweils sei $\sqrt{1 + x^2}$ über Label A, $\frac{2}{\sqrt{1 + x^2}}$ über Label B aufzurufen.

Flag 0 (wird durch h CF 0 gelöscht):




```
 1          f LBL B
            h SF 0
            f LBL A
              g x²
 5            1
              +
            f √x
            h F ? 0
      ja ↓  GTO 0      nein
10          h RTN
            f LBL 0
            h 1/x
              2
              x
15          h CF 0
            h RTN
```

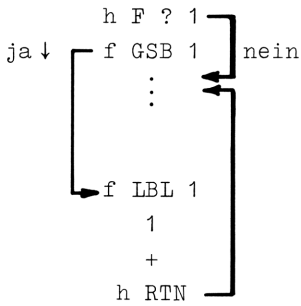
Flag 2 (wird selbstätig bei Abfrage gelöscht):



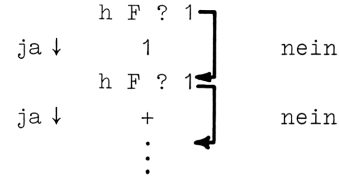

```
 1          f LBL A
            h SF 2
            f LBL B
              g x²
 5            1
              +
            f √x
            h F ? 2
      ja ↓  h RTN      nein
10          h 1/x
              2
              x
            h RTN
```

Eine wiederholte Prüfung der Flags 0 oder 1 gestattet für die "nein"-Aussage das Überspringen mehrerer Programmschritte. Häufig lassen sich so statt eines Verzweigens zum Unterprogramm direkt die benötigten Schritte eintragen. Falls z.B. bei "ja" zu einem Ausdruck 1 hinzugezählt werden soll, wäre die Schrittfolge

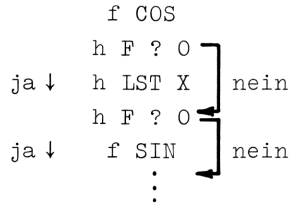
mit Unterprogramm



mit wiederholter Flagprüfung



Durch die Mehrfachanwendung von Flags wird ein Programm also schneller, übersichtlicher und kürzer. Ähnlich kann bei einem komplizierten Ausdruck, von dem z.B. bei "ja" der Sinus, bei "nein" der Cosinus gebildet werden soll, verfahren werden:



Eine besondere Möglichkeit bietet beim HP 67 und 97 das Flag 3. Dieses wird vom Rechner gesetzt, wenn Zahlentasten gedrückt werden. Beim Drücken einer Labeltaste unterscheidet der Rechner also zwischen einer Eingabe und dem Abfragen eines Ergebnisses.

Beispiel 4.9: Seitenlängen im rechtwinkeligen Dreieck

Die Label A, B, C sollen den Seiten a, b, c entsprechen. Nach Eingabe der zwei bekannten Seiten über zwei Labeltasten ist durch Drücken der dritten Labeltaste die unbekannte Seite anzugeben.

Mit c als größter Seite im rechtwinkligen Dreieck gilt

$a = \sqrt{c^2 - b^2}$, $b = \sqrt{c^2 - a^2}$, $c = \sqrt{a^2 + b^2}$

1	f LBL A		11	f LBL B		21	f LBL C	
	g x^2			g x^2			g x^2	
	STO A	a		STO B	b		STO C	c
	h F ? 3			h F ? 3			h F ? 3	
5	ja↓ R/S	nein	15	ja↓ R/S	nein	25	ja↓ R/S	nein
	RCL C			RCL C			RCL A	
	RCL B			RCL A			RCL B	
	−			−			+	
	f $\sqrt{x}$			f $\sqrt{x}$			f $\sqrt{x}$	
10	h RTN		20	h RTN		30	h RTN	

Wegen der besseren Übersichtlichkeit wurden zum Abspeichern von a, b, c die Speicher A, B, C gewählt. Im Rechengang ist ersichtlich, daß nach Dateneingabe (selbständigem Setzen von Flag 3) die Programmausführung jeweils bei R/S anhält. Soll das Ergebnis durch Drücken einer Labeltaste angegeben werden (Flag 3 nicht gesetzt), so wird R/S übersprungen. Wie erwähnt wird Flag 3 nach einer Abfrage selbstätig gelöscht.

Schließlich sei die Verwendung von Flags in einem Spiel von J.R. Jacobs+ gezeigt.

+ J.R. Jacobs, HP 65 Users' Library Nr. 406 A

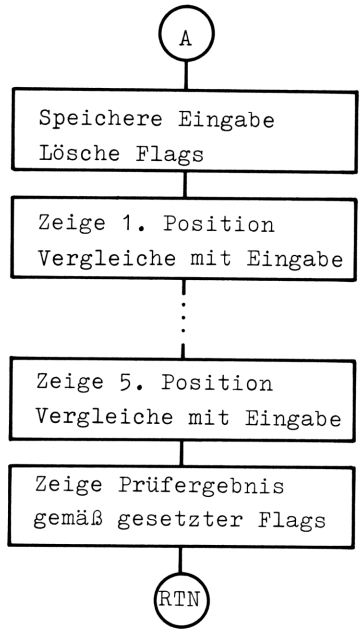
Beispiel 4.10: U-Boot-Jagd

Auf einer Spielfläche von 9 x 9 Feldern werden vom Rechner fünf U-Boote als Feldnummern versteckt. Das Feld Nr. 2.3 bedeutet z.B. ein U-Boot in Zeile 2 und Spalte 3. Wird als "Schuß" versuchsweise eine Feldnummer eingetastet, so wird diese vom Rechner überprüft. Je nach Ergebnis wird angezeigt:

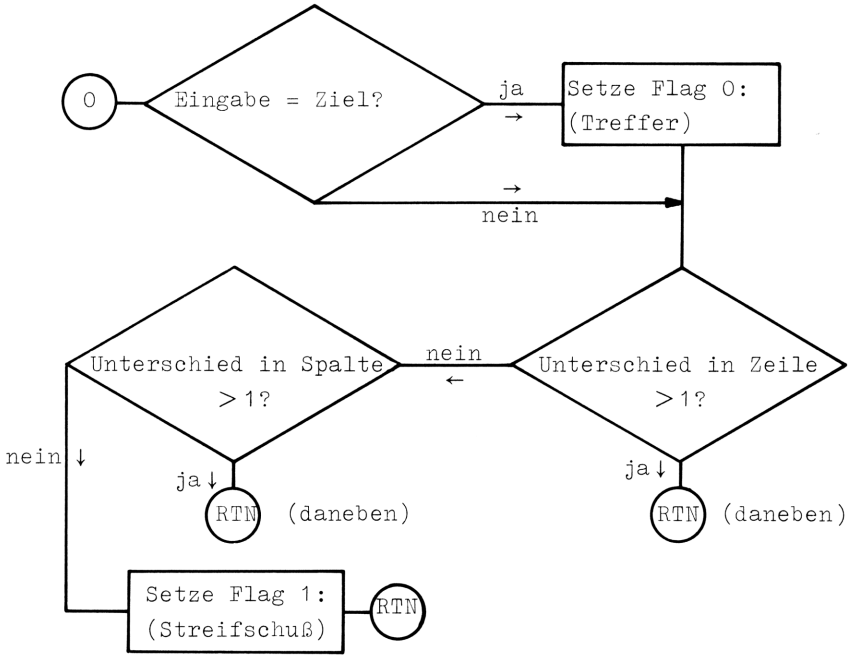
Anzeige	Bedeutung
3333333333	Spielvorbereitung nach Drücken der Labeltaste E abgeschlossen
0	Kein U-Boot auf diesem oder einem der acht angrenzenden Felder
111	Auf einem der acht angrenzenden Felder befindet sich ein U-Boot
1111111111	Treffer: ein (oder mehrere) U-Boote befinden sich auf diesem Feld

Mit möglichst wenigen Schüssen (d.h. möglichst wenigen eingegebenen Feldnummern) sollen die fünf U-Boote versenkt werden.

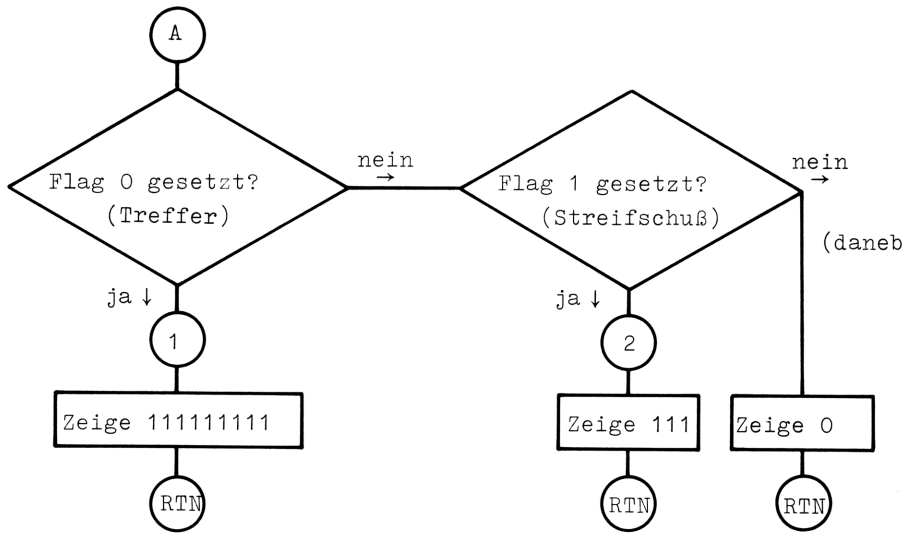
Im Programmablauf wird die eingegebene Feldnummer der Reihe nach mit sämtlichen fünf gespeicherten Positionen verglichen. Der Prüfplan hat damit folgendes Aussehen:



Beim Vergleich der jeweiligen U-Boot-Position mit dem Eingabewert werden Flags gesetzt: Flag 0 bei Treffer und Flag 1 bei benachbarter Lage. Diese liegt vor, wenn sich Zeile und Spalte höchstens um eins unterscheiden. Der Ablaufplan hierfür hat also folgendes Aussehen:



Je nach dem Ergebnis des Vergleichs erfolgt die Anzeige:



Das Teilprogramm für den Spielstart soll auf Tastendruck jeweils fünf neue Positionen in den Speichern unterbringen. Dazu wird vor dem ersten Spiel eine willkürliche Zahl eingegeben. Bei weiteren Spielen genügt es zur Spielvorbereitung die Labeltaste E zu drücken. Die Schrittliste ist im Anhang aufgeführt.

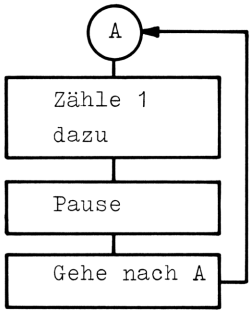
5. Programmschleifen

5.1 Schleifen ohne Bedingung

Programmschleifen (englisch: loops) sind Teile des Programms, die mehrfach durchlaufen werden. Dies erfolgt durch einen GTO-Befehl am Ende des Programmteils, worauf die Ausführung bei der angegebenen Programm-Marke wiederholt wird. Das Programm hält an infolge eingebauter R/S-Befehle oder auch infolge erfüllter Bedingungen bei Vergleichs- oder Flagprüfungen. Ein Zwischen-Stop kann durch den h PAUSE-Befehl bewirkt werden.

Beispiel 5.1: Ansteigende Zahlenreihe

Die Zahlen 1, 2, 3, ... sollen der Reihe nach gezeigt werden.




```
f LBL A
   1
   +
h PAUSE
 GTO A
```

Gestartet wird das Programm durch Drücken von A. Es kann unterbrochen werden durch Drücken einer Taste wie CLX während der Programmausführung (also nicht während des Zwischen-Stops).

In Stellung RUN wird gedrückt DSP 0, dann 0
A

Die Anzeige zeigt 1, 2, 3, 4, ...

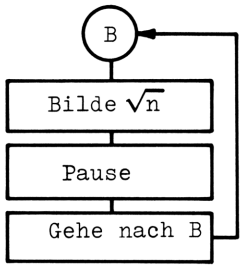
Nach dem Anhalten des Programms wird z.B. eingegeben

```
1
7
CHS
A
```

Auf der Anzeige erscheint -16, -15, -14, -13, ...

Beispiel 5.2: Mehrfache Wurzel einer Zahl

Es soll aus einer beliebigen Zahl n größer als Null der Reihe nach $\sqrt{n}$, $\sqrt[4]{n}$, $\sqrt[8]{n}$, ... gebildet werden, also immer die Quadratwurzel des vorhergegangenen Wertes. Die Ergebnisse sollen selbstätig angezeigt werden.




```
f LBL B
f √x
h PAUSE
GTO B
```

Es werde eingegeben:

```
EEX
 8   (= 10^8)
 B
```

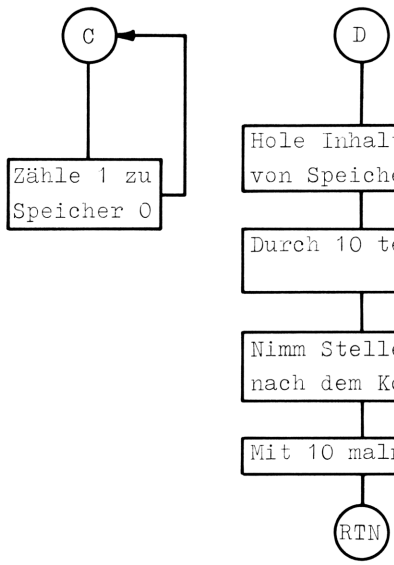
dann nach Stoppen des Programms:

```
DSP
 4
EEX
CHS
 8    (= 4 · 10^-8)
 B
```

Bereits nach 12 Programmschleifen sind die Werte 1.0045 bzw. 0.9955 erreicht. Der Grenzwert für $\sqrt[a]{n}$ mit n als beliebiger Zahl und a über alle Grenzen wachsend ist erwartungsgemäß genau 1.

Beispiel 5.3: Glücksrad

Das Programm gibt zufällige Zahlen von 0 bis 9 durch Stoppen eines laufenden Zählwerks unter Label C. Nach dem Anhalten des Zählers wird die erste Stelle vor dem Komma als einstellige Zahl durch Drücken der Labeltaste D angezeigt.



Format: DSP 0

```
1     f LBL C
        1
      STO + 0
      GTO C
5     f LBL D
      RCL 0
        1
        0
        ÷
10    g FRAC
        1
        0
        x
      h RTN
```

In Stellung RUN wird C gedrückt. Nach ca. 20 Sekunden wird die Programmausführung durch Drücken einer beliebigen Taste (z.B. R/S) unterbrochen. Drücken von D gibt eine einstellige zufällige Zahl.

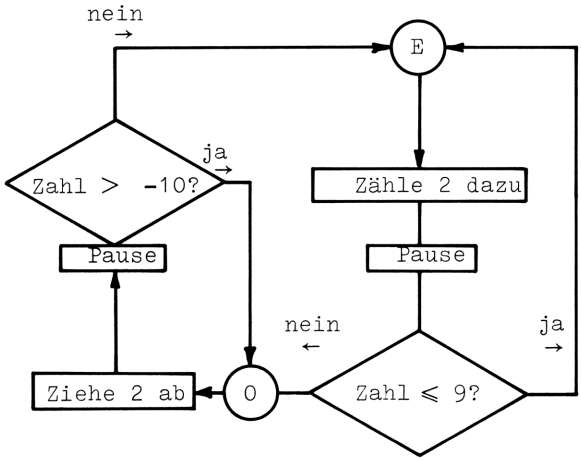
5.2 Schleifen mit Vergleichsprüfungen

Wird vor dem GTO-Befehl eine Vergleichsprüfung durchgeführt, so verzweigt der Rechengang bei erfülltem Vergleich zum angezeigten Label. Bei nicht erfülltem Vergleich können h RTN, h PAUSE, R/S, beliebige Programmzeilen oder GTO weiterer Label folgen. Das aufgeführte Beispiel 5.4 ist eine Erweiterung von Beispiel 5.1.

5.2.1 Vergleich gegenüber einer festen Zahl

Beispiel 5.4: Auf- und absteigende Zahlenreihe

Die Zahlenreihe soll nur die geraden Zahlen umfassen. Sie soll zu einem Höchstwert ansteigen und dann bis zu einem Niedrigstwert abfallen. Diese Werte seien +10 und -10.



Format: DSP 0

```
                f LBL E
                   2
                   +
                h PAUSE
                   9
                h x ⇄ y          jetzt ist Zahl im X-Register
                g x ≤ y
      ja ↓       GTO E     nein
                f LBL 0
                   2
                   -
                h PAUSE
                   1
                   0
                  CHS
                h x ⇄ y          jetzt ist Zahl im X-Register
                g x > y
      ja ↓       GTO 0     nein
                 GTO E
```

In Stellung RUN wird gedrückt CLX, dann E.

Eine Schranke für die zu prüfende Zahl läßt sich bei der Bildung von Reihen einsetzen. Bekanntlich können so unterschiedliche Ausdrücke wie e^x, cos x und Ei(x) durch Reihen dargestellt werden. Die Summierung ist abzubrechen, wenn die Glieder der Reihe hinreichend klein sind.

<u>Beispiel 5.5: e^x, cos x, Ei(x) als Reihen</u>

Es ist
$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

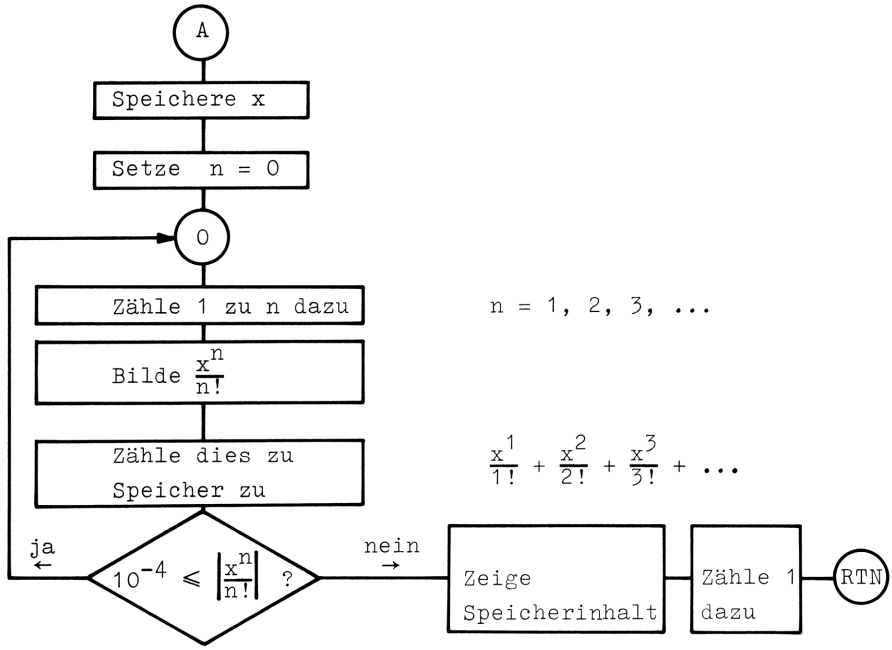
$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots$$

$$Ei(x) = C + \ln |x| + \frac{x}{1 \cdot 1!} + \frac{x^2}{2 \cdot 2!} + \frac{x^3}{3 \cdot 3!} + \cdots$$

wobei n! (gesprochen: n -Fakultät) das Produkt bedeutet
$n! = 1 \times 2 \times 3 \times \cdots \times n$.

Funktionen wie e^x oder cos x können bei vielen Rechnern durch einfachen Tastendruck aufgerufen werden. Andere Funktionen wie die bei Aufgaben der Wärmeleitung auftretende Integral-exponentialfunktion $Ei(x) = -\int_{\infty}^{x} \frac{e^t}{t}\,dt$ sind wenigstens tabelliert, manche dagegen nicht einmal dies. Für solche Fälle läßt sich jederzeit ein Programm aufstellen:

Werden die Glieder einer Reihe hinreichend rasch kleiner, so kann die Reihe bei einem n-ten Glied abgebrochen werden. Das n-te Glied ist dabei kleiner als eine vorgegebene Schranke, z.B. 10^{-4}. Die Reihe wird vom ersten bis zum n-ten Glied aufsummiert. Ein Ablaufplan z.B. für e^x hat folgendes Aussehen:



Zur Prüfung auf genügende Kleinheit des n-ten Gliedes muß der Absolutbetrag überprüft werden. Andernfalls wird bei negativen Gliedern irrtümlich entschieden.

Beispiel:

Als Grenzwert sei 10^{-4} gewählt, das zu prüfende Glied betrage -10. Die Frage $-10 \leqslant 10^{-4}$? wird also mit "ja" beantwortet. Die Reihe wird abgebrochen, ohne daß dieses Glied vernachlässigbar klein wäre. Durch Prüfen des Absolutbetrages wird die Frage sinnrichtig beantwortet: $|-10| \leqslant 10^{-4}$? ergibt "nein".

Im Anhang sind die Programme e^x, cos x und Ei(x) aufgelistet. Bei Ei(x) ist die Konstante C der Grenzwert von $(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \cdots - \ln n)$ für n über alle Grenzen steigend. Diese sogenannte Eulersche Konstante beträgt

$$C = 0.\ 577\ 215\ 664\ 9\ \ldots$$

Die Vergleichsprüfung auf hinreichende Kleinheit kann allgemein bei Näherungsverfahren verwendet werden. So z.B. bei der Lösung einer Gleichung f(x) = 0 in Beispiel 5.6.

Beispiel 5.6: Newtonsche Näherungslösung für f(x) = 0

Es sei f(x) eine gegebene beliebig komplizierte Funktion. Gesucht sind die Lösungen f(x) = 0, d.h. diejenigen x-Werte, für die f(x) verschwindet (Nullstellen). Ausgegangen wird von einer Wertetabelle. Haben zwei Funktionswerte $f(x_1)$ und $f(x_2)$ ein unterschiedliches Vorzeichen, so liegt bei stetigen eindeutigen Funktionen eine Nullstelle dazwischen. Unstetigkeiten wie Verschwinden des Nenners oder mehrdeutige Funktionswerte z.B. bei der Kreisgleichung erfüllen diesen Tatbestand nicht. Sie sind daher ausgeschlossen.

Nullstellen können bequem mit dem Newtonschen Näherungsverfahren ermittelt werden. Ist x_0 ein Wert in der Nähe der Nullstelle ohne Maxima oder Minima zwischen x_0 und dieser, so ist die Steigung an der Stelle x_0

$$f'(x_o) = \frac{f(x_o)}{x_o - x_1} .$$

Der Wert x_1, der näher an der Nullstelle liegt, ist damit

$$x_1 = x_o - \frac{f(x_o)}{f'(x_o)} .$$

Ist x_1 bereits ein brauchbarer Näherungswert, also $f(x_1)$ hinreichend klein, so ist die Rechnung bereits beendet. Andernfalls wird sie fortgesetzt, wobei nun ein

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

bestimmt wird. Die Rechnung wird wiederholt, bis beim gesuchten Näherungswert x_n der Funktionswert $f(x_n)$ hinreichend klein ist. Die Steigung $f'(x)$ wird gebildet als

$$f'(x) = \frac{f(x + h) - f(x - h)}{2\ h}$$

mit h als kleiner Größe, z.B. $h = 10^{-4}$ (vgl. Beispiel 3.3: Ableitungen einer Funktion).

Das Newtonsche Näherungsverfahren führt mit den angegebenen Einschränkungen bereits nach einigen Schritten zu einem brauchbaren Näherungswert. Von Nachteil ist manchmal, daß außer Extremwerten (also Maxima und Minima) auch Wendepunkte mit der Steigung Null zum Programmabbruch führen. Als Ausweg läßt sich meist ein anderer Ausgangspunkt x_o verwenden.

Ein anderes Verfahren, das auch Extremwerte und Wendepunkte zuläßt, ist das Sekantenverfahren. Dieses erfordert nicht einmal eine erste Ableitung $f'(x)$ im untersuchten Bereich: Bei diesem Verfahren werden zwei Funktionswerte mit unterschiedlichem Vorzeichen durch eine Gerade verbunden. Der Schnitt mit der x-Achse ist der erste Näherungswert x_1. Vom Funktionswert $f(x_1)$ wird eine weitere Gerade gezogen zum Funktionswert mit anderem Vorzeichen und so ein x_2 gewonnen. Ist schließlich nach n Schritten $f(x_n)$ hinreichend klein, so ist das Verfahren abgeschlossen. Da bei Maxima und Minima im untersuchten Bereich sich das Verfahren nur langsam einem Endwert

nähert, sind die erforderlichen Rechenzeiten recht lang. Im Anhang sind beide Verfahren wiedergegeben.

Die Funktion f(x) wird unter Label E eingegeben mit x als direkter Enter-Eingabe oder als STO 0 und anschließendem RCL 0.

Beispiel:

f(x) = x - 2. Gesucht ist die Nullstelle im Bereich x = -5 bis x = 5.

In Stellung RUN wird gedrückt GTO E, dann in Stellung W/PRGM:

```
ENT
 2
 -
```

Damit ist die Funktion f(x) = x - 2 eingegeben. In Stellung RUN können nun beliebige x-Werte auf Label E eingegeben und eine Wertetabelle erstellt werden:

x	-5	5	10	...
f(x)	-7	3	8	...

Die Nullstelle wird errechnet beim

Newton-Verfahren	Sekanten-Verfahren
5	5
CHS	CHS
A	B
5	5
R/S	R/S

Zur Eingabe einer anderen Funktion kann die Programmkarte erneut eingegeben oder die bisherige Funktion mit h DEL gelöscht werden. Dazu wird in Stellung RUN E gedrückt. Der Rechner durchläuft das Unterprogramm unter Label E einschließlich h RTN, der angezeigte Schritt ist R/S. Durch zweimaliges h BST wird der letzte Schritt der Funktion f(x) erreicht. Im Beispiel wird diese mit dreimaligem h DEL gelöscht.

Beispiel:

$f(x) = x^3 - 5x^2 - 12x + 5$. Die Gleichung wird umgeformt zu
$f(x) = (x^2 - 5x - 12)x + 5$.

Im Stellung W/PRGM wird unter Label E eingegeben

STO 0	x	2
g x^2		-
RCL 0		RCL 0
5		x
x		5
-		+
1		

Anschließend wird in Stellung RUN eine Wertetabelle erstellt. Die Wertetabelle zeigt drei Nullstellen in den Bereichen $x = -3$ bis -2, 0 bis 1 sowie 6 bis 7. Eine Eingabe entweder über die Tasten A, R/S oder über B, R/S ergibt

$$x_1 = -2.0487 \quad x_2 = 0.3652 \quad x_3 = 6.6835.$$

Beispiel:

$f(x) = 5 \ln x - x$

Nach Löschen der vorigen Funktion wird in Stellung W/PRGM unter Label E eingegeben

STO 0	x
f LN	
5	
x	
RCL 0	
-	

Nach Wertetabelle befinden sich zwei Nullstellen zwischen $x = 1$ und 2 bzw. $x = 12$ und 13. Die Nullstellen werden berechnet zu $x_1 = 1.2959$ und $x_2 = 12.7132$.

5.2.2 Vergleich zweier Zwischenergebnisse

Die bisher gezeigten Vergleichsprüfungen prüften Zahlen gegenüber einem vorgegebenem Wert. Dieser betrug z.B. $\pm$ 10 im Beispiel 5.4, 10^{-4} in den übrigen Fällen. Ebenso kann eine Vergleichsprüfung auch die Prüfung zweier aufeinander folgender Zwischenergebnisse umfassen. Dies wird im folgenden Beispiel gezeigt.

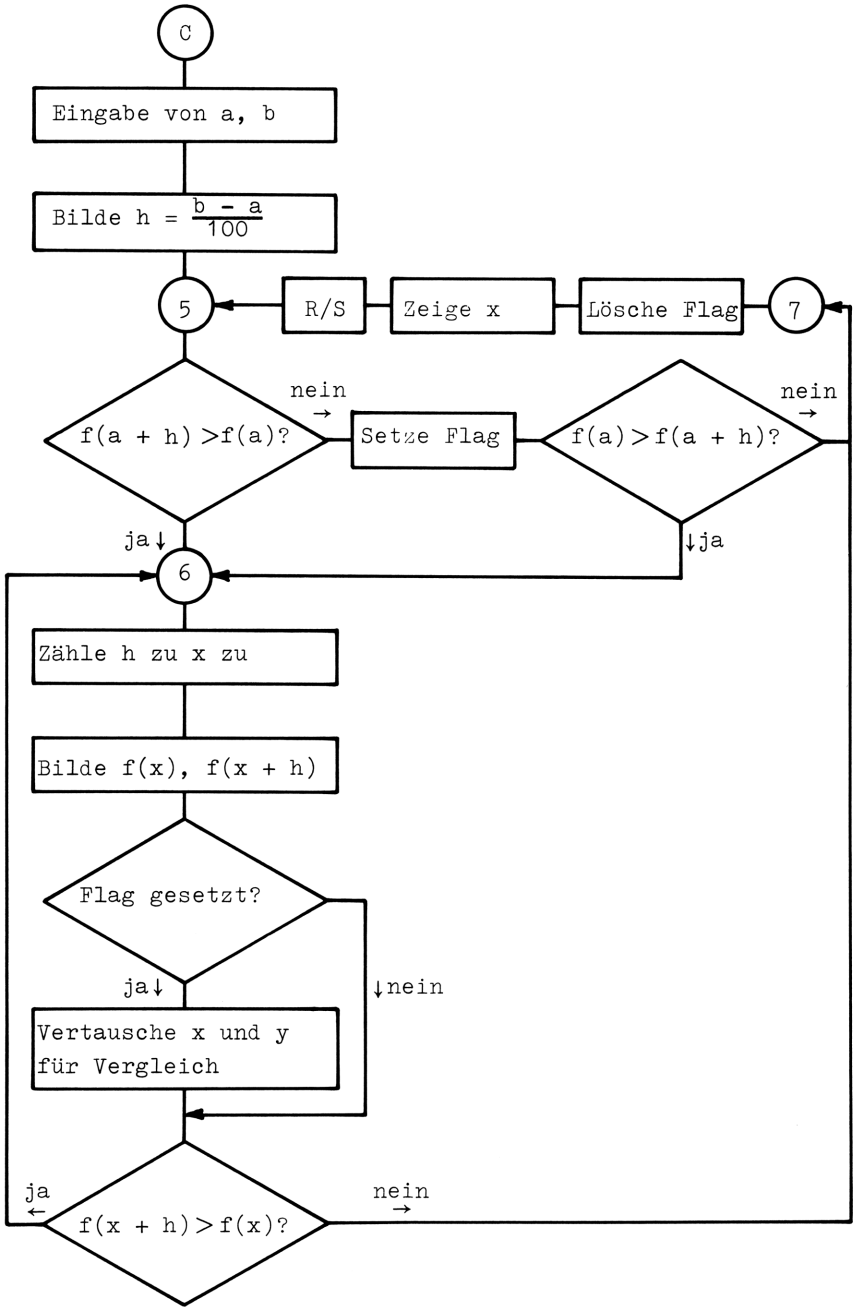
Beispiel 5.7: Extremwertefinder

Extremwerte (Maxima und Minima) einer Funktion haben höhere bzw. niedrigere Funktionswerte als ihre Umgebung. Beim Prüfen einer solchen Funktion wird also bei schrittweisem Erhöhen von x um h die Antwort auf f(x + h) > f(x)? vor Erreichen eines Maximums "ja" lauten, nachher "nein". Entsprechend ist vor Erreichen eines Minimums für f(x) > f(x + h)? die Antwort "ja", nachher "nein". Das dargestellte Programm bildet bei einem zu prüfenden Abschnitt von a bis b eine Schrittweite $h = \frac{b - a}{100}$. Es prüft die Funktion schrittweise, bei x = a beginnend, auf Maxima und Minima. Nach dem Auffinden eines Extremwertes hält das Programm an und zeigt den letzten x-Wert mit erfüllter Bedingung. Mit R/S startet die Programmausführung zum nächsten Extremwert. Enthält die Funktion im untersuchten Bereich keine Maxima oder Minima, so läuft die Programmausführung weiter (Stop durch R/S). Der Ablaufplan ist nebenstehend wiedergegeben

Wie in Beispiel 5.6 ist die Funktion f(x) unter Label E untergebracht. Gestartet wird das Programm durch Labeltaste C. Eine Schrittliste befindet sich im Anhang.

Beispiel:

$f(x) = x^3 - 5x^2 - 12x + 5$. Dies ist die gleiche Funktion wie in Beispiel 5.6. Nach der Wertetabelle liegen die Extremwerte zwischen x = -2 und 0 sowie zwischen x = 3 und 5.


C
Eingabe von a, b
Bilde h = (b - a)/100
5
R/S
Zeige x
Lösche Flag
7
f(a + h) > f(a)?
nein
Setze Flag
f(a) > f(a + h)?
nein
ja
ja
6
Zähle h zu x zu
Bilde f(x), f(x + h)
Flag gesetzt?
ja
nein
Vertausche x und y
für Vergleich
ja
f(x + h) > f(x)?
nein

In RUN wird eingegeben	2	sowie anschließend	3
	CHS		C
	C		5
	0		R/S
	R/S		

Für die Extremwerte ergibt sich $x_{max} = -0.94$, $x_{min} = 4.28$. Diese Werte sind innerhalb der Streifenbreite genau, also hier wegen h = 0.02 auf $\pm$ 0.02 genau. Eine erneute Eingabe mit den Grenzen x = -0.96 und -0.92 bzw. 4.26 und 4.30 verbessert die Angabe zu $x_{max} = -0.9368 \pm 0.0004$ und $x_{min} = 4.2700 \pm 0.0004$. Allerdings kann der Prüfbereich wegen der annähernd konstanten Funktionswerte in der Nähe eines Maximums oder Minimums nur begrenzt verringert werden.

Beispiel:

$f(x) = 5 \ln x - x$. Nach der Wertetabelle befindet sich das Maximum zwischen x = 4 und 6. Die Eingabe führt auf 5.00 $\pm$ 0.02. Eine erneute Eingabe mit 4.98 und 5.02 ergibt $x_{max} = 5.0000 \pm 0.0004$. Die direkte Ableitung $f'(x) = \frac{5}{x} - 1$ führt mit der Bedingung für einen Extremwert f'(x) = 0 auf genau $x_{max} = 5$.

Beispiel:

$f(x) = e^{-x} \cdot \sin x$. In Stellung W/PRGM wird unter Label E eingegeben

f	SIN
h	LST X
	CHS
g	e^x
	x

In Stellung RUN wird gedrückt h RAD, dann 0

C

h π

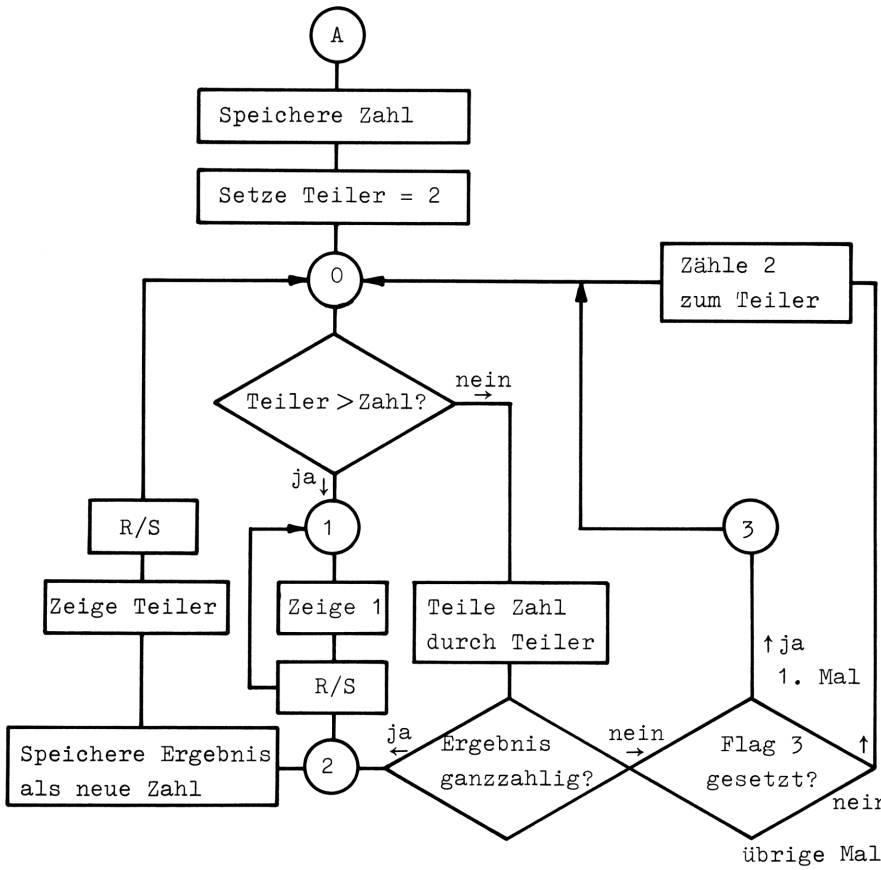
R/S

Nach jedem Ergebnis wird R/S erneut gedrückt. Es ergibt sich

$$x_{max} = \frac{\pi}{4}$$
$$x_{min} = \frac{5\pi}{4}$$
$$x_{max} = \frac{9\pi}{4}$$
$$\vdots$$

Beispiel 5.8: Primfaktorenzerleger

Die auf Labeltaste **A** eingegebene Zahl ist in kleinste, nicht mehr teilbare Faktoren zu zerlegen, also 15 in 3 x 5 oder 221 in 13 x 17 und so fort. Zur Lösung wird die eingegebene Zahl probehalber durch 2 geteilt. Bei einem ganzzahligen Ergebnis wird nochmals geteilt. Läßt sich das Ergebnis nicht mehr ohne Rest durch 2 teilen, so wird wiederholt durch 3 geteilt, dann durch 5, durch 7 und die weiteren ungeraden Zahlen. Das Ablaufschema sieht also folgendermaßen aus:



Nachstehend ist die Programmschrittliste wiedergegeben.

Format: DSP 0

```
1         f LBL A
           STO 0        Zahl
             2
           STO 1        Teiler
-------------------------------
5         f LBL 0
           RCL 0
           RCL 1
          g x>y
   ja ↓    GTO 1       nein
10           ÷
          g FRAC
          f x = 0
   ja ↓    GTO 2       nein
          h F ? 3
15 ja ↓    GTO 3       nein
             2
          STO + 1
           GTO 0
-------------------------------
          f LBL 1
20           1
            R/S
           GTO 1
-------------------------------
          f LBL 2
          h LST X
25         STO 0        Ergebnis
           RCL 1
            R/S
           GTO 0
-------------------------------
          f LBL 3
30           3
           STO 1
           GTO 0
```

Die zu teilende Zahl wird auf Label A eingegeben. Nach der Berechnung jedes Primfaktors wird R/S gedrückt, bis schließlich 1 erscheint. Es werden z.B. die Zahlen 8, 82, 143, 1081 eingegeben.

Zur Prüfung auf Ganzzahligkeit ist die HP 67/97-Funktion g FRAC verwendet worden. Die Funktion g FRAC zeigt die Stellen einer Zahl nach dem Komma. Sind diese gleich Null (geprüft durch f x = 0), so ist die Zahl ganzzahlig. Rechner, welche die Funktion INT/FRAC nicht besitzen, können dennoch Zahlen auf Ganzzahligkeit prüfen. Dazu wird die endliche Stellenzahl benutzt. Wie in Kapitel 1.4.2 beschrieben, führt die endliche Stellenzahl von Digitalrechnern zu ungewollten Rundungen bei Zahlenverknüpfungen. So wird bei 10-stelligen Rechnern der Ausdruck $2.18 + 10^9 - 10^9$ als 2.0 wiedergegeben. $7.82 + 10^9 - 10^9$ erscheint als 8.0. Durch Vergleich der ursprünglichen Zahl mit der gerundeten kann also auf Stellen nach dem Komma geschlossen werden. Die Prüfung auf Ganzzahligkeit im Beispiel 5.8 erfolgt beim Rechnen

mit g FRAC

```
           ÷
       g FRAC
       f x = 0 ─┐
  ja↓  GTO 2    │ nein
         .   ◄──┘
         .
         .
───────────────────
       f LBL 2
       h LST X
         .
         .
         .
```

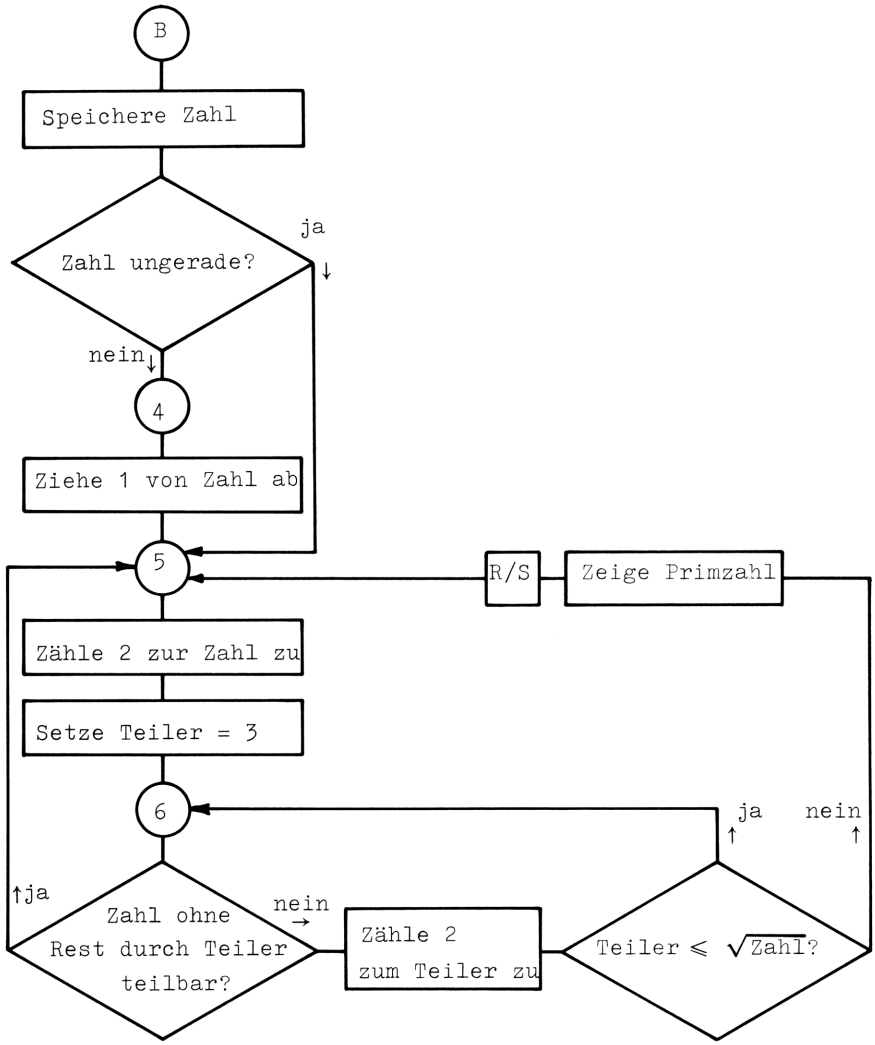
ohne g FRAC

```
           ÷
       STO 8        ursprüngliche
        EEX         Zahl
         9
         +
        EEX
         9
         -          gerundete Zahl
       RCL 8        ursprüngliche
      g x = y ─┐    Zahl
  ja↓  GTO 2   │ nein
         .  ◄──┘
         .
───────────────────
       f LBL 2
       RCL 8
         .
         .
```

Im Beispiel muß außerdem in Label 2 der Aufruf der ursprünglichen Zahl mit h LST X durch RCL 8 ersetzt werden, da sonst 10^9 aufgerufen würde. Das abgeänderte Programm kann nun an Stelle des ursprünglichen verwendet werden.

Beispiel 5.9: Primzahlenfinder

Mit diesem Programm läßt sich zu einer vorgegebenen Zahl die jeweils nächste Primzahl finden. Ähnlich wie im Beispiel 5.8 wird die eingegebene Zahl auf gerade/ungerade geprüft und je nach Ergebnis eine Eins oder eine Zwei hinzugezählt. Die neue ungerade Zahl wird der Reihe nach durch alle ungeraden Teiler 3, 5, 7, ... $\leqslant \sqrt{\text{Zahl}}$ zu teilen versucht. Gelingt dies ohne Rest nicht, so wird die ungerade Zahl als neue Primzahl angezeigt. Gelingt dies ohne Rest, so wird zur ungeraden Zahl 2 hinzugefügt und wiederum durch die ungeraden Teiler geteilt:



Format: DSP 0

```
1          f LBL B
            STO 0
              2
              ÷
5          g FRAC
           f x ≠ 0 ┐
    ja ↓    GTO 5  ┘ nein
           f LBL 4 ◄┘
              1
10         STO - 0
           f LBL 5
              2
           STO + 0
              3
15          STO 1
```

```
16         f LBL 6
            RCL 0
            RCL 1
              ÷
20         g FRAC
           f x = 0 ┐
        ja ↓ GTO 5 ┘ nein
              2   ◄┘
           STO + 1
25          RCL 0
           f √x
            RCL 1
           g x≤y   ┐
        ja ↓ GTO 6 ┘ nein
30          RCL 0 ◄┘ neue Prim-
            R/S      zahl
           GTO 5
```

Die Ausgangszahl wird auf Label B eingegeben. Nach jedem Programmstop wird R/S gedrückt.

Es erscheinen bei 12 : 13, 17, 19, 23, ...
bei 178 : 179, 181, 191, ...
bei 10^3: 1009, 1013, 1019, ...
bei 10^5: 100003, 100019, ...

Mit dem Programm werden Primzahlen ab einschließlich 5 angezeigt. Die Primzahl 3 wird wegen der Programmschritte 11 bis 19 (Teiler = 3, Zahl durch Teiler teilbar?) nicht berücksichtigt. Auffallend ist bei größeren Zahlen die erhebliche Rechenzeit, z.B. bei 10^5 bereits 2 bis 3 min je Primzahl.

Beispiel 5.10: Zahlenumwandlung

Das Programm wandelt Zahlen eines Basissystems in die eines anderen um. Als Basissysteme sind das Dual-, das Oktal- und das Dezimalsystem angegeben. Andere Systeme können zusätzlich eingeführt werden. Die Zahlensysteme unterscheiden sich durch ihre jeweilige Basis.

Im Dualsystem haben die Zahlen die Basis 2. Die Zahlen werden durch eine Folge von 0 und 1 dargestellt. Stellenwechsel ist bei 2, 4, 8, 16, .. Im Oktalsystem ist die Basis 8, ein Wechsel der Stelle findet bei 8, 16, 24, 32, ... statt. Die Tabelle zeigt das unterschiedliche Aussehen von Zahlen verschiedener Basis:

Dezimalsystem (Basis 10)	Oktalsystem (Basis 8)	Dualsystem (Basis 2)
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
⋮	⋮	⋮
7	7	111
8	10	1000
9	11	1001
⋮	⋮	⋮
15	17	1111
16	20	10000
17	21	10001
⋮	⋮	⋮
23	27	10111
24	30	11000
25	31	11001
⋮	⋮	⋮

Aus der Aufstellung wird auch das Bildungsgesetz erkennbar, nach dem Zahlen umgewandelt werden. Zum Umwandeln einer Dezimalzahl in eine Zahl mit anderer Basis wird die Dezimalzahl durch deren Basis geteilt. Der nicht teilbare Rest bildet die Einerstelle der gesuchten Zahl. Der ganzzahlige Teil des Ergebnisses wird erneut durch die Basis geteilt, der nicht teilbare Rest als nächsthöhere Stelle (Zehnerstelle) abgespeichert. Nach wiederholten Teilungen ist der ganzzahlige Teil kleiner als die Basis. Er bildet die höchste Stelle der gewonnenen Zahl.

Es soll z.B. die Dezimalzahl 811 in die entsprechende Oktalzahl verwandelt werden:

Dezimalzahl	Aufteilung der Zahl	
811	= 101 x 8	+ 3
101	= 12 x 8	+ 5
12	= 1 x 8	+ 4
1	= 0 x 8	+ 1

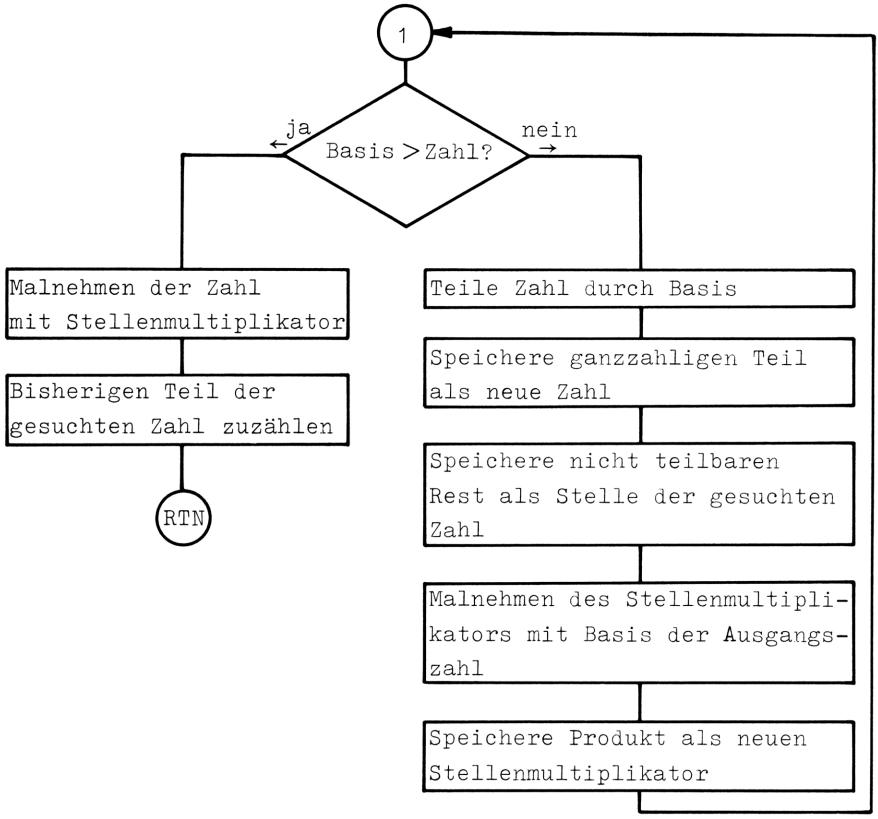
Die gesuchte Oktalzahl ist also 1453.

Es soll die Dezimalzahl 23 in die entsprechende Dualzahl umgewandelt werden:

Dezimalzahl	Aufteilung der Zahl	
23	= 11 x 2	+ 1
11	= 5 x 2	+ 1
5	= 2 x 2	+ 1
2	= 1 x 2	+ 0
1	= 0 x 2	+ 1

Die gesuchte Dualzahl ist damit 10111.

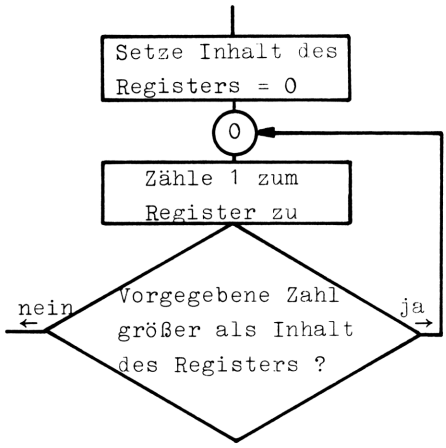
Das vollständige Programm wandelt Zahlen der Systeme Dezimal ⇄ Oktal, Dezimal ⇄ Dual sowie Oktal ⇄ Dual ineinander um. Als Ablaufschema ist zur besseren Übersicht lediglich die Wiederholschleife dargestellt.



Der im Ablaufschema verwendete Stellenmultiplikator ermöglicht die Bildung der gesuchten Zahl als Summe mehrerer Teilzahlen. Die vollständige Schrittliste für das Programm ist im Anhang aufgeführt. Die Umwandlung von Dezimalzahlen in weitere Zahlensysteme ist nach dem gezeigten Muster ebenfalls möglich. Eine Umwandlung z.B. Oktal ⇄ Dual ist über das Dezimalsystem besonders einfach. Die Zahlen werden umgewandelt gemäß Oktal ⇄ Dezimal ⇄ Dual (vgl. die Label E und f e).

5.3 Schleifen mit fester Zahl von Durchläufen (DSZ und ISZ)

In Kapitel 5.2 wurden Schleifen mit Vergleichsprüfungen besprochen, welche die durchlaufende Zahl betreffen. In ähnlicher Weise kann geprüft werden, zum wievielten Mal eine Schleife durchschritten wird. Bei Erreichen einer vorgegebenen Zahl von Durchläufen wird dann zu einem anderen Programmteil verzweigt. Allgemein kann dies durch folgende Schritte geschehen:




```
      0
    STO 1
f LBL 0
      .
      .
      .
      1
STO + 1   bisherige
 RCL 1    Durchläufe
 RCL 2    Zahl
   g x>y
ja↓ GTO 0  nein
      .
      .
      .
```

Die Schrittfolge wird besonders übersichtlich durch die Tastenfunktionen f DSZ und f ISZ. f DSZ (decrement and skip if zero) bedeutet dabei verringere und überspringe falls Null, f ISZ (increment and skip if zero) wachse an und überspringe falls Null.

Der HP 65 verfügt nur über die Funktion g DSZ mit Speicherung im Register 8. Der HP 67/97 besitzt beide Funktionen. Als Speicher für die Vergleichszahl dient beim HP 67/97 für DSZ und ISZ das I-Register. Für h DSZ wird eine ganze Zahl grösser als Null mit h STI im I-Register gespeichert. Beim Schritt f DSZ wird davon Eins abgezogen und anschließend der Inhalt des I-Registers auf ungleich Null geprüft. Ist der Inhalt ungleich Null, so wird zur nächsten Programmzeile vorgerückt. Im allgemeinen enthält diese den Befehl GTO eigener Label, so daß dieser Programmteil erneut durchlaufen wird (Schleife). Nach n Durchläufen, wobei n die anfangs im I-Register gespeicherte Zahl ist, wird durch das Abziehen der Inhalt des I-Registers Null. Die nächste Programmzeile (mit dem GTO-Befehl) wird übersprungen, die Schleife braucht nicht mehr durchlaufen werden. Für h ISZ wird entsprechend eine negative ganze Zahl (also kleiner als Null) im I-Register gespeichert. Beim Schritt f ISZ wird entsprechend jeweils Eins hinzugezählt. Ist der Inhalt des I-Registers zu Null geworden, so wird die nächste Programmzeile übersprungen. Dies wird in den beiden folgenden Programmen gezeigt.

Beispiel:

Die Programme geben für die jeweils eingegebene ganze positive Zahl die Anzahl der durchlaufenen Schleifen an. Die Schleifenzahl ist hier gleich der eingegebenen Zahl.

Programm mit DSZ

```
1        f LBL A
         f CL REG
          h STI      n
         f LBL 4
5          1
         STO + 4
          f DSZ
 ja, I≠0↓ GTO 4      I = 0
         RCL 4
10       h RTN
```

Programm mit ISZ

```
11       f LBL B
         f CL REG
           CHS
          h STI      -n
15       f LBL 5
           1
         STO + 5
          f ISZ
ja, I ≠ 0 ↓ GTO 5    I = 0
 20       RCL 5
           h RTN
```

Damit auch im ISZ-Programm positive Zahlen eingegeben werden können, wird als Schritt 13 CHS verwendet.

Eine Anwendung von f DSZ bei der Summenbildung geben die folgenden Programme.

Beispiel 5.11: $\sum n$, $\sum \frac{1}{n}$, $\sum \frac{1}{n^2}$, $n!$

Die Summenbildungen (Zeichen $\sum$) sowie die Fakultät (Zeichen !) bedeuten die Abkürzung der folgenden Ausdrücke:

$$\sum n = 1 + 2 + 2 + \cdots + n$$

$$\sum \frac{1}{n} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$$

$$\sum \frac{1}{n^2} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}$$

$$n! = 1 \times 2 \times 3 \times \cdots \times n$$

	$\sum n$		$\sum \frac{1}{n}$		$\sum \frac{1}{n^2}$		$n!$
1	f LBL A		f LBL B		f LBL C		f LBL D
	h STI n		h STI	25	h STI		h STI
	0		0		0		1
	STO 0	15	STO 0		STO 0	40	STO 0
5	f LBL 0		f LBL 1		f LBL 2		f LBL 3
	h RCI		h RCI		h RCI		h RCI
	STO + 0		h $\frac{1}{x}$	30	h $\frac{1}{x}$		STO x 0
	f DSZ		STO + 0		g x^2		f DSZ
	GTO 0	20	f DSZ		STO + 0	45	GTO 3
10	RCL 0		GTO 1		f DSZ		RCL 0
	h RTN		RCL 0		GTO 2		h RTN
			h RTN	35	RCL 0		
					h RTN		

Im Programm n-Fakultät wird die Speicherarithmetik STO x 0 verwendet. Daher muß in Register 0 zu Beginn eine Eins eingeschrieben werden.

Beispiel 5.12: Numerische Integration nach Simpson

Das bestimmte Integral $\int_{x=a}^{x=b} f(x)\,dx$ berechnet die Fläche zwischen der Kurve f(x) und der x-Achse im Abschnitt a bis b der x-Achse. Entsprechend den Regeln der Integrationsrechnung gelingt es dabei oft, eine Funktion zu finden, die das Integral von f(x) darstellt. Allgemein kann die Fläche auch bestimmt werden, indem man sie in schmale senkrechte Streifen unterteilt mit der jeweiligen Höhe f(x) und der Breite h. Die Gesamtfläche ist dann $F = h\,[\,f(a) + f(a+h) + \cdots + f(b)\,]$. Die Streifenoberkante wird bei dieser einfachen Integration als waagerecht angenommen. Bei der Integration nach der Simpsonschen Regel wird nun auch die Form der Streifenoberkante berücksichtigt. Durch je zwei an einander grenzende Streifen wird eine Parabel gelegt. Bei gleicher Streifenbreite ist dadurch die Integration merklich genauer. Die Fläche des ersten Doppelstreifens ist nach dem Bestimmen der Parabelgrößen

$$F_1 = \frac{h}{3}\,[\,f(a) + 4\,f(a+h) + f(a+2\,h)\,]\,,$$

die des nächsten entsprechend

$$F_2 = \frac{h}{3} [f(a + 2h) + 4 f(a + 3h) + f(a + 4h)]$$

und so fort. Die Gesamtfläche ist

$$F = F_1 + F_2 + F_3 + \cdots + F_n.$$

Damit ist

$$F = \frac{h}{3} \{ f(a) + f(b) + 4 [f(a + h) + f(a + 3h) + \cdots + f(a + (n - 1)h)] + 2 [f(a + 2h) + f(a + 4h) + \cdots + f(a + (n - 2)h)] \}.$$

Es sind also zwei Summationen durchzuführen mit

$$h = \frac{b - a}{n}, \quad n \text{ geradzahlig.}$$

Die erste Summe enthält dabei $\frac{n}{2}$ Glieder, die zweite $(\frac{n}{2} - 1)$ Glieder. Wird in der ersten Summe f(a + h) getrennt berechnet, so kann für beide Summen die selbe Programmschleife verwendet werden. Die Programmliste ist im Anhang aufgeführt. Wie bisher ist f(x) unter Label E einzufügen. Die Eingabe der Grenzen a und b erfolgt mit A und R/S.

Beispiel:

Es soll das bestimmte Integral von $f(x) = \sin^2 x$ im Abschnitt x = 0 bis $x = \pi$ berechnet werden, also

$$\int_{x=0}^{x=\pi} \sin^2 x \, dx.$$

In Stellung RUN wird gedrückt h RAD, dann GTO E. Umschalten auf W/PRGM, dann f SIN, g x^2. Es wird auf RUN zurückgeschaltet und eingegeben

```
0
A
h π
R/S
```

Erwartungsgemäß ergibt sich als Wert des Integrals 1.5708. Die Abweichung zum genauen Wert $\frac{\pi}{2}$ beträgt 10^{-9}.

Beispiel:

Es soll das Elliptische Integral

$$F(k, \varphi) = \int_{\varphi_1 = 0}^{\varphi_2 = \varphi} \frac{d\varphi}{\sqrt{1 - k^2 \cdot \sin^2 \varphi}}, \quad k = \sin \alpha$$

für verschiedene Werte von φ und α angegeben werden. Für eine Eingabe in Winkelgrad wird als Label 2 ein Unterprogramm eingeführt, α wird unter Label B eingegeben.

f LBL 2	Umwandlung	f LBL B	α	f LBL E	f(x)	f LBL A	φ_1
ENT	Winkelgrad	f GSB 2		f SIN		STO 1	
h π	in Radian	f SIN		g x^2		R/S	φ_2
x		g x^2		RCL 7	k^2	f GSB 2	
1		STO 7	k^2	x		STO 2	
8		h RTN		CHS		⋮	
0				1		⋮	
÷				+			
h RTN				f $\sqrt{x}$			
				h $\frac{1}{x}$			
				h RTN			

Das Zusatzprogramm unter Label B, das Umwandlungsprogramm unter Label 2, die Funktion f(x) unter Label E sowie die Programmänderung unter Label A werden eingefügt. Anschließend kann α unter Label B, $\varphi_1 = 0$ unter Label A sowie $\varphi_2 = \varphi$ als nachfolgendes R/S gedrückt werden. Für das Elliptische Integral $F(k, \varphi)$ ergibt sich:

α	φ	$F(k, \varphi)$
30°	30°	0.5294
45°	60°	1.1424
60°	90°	2.1565
⋮	⋮	⋮

Beispiel 5.13: Numerische Lösung der Differentialgleichung $y' = f(x, y)$

Die erste Ableitung y' sei eine beliebig komplizierte Funktion von x und y, gegeben sei ferner für den Ausgangswert x_0 der zugehörige Funktionswert y_0. Für weitere Werte von x werden dann die Funktionswerte y berechnet, indem man schrittweise x um h erhöht und das jeweilige y bestimmt. Nach Runge und Kutta läßt sich eine besonders genaue Näherung erreichen, wenn ähnlich wie bei der numerischen Integration nach Simpson eine Parabel durch jeden Abschnitt der Breite h gelegt wird. Es ist

$$y_1 = y_0 + \frac{1}{6}(k_1 + 2\,k_2 + 2\,k_3 + k_4)\ ,$$

$$k_1 = h \cdot f(x_0,\ y_0)$$

$$k_2 = h \cdot f(x_0 + \frac{h}{2},\ y_0 + \frac{k_1}{2})$$

$$k_3 = h \cdot f(x_0 + \frac{h}{2},\ y_0 + \frac{k_2}{2})$$

$$k_4 = h \cdot f(x_0 + h,\ y_0 + k_3).$$

Nach Berechnung von y_1 wird $(x_0 + h)$ um das nächste h erhöht, y_2 bestimmt und so fort, bis schließlich das gesuchte y_n an der Stelle x_n berechnet ist. Zweckmäßig wird die Zahl der Unterteilungen n ins I-Register gespeichert, so daß für die n Durchläufe die Funktion f DSZ verwendet werden kann.

Eine Programmschrittliste ist im Anhang angegeben. Die Funktion f(x, y) wird unter Label E eingefügt mit x als RCL 1 und y als RCL 2. x_0, y_0 und x_1 werden über Label A, gefolgt von zweimaligem R/S, eingegeben. Als Ergebnis wird y_1 sowie nach Drücken von R/S y_1' berechnet.

Beispiel:

Für $y' = 2\,x \cdot \sqrt{y}$ sei $x_0 = 1$, $y_0 = \frac{1}{4}$. Für $x_1 = 10$ wird y_1 gesucht. Zur Eingabe von f(x) wird in Stellung RUN gedrückt GTO E, dann auf W/PRGM umgeschaltet. Es wird eingegeben

```
RCL 2        y
f √x
RCL 1        x
 x
 2
 x
```

Die Werte von x_0, y_0 und x_1 werden in Stellung RUN eingegeben als

```
1            x0
A
.
2
5            y0
R/S
1
0            x1
R/S
```

Nach etwa 5 min wird angezeigt $y_1 = 2499.9581$

sowie nach R/S $y_1' = 999.9916.$

Die geschlossene Lösung mit $y = \frac{x^4}{4}$ bzw. $y' = x^3$ ergibt $y_1 = 2500$ und $y_1' = 1000.$

Bei gegebenem x_0 und x_1 ist das Ergebnis um so genauer, je feiner der Abschnitt $(x_1 - x_0)$ unterteilt wird, also je kleiner die Schrittweite h ist. Entsprechend steigt auch die benötigte Rechenzeit. Im Beispiel ist:

Zahl der Unterteilungen	Rechenzeit	Ergebnis	Relativer Fehler $(= \frac{\text{Ergebnis-genauer Wert}}{\text{genauer Wert}})$
2	9 s	2197	$-1.2 \cdot 10^{-1}$
20	90 s	2499.5	$-2.0 \cdot 10^{-4}$
40	3 min	2499.9581	$-1.7 \cdot 10^{-5}$
100	$7\frac{1}{2}$ min	2499.9987	$-5.2 \cdot 10^{-7}$
⋮	⋮	⋮	⋮

Die Zahl der Unterteilungen wird geändert durch Ändern der Programmschritte 9 und 10 (Unterteilungszahl n = 40). Wird eine Rechnung mit der Unterteilungszahl n und eine weitere mit 2 n durchgeführt, so ist der Restfehler der genaueren Rechnung etwa $\frac{1}{15}$ des Unterschiedes der beiden Ergebnisse.

Für eine Tabelle mit einer Anzahl von x-Werten wird zweckmäßig mit möglichst feiner Unterteilung der erste y-Wert berechnet und gespeichert. Dieser Wert wird gemeinsam mit dem zugehörigen x-Wert als neuer Ausgangspunkt x_0, y_0 verwendet. Dadurch kann mit gleichbleibend feiner Unterteilung gerechnet und ein entsprechend genaues Ergebnis erzielt werden.

Beispiel:

$y' \, x^3 + y^2 \, (2 - x) = 0$. Der Ausgangswert sei $x_0 = 1.1$, $y_0 = 12.1$. Gesucht ist y_1 und y_1' bei $x_1 = 5$. Die Gleichung wird umgeformt zu

$$y' = \frac{y^2}{x^3} \, (x - 2).$$

In Stellung RUN wird gedrückt GTO E, dann in W/PRGM

RCL 2	y
RCL 1	x
÷	
g x^2	
RCL 1	
2	
−	
x	
RCL 1	
÷	

Mit der Unterteilung in n = 40 Abschnitte wird berechnet
$y_1 = 6.1971$, $y_1' = 0.9217$.
Bei einer Unterteilung in 100 Abschnitte folgt
$y_1 = 6.2496$, $y_1' = 0.9374$.
Eine direkte Integration führt über $y = \frac{x^2}{x - 1}$ und $y' = \frac{x(x - 2)}{(x - 1)^2}$

für $x_1 = 5$ auf
$y_1 = 6.2500$, $y_1' = 0.9375$.

Beispiel 5.14: Numerische Lösung der Differentialgleichung
$y'' = f(x, y, y')$

Die Differentialgleichung enthalte y", y', y und x. Läßt sich die Gleichung nach y" auflösen, so kann bei bekannten Anfangswerten x_0, y_0 und y_0' zu einem beliebigen x_1 das zugehörige y_1, y_1' und y_1'' angegeben werden. Das Runge-Kutta-Verfahren gibt bei Unterteilung in Abschnitte der Breite h

$$x_{n+1} = x_n + h$$
$$y_{n+1} = y_n + h \cdot y_n' + \frac{h}{6}(k_1 + k_2 + k_3)$$
$$y_{n+1}' = y_n' + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

Hierbei ist

$$k_1 = h \cdot f(x_n, y_n, y_n')$$
$$k_2 = h \cdot f\left[x_n + \frac{h}{2}, y_n + \frac{h}{2}\left(y_n' + \frac{k_1}{4}\right), y_n' + \frac{k_1}{2}\right]$$
$$k_3 = h \cdot f\left[x_n + \frac{h}{2}, y_n + \frac{h}{2}\left(y_n' + \frac{k_2}{4}\right), y_n' + \frac{k_2}{2}\right]$$
$$k_4 = h \cdot f\left[x_n + h, y_n + \frac{h}{2}(2y_n' + k_3), y_n' + k_3\right].$$

Die Programmschrittliste ist im Anhang wiedergegeben. Wie in Beispiel 5.13 wird die Funktion - hier f(x, y, y') - unter Label E eingeschrieben. Dabei ist zu setzen x als RCL 0, y als RCL 2 und y' als RCL 4. x_0 wird anschließend auf Label A eingegeben, y_0, y_0' und x_1 jeweils mit R/S. Als Ergebnis wird y_1 angezeigt, dann jeweils nach R/S y_1' und y_1''.

Beispiel:

$(y'' + y) \cdot x + 2y' = 0$. Gegeben sei $x_0 = \pi$, $y_0 = 0$ und $y_0' = -\frac{1}{\pi}$, gesucht sind y_1, y_1' und y_1'' für $x_1 = \frac{5\pi}{2}$. Die Gleichung wird umgestellt zu

$$y'' = -\frac{2y'}{x} - y.$$

In Stellung RUN wird gedrückt GTO E, dann in W/PRGM:

RCL 4	y'
2	
x	
RCL 0	x
÷	
CHS	
RCL 2	y
−	

In Stellung RUN wird eingegeben

h π	x
A	
0	y_0
R/S	
h π	
h $\frac{1}{x}$	
CHS	y'_0
R/S	
h π	
5	
x	
2	
÷	x_1
R/S	.

Es wird berechnet $y_1 = 0.1273$, dann nach R/S $y'_1 = -0.0162$ und $y''_1 = -0.1232$. Die exakten Werte folgen aus $y = \frac{\sin x}{x}$, die relative Abweichung beträgt ca. $2 \cdot 10^{-4}$. Wie in Beispiel 5.13 wird das Ergebnis genauer, wenn die Unterteilung verfeinert wird. Im Programm werden dazu die Schritte 12 und 13 ($n = 40$) durch eine entsprechend höhere Zahl (z.B. $n = 100$ oder 200) ersetzt. In gleichem Maße erhöht sich auch hier die benötigte Rechenzeit.

Beispiel:

$y'' = 2y(2x^2 - 1)$. Für $x_0 = 1$ ist $y_0 = \frac{1}{e}$, $y'_0 = -\frac{2}{e}$. Gesucht sind für $x_1 = 2$ die Werte von y_1, y'_1 und y''_1. Nach GTO E wird in W/PRGM eingetastet

```
RCL 0          x
 g x^2
  2
  x
  1
  -
RCL 2          y
  x
  2
  x
```

x_0, y_0, y_0' und x_1 werden eingegeben. Es folgt $y_1 = 0.0183$, $y_1' = -0.0733$ und $y_1'' = 0.2564$. Wegen $y = e^{-x^2}$ sind die exakten Werte $y_1 = e^{-4}$, $y_1' = -4 \cdot e^{-4}$ und $y_1'' = 14 \cdot e^{-4}$. Mit n = 40 beträgt die relative Abweichung ca. 10^{-4}.

6. Indirekte Adressierung

Der Einsatz des I-Registers zum Abspeichern von Daten sowie zum Zählen von Schleifen mit DSZ und ISZ wurde bereits beschrieben. Hier soll eine weitere wichtige Anwendung des I-Registers behandelt werden: die indirekte Adressierung. Durch sie können Speicher oder Label aufgerufen werden, deren Nummer im I-Register gespeichert ist. Die entsprechenden Befehle lauten STO (i) und RCL (i) bzw. GTO (i) und f GSB (i). Der Vorteil dieser indirekten Adressierung ist, daß das Programm von einer Programmstelle zu verschiedenen Labeln verzweigt oder Zahlenwerte in unterschiedlichen Registern gespeichert werden, entsprechend zu der gerade im I-Register gespeicherten Zahl. Wie bereits erwähnt, werden Zahlen im I-Register gespeichert oder daraus abgerufen durch h STI bzw. h RCI.

6.1 Speichern und Zurückrufen von Daten

Wie bereits in Kapitel 2.10.3 besprochen, sind die 26 Speicherregister des HP 67/97 durchnumeriert. Die Speicher tragen folgende Nummern:

Speichername	Nummer
Primärregister 0 bis 9	0 bis 9
Sekundärregister S 0 bis S 9	10 bis 19
A bis E	20 bis 24
I-Register	25

Wird durch h STI eine Zahl im I-Register gespeichert, so bewirkt die Anweisung STO (i), daß Daten im Register Nr. (i) gespeichert werden.

Beispiel:

Die Zahl 33 soll im Sekundärregister S 2 durch indirekte Adressierung gespeichert werden.

```
  1
  2
h STI
  3
  3
STO (i)
```

Beispiel:

Die Zahl 17 soll im Register C gespeichert werden.

```
  2
  2
h STI
  1
  7
STO (i)
```

Zur Kontrolle wird gedrückt

```
 CLX
RCL C
```

Mit indirekter Adressierung kann die Speicherarithmetik auf alle Speicher angewendet werden, auch auf die Speicher A bis E.

Beispiel:

Die Zahl 43 soll Register C zugezählt werden. Das I-Register enthalte weiterhin die Zahl 22.

```
  4
  3
STO + (i)
```

Zur Kontrolle wird gedrückt

```
 CLX
RCL C
```

Beispiel:

Der Inhalt des I-Registers soll verdoppelt werden.

```
  2
  5
h STI
  2
STO x (i)
```

Nach Drücken von h RCI erscheint 50. RCL (i) ist hier nicht zulässig, da der Inhalt des I-Registers keine der Zahlen 0 bis 25 enthält:

Auf

```
  CLX
RCL (i)
```

erfolgt die Anzeige "Error". Zum weiteren indirekten Adressieren muß erst eine Zahl zwischen 0 und 25 ins I-Register geschrieben werden.

Beispiel 6.1: Sortieren von Zahlen

In Zahlenfolgen muß öfters die Häufigkeit der auftretenden Zahlen festgestellt werden. Dazu müssen sie sortiert werden. Für die Zahlen 0 bis 24 wird dies durch folgendes Programm bewirkt:

```
f LBL E
 h STI
   1
STO + (i)
 h RTN
```

Zu Beginn einer Zahlenfolge wird f CL REG, f P ⇄ S, f CL REG gedrückt. Anschließend werden die Zahlen auf Label E eingegeben. Ersichtlich erhöht eine "3" den Inhalt von Register 3 um Eins, eine "17" den Inhalt von Register 17 um Eins und so fort. Nach Eingabe aller Zahlen werden durch h REG die Inhalte

der Register aufgerufen. Der erste Durchlauf ruft dabei die Primärregister 0 bis 9, dann A bis E und I. Nach Drücken von f P ⇄ S werden durch h REG die Sekundärregister S 0 bis S 9, gefolgt von A bis E und I aufgerufen. Die Belegung der Register gibt direkt die Zahl der Nennungen wieder. Z.B. bedeutet eine 13 im Register 7, dreizehnmal wurde die Sieben genannt. Als Anwendung können Würfel überprüft werden: bei einer größeren Zahl von Würfen sollte jede der sechs Augenzahlen annähernd gleich häufig auftreten, also z.B. bei 300 Würfen etwa $\frac{300}{6} = 50$ betragen. Ebenso kann ein Zufallszahlen-Generator (vgl. Beispiel 5.3) überprüft werden. Eine weitere Anwendung wäre die Aufstellung eines Histogramms. Dazu werden die zu gruppierenden Werte entsprechend ihrer Größe in Klassen eingeteilt (hier: bis zu 25 Klassen). Anschließend wird die Häufigkeit ihres Auftretens festgestellt und grafisch aufgetragen. Als Histogramme auftragen lassen sich sowohl Monatseinkommen einer Belegschaft wie auch Meßergebnisse einer Fertigungskontrolle.

Beispiel 6.2: Nachtgefecht

Dieses Spiel ist ähnlich Beispiel 4.10 U-Boot-Jagd. Auf dem Spielfeld von 9 x 9 Feldern sind 7 Gegner versteckt. Der Spieler verfügt über die 4 Geschütz-Batterien Label A bis D mit je 6 Schuß. Ein gezielter Schuß bedeutet Eingabe eines Feldes auf der Spielfläche. Z.B. bedeutet 4.6 das Feld Zeile 4 Spalte 6. Ist ein Gegner auf diesem Feld, so erfolgt die Anzeige "Treffer". Dadurch wird zugleich die Position des Gegners gelöscht. Ein weiterer Schuß auf dieses Feld zeigt "daneben". Wurde der Gegner nicht entdeckt, so ergibt ein Schuß auf eines der vier Nachbarfelder "Streifschuß" ohne Löschung. Beim Feld 4.6 sind dies die Felder 3.6, 4.5, 4.7 und 5.6. Gespielt wird auf kariertem Papier, in das die Spielfläche eingezeichnet ist. Wie in Beispiel 4.10 wird zu Spielbeginn eine beliebige Zahl (z.B. 555, 123 oder 368) auf Label E eingegeben. Dadurch werden 7 Positionen "zufällig" d.h. nicht durch den Spieler erkennbar auf dem Spielfeld besetzt. Es soll jetzt versucht werden, mit den 4 x 6 Schuß sämtliche versteckten Gegner ausfindig zu machen. Nach Spielende können nicht entdeckte Positionen mit h REG der Speicher 0 bis 6 überprüft werden. Für das nächste Spiel genügt es, E ohne weitere Eingabe zu drücken. Der Rechner setzt selbsttätig 7 neue Positionen ein. Da die 7 Koordinaten

unabhängig von einander sind, ist also gelegentlich ein Feld von zwei Gegnern gleichzeitig besetzt. Bei einem Treffer auf dieses Feld bleibt also ein Gegner übrig. Es bedeutet

Anzeige	Bedeutung
1111111111	Spielvorbereitung abgeschlossen
0	Kein Gegner auf diesem Feld oder einem der vier direkt angrenzenden Felder
333	"Streifschuß". Gegner sind auf einem oder mehreren der vier angrenzenden Felder
3333333333	"Treffer". Die Position dieses Gegners wird gelöscht.
11111	"Batterie leer". Nächste Geschütz-Batterie verwenden.

Die Programmschrittliste ist im Anhang wiedergegeben.

6.2 Aufruf von Unterprogrammen

Ist eine der Zahlen 0 bis 19 im I-Register gespeichert, so wird durch den Befehl f GSB (i) oder GTO (i) das Unterprogramm mit diesem Label aufgerufen bzw. zu diesem Label gesprungen. Wie die Speicher sind beim HP 67 und 97 auch die Label durchnumeriert. Es bedeutet dabei

Labelname	Nummer
0 bis 9	0 bis 9
A bis E	10 bis 14
f a bis f e	15 bis 19

Beispiel:

Unter Label 3 sei das Unterprogramm g x^2 untergebracht. Es soll durch f GSB (i) aufgerufen werden.

```
1       f LBL A
        f GSB (i)
        h RTN
        f LBL 3
5       g x²
        h RTN
```

In Stellung RUN wird eingegeben

```
  3
h STI
```

dann

```
  5          13          10
  A          A           A
```

Wird gedrückt f CL REG oder

```
  4
h STI
```

so erfolgt bei Eingabe auf A die Meldung "Error".

Auf die gleiche Art kann der Sprung zum Unterprogramm Nr. (i) erfolgen. So werden 19 Label von einer Haupttaste aus erreichbar.

Beispiel 6.3: Direktes Anwählen von Unterprogrammen

Jede der 19 Programm-Marken 0 bis 9, B bis E und f a bis f e kann durch Drücken der Labeltaste A angewählt werden. Dazu wird lediglich die Nummer des gewünschten Programms gedrückt. Infolge des Programmstops durch R/S nach jeder Labelmarke und dem GTO-Befehl am jeweiligen Programmende können nach dem Anwählen über R/S beliebig oft Werte in ein Programm eingegeben werden.

```
 1      f LBL A
         h STI
        GTO (i)
        f LBL 0
 5        R/S
         g x²
        GTO 0
        g LBL e
          R/S
10        f√x
        GTO f e
          ⋮
```

Es wird gedrückt

```
0
A
```

dann

```
 2
R/S
```

gefolgt von weiteren R/S.

Anschließend wird gedrückt

```
1
9
A
```

dann

```
EEX
 8
```

und weitere R/S.

Schließlich werden Zahlen gedrückt, gefolgt von R/S. Ersichtlich ist jedes Programm für sich funktionsfähig.

7. Rechenzeiten

Während einfache Geradeausprogramme meist nur wenige Sekunden zur Ausführung benötigen, sind bei Wiederholschleifen die langen Rechenzeiten oft störend. Sollen die Rechenzeiten vertretbar bleiben, so ist zu berücksichtigen:

a) Zahl der Wiederholschleifen. Häufig genügen für die geforderte Genauigkeit 20 oder 40 Schleifen statt 100 oder 200.

b) Länge des Unterprogramms. Benötigte Teilergebnisse können in einem Vorprogramm gerechnet und gespeichert werden.

c) Aufbau des Unterprogramms. Enthält das Unterprogramm Funktionen, so sollten diese so kurz wie möglich sein. Winkel- oder Exponentialfunktionen sollten möglichst durch Umformung zusammengefaßt werden. Für Ausdrücke wie x^3 oder x^4 sollten ENT ENT x x bzw. g x^2 g x^2 gewählt werden, nicht die langsame Funktion h y^x.

d) Lage des Unterprogramms. Nach Aufruf eines Unterprogramms werden die dem Aufrufbefehl folgenden Schritte der Reihe nach überprüft, ob sie Label des gesuchten Unterprogramms sind. Das Unterprogramm sollte also hinter dem aufrufenden Programm stehen.

8. Weitere Informationsmöglichkeiten

Der Leser, der bis hierher vorgedrungen ist, wird vielleicht noch an weiterer Information interessiert sein.

Als Auskunftstelle sind zunächst die HP-Vertragshändler zu nennen. Diese sind in fast allen größeren Städten zu finden. Auskünfte über Batteriewechsel, Schutz von Karten oder Zubehörteile können hier am einfachsten gegeben werden. Auch Fragen zur Tastenbelegung oder über notwendige Schrittfolgen werden meist sofort beantwortet.

Wer zu einem bestimmten Arbeitsgebiet eine Zusammenstellung überprüfter Programme sucht, wird diese als HP-Programmpaket finden. Dazu kann man bei einer HP-Niederlassung oder einem Vertragshändler das Verzeichnis der offiziellen Programmsammlungen einsehen. Diese nach Sachgebieten geordneten "Pakete" enthalten jeweils ca. 20 programmierte Magnetkarten mit entsprechender Bedienungsanleitung (Handbuch). Preis je Paket ca. 100.- DM.

Eine andere interessante Sammlung bilden die Programme des HP-eigenen Users' Club Europe. Die Sammlung umfaßt mehr als 800 Programme. Alle sind von HP-Benutzern eingesandt und zur Verfügung gestellt worden. Der Clubbeitrag beträgt z.Zt. 40 SFR (ca 50.- DM) im Jahr. Programme können dann für 8 SFR (ca 10.- DM) je Titel bestellt werden. Die Lieferung umfaßt die Programmschrittlisten mit Erläuterungen. Die Anschrift: HP Users' Club Europe, P.O. Box, CH 1217 Meyrin 2, Genf, Schweiz.

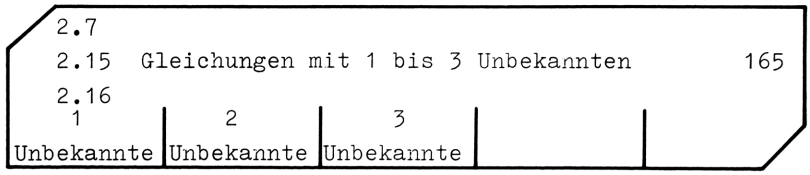
Eine weitere interessante Möglichkeit bietet der Zusammenschluß MICAC für Taschenrechner-Freunde. Die von H. Schnepf herausgegebene Zeitschrift "display" berichtet über Rechenprobleme, Schrittfolgen und Neues vom Rechnermarkt. Jede Ausgabe enthält eine Anzahl Programme als Schrittlisten. Über die Anschriftenliste der Mitglieder ist auch ein direkter persönlicher Kontakt mit anderen Taschenrechner-Fans möglich. Gegen

einen Jahresbeitrag von 48.- DM wird die Zeitschrift kostenfrei geliefert. Die Anschrift:

Heinrich Schnepf, Buchenweg 24, 5000 Köln 40.
Tel. (0221) 50 26 76.

9. Programmschrittlisten

Im Anhang sind die Schrittlisten der längeren Programme zusammengestellt. Nummer und Titel entsprechen dem jeweiligen Programmbeispiel im Text. Die Schrittlisten sind als HP-97-Programmausdruck wiedergegeben. Ein solcher Ausdruck läßt sich von jeder HP 67/97-Programmkarte herstellen. Das Programm ist in Stellung W/PRGM schrittweise in einen HP 67 oder HP 97 einzutasten und auf einer Magnetkarte aufzuzeichnen. Dann wird die Karte in Stellung RUN in einen HP 97 eingelesen. Der obere Schiebeschalter wird in Stellung TRACE gesetzt, der untere Schalter in Stellung PRGM. Nun wird gedrückt f PRGM. Der HP 97 druckt jetzt das eingelesene Programm schrittweise aus.



Eingabe: Labeltaste, R/S, R/S, ...

Register	2 Unbekannte	3 Unbekannte
0	a_0	a_0
1	a_1	a_1
2	a_2	a_2
3	b_0	a_3
4	b_1	b_0
5	b_2	b_1, $(b_3\ c_1 - c_3\ b_1)$
6	$(a_0\ b_1 - a_1\ b_0)$	b_2
7		b_3
8		c_0
9		c_1
A		c_2
B		c_3
C		$(b_1\ c_2 - c_1\ b_2)$, $(b_3\ c_2 - c_3\ b_2)$
D		$(b_0\ c_2 - c_0\ b_2)$, $(b_0\ c_3 - c_0\ b_3)$
E		$(b_0\ c_1 - c_0\ b_1)$
I		D

Schritt	Inhalt
5	x (1 Unbekannte)
34	x (2 Unbekannte)
44	y "
101	D (3 Unbekannte)
129	x "
150	y "
164	z "

```
      *LBLA
      ENT↑
       R/S
        ÷
000    1/X
       RTN
      *LBLB
      STO0
       R/S
      STO1
       R/S
      STO2
       R/S
      STO3
       R/S
      STO4
       R/S
      STO5
       CHS
      RCL1
        ×
      RCL2
      RCL4
        ×
        +
      RCL0
      RCL4
        ×
      RCL1
      RCL3
        ×
        -
      STO6
        ÷
       R/S
      RCL0
      RCL5
        ×
      RCL2
      RCL3
        ×
        -
      RCL6
        ÷
       RTN
      *LBLC
      STO0
       R/S
      STO1
       R/S
      STO2
       R/S
      STO3
       R/S
      STO4
       R/S
      STO5
       R/S
      STO6
       R/S
      STO7
       R/S
      STO8
       R/S
      STO9
       R/S
      STOA
       R/S
      STOB
      RCL5
      RCLA
        ×
      RCL9
      RCL6
        ×
        -
      STOC
      RCL0
        ×
      RCL4
      RCLA
        ×
      RCL8
      RCL6
        ×
        -
      STOD
      RCL1
        ×
        -
      RCL4
      RCL9
        ×
      RCL8
      RCL5
        ×
        -
      STOE
      RCL2
        ×
        +
      STOI
      RCLC
      RCL3
        ×
      RCL7
      RCLA
        ×
      RCLB
      RCL6
        ×
        -
      STOC
      RCL1
        ×
        -
      RCL7
      RCL9
        ^
      RCLB
      RCL5
        ×
        -
      STO5
      RCL2
        ×
        +
      RCLI
        ÷
       R/S
      RCLC
      RCL0
        ×
      RCLD
      RCL3
        ×
        -
      RCL4
      RCLB
        ×
      RCL8
      RCL7
        ×
        -
      STOD
      RCL2
        ×
        +
      RCLI
        ÷
       R/S
      RCLE
      RCL3
        ×
      RCL5
      RCL0
        ×
        -
      RCLD
      RCL1
        ×
        -
      RCLI
        ÷
      RTN
```

2.27 Mondlandespiel				70
(Schubmasse) Abstand km	Geschwindigkeit m/s	Treibstoff kg		Vorbereitung

Zum Spielbeginn Drücken der Spielvorbereitung auf Taste E.

Eingabe der Schubmasse m_S auf Taste A, der Rechner ermittelt den verbleibenden Abstand zur Mondoberfläche. Zur Kontrolle kann der Spieler jeweils Geschwindigkeit und Treibstoffvorrat aufrufen.

Register	Inhalt
1	$m_T = m_{To} - \Sigma\, m_S$
2	m_o, m_v
3	$a = a_o - \frac{\Sigma\, \bar{v}}{100}$, $\bar{v} = \frac{v_1 + v_2}{2}$
4	$\frac{m_S}{m_o + m_T}$
5	m_S
6	$v = v_o - v_s \cdot \sum \frac{m_S}{m_o + m_T} + b_o\, t_o \cdot \sum \frac{1}{(1 + \frac{a}{r_o})^2}$
7	v für $\bar{v} = \frac{v_1 + v_2}{2}$

Schritt	Inhalt
5, 6	Treibstoffvorrat < 0 ?
44, 45	Abstand < 0 ?

DSP 0

```
      *LBLA
       STO5
       ST-1
       RCL1
005      √X
         X²
       RCL2
         +
        1/X
010    RCL5
         X
       STO4
       RCL6
       STO7
015    RCL3
          1
          7
          3
          8
020       ÷
          1
          +
         X²
        1/X
025       1
          6
          .
          2
          X
030    ST+6
       RCL4
       RCL2
          X
       ST-6
035    RCL6
       RCL7
          +
          2
          0
040       0
          ÷
       ST-3
       RCL3
         √X
045      X²
       RTN
      *LBLB
       RCL6
       RTN
050   *LBLC
       RCL1
       RTN
      *LBLE
       CLRG
055       1
          2
         EEX
          3
       STO1
060       4
          ÷
       STO6
       STO7
          2
065       ÷
       STO2
          3
          ÷
       STO3
070    RTN
```

2.28 Optimale Gerade				65
a, b, m_s	y = ax + b			Vorbereitung

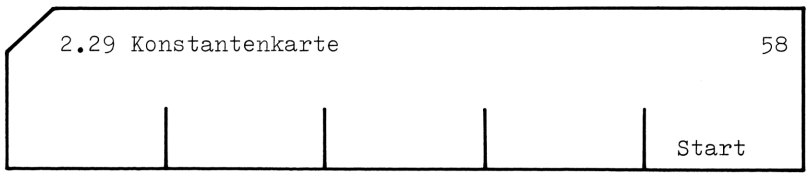
Die Wertepaare werden als x_i ↑ y_i Σ+ eingegeben. Nach Eingabe sämtlicher Wertepaare wird durch Drücken von Labeltaste A die Geradensteigung a berechnet, dann nach R/S der Achsabschnitt b und nach nochmaligem R/S die mittlere Streuung in y-Richtung m_s. Anschließend können x-Werte auf Label B eingegeben werden. Der Rechner gibt die zugehörigen y-Werte an. Vor Eingabe der nächsten Gruppe von Wertepaaren ist Labeltaste E zu drücken. Dadurch werden die Speicher gelöscht.

Register	Inhalt	Register	Inhalt
S 0	a	S 6	Σx_i
S 1	b	S 7	Σx_i^2
S 2	m_s	S 8	$\Sigma x_i \cdot y_i$
S 4	Σy_i	S 9	n
S 5	Σy_i^2		

```
     *LBLA
      P⇄S
     RCL5
     RCL8
005    ×
     RCL6
     RCL4
       ×
       -
010  RCL9
     RCL7
       ×
     RCL6
       X²
015    -
       ÷
     STO0
       R/S
     RCL4
020  RCL0
     RCL6
       ×
       -
     RCL9
025    ÷
     STO1
       R/S
     RCL5
     RCL1
030    X²
     RCL9
       ×
       -
     RCL0
035  RCL7
       ×
     RCL8
       2
       ×
040    -
     RCL0
       ×
       +
     RCL9
045    2
       -
       ÷
      √X
     STO2
050   P⇄S
     RTN
     *LBLB
     ENT↑
      P⇄S
055  RCL0
       ×
     RCL1
       +
      P⇄S
060   RTN
     *LBLE
     CLRG
      P⇄S
     CLRG
065   RTN
```

2.29 Konstantenkarte 58

Start

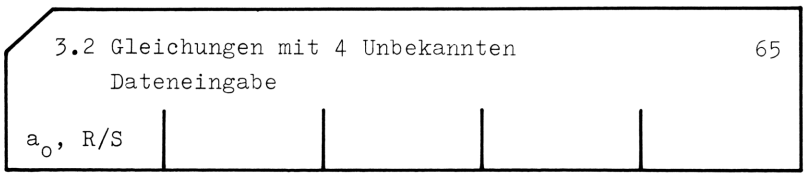


Die Konstanten h bis m_e werden durch Drücken von E in den Registern 1 bis 6 gespeichert. Das folgende Einlesen einer Programmkarte überschreibt das Konstantenprogramm. Die Speicherinhalte bleiben dabei unverändert.

Register	Inhalt
1	Plancksches Wirkungsquantum h
2	Boltzmann-Konstante k
3	Lichtgeschwindigkeit im Vakuum c
4	Stefan-Boltzmannsche Konstante σ
5	Ladung des Elektrons (Elementarladung) e
6	Ruhemasse des Elektrons m_e

```
     *LBLE
         6
         .
         6
005      2
         5
       EEX
       CHS
         3
010      4
      STO1
         1
         .
         3
015      8
         1
       EEX
       CHS
         2
020      3
      STO2
         2
         .
         9
025      9
         8
       EEX
         8
      STO3
030      5
         .
         6
         7
       EEX
035    CHS
         8
      STO4
         1
         .
040      6
         0
         2
       EEX
       CHS
045      1
         9
      STO5
         9
         .
050      1
         0
         9
       EEX
       CHS
055      3
         1
      STO6
       RTN
```

3.2 Gleichungen mit 4 Unbekannten Dateneingabe				65
a_o, R/S				



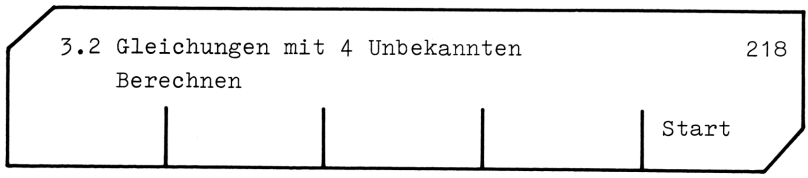
a_o wird auf Label A eingegeben, die weiteren Koeffizienten a_1 bis a_4, b_o bis b_4, c_o bis c_4 sowie d_o bis d_4 der Reihe nach mit R/S. Zur Kontrolle erscheinen nach dem Eingeben der letzten Koeffizienten einer Zeile jeweils Ziffern, und zwar:

Koeffizient	Anzeige
a_4	1111111111
b_4	2222222222
c_4	3333333333
d_4	4444444444

Nach der Eingabe von d_4 werden beide Seiten der Rechenkarte eingelesen.

Register	Inhalt	Register	Inhalt
0	a_0	8	c_0
1	a_1	9	c_1
2	a_2	A	c_2
3	a_3	B	c_3
S 0	a_4	S 2	c_4
4	b_0	C	d_0
5	b_1	D	d_1
6	b_2	E	d_2
7	b_3	S 8	d_3
S 1	b_4	S 3	d_4

```
      *LBL A
       STO 0
        R/S
       STO 1
005     R/S
       STO 2
        R/S
       STO 3
        R/S
010     P≠S
       STO 0
        P≠S
         9
         1/X
015     EEX
         1
         0
         X
       STO I
020     R/S
       STO 4
        R/S
       STO 5
        R/S
025    STO 6
        R/S
       STO 7
        R/S
        P≠S
030    STO 1
        P≠S
       RCL I
         2
         X
035     R/S
       STO 8
        R/S
       STO 9
        R/S
040    STO A
        R/S
       STO B
        R/S
        P≠S
045    STO 2
        P≠S
       RCL I
         3
         X
050     R/S
       STO C
        R/S
       STO D
        R/S
055    STO E
        R/S
        P≠S
       STO 8
        R/S
060    STO 3
        P≠S
       RCL I
         4
         X
065     RTN
```



Nach der Dateneingabe werden beide Seiten der Rechenkarte eingelesen. Das Programm startet durch Drücken der Labeltaste E. Der Rechner berechnet die Determinante D. Ist D ungleich Null, so hat das Gleichungssystem eindeutige Lösungen. Durch wiederholtes Drücken von R/S werden dann der Reihe nach x, y, z und u

berechnet und angezeigt. Ein nochmaliges Drücken von R/S bewirkt eine Rückkehr zur Berechnung von D.

Zum Lösen des nächsten Gleichungssystems müssen die Karten Dateneingabe und Berechnen erneut eingelesen werden.

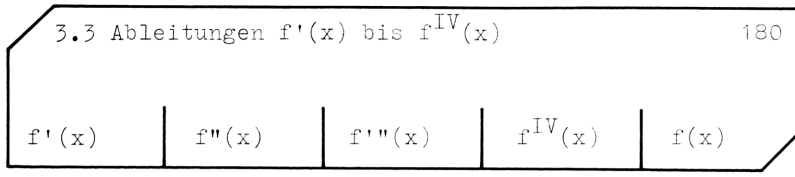
Register	Inhalt
0 bis 9	wie in Schrittliste Dateneingabe
S 0 bis S 3, S 8	"
S 9	Teilsummen der Determinante D
I	D, dann u = $\frac{D_4}{D}$

Label	Inhalt
E	Berechnen von D, x, y, z, u
0	Berechnen von D, D_1 bis D_4
1	Stack abspeichern in S 4 bis S 7, dann S 0 bis S 3 in Stack einschreiben
2	S 4 bis S 7 in Stack einschreiben
3	Stack abspeichern in 0 bis C
4	" 1 bis D
5	" 2 bis E
6	" 3 bis S 8
7	Schritte sparen, Teilsummen von D

Schritt	Inhalt
2	Determinante D
11	" D_1
23	" D_2
35	" D_3
49	" D_4
75	1. Teilsumme von D
91	2. "
104	3. "
117	4. "
132	5. "
145	6. "

```
     *LBLE
      GSB0
      STOI
       R/S
005   RCL0
      RCL4
      RCL8
      RCLC
      GSB1
010   GSB3
      GSB0
      RCLI
       ÷
       R/S
015   GSB2
      GSB3
      RCL1
      RCL5
      RCL9
020   RCLD
      GSB1
      GSB4
      GSB0
      RCLI
025    ÷
       R/S
      GSB2
      GSB4
      RCL2
030   RCL6
      RCLA
      RCLE
      GSB1
      GSB5
035   GSB0
      RCLI
       ÷
       R/S
      GSB2
040   GSB5
      RCL3
      RCL7
      RCLB
       P≠S
045   RCL8
       P≠S
      GSB1
      GSB6
      GSB0
050   RCLI
       ÷
      STOI

      GSB2
      GSB6
055   RCLI
       R/S
      GTOE
      *LBL0
      RCL0
060   RCL5
       x
      RCL4
      RCL1
       x
065    -
      RCLA
      P≠S
      RCL8
       x
070   RCLE
      RCLB
       x
       -
       x
075   STO9
       P≠S
      RCL1
      RCL6
       x
080   RCL5
      RCL2
       x
       -
      RCL8
085   P≠S
      RCL8
       P≠S
       x
      RCLC
090   RCLB
      GSB7
      RCL2
      RCL7
       x
095   RCL3
      RCL6
       x
       -
      RCL8
100   RCLD
       x
      RCLC
      RCL9
      GSB7

105   RCL3
      RCL5
       x
      RCL7
      RCL1
110    x
       -
      RCL8
      RCLE
       x
115   RCLC
      RCLA
      GSB7
      RCL2
      RCL4
120    x
      RCL6
      RCL0
       x
       -
125   RCL9
       P≠S
      RCL8
       P≠S
       x
130   RCLD
      RCLB
      GSB7
      RCL0
      RCL7
135    x
      RCL4
      RCL3
       x
       -
140   RCL9
      RCLE
       x
      RCLD
      RCLA
145   GSB7
       P≠S
      RCL9
       P≠S
      RTN
150   *LBL1
       P≠S
      STO7
       R↓
      STO6
       R↓
155   STO5

       R↓
      STO4
      RCL0
160   RCL1
      RCL2
      RCL3
       P≠S
      RTN
165   *LBL2
       P≠S
      RCL4
      RCL5
      RCL6
170   RCL7
       P≠S
      RTN
      *LBL3
      STOC
175    R↓
      STO8
       R↓
      STO4
       R↓
180   STO0
      RTN
      *LBL4
      STOD
       R↓
185   STO9
       R↓
      STO5
       R↓
      STO1
190   RTN
      *LBL5
      STOE
       R↓
      STOA
195    R↓
      STO6
       R↓
      STO2
      RTN
200   *LBL6
       P≠S
      STO8
       P≠S
       R↓
205   STOB
       R↓
      STO7
       R↓

      STO3
210   RTN
      *LBL7
       x
       -
       x
215    P≠S
      ST+9
       P≠S
      RTN
```

3.3 Ableitungen f'(x) bis $f^{IV}(x)$				180
f'(x)	f"(x)	f'"(x)	$f^{IV}(x)$	f(x)



Nach dem Einlesen der Karte in Stellung RUN speichert der Anwender die Größe h in Register 2 (z.B. $h = 10^{-4}$). Dann wird die Funktion f(x) eingegeben. Es wird gedrückt GTO · 179 und in W/PRGM f(x) eingetastet. Die Veränderliche x ist als RCL 1 aufzurufen. In Stellung RUN läßt sich zu jedem eingetasteten x die Funktion oder eine ihrer Ableitungen aufrufen.

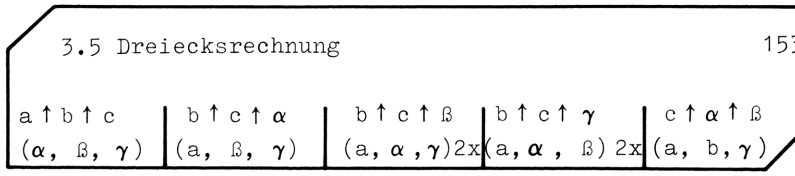
Register	Inhalt
1	x, x + h, ...
2	h (zu Beginn einzugeben)
3	f(x) + f(x + h) - ...

Schritt	Inhalt
5	f(x + h)
13	f(x - h)
179 ff	f(x)

DSP 4

```
     *LBLA
      STO1
      RCL2
      ST+1
005   GSBe
         8
        x
      STO3
      RCL2
010      2
        x
      ST-1
      GSBe
         8
015     x
      ST-3
      RCL2
      ST-1
      GSBe
020   ST+3
      RCL2
         4
        x
      ST+1
025   GSBe
      ST-3
      RCL3
      RCL2
        ÷
030     1
         2
        ÷
       RTN
     *LBLB
035   STO1
      GSBe
         3
         0
        x
040    CHS
      STO3
      RCL2
      ST+1
      GSBe
045     1
         6
        x
      ST+3
      RCL2
050   ST+1
      GSBe
      ST-3
      RCL2
         4
055     x
      ST-1
      GSBe
      ST-3
      RCL2
060   ST+1
```

```
      GSBe
        1
        6
        x
065   RCL3
        +
      RCL2
       X²
        ÷
070     1
        2
        ÷
      RTN
    *LBLC
075   STO1
      RCL2
      ST-1
      GSBe
        1
080     3
        x
      STO3
      RCL2
      ST-1
085   GSBe
        8
        x
      ST-3
      RCL2
090   ST-1

      GSBe
      ST+3
      RCL2
        6
095     x
      ST+1
      GSBe
      ST-3
      RCL2
100   ST-1
      GSBe
        8
        x
      ST+3
      RCL2
105   ST-1
      GSBe
        1
        3
110     x
      CHS
      RCL3
        +
      RCL2
115    X²
        ÷
      RCL2
        ÷
        8
120     ÷

      RTN
    *LBLD
      STO1
      GSBe
125     5
        6
        x
      STO3
      RCL2
130   ST-1
      GSBe
        3
        5
        x
135   ST-3
      RCL2
      ST-1
      GSBe
        1
140     2
        x
      ST+3
      RCL2
      ST-1
145   GSBe
      ST-3
      RCL2
        6
        x
150   ST+1

      GSBe
      ST-3
      RCL2
      ST-1
155   GSBe
        1
        2
        x
      ST+3
160   RCL2
      ST-1
      GSBe
        3
        9
        x
165     CHS
      RCL3
        +
      RCL2
170    X²
       X²
        ÷
        6
        ÷
175    RTN
    *LBLE
      STO1
    *LBLe
      RCL1
180    RTN
```



Register	Inhalt			
	A, B	C	D	E
1	a	-	-	-
2	a^2	$\frac{b}{\sin ß}$	$\frac{c}{\sin \gamma}$	$\frac{c}{\sin \gamma}$
3	b	-	-	-
4	b^2	-	-	-
5	c	c	b	c
6	c^2	-	-	-
7	α	α_1, α_2	α_1, α_2	α
8	ß	ß	γ	ß
9	γ	γ_1, γ_2	$ß_1$, $ß_2$	γ

Label	Inhalt
A	$\cos\alpha = \frac{b^2 + c^2 - a^2}{2\ b\ c}$
B	$a = \sqrt{b^2 + c^2 - 2\ b\ c\ \cos\alpha}$
C	$\sin\gamma_1 = \frac{c}{b} \cdot \sin\beta$
D	$\sin\beta = \frac{b}{c} \cdot \sin\gamma$
E	$a = c \cdot \frac{\sin\alpha}{\sin\gamma}$
0	$\cos\beta = \frac{a^2 + c^2 - b^2}{2\ a\ c}$
1	$\gamma = 180^\circ - \alpha - \beta$
2	$\alpha = 180^\circ - \gamma - \beta$

DSP 4

```
     *LBLA
      STO5
        X²
      STO6
005     R↓
      STO3
        X²
      STO4
        R↓
010   STO1
        X²
      STO2
       CHS
      RCL4
015     +
      RCL6
        +
      RCL3
        ÷
020   RCL5
        ÷
         2
        ÷
      COS⁻¹
025   STO7
       R/S
      GSB0
       R/S
      GTO1
030   *LBLB
      STO7
        R↓
      STO5
        X²
035   STO6
        R↓
      STO3
        X²
      STO4
040   RCL6
        +
      RCL3
      RCL5
        x
045   RCL7
       COS
        x
         2
        x
050     -
      STO2
       √X
      STO1
       R/S
055   GSB0
       R/S
      GTO1
     *LBL0
      RCL6
060   RCL2
        +
      RCL4
        -
      RCL5
065     ÷
      RCL1
        ÷
         2
        ÷
070   COS⁻¹
      STO8
       RTN
     *LBL1
         1
075      8
         0
      RCL7
        -
      RCL8
080     -
```

```
      STO9
       RTN
    *LBL C
      STO8
       R↓
      STO5
       R↓
      RCL8
       SIN
 090    ÷
      STO2
       1/X
      RCL5
       X
 095  SIN-1
      GTO2
    *LBL D
      STO8
       SIN
 100    ÷

      STO2
       X⇄Y
      STO5
       ÷
 105   1/X
      SIN-1
    *LBL 2
      STO9
       CHS
 110    1
       8
       0
       +
      RCL8
 115    -
      STO7
       SIN
      RCL2
       X
 120   R/S

      RCL7
       R/S
      RCL9
       R/S
 125   CHS
       1
       8
       0
       +
 130  GTO2
    *LBL E
      STO8
       R↓
      STO7
 135    R↓
      STO5
      GSB1
       SIN
      RCL5
 140    ÷

       1/X
      STO2
      RCL7
       SIN
 145    X
       R/S
      RCL8
       SIN
      RCL2
 150    X
       R/S
      RCL9
       RTN
```

4.5 Kreis durch drei Punkte 158

	x			
Eingabe x_1	$(\rightarrow y_1, y_2)$			

Das Programm startet durch Eingeben von x_1 auf Label A. Die anderen fünf Werte y_1, x_2, y_2, x_3, y_3 werden der Reihe nach über R/S eingegeben. Das Programm berechnet selbständig a, dann nach R/S b und nach nochmaligem R/S den Kreisradius R. Anschließend lassen sich zu x-Werten innerhalb des Kreises über Label B und R/S die zugehörigen Doppelwerte y_1, y_2 abrufen.

Register	Inhalt	Register	Inhalt
0	x_1	8	x_3
1	x_1^2	9	x_3^2
2	y_1	A	y_3
3	y_1^2	B	y_3^2
4	x_2	C	a
5	x_2^2	D	b
6	y_2	E	R
7	y_2^2		

Label	Inhalt
A	Dateneingabe, Berechnen von a, b, R
B	Berechnen von y_1, y_2
0	Berechnen von b auf 1. Art
1	Berechnen von R

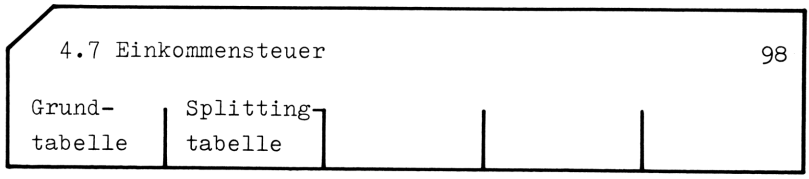
Schritt	Inhalt
52	Zähler von 2 a
69	Nenner von 2 a
73	a
99	b auf 2. Art
112	R
130	Zähler von 2 b
136	b auf 1. Art
151	$\sqrt{\ldots}$ ins X-Register

DSP 4

```
    *LBLA
    STO0
     X²
    STO1
005  R/S
    STO2
     X²
    STO3
     R/S
010 STO4
     X²
    STO5
     R/S
    STO6
015  X²
    STO7
     R/S
    STO8
     X²
020 STO9
     R/S
    STOA
     X²
    STOB
025 RCL7
     -
    RCL9
     +
    RCL5
030  -
    RCL2
     X
    RCL1
    RCL9
035  -
    RCL3
     +
    RCLB
     -
040 RCLE
     X
     +
    RCL5
    RCL1
045  -
    RCL7
     +
    RCL3
     -
050 RCLA
     X
     +
    RCL8
    RCL4
055  -
    RCL2
     X
    RCL0
    RCL8
060  -
    RCL6
     X
     +
    RCL4
065 RCL0
     -
    RCLA
     X
     +
070  ÷
     2
     ÷
    STOC
     R/S
075 RCLE
    RCLA
    X=Y?
    GTO0
    RCL5
080 RCL9
     -
    RCL7
     +
    RCLB
085  -
    RCL8
    RCL4
     -
    RCLC
090  X
     2
     X
     +
    RCL6
095 RCLA
     -
     ÷
     2
     ÷
100 STOD
     R/S
    *LBL1
    RCL0
    RCLC
105  -
     X²
    RCL2
    RCLD
     -
110  X²
     +
     √X
    STOE
    RTN
115 *LBL0
    RCL1
    RCL9
     -
    RCL3
     +
120 RCLB
     -
    RCL8
    RCL0
125  -
    RCLC
     X
     2
     X
     +
130 RCL2
    RCLA
     -
     ÷
135  2
     ÷
    STOD
     R/S
    GTO1
140 *LBLB
    ENT↑
    RCLC
     -
     X²
145  CHS
    RCLE
     X²
     +
     √X
150 RCLD
    X≷Y
     +
     R/S
    LSTX
155  CHS
    RCLD
     +
    RTN
```

4.7 Einkommensteuer 98

Grund-tabelle	Splitting-tabelle			



Das zu versteuernde Jahreseinkommen wird für Alleinstehende nach der Grundtabelle, für Verheiratete nach der Splittingtabelle berechnet. Die Abweichungen zur maßgeblichen Tabelle des Finanzamtes bleiben unter 50.- DM.

Register	Inhalt
1	Einkommen abzüglich 16000.- DM

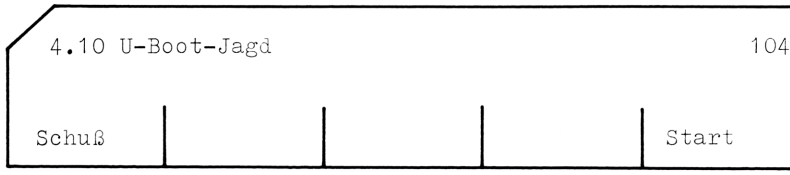
Label	Inhalt
A	Steuer nach Grundtabelle
B	" Splittingtabelle
0	Einkommen kleiner als 16000.- DM
1	" 3690.- DM
2	" 130000.- DM

DSP 0

```
     *LBLA          GTO0          9          CHS
      ENT↑            1          1            .
         3            3          3            2
         0          EEX          3            2
005      ÷      020    4      035    +      050    ×
       INT          X⇄Y         RTN          RTN
         3         X≤Y?       *LBL0        *LBL1
         0         GTO2          3            0
         ×           -           6          RTN
010      1      025  CHS      040    9      055 *LBL2
         6            .           1            1
       EEX            5          X⇄Y           6
         3            6         X≤Y?           0
       X⇄Y            ×         GTO1           2
015   X≤Y?      030    5      045    -      060    0
```

```
        -
      STO1
         3
         7
065      5
         3
         0
        +
      1/X
070      1
         3
         2
         9
         4
075      x
       CHS
         .
         5
         6
080      2
         8
        +
      RCL1
        x
085      2
         7
         1
         2
        +
090    RTN
*LBLB
 ENT1
    2
    ÷
095 GSBA
    2
    x
  RTN
```

4.10 U-Boot-Jagd				104
Schuß				Start



Zur Spielvorbereitung wird eine willkürliche Zahl (z.B. 256, 8001, ...) auf Label E eingegeben. Darauf versteckt der Rechner fünf U-Boote im 9 x 9-Spielfeld. Als Schuß wird eine Feld-Nr. eingegeben, also z.B. 2.3 für die zweite Zeile Feld Nr. 3. Zweckmäßig verfolgt der Spieler die Anzeige auf einem eigenen 9 x 9-Spielfeld, z.B. auf kariertem Papier. Nach Spielende können die U-Boot-Positionen mit RCL 1 bis RCL 5 aufgerufen werden.

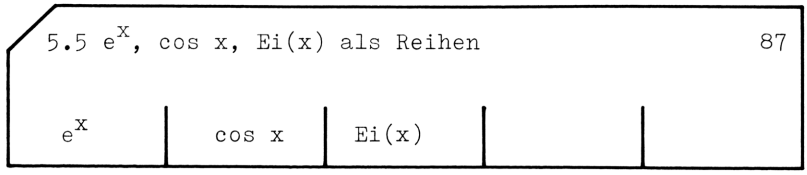
Register	Inhalt
1	Position von U-Boot Nr. 1
2	" 2
3	" 3
4	" 4
5	" 5
6	jeweilige Feldeingabe (Schuß)
7	Position des geprüften U-Boots

Label	Inhalt
A	Prüfung der fünf U-Boot-Positionen
B	Spielvorbereitung
0	Vergleichen der Feldeingabe mit Position des jeweiligen U-Boots
1	Anzeige Treffer
2	Anzeige Streifschuß
3	Feldnummern 1.1 bis 9.9

Flag	Inhalt
0	gesetzt: Treffer
1	gesetzt: Streifschuß

DSP 1

```
    *LBLA
     STO6
      CF0
      CF1
005   RCL1
     GSB0
     RCL2
     GSB0
     RCL3
010  GSB0
     RCL4
     GSB0
     RCL5
     GSB0
015   F0?
     GTO1
      F1?
     GTO2
        0
020   RTN
    *LBL0
     STO7
     RCL6
     X=Y?
025   SF0
      INT
     RCL7
      INT
        -
030   ABS
        1
      X≠Y
     X>Y?
      RTN
035   RCL6
      FRC
     RCL7
      FRC
        -
040   ABS
        .
        1
      X≠Y
     X≤Y?
045   SF1
      RTN
    *LBL1
        9
      1/X
050   EEX
        1
        0
        x
      RTN
055 *LBL2
        1
        1
        1
      RTN
060 *LBLE
     X≠0?
       LN
      FRC
     ST+7
065  GSB3
     STO1
     GSB3
     STO2
     GSB3
070  STO3
     GSB3
     STO4
     GSB3
     STO5
075     3
      1/X
      EEX
        1
        0
080     x
      RTN
    *LBL3
     RCL7
       Pi
085    +
      X²
      FRC
     STO7
        8
090     9
        x
      INT
        .
        1
095     x
        1
        .
        1
        +
100   INT
     LSTX
     X=Y?
     GTO3
      RTN
```



x wird eingegeben und der jeweilige Funktionswert aufgerufen. Bei cos x ist x in Radian einzugeben mit $\frac{\pi}{2} = 90^o$. In den Funktionen e^x und Ei(x) ist x beschränkt auf Werte kleiner gleich 23. Für diesen Wert wird die Funktion n! als 69! aufgerufen, dem höchsten $n! < 10^{100}$. Diese Beschränkung ist für e^x unwesentlich, da sich dieses auch als Tastenfunktion für $-227 < x < 230$ aufrufen läßt. Die Funktion Ei(x) wird zum Berechnen von $|x| > 23$ aufgeteilt. Als Integral läßt sich

$$Ei(x) = \int_{-\infty}^{x} \frac{e^t}{t}\,dt \quad \text{aufteilen in} \quad \int_{-\infty}^{x} = \int_{-\infty}^{x_o} + \int_{x_o}^{x} \quad \text{mit } |x_o| < 23.$$

Das erste Teilintegral wird als Reihe bestimmt, das zweite durch zahlenmäßige Integration nach Simpson, vgl. Beispiel 5.12.

So läßt sich z.B. das Integral von $-\infty$ bis 23 mit $x_o = 20$ auf zwei Arten bestimmen:

Als Reihe gerechnet von $-\infty$ bis 23 = $4.4397 \cdot 10^8$
Als Reihe gerechnet von $-\infty$ bis 20 = $0.2562 \cdot 10^8$
Nach Simpson von 20 bis 23 = $4.1835 \cdot 10^8$.

Für $x < -23$ wird ein x_o kleiner Null gewählt, z.B. $x_o = -20$. Entsprechend ist

$$\int_{-\infty}^{x} = \int_{-\infty}^{x_o} - \int_{x}^{x_o} .$$

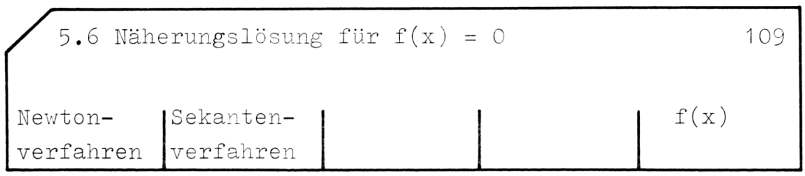
Zur Veranschaulichung lassen sich die Teilintegrale (Flächen unter der Kurve) aufzeichnen.

Register	Inhalt		
	A	B	C
1	x	$-x^2$	x
2	$x, x^2, x^3, \ldots$	$2, 4, 6, \ldots$	$x, x^2, x^3, \ldots$
3	$1, 2, 3, \ldots$	$-x^2, x^4, -x^6, \ldots$	$1, 2, 3,$
4	$1 + \frac{x}{1!} + \frac{x^2}{2!} \ldots$	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \ldots$	$\frac{x}{1 \; 1!} + \frac{x^2}{2 \; 2!} + \frac{x^3}{3 \; 3!} + \ldots$

Label	Inhalt
A	Berechnen von e^x
B	" cos x
C	" Ei(x)
0	Wiederholschleife für e^x
1	" cos x
2	" Ei(x)

DSP 4

```
      *LBLA
       CLRG
       STO1
          1
005    STO2
       STO4
      *LBL0
       RCL1
       STx2
010       1
       ST+3
       RCL2
       RCL3
         N!
015       ÷
       ST+4
        ABS
        EEX
        CHS
020       4
       X≤Y?
       GTO0
       RCL4
        RTN
025   *LBLB
       CLRG
         X≥
        CHS
       STO1
030       1
       STO3
       STO4
      *LBL1
       RCL1
035    STx3
          2
       ST+2
       RCL3
       RCL2
040      N!
          ÷
       ST+4
        ABS
        EEX
045     CHS
          4
       X≤Y?
       GTO1
       RCL4
050     RTN
      *LBLC
       CLRG
       STO1
          1
055    STO2
      *LBL2
       RCL1
       STx2
          1
060    ST+3
       RCL2
       RCL3
         N!
          ÷
065    RCL3
          ÷
       ST+4
        ABS
        EEX
070     CHS
          6
       X≤Y?
       GTO2
       RCL4
075    RCL1
        ABS
         LN
          +
          .
080       5
          7
          7
          2
          1
085       6
          +
        RTN
```



In Stellung RUN wird gedrückt GTO E, dann wird in W/PRGM f(x) eingegeben. Die Veränderliche x ist mit ENT oder mit STO 0 einzugeben. Zunächst wird eine Wertetabelle erstellt und die Funktion auf Nullstellen geprüft. Ein Vorzeichenwechsel zwischen zwei f(x)-Werten bedeutet bei stetigen Funktionen eine Nullstelle in diesem Bereich. Die beiden x-Werte links und rechts der gesuchten Nullstelle werden eingegeben mit Drücken von A und R/S beim Newtonverfahren bzw. B und R/S beim Sekantenverfahren. Der berechnete x-Wert der Nullstelle kann durch Drücken von E den Funktionswert f(x) an dieser Stelle feststellen. Dieser sollte mit guter Näherung Null sein.

Es ist im Newtonverfahren (Tangentennäherung)

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}, \qquad f'(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2\,h}$$

Beim Sekantenverfahren ist der entsprechende x-Wert für die angenäherte Nullstelle

$$c = \frac{b \cdot |f(a)| + a \cdot |f(b)|}{|f(a)| + |f(b)|}$$

Register	Inhalt	
	Newtonverfahren	Sekantenverfahren
0	vorgesehen für x in f(x)	vorgesehen für x in f(x)
1	x_0	a
2	$h = 10^{-4}$	f(a)
3	$f(x_0)$	$\lvert f(a) \rvert$
4	$f(x_0 + h)$	b
5	-	f(b)
6	-	$\lvert f(b) \rvert$
7	-	c
8	-	f(c)

Label	Inhalt
A	Newtonverfahren
B	Sekantenverfahren
E	f(x)
0	Wiederholschleife Newtonverfahren
1	Programmende "
2	Wiederholschleife Sekantenverfahren
3	Programmende "
4	Austausch der Grenzen "

Schritt	Inhalt
11	Errormeldung falls kein Vorzeichenwechsel
63	"

DSP 4

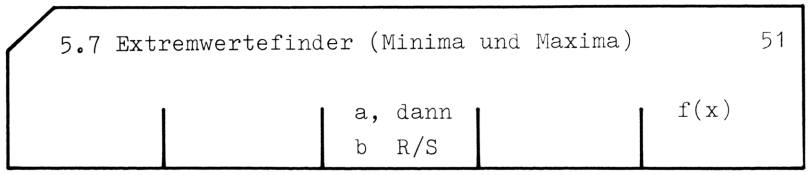
```
     *LBLA
      STO1
      GSBE
      STO3
005    R/S
      GSBE
      RCL3
        x
         0
010   X≤Y?
```

```
         ÷
        EEX
        CHS
          4
015    STO2
      *LBL0
       RCL1
       GSBE
       STO3
020     ABS
```

```
       RCL2
       X>Y?
       GTO1
       RCL1
025      +
       GSBE
       STO4
       RCL1
       RCL2
030      -
```

```
       GSBE
       RCL4
         -
        1/X
035    RCL3
         x
          2
         x
       RCL2
040      x
```

```
      ST+1
      GTO0
      *LBL1
      RCL1
045   RTN
      *LBLB
      STO1
      GSBE
      STO2
050   ABS
      STO3
      R/S
      STO4
      GSBE
055   STO5
      ABS
      STO6
      RCL2
      RCL5
060   x
      0
      X≤Y?
      ÷
      *LBL2
065   RCL1
      RCL6
      x
      RCL4
      RCL3
070   x
      +
      RCL3
      RCL6
      +
075   ÷
      STO7
      GSBE
      STO8
      ABS
080   EEX
      CHS
      4
      X>Y?
      GTO3
085   RCL2
      RCL8
      x
      X<0?
      GTO4
090   RCL7
      STO1
      RCL8
      STO2
      ABS
095   STO3
      GTO2
      *LBL4
      RCL7
      STO4
100   RCL8
      STO5
      ABS
      STO6
      GTO2
105   *LBL3
      RCL7
      RTN
      *LBLE
      RTN
```



In RUN wird nach GTO E auf W/PRGM umgeschaltet und f(x) eingegeben. x ist mit ENT oder mit STO 0 einzugeben. In RUN wird die untere Bereichsgrenze a auf C eingegeben, dann mit R/S b. Das Programm prüft mit einer Schrittweite $h = \frac{|b - a|}{100}$ den Bereich auf Extremwerte. Bei Auffinden eines Extremwertes hält es an. Nach R/S läuft das Programm weiter zum nächsten Minimum oder Maximum. Enthält der Bereich keine weiteren Extremwerte, so sucht das Programm weiter. Die Ausführung kann durch Tastendruck (z.B. CLX) unterbrochen werden.

Register	Inhalt
0	vorgesehen für x in f(x)
1	a
2	$h = \frac{\lvert b - a \rvert}{100}$
3	f(x)
4	f(x + h)

Label	Inhalt
C	Rechenprogramm
E	f(x)
5	Prüfen f(x + h) > f(x)?
6	Bilden des nächsten Funktionswertes
7	Flag löschen

Flag	Inhalt
0	Setzen, falls f(a + h) ⩽ f(a)

DSP 4

```
    *LBL C
     STO 1
      R/S
       -
005   ABS
      EEX
       2
       ÷
      ABS
010  STO 2
    *LBL 5
     RCL 1
     GSBE
     STO 3
015  RCL 2
     ST+ 1
     RCL 1
     GSBE
     STO 4
020  RCL 3
     X≷Y
     X>Y?
     GTO 5
      SF 0
025  X≷Y
     X>Y?
     GTO 6
    *LBL 7
      CF 0
030  RCL 1
     RCL 2
       -
      R/S
     GTO 5
035 *LBL 6
     RCL 4
     STO 3
     RCL 2
     ST+ 1
     GTO 6
040  RCL 1
     GSBE
     STO 4
     RCL 3
     RCL 4
045   F0?
     X≷Y
     X>Y?
     GTO 6
     GTO 7
050 *LBL E
      RTN
```

5.10 Zahlenumwandlung 71

Dez → Okt	Okt → Dez	Dez → Dual	Dual → Dez	Okt → Dual Dual → Okt (mit f e)

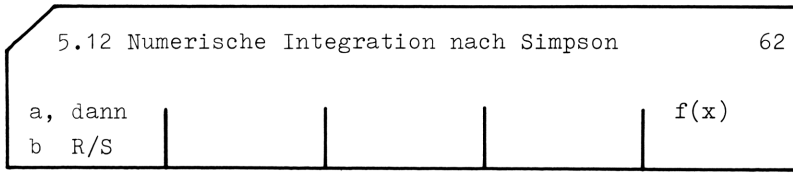
Die umzuwandelnde Zahl wird auf dem entsprechenden Label eingegeben, also z.B. Okt → Dual auf E, Dual → Okt auf f e.

Register	Inhalt
0	eingegebene Zahl
1	Basis der gesuchten Zahl
2	gesuchte Zahl
3	Stellenmultiplikator
4	Basis der eingegebenen Zahl

Label	Inhalt
A bis E, f e	Zahlenumwandlung
0	Vorbereitung
1	Wiederholschleife
2	Bilden der Zahl

DSP 0

```
      *LBLA
       STO0
          8
       STO1
065       1
          0
       STO4
      *LBL0
          0
010   STO2
          1
       STO3
      *LBL1
       RCL0
015    RCL1
       X>Y?
       GTO2
          ÷
          INT
020    STO6
       LSTX
        FRC
       RCL1
          ×
025    RCL3
          ×
       ST+2
       RCL4
       ST×3
030    GTO1
      *LBL2
       X≠Y
       RCL3
          ×
035    RCL2
          +
        RTN
      *LBLB
       STO0
040       1
          0
       STO1
          8
       STO4
045    GTO0
      *LBLC
       STO0
          2
       STO1
050       1
          0
       STO4
       GTO0
      *LBLD
055    STO0
          1
          0
       STO1
          2
060    STO4
       GTO6
      *LBLE
       GSBB
        PSE
065    GSBC
        RTN
      *LBLe
       GSBD
        PSE
070    GSBA
        RTN
```



In RUN wird gedrückt GTO E, dann wird in W/PRGM f(x) eingegeben. Die untere Integrationsgrenze ist durch Drücken von A einzugeben, dann die obere mit R/S. Für eine genauere Integration bei entsprechend erhöhter Rechenzeit läßt sich die Zahl der Unterteilungen n erhöhen. Z.B. wird statt $\frac{n}{2} = 10$ in Schritt 7 und 8 gewählt $\frac{n}{2} = 50$ oder 100. Entsprechend erhöht sich die erforderliche Rechenzeit.

Register	Inhalt
0	a, a + 2 h, a + 4 h, ..., a + h, a + 3 h, ...
1	a
2	b
3	$\frac{n}{2} = 10$
4	$2\,h = \frac{\lvert b - a \rvert}{10}$
5	f(a + h, 4 · [f(a + h) + f(a + 3 h) + ...] + 2 · [f(a + 2 h) + f(a + 4 h) + ...] + f(a)
6	0, [f(a + 3 h) + f(a + 5 h) + ...] , [f(a + 2 h) + f(a + 4 h) + ...]

Label	Inhalt
A	Numerische Integration von f(x)
E	f(x)
0	Vorbereitung
1	Wiederholschleife

Schritt	Inhalt
18	f(a + h)
24	4 [f(a + h) + f(a + 3 h) + ...]
30	+ 2 [f(a + 2 h) + f(a + 4 h) + ...]
33	+ f(a)

```
      *LBLA
       STO1
        R/S
       STO2
005      -
        CHS
         1
         0
       STO3
010      ÷
        ABS
       STO4
         2
         ÷
015    RCL1
         +
       STO0
       GSBE
       STO5
020    GSB0
       RCL5
         +
         4
         ×
025    STO5
       RCL1
       STO0
       GSB0
         2
030      ×
       ST+5
       RCL1
       GSBE
       ST+5
035    RCL2
       GSBE
       RCL5
         +
       RCL4
040      ×
         6
         ÷
       RTN
      *LBL0
045      0
       STO6
       RCL3
         1
         -
050    STO1
      *LBL1
       RCL4
       ST+0
       RCL0
055    GSBE
       ST+6
       DSZI
       GTO1
       RCL6
060    RTN
      *LBLE
       RTN
```

5.13 Numerische Lösung von y' = f(x, y) 66

x_0, dann
y_0, R/S

f(x, y)

In Stellung RUN wird gedrückt GTO E, dann wird in W/PRGM f(x, y) eingegeben mit x als RCL 1, y als RCL 2. Der Anfangswert x_0 wird auf A eingegeben, das zugehörige y_0 mit R/S. Bei Eingabe des gewünschten x_1 mit R/S berechnet das Programm y_1 und mit einem weiteren R/S y_1'. Die Rechnung wird genauer, wenn die Zahl der Unterteilungen n = 40 in Schritt 9 und 10 durch eine größere gerade Zahl (z.B. n = 100 oder 200) ersetzt wird. Entsprechend erhöht sich die erforderliche Rechenzeit.

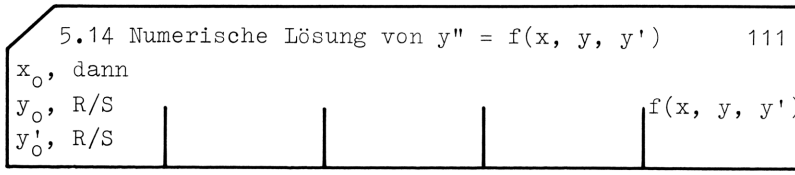
Register	Inhalt
1	x_o, $x_o + \frac{h}{2}$, $x_o + h$
2	y_o, $y_o + \frac{k_1}{2}$, $y_o + \frac{k_2}{2}$, $y_o + k_3$
3	y_o, später y_n
4	$\frac{h}{2} = \frac{\lvert x_1 - x_o \rvert}{80}$
5	$\frac{k_1}{2} = \frac{h}{2} \cdot f(x_o, y_o)$
6	$\frac{k_2}{2} = \frac{h}{2} \cdot f(x_o + \frac{h}{2}, y_o + \frac{k_1}{2})$
7	$k_3 = h \cdot f(x_o + \frac{h}{2}, y_o + \frac{k_2}{2})$
I	n = 40

Label	Inhalt
A	Numerische Lösung von y' = f(x, y)
E	f(x, y)
0	Wiederholschleife

Schritt	Inhalt
62	y_1
63	y_1'

DSP 4

```
    *LBLA
     STO1
      R/S
     STO2
005  STO3
      R/S
     RCL1
       -
       4
010    0
     STO1
       ÷
       2
       ÷
015  ABS
     STO4
    *LBL0
     GSBE
     RCL4
020    x
     STO5
     ST+2
     RCL4
     ST+1
025  GSBE
     RCL4
       x
     STO6
     RCL3
030    +
     STO2
     GSBE
     RCL4
       x
035    2
       x
     STO7
     RCL3
       +
040  STO2
     RCL4
     ST+1
     GSBE
     RCL4
045    x
     RCL7
       +
     RCL6
       2
050    x
       +
     RCL5
       +
       3
055    ÷
     RCL3
       +
     STO2
     STO3
060  DSZI
     GTO0
      R/S
     GSBE
      RTN
065 *LBLE
      RTN
```



In RUN wird gedrückt GTO E, dann wird in W/PRGM f(x, y, y') eingegeben. In der Funktion ist x als RCL 0 aufzurufen, y als RCL 2 und y' als RCL 4. Der Anfangswert x_0 wird auf Label A eingegeben, dann folgen y_0, y'_0 und x_1 mit R/S. Das Programm berechnet zum gewünschten Endwert x_1 den zugehörigen Funktionswert y_1. Es folgen mit R/S y'_1 und y''_1. Die Rechnung wird genauer, wenn statt der Zahl der Unterteilungen n = 40 in den Schritten 12 und 13 eine höhere gerade Zahl gewählt wird, z.B. n = 100 oder 200. Entsprechend steigt damit die erforderliche Rechenzeit.

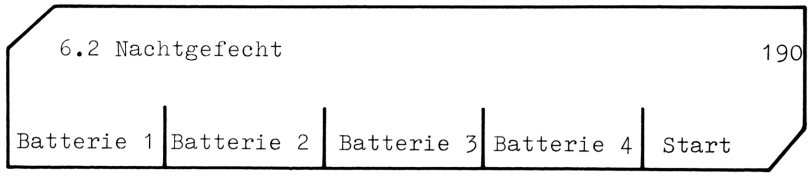
Speicher	Inhalt
0	x_0, x für f(x, y, y')
1	y_0, y_n
2	y für f(x, y, y')
3	y'_0, y'_n
4	y' für f(x, y, y')
5	$\frac{h}{2} = \frac{\lvert x_1 - x_0 \rvert}{80}$
6	$\frac{k_1}{2}$
7	$\frac{k_2}{2}$
8	k_3
9	$\frac{k_4}{2}$
I	n = 40

Label	Inhalt
A	Numerische Lösung von y" = f(x, y, y')
E	f(x, y, y')
0	Wiederholschleife

Schritt	Inhalt
104	y_1
106	y_1'
108	y_1''

DSP 4

```
    *LBLA
     STO0
      R/S
     STO1
005  STO2
      R/S
     STO3
     STO4
      R/S
010  RCL0
       -
       4
       0
     STO1
015    ÷
       2
       ÷
     ABS
     STO5
020  *LBL0
     GSBE
     RCL5
       x
     STO6
025  ST+4
       2
       ÷
     RCL3
       +
030  RCL5
     ST+0
       x
     ST+2
     GSBE
035  RCL5
       x
     STO7
     RCL3
       +
040  STO4
     RCL7
       2
       ÷
       -
045  RCL5
       x
     RCL1
       +
     STO2
050  GSBE
     RCL5
       x
       2
       x
055  STO8
     RCL3
       +
     STO4
     RCL3
060    +
     RCL5
       x
     RCL1
       +
065  STO2
     RCL5
     ST+0
     GSBE
     RCL5
070    x
     STO5
     RCL6
       2
       x
075  RCL7
       2
       x
       +
     RCL8
080    +
       3
       ÷
     RCL3
       2
085    x
       +
     RCL5
       x
     ST+1
090  RCL7
       2
       x
     RCL6
       +
095  RCL8
       +
     RCL9
       +
       3
100    ÷
     ST+3
     DSZI
     GTO0
     RCL1
105  R/S
     RCL3
     R/S
     GSBE
     RTN
110  *LBLE
     RTN
```



Nach dem Eintasten einer beliebigen Zahl und Drücken von E (Start) versteckt der Rechner 7 Gegner auf einem 9 x 9-Spielfeld. Die Geschützbatterien 1 bis 4 verfügen über je 6 Schuß als Zielangaben. So bedeutet 2.6 A den Schuß aus Batterie A auf die 2. Zeile 6. Feld. Zweckmäßig wird zuerst Batterie A leergeschossen, dann B und so fort.

Register	Inhalt
0	Position von Gegner Nr. 0
1	" 1
2	" 2
3	" 3
4	" 4
5	" 5
6	" 6
7	Position des (i)-ten Gegners
8	Zieleingabe
9	Bilden der Position
A	Schußzahl in Batterie Nr. A
B	" B
C	" C
D	" D
E	Zeilenunterschied
I	Nr. des überprüften Gegners

Label	Inhalt
A	Schuß aus Batterie A
B	" B
C	" C
D	" D
E	Spielvorbereitung
0	Vergleich der Schußkoordinate mit jeweiliger Gegnerposition
1	"Treffer"
2	"Streifschuß"
3	Erzeugung der 7 Positionen
4	"daneben"
5	Prüfvorgang
6	"Batterie leer"

Flag	Inhalt
0	Setzen bei Streifschuß

Schritt	Inhalt
91	Unterschied in Zeile im X-Register
102	Unterschied in Spalte im X-Register
113	bei Treffer Löschen der Gegnerposition
122	Stop ohne Rückkehr zu Label 5
156	Schußzahl je Batterie
188	falls Gegnerposition hinten Null: nochmals Position berechnen

```
DSP 0
```

```
    *LBLA
     STO6
     RCLA
        1
005     -
     X<0?
     GTO6
     STOA
     GTO5
010 *LBLB
     STO6
     RCLB
        1
        -
015  X<0?
     GTO6
     STO6
     GTO5
     *LBLC
020  STO6
     RCLC
        1
        -
     X<0?
025  GTO6
     STOC
     GTO5
     *LBLD
     STO6
030  RCLD
        1
        -
     X<0?
     GTO6
035  STOD
     *LBL5
      CF0
       0
     STOI
040  RCL0
     X≠0?
     GSB0
        1
     STOI
045  RCL1
     X≠0?
     GSB0
        2
     STOI
050  RCL2
     X≠0?
     GSB0
        3
     STOI
055  RCL3
     X≠0?
     GSB0
        4
     STOI
060  RCL4
     X≠0?
     GSB0
        5
     STOI
065  RCL5
     X≠0?
     GSB0
        6
     STOI
070  RCL6
     X≠0?
     GSB0
      F0?
     GTO2
075  *LBL4
       0
     RTN
     *LBL0
     STO7
080  RCL8
     X=Y?
     GTO1
     RCL7
      INT
085  RCL8
      INT
       -
     ABS
     STOE
090     1
     X≠Y
     X>Y?
     GTO4
     RCL7
095   FRC
     RCL8
      FRC
       -
     ABS
100     .
        1
     X≠Y
     X>Y?
     GTO4
105  X=0?
      SF0
     RCLE
     X=0?
      SF0
110  RTN
     *LBL1
        0
     STOi
        3
115  1/X
     EEX
        1
        0
        ×
120  PSE
     PSE
     R/S
     *LBL2
      CF0
125     3
        3
        3
     RTN
     *LBL6
130     1
        1
        1
        1
        1
135  RTN
     *LBLE
     ABS
     X≠0?
      LN
140   FRC
     3T+9
     GSB3
     STO0
     GSB3
145  STO1
     GSB3
     STO2
     GSB3
     STO3
150  GSB3
     STO4
     GSB3
     STO5
     GSB3
155  STO6
        6
     STOA
     STOB
     STOC
160  STOD
        9
      1/X
      EEX
        1
165     0
        ×
      RTN
     *LBL3
     RCL9
170   Pi
        +
      X²
     FRC
     STO9
175     8
        9
        ×
      INT
        .
180     1
        ×
        1
        .
        1
185     +
      INT
     LSTX
     X=Y?
     GTO3
190   RTN
```

10. Rechnen mit dem HP-41 C

Bereits äußerlich unterscheidet sich der HP-41 C von der Gruppe HP-33 E bis 97. Das helle Anzeigefeld mit den dunklen Zeichen der Flüssigkristall-Anzeige verbraucht weniger Energie als eine Leuchtdioden-Anzeige. Dadurch entfällt das sonst notwendige Ladegerät. Die einfachen eingesetzten Batterien reichen für ca. neun Monate.

Eine weitere wichtige Neuerung ist die Verwendung steckbarer Zusätze wie Kartenleser oder Speichermodul. Der Rechner wird damit zum Zentrum eines kleinen Systems. Der Verzicht auf den Kartenleser in der Grundausführung ist evtl. zu verschmerzen, da eingetastete Programme auch nach dem Ausschalten erhalten bleiben (C = continuous memory). Außerdem ist der Speicherplatz so groß, daß eine ganze Anzahl auch schwieriger Programme Platz finden. Allerdings ist die etwas höhere Empfindlichkeit des Rechners gegenüber elektrischen Entladungen zu beachten. So darf ein Batteriewechsel und ein Stecken der Zusätze nur bei ausgeschaltetem Gerät erfolgen. Entsprechend dürfen Kontakte am Rechner und an den Steckmoduln nicht berührt werden.

An den Tasten fällt auf, daß die Zahl der fest mit den Tasten verknüpften Funktionen zurückgegangen ist. Dafür tragen die Tasten auf ihrer vorderen Schrägfläche die Buchstaben des Alphabets. Zu Buchstabenketten (Worten) zusammengestellt, lassen sich mit ihnen Funktionen aufrufen sowie in Programmen die Ein- und Ausgabe genauer kennzeichnen. Der Dialog mit dem Rechner wird dadurch leichter.

Der HP-41 C überspringt so die bisherige Lücke zwischen den Taschen- und den Tischrechnern.

Der Lieferumfang

In der Grundausstattung verfügt der HP-41 C über einen Block von 63 Registern. Diese lassen sich wahlweise als Speicherregister oder als Programmzeilen verwenden. Ein Speicherregister

entspricht dabei etwa sieben Programmzeilen. Außerdem können als Zubehör 4 Speichermodule mit je 64 Registern erworben werden. Auch diese lassen sich wahlweise als Speicher oder als Programmzeilen verwenden.

	Verwendung als	
Zahl der Steckmodule	Speicherregister	Programmzeilen
0 (Grundausstattung)	63	440
1	127	880
2	191	1320
3	255	1760
4	319	2200

Ein solcher Steckmodul mit seiner Verdoppelung des Speicherplatzes könnte - bei begrenzten Geldmitteln - fürs erste den Kartenleser ersetzen.

10.1 Das Tastenfeld

Oben unter dem Anzeigefeld befinden sich vier Drucktasten: ON, USER, PRGM und ALPHA. Einmaliges Drücken schaltet den Bereich ein, nochmaliges Drücken wieder aus. Zu Beginn sei nur ON (= ein) gedrückt. Das Tastenfeld ist damit betriebsbereit. Wie die anderen HP-Rechner verwendet auch der HP-41 C die Umgekehrte Polnische Notation (UPN, siehe Kapitel 1.3).

Es ist

$3 + 4$	5×6	$\frac{4}{7}$
3	5	4
ENT	ENT	ENT
4	6	7
+ (= 7)	x (= 30)	÷ (= 0.5714)

Direkte Funktionen (mit weißem Aufdruck auf der Taste) benötigen zur Ausführung lediglich die Schrittfolge

einzugebende Zahl
Druck auf die Funktionstaste

Also z.B.

$\sqrt{3}$	$\frac{1}{7}$	sin 45°
3 $\sqrt{x}$ (= 1.7321)	7 1/x (= 0.1429)	45 SIN (= 0.7071)

Die Funktionen in goldener Schrift jeweils oberhalb der Tasten werden mit der goldenen Vortaste - hier f genannt - aufgerufen. Hier ist die Reihenfolge

einzugebende Zahl
goldene Vortaste (f)
Funktionstaste

Es ist z.B.

12^2	$10^{2.5}$	3.2^5
12 f x^2 (= 144)	2.5 f 10^x (= 316.23)	3.2 ENT 5 f y^x (= 335.54)
e^7	3 π	Anzeige von 3 π mit
7 f e^x (= 1097)	3 f π x (= 9.42)	vier Dezimalen nach dem Punkt f FIX 4 (= 9.4248)

Das Speichern und Rückrufen von Daten geschieht in der gewohnten Weise (vgl. Kapitel 2.7). Es ist z.B. die Zahl 33 in Speicher 5 abzuspeichern.

```
  33
STO 05
```

Nach dem Löschen der Anzeige mit ← ist die Zahl wieder zurückzuholen.

```
RCL 05   (= 33).
```

10.2 Das Alpha-Register

Wird die Taste ALPHA - im folgenden AL genannt - gedrückt, so läßt sich die blaue Tastenbelegung aufrufen.

So läßt sich z.B. "START" eingeben als

```
AL  S T A R T
```

Ein ← löscht das jeweils letzte Zeichen, ein f CLX alles. Auch ein AL AL← löscht den gesamten Ausdruck.

Hierzu einige Beispiele:

1. APPEND

Als APPEND (hänge an) ist in Stellung Alpha f XEQ zu drücken.

Es werde in Stellung ALPHA die Zeichenkette "Start" eingegeben und durch Drücken von ALPHA abgeschlossen, also

```
AL  S T A R T  AL
```

Nun soll die Zeichenkette " , Spieler 1" angehängt werden.

Ein direktes Eintasten würde die erste Kette zerstören. Daher

ist einzutasten

AL f XEQ , S P I E L E R f 1 AL

2. ASTO, ARCL

Die Zeichenkette "Start," soll in Register 7 gespeichert werden. Die vorige Zeichenkette stehe noch in der Anzeige.

In Stellung ALPHA sind dies die Schritte

f STO 07 (also ASTO 07).

Nach f ← (also CLA) ist die Zeichenkette wieder zurückzuholen.

f RCL 07 (= Start,) (also ARCL 07)

Ein weiteres ARCL hängt die Zeichenkette an eine eventuelle Kette in der Anzeige an.

f RCL 07 (= Start, Start,).

Ein weiteres Beispiel:

Die Zeichenkette "Strom:" ist in Register 00 zu speichern, die Kette " 22 A" in Register 01.

In Stellung ALPHA:

S T R O M :
f STO 00

Nach f ← wird gedrückt

Space Space f 2 f 2 Space A (= 22 A)
f STO 01

Nach f ← ist der Inhalt beider Speicher wieder zu zeigen

```
f   RCL   00
f   RCL   01     (S T R O M :   22 A)  .
```

Hierbei kann ein Register nur bis zu sechs Alphazeichen speichern. Beim Versuch, die Zeichenkette Space Space Space f2 f2 Space A im Register 01 zu speichern fällt also A heraus.

Es werde z.B. in Stellung ALPHA eingegeben "Rechnen macht oft Freude" mit SPACE (= Abstand) für die Zwischenräume. Bei Eingabe des letzten Buchstabens ertönt ein Signal: Das Alpha-Register ist mit 24 Zeichen voll besetzt. Bei Eingabe weiterer Zeichen fallen die ersten Zeichen heraus. In Stellung ALPHA zeigt der Befehl f VIEW (= Blick) die ganze Zeichenkette.

Weitere Zeichen lassen sich in Stellung ALPHA mit der goldenen Vortaste f einschreiben. So ergibt die obere Reihe f Σ+ bis f LN die kleinen Buchstaben a bis e, die Zahlentasten f 0 bis f 9 auch in Stellung ALPHA die Zahlen 0 bis 9.

Das Tastenfeld bildet in Stellung ALPHA mit der goldenen Vortaste f die Zeichen bzw. Funktionen

a	b	c	d	e
Σ	%	≠	<	>
	APPEND	ASTO	ARCL	BST
↑		✲	∅	CLA
−	7	8		9
+	4	5		6
x	1	2		3
÷	0	.		AVIEW

Es ist hierbei

Tastendruck	Funktion	Bedeutung
f XEQ	APPEND	hänge an die Alpha-Zeichenkette an
f STO	ASTO	speichere die "
f RCL	ARCL	hole die " zurück
f ←	CLA	lösche die "
f VIEW	AVIEW	zeige die "

10.3 Die Standardfunktionen

Der 41 C enthält im Katalog 3 ein Verzeichnis aller verfügbaren Funktionen. Das Verzeichnis läßt sich mit f CATALOG 3 aufrufen. In der Anzeige erscheint alphabetisch geordnet in rascher Folge

```
+
-
.
.
.
X↑2
Y↑X
```

Die Ausgabe läßt sich stoppen durch ein R/S. Dann kann durch wiederholtes SST (single step, Einzelschritt) die Ausgabe schrittweise fortgesetzt werden. Also z.B.

Tastendruck	Anzeige
f CATALOG 3	+
	-
	.
	.
R/S	FC?C
SST	FIX
SST	FRC
SST	FS?
.	.
.	.
.	.

Entsprechend ruft der Befehl f BST (back step, Rückwärtsschritt) die jeweils vorige Funktion auf, also

f BST	FRC
f BST	FIX
f BST	FC?C
.	.
.	.
.	.

Etwa die Hälfte dieser Funktionen sind fest an eine Taste gebunden. Sie lassen sich wie besprochen verwenden.

Die anderen eingebauten Funktionen werden mit XEQ (execute, ausführen) über das Alpha-Register aufgerufen. Es sind dies die folgenden Funktionen:

Funktions-bezeichnung	Abkürzung von	Bedeutung
ABS	absolute	Absolutwert, Betrag
ADV	advance	Papiervorschub bei angeschlossenem Drucker
AOFF	alpha off	Alpha-Modus abschalten
AON	alpha on	Alpha-Modus einschalten
ASHF	alpha shift	Alpha-Anzeige um sechs Zeichen nach links verschieben
CLD	clear display	Löschen der Anzeige (im Programm)
CLP	clear program	Löschen des Programms
CLRG	clear registers	Löschen aller Speicherregister
CLST	clear stack	Löschen des Stack (Registerstapel)
COPY	copy	Kopieren von Programmen
D - R	degrees to radians	Grad → Radiant-Umwandlung
DEC	decimal	Oktal→Dezimal-Umwandlung
DEG	degrees	Grad-Modus
DEL	delete	Löschen von Programmzeilen
DSE	decrement and skip if equal	Ziehe eins ab und überspringe falls gleich
END	end	Ende des Programms
E ↑ X - 1	$e^x - 1$	$e^x - 1$ (für x nahe Null)
FACT	factorial	Fakultät $n! = 1 \cdot 2 \cdot 3 \cdots n$
FC?	flag clear?	Flag gelöscht?
FC?C	flag clear? clear	Flag gelöscht? Löschen
FRC	fraction	Dezimalteil einer Zahl
FS?C	flag set? clear	Flag gesetzt? Löschen
GRAD	grads	Neugrad-Modus

HMS	hours minutes seconds	Umwandlung von Stunden (dezimal) in Stunden, Minuten, Sekunden
HMS+	" +	Addition von h, min, s
HMS-	" -	Subtraktion von h, min, s
HR	hours	Umwandlung von " im Stunden (dezimal)
INT	integer	Ganzzahliger Teil einer Zahl
LN 1 + X	ln(1+x)	ln(1+x) (für x nahe Null)
MEAN	mean	Mittelwert
MOD	modulo	Modulo-Funktion (Rest)
OCT	octal	Dezimal → Oktal-Umwandlung
PACK	pack	Dichter packen des Programmspeichers
%CH	% change	Prozentualer Unterschied
PROMPT	prompt	Ausgabe von Zeichenketten
PSE	pause	Programm-Pause (Anzeigedauer ca 1 s)
R ↑	roll up	Rolle den Stack hoch
R - D	radians to degrees	Radiant → Grad-Umwandlung
RAD	radians	Radiant-Modus
RND	round	Runden
SDEV	standard deviation	Standardabweichung
Σ REG	sum of registers	Definieren des Statistikblocks
SIGN	sign	Vorzeichen von x
SIZE	size of data register allocation	Speicherregisterzuweisung
TONE	tone	Tonhöhe des Summers
X ≠ 0?	wie nebenstehend	Vergleichsprüfung wie nebenstehend
X < 0?	"	"
X ≤ 0?	"	"
X > 0?	"	"
X ≠ Y?	"	"
X < Y?	"	"
X < >	x ⇄	Vertausche X- und Speicherregister

Zur Darstellung dieser inneren Funktionen dienen im ALPHA-Modus die blauen Buchstaben sowie die mit der goldenen Verschiebetaste aufrufbaren Zeichen (im Schaubild weiter vorn gezeigt). Die Ausführung erfolgt mit XEQ (execute, ausführen) mit der Tastenfolge

Eingabewert

XEQ Einschalten von Alpha Funktionsname Ausschalten von Alpha

Beispiel:

Es soll der Dezimalteil von π gebildet werden. Nach der Tabelle ist die wirksame Funktion FRC. Die Tastenfolge ist

f π

XEQ AL FRC AL (= 0.1416)

Die Abkürzung AL bedeutet hier und im folgenden Drücken der Alpha-Taste. Die Zeichen zwischen den beiden AL sind dabei einzeln einzutasten.

Beispiel:

Es soll 7! gebildet werden, also das Produkt 1 x 2 x 3 x ... x 7. Die entsprechende Funktion heißt im HP-41 C FACT. Die Tastenfolge ist

7

XEQ AL FACT AL (= 5040)

Beispiel:

Es soll für x=1 die innere Funktion ln (1+x) aufgerufen werden. Im Normal-Modus ist der natürliche Logarithmus als Taste aufrufbar, also

2

LN (=0.6931)

Der Aufruf der inneren Funktion erfolgt als

$$\text{XEQ} \quad \text{AL} \quad \text{LN} \quad f\,1 \quad f + \quad \text{X} \quad \text{AL} \quad (= 0.6931).$$

Beispiel:

Es werde die Zahl 78 im Register 13 gespeichert. Sie soll durch Vertauschen mit dem X-Register zurückgeholt werden. Die gesuchte Funktion ist also $X \rightleftarrows R13$.

Zuerst wird getastet

78
STO 13

Nun wird die innere Funktion $X \rightleftarrows$ aufgerufen.

Wie auf dem Schaubild gezeigt, werden im ALPHA-Modus mit f COS und f TAN die Zeichen < und > aufgerufen zur Darstellung von $\rightleftarrows$.

$$\text{XEQ} \quad \text{AL} \quad \text{X} \quad f\,\text{COS} \quad f\,\text{TAN} \quad \text{AL}$$
$$13 \qquad\qquad (= 78).$$

Das etwas mühevolle Eintippen der inneren Funktionen läßt sich mit f ASN (assign, zuordnen) für häufiger gebrauchte Funktionen vereinfachen. Dadurch ordnet der Rechner im USER-Modus die gewünschte innere Funktion - oder auch ein eigenes Programm - der gewählten Taste zu. Günstige Tasten sind z.B. die oberen beiden Reihen, nicht so sehr das Zahlenfeld. Hier würde bei Zahleneingabe im USER-Modus gleich die Funktion aufgerufen.

Beispiel:

FRC ist im USER-Modus der Taste $\Sigma+$ zuzuordnen.

In Normalstellung wird gedrückt

$$f \quad \text{ASN} \quad \text{AL} \quad \text{FRC} \quad \text{AL} \quad \Sigma +$$

Nun wird auf USER umgeschaltet und gedrückt $\Sigma+$.

Zur Kontrolle erscheint beim Drücken und Halten von $\Sigma+$ FRC, dann führt der Rechner die Funktion aus. Bei längerem Drücken schaltet der Rechner die Funktion jeweils ab, es erscheint die Anzeige NULL.

Mit f FIX 4 wird zur Kontrolle von FRC auf der Taste $\Sigma+$ eingegeben

Eingabe (auf Taste $\Sigma+$)	Anzeige (FRC)
7.2486	0.2486
f π	0.1416
1 f e^x	0.7183
⋮	⋮

Nach Wegschalten des USER-Modus erscheint beim Drücken der Taste $\Sigma+$ auch in der Anzeige wieder $\Sigma+$.

Zum Löschen der USER-Tastenbelegung wird im Normalmodus gedrückt

f ASN AL AL Funktionstaste

also hier

f ASN AL AL $\Sigma+$

In Stellung USER zeigt das Display beim Gedrückthalten der Taste $\Sigma+$ wieder $\Sigma+$.

10.4 Programmieren mit dem HP-41 C

Wie erwähnt verfügt der 41 C in der Grundausstattung über einen Speicher für wahlweise 63 Register oder 440 Programmzeilen. Sieben Programmzeilen belegen also nur soviel Platz wie ein Speicherregister. Mit der Funktion SIZE läßt sich die Zahl

der verfügbaren Speicherregister festlegen. Es sollen z.B. 40 Register verfügbar bleiben, die übrigen 23 ergeben dann etwa 160 verfügbare Programmzeilen. Die Tastenfolge

XEQ AL SIZE AL 040

legt den Speicherbereich fest.

Zur Probe wird getastet

45
STO 39

dann nach Löschen der Anzeige mit ←

RCL 39 (= 45.00)

Dagegen erscheint auf STO 40 die Anzeige NONEXISTENT.

Die Zählung mit dem Speicherregister Nr. 00 beginnend wäre dies auch der Speicher Nr. 41.

Die Tastenfolge bei der Eingabe eines Programms ist ähnlich der bei den übrigen HP-Rechnern. Zunächst geht der Rechner mit f GTO .. an den Beginn des Programmbereichs. Hier wird in Stellung ALPHA der Programm-Name eingegeben, zweckmäßig etwa drei Buchstaben. Das Eintasten der Schrittfolge geschieht ebenso wie bei den übrigen HP-Rechnern. Ein wichtiger Fortschritt: Zahlen belegen als Ganzes jeweils nur eine Programmzeile. So wird z.B. $1.023456789 \cdot 10^{-17}$ als eine Zeile gespeichert, bei den übrigen HP-Rechnern immerhin als fünfzehn Zeilen. Nach Eingabe der Tastenfolge wird erneut gedrückt f GTO.. Der Rechner behandelt das Programm als abgeschlossen. Er fügt selbsttätig eine END-Anweisung ein. Diese entspricht etwa dem RTN bei den anderen Rechnern. Sie besagt "Ende des Programmbereichs unter dem vorstehenden Programm-Namen". Anschließend wird mit f ASN das Programm einer Funktionstaste zugeordnet.

Beispiel:

Die eingegebene Zahl ist zu verdoppeln (vgl. Beispiel 2.1).

In Stellung PRGM wird eingetastet

	f GTO ..	Der Rechner geht an den Anfang des Programmbereichs
f LBL	AL DOP AL	Programm-Name "DOP"
	2	Die eingegebene Zahl wird verdoppelt
	x	
	f GTO ..	Ende des Programms. Der Rechner fügt eine END-Anweisung ein.

Das Programm sei der $\frac{1}{x}$-Taste zuzuordnen.

In Normalstellung:

f ASN AL DOP AL $\frac{1}{x}$

In Stellung USER wird gedrückt f FIX 0 , dann 1 und wiederholt die $\frac{1}{x}$-Taste. Beim Gedrückt-Halten erscheint erwartungsgemäß DOP , dann NULL.

Mit der Funktion f CATALOG 1 zeigt der Rechner der Reihe nach die gespeicherten Programm-Namen und die jeweilige END-Anweisung. Wie bei f CATALOG 3 hält nach Drücken von R/S die Ausgabe an. Nach Drücken von SST zeigt der Rechner das jeweils nächste Programm. Mit f BST geht der Rechner jeweils um eine Anzeige zurück. Ein ← beendet die unterbrochene Ausgabe.

Beispiel:

Der Programmspeicher enthalte mehrere Programme mit den Labeln DREI, DOP, LNA, SORT, ... Die Anzeige des X-Registers sei 0.23.

In Normalstellung zeigt der Rechner:

Tastendruck	Anzeige
f CATALOG 1	DREI
	END
	DOP
	END
R/S	LNA
SST	END
SST	SORT
SST	END
f BST	SORT
f BST	END
f BST	LNA
.	.
.	.
←	0.23 (X-Register)

Aufsuchen eines Programms

Enthält der Programmspeicher mehrere Programme, so muß das gewünschte Programm aufgerufen werden (der Befehl f GTO · nnn führt nur im vorliegenden Programm herum).

Ist das Programm mit f ASN einer Funktionstaste zugeordnet, so genügt in Stellung USER die Tastenfolge

Eingabewert
Funktionstaste .

Allgemein wird ein Programm aufgerufen mit der Tastenfolge

Eingabewert
XEQ AL Programm-Name AL

Eine andere Möglichkeit ist, mit f CATALOG 1 das Programmverzeichnis wiederzugeben und das gewünschte Programm einzustellen. Nach dem Umschalten auf PRGM ist der gesuchte Programmbereich sichtbar.

Erreichen von Programmzeilen

In Stellung PRGM kann der Benutzer jede Zeile im Programm mit f GTO · nnn (nnn = dreistellige Zahl) aufrufen. Mit wiederholtem SST bzw. f BST läßt sich das Programm vorwärts und rückwärts durchtasten.

Ändern von Programmen

In Stellung PRGM lassen sich neue Programmzeilen einfach eintasten, andere mit ← problemlos entfernen.

Beispiel:

Im Programm DOP soll statt des Verdoppelns der Eingabe jede Zahl durch 37 geteilt werden. Das Programm sei weiterhin auf Taste $\frac{1}{x}$ aufzurufen.

In Normalstellung wird auf USER geschaltet und getastet

7
Taste $\frac{1}{x}$ (verdoppelt: = 14)

Nach Umschalten auf PRGM zeigt die Anzeige

REG nn

Es wird getastet

f BST
f BST
←
←
37
+

Das Programm hat nun die Gestalt

```
LBL  DOP
  37
  +
  END
```

In Normalstellung im USER-Bereich wird getastet

Eingabe (auf $\frac{1}{x}$)	Anzeige
37	1
74	2
370	10
⋮	⋮

Löschen von Programmteilen

Ein Programmteil, also eine Reihe Programmzeilen, läßt sich im Prinzip auch mit wiederholtem ← löschen. Bei einer größeren Zeilenzahl empfiehlt sich die DEL-Funktion (delete, durchstreichen). In Stellung PRGM wird die Anzeige mit f GTO · nnn oder SST auf die erste zu entfernende Programmzeile gesetzt, die DEL-Funktion aufgerufen und die Zahl der zu entfernenden Zeilen dreistellig eingegeben (z.B. 9 Zeilen als 009), also

XEQ AL DEL AL dreistellige Zeilenzahl

Ist die angegebene Zahl größer als die Zeilenzahl bis zum Programmende, so wird nur bis zum Programmende END gelöscht.

Beispiel:

Im Programm DOP sind einschließlich Schritt 2 alle Schritte zu löschen.

In Normalstellung wird gedrückt

XEQ AL DOP AL
f GTO · 001

Nun wird auf PRGM geschaltet. Die Anzeige enthält LBL DOP, also das gewünschte Programm. Es wird getastet

SST
XEQ AL DEL AL 088

Also 88 Schritte seien zu entfernen. Das Programm besteht jetzt nur noch aus

```
LBL  DOP
  END
```

Löschen von ganzen Programmen

Ganze Programme lassen sich mit der Funktion CLP (clear program, lösche das Programm) aus dem Programmspeicher entfernen. In Normalstellung ist dies die Tastenfolge

XEQ AL CLP AL AL Programm-Name AL

Eine etwaige Zuordnung zum Tastenfeld mit ASN wird zugleich aufgehoben.

Beispiel:

Das Programm DOP ist zu löschen.

In Normalstellung ist zu tasten

XEQ AL CLP AL AL DOP AL

Beispiel 10.1: Umrechnen von Tagen in Wochen

Tage sollen in Wochen umgerechnet werden. Hierbei sollen als Ergebnis in der Anzeige die Zahl der Wochen links vom Dezimalpunkt stehen, die verbleibenden restlichen Tage rechts. Z.B. sind 9 Tage darzustellen als 1 Woche 2 Tage, also 1.2 . Im Beispiel werden die Funktionen INTEGER und FRACTIONAL verwendet. Die Schrittlisten sind für den HP 67 und den HP-41 C angegeben.

HP 67		HP-41 C
		f GTO ··
f LBL C		f LBL AL WOC AL
7		7
+		+
f INT		XEQ AL INT AL
STO 0	Zahl der Wochen	STO 00
h LST X		f LAST X
g FRAC		XEQ AL FRC AL
7		7
x	Restzahl der Tage	x
1		10
0		+
+		RCL 00
RCL 0		+
+		f GTO ··
h RTN		Sodann in Normalstellung

f ASN AL WOC AL $\sqrt{x}$

HP 67	HP-41 C
Stellung RUN	Stellung USER
DSP 1	f FIX 1
Tage	Tage
Taste C	Taste $\sqrt{x}$

Tage	Wochen. Tage
13	1.6
29	4.1
333	47.4
·	·
·	·

Im Beispiel erscheint der HP-41 C schwerfälliger als der HP 67. Dafür bleiben die Programme nach dem Ausschalten auch ohne Kartenleser erhalten.

In entsprechender Weise lassen sich die Programm-Beispiele 2.2 bis 2.30 des vorderen Buchteiles eingeben.

Programm-Marken

Die Programm-Marken umfassen zwei Gruppen: die bereits verwendeten Alpha-Marken sowie die numerischen Marken und lokalen Alpha-Marken.

Alpha-Marken bezeichnen Programmblöcke, die aus einem Hauptprogramm und ggf. aus mehreren Unterprogrammen bestehen. Bei den Alpha-Marken sind mehrere Buchstaben zu einem Programm-Namen zusammengefaßt, wie z.B. TEST , DOP , SPAR oder WOC . Der Programmblock wird durch f GTO·· mit einer End-Anweisung abgeschlossen.

Numerische Marken und lokale Alpha-Marken bezeichnen Unterprogramme innerhalb eines Programmblocks. Numerische Marken sind die 100 Zahlen 00 bis 99. Lokale Marken sind im Alpha-Modus die Buchstaben A bis J sowie f a bis f e. Die Unterprogramme sind jeweils mit f RTN abzuschließen.

Beim Aufruf einer der regulären Alpha-Marken prüft der Rechner den gesamten Programmspeicher. Bei den numerischen Marken und lokalen Alpha-Marken wird nur das in Stellung PRGM sichtbare Hauptprogramm durchgesehen. Dadurch lassen sich in jedem Hauptprogramm dieselben numerischen Marken verwenden.

Aufruf von Programmen in Normalstellung

Programme mit eigenem Namen werden in Normalstellung aufgerufen mit

XEQ AL Programm-Name AL

Enthält das Programm Unterprogramme, so können diese gleichfalls aufgerufen werden mit

XEQ Programm-Nr.

Dabei ist vorausgesetzt, daß das Hauptprogramm in der Anzeige steht.

Beispiel:

Zur Eingabe soll jeweils eins hinzugezählt werden, das Ergebnis ist zu quadrieren. Das Quadrieren soll auch durch Aufruf eines Unterprogramms erfolgen.

In Stellung PRGM wird getastet

```
                f GTO··
f LBL   AL   LERN   AL
                1
                +
-----------------------
                f LBL 01
                 f x²
                f GTO··
```

In Normalstellung wird eingegeben

```
f   ASN   AL   LERN   AL   LN
```

dann in Stellung USER

```
   3
  LN          (= 16)
```

anschließend

```
   3
XEQ 01        (=  9)
```

Aufruf von Programmen in Programmen

Ein Aufruf mit XEQ kann sowohl vom Tastenfeld aus wie auch durch eine Programmzeile des Hauptprogramms erfolgen.

Z.B. soll das Programm LERN ein XEQ 01 enthalten.

In Stellung PRGM wird getastet

```
f GTO · 001
  XEQ 01
   SST
   SST
  f RTN
```

Das Programm lautet damit

```
1         f LBL  LERN
             XEQ 01
               1
               +
5            f RTN
          f LBL 01
            f x²
8           END
```

In Stellung USER wird gedrückt

```
3
LN          (= 10)
```

Das Unterprogramm wird also ausgeführt, dann setzt der Rechner die Ausführung des Hauptprogramms fort. Im aufgerufenen Unterprogramm kann wieder ein XEQ Programm-Marke stehen, in diesem wieder eins und so fort. Bis zu sechs XEQ Programm-Marken können so ineinander verschachtelt stehen.

Sprung zu Programmen

Der Sprung zu Programmen erfolgt durch den Befehl GTO Programm-Marke. Diese kann eine Alpha-Marke, eine lokale Alpha-Marke oder eine numerische Marke sein. Siehe hierzu auch den vorderen Buchteil, Kapitel 3.

Beispiel:

Im Programm LERN soll nur das Unterprogramm ausgeführt werden. In Stellung PRGM wird getastet

```
f GTO · 002
      ←
 f GTO 01
```

Das Programm lautet jetzt

```
f LBL  LERN
 f GTO 01
     1
     +
  f RTN
 f LBL 01
  f x²
   END
```

Anschließend wird in Stellung USER gedrückt

```
   3
  LN          (= 9)
```

Ersichtlich erfolgt im Programmschritt 2 der Sprung zum Unterprogramm 01. Dieses wird ausgeführt. Die Programmausführung endet dort.

Bei Verwendung von Programmaufruf und Sprung lassen sich die Beispiele des vorderen Buchteils 3.1 bis 4.10 rechnen.

Steuerung von Programmschleifen (DSE und ISG)

Die Funktion DSE (decrement and skip if equal, ziehe ab und spring falls gleich) und ISG (increment and skip if greater, zähle zu und spring falls größer) steuern die Zahl der zu durchlaufenden Programmschleifen. Dazu wird eine Zahl der Form iiiii.fffcc in einem beliebigen Register gespeichert und durch DSE Register-Nr. bzw. ISG Register-Nr. aufgerufen. Es ist hierbei

iiiii = Anfangswert der Schleifenzahl, ein- bis fünfstellig
fff = Endwert der Schleifenzahl, dreistellig
cc = Schrittweite. Falls nichts eingegeben wird (cc = 0) setzt der Rechner die Schrittweite 1.

Beispiel 10.2: Überprüfen der Schleifenzahl

Eine Schleife soll zehnmal durchschritten werden, die Funktion DSE und Register 5 sind zu verwenden. Die Zahl der Durchgänge ist anzuzeigen.

In Stellung PRGM:

```
                        f GTO··
1              f LBL  AL  LOOP  AL
                        STO 05
                          0
                        STO 00
5                       f LBL 01
                          1
                        STO + 00
                  XEQ  AL  DSE  AL  05 ┐
ja, Reg 5 ≠ 0 ↓         f GTO 01      │   nein, Reg 5 = 0
10                       RCL 00       ↵
                        f GTO··
```

In Normalstellung wird das Programm der LOG-Taste zugewiesen, also

```
f  ASN  AL  LOOP  AL  LOG
```

Eingabe DSE	Zahl der Durchläufe durch LBL 01
5	5
10	10
20	20
⋮	⋮

Nun sollen die Funktion ISG und Register 7 verwendet werden.

In Stellung PRGM:

```
f GTO · 002
     ←
  STO 07
f GTO · 008
     ←
XEQ  AL  ISG  AL  07
```

Eingabe ISG	Zahl der Durchläufe durch LBL 01
.004	5
.009	10
.019	20
:	:

Schließlich seien mit DSE 20 Schleifen der Schrittweite 2 zu durchlaufen, der Startwert sei 50, Speicherung in Register 8.

In Stellung PRGM:

```
f GTO · 002
     ←
  STO 08
f GTO · 008
     ←
XEQ  AL  DSE  AL  08
```

In Stellung USER wird gedrückt

```
50 . 01002
   LOG          (= Eingabe ins Programm LOOP,
                   Ausgabe = 20).
```

Indirekte Adressierung

Die indirekte Adressierung (siehe Kapitel 6) läßt sich beim HP-41 C mit allen Registern durchführen. Sie wird u.a. zum Abspeichern von Zahlen in den Registern R 100 und höher benötigt. Die Tastenfolge STO f Register-Nr. (f = goldene Verschiebetaste) bewirkt ein Abspeichern in einem Register, dessen Adresse im genannten Register enthalten ist, also

```
Erste Zahl
STO  Register-Nr.
Zweite Zahl
STO  f Register-Nr.
```

Beispiel:

Die Zahl 13 soll in Register 7 abgespeichert werden. Die indirekte Adresse 7 soll Register 3 entnommen werden.

```
  7
STO 03          Die Register-Nr. 7 steht jetzt in
                Register 3
   13
STO f 03        Die Speicheradresse 7 wird Register 3
                entnommen.
```

Anschließend wird gedrückt ←

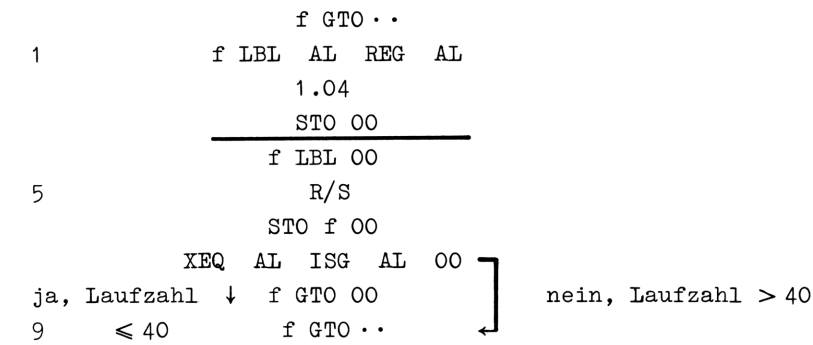
```
RCL 07          (= 13) .
```

Das folgende Programm verwendet die indirekte Adressierung.

Beispiel 10.3: Abspeichern von Zahlen

Bei eingesetzten Speichermoduln sind Zahlen in den Zusatzregistern R 100 bis R 200 abzuspeichern. Die Zahlen sind mit R/S einzugeben. Bequem läßt sich die Funktion ISG verwenden, deren ganzzahliger Teil als Registeradresse Verwendung findet. Für eine einfache Kontrolle sind fürs erste die Register 01 bis 41 zu belegen. Register 00 diene der Funktion ISG.

In Stellung PRGM wird getastet



In Normalstellung wird gedrückt

XEQ AL REG AL

dann eingegeben

11
R/S
22
R/S
:
66
R/S
:

Mit RCL 01 bis RCL 06 werden die Registerinhalte nachgewiesen.

Sind z.B. drei Speichermodule eingesetzt (also insgesamt 255 Register), so ist mit SIZE der Speicherbereich neu zu definieren.

In Normalstellung:

XEQ AL SIZE AL 200

dann

f GTO · 002

sowie in PRGM

←
100.199

Nun lassen sich Zahlen der Reihe nach in den Registern 100 bis 200 abspeichern.

Programmieren mit Alphaketten

Alphaketten, also Worte, bei der Ein- und Ausgabe machen Programme in der Bedienung sicherer. Wer den Mehraufwand beim Programmieren nicht scheut, wird feststellen, die Programme werden auch bequemer.

Die einzelne Programmzeile kann bis zu 15 Alphazeichen enthalten. Mit der APPEND-Funktion AL f XEQ AL lassen sich 9 weitere Zeichen aus der nächsten Programmzeile anfügen. Die Zeichenketten werden durch die Anweisung PROMPT in die Anzeige geholt, das Programm hält an. Die gleiche Wirkung hat übrigens ein f VIEW im Alpha-Modus vor einem Programm-Stop. Zeichenketten aus den Registern werden mit f RCL Register-Nr. zurückgeholt, aus dem Stack mit f RCL Stackregister-Name (X, Y, Z oder T).

Für den Anfänger verwirrend ist, daß beim Durchtasten eines Programms in Stellung PRGM sich Befehle wie XEQ AL PROMPT AL nur wenig von einfachen Alphaketten wie AL PROMPT AL unterscheiden. Diese sind durch ein hochgestelltes T (= Text) vor Beginn der Zeichenkette gekennzeichnet, also beim Durchtasten

```
   :
ARCL 07
PROMPT          Befehl
TPROMPT         einfacher Text
   :
```

Beispiel 10.4: Integer und Fractional

Der ganzzahlige Teil und der Dezimalteil einer Eingabe sind zu bilden und anzuzeigen.

In Stellung PRGM wird getastet

```
        f GTO ··
f  LBL  AL  INFR  AL
        AL  ZAHL?  AL
XEQ  AL  PROMPT  AL          Eingabe der Zahl, dann
                             Drücken von R/S
        ENT
XEQ  AL  INT  AL
     AL  INT=
   f  RCL · X  AL
XEQ  AL  PROMPT  AL          Programmstop und Anzei-
                             ge des ganzzahligen
                             Teils, dann Drücken
                             von R/S
   f  LAST  X
XEQ  AL  FRC  AL
     AL  FRC=
    f  RCL · X
    f  VIEW  AL              Anzeige des Dezimal-
                             teils
    f GTO ··
```

In Normalstellung wird getastet

```
f  ASN  AL  INFR  AL     x ⇄ y
```

In Stellung USER ist damit die Funktion der Taste $x \rightleftarrows y$ zugeordnet. Mit f FIX 4 wird getastet

Eingabe	INT=	FRC=
11.2286	11.0000	0.2286
f π	3.0000	0.1416
1 f e^x	2.0000	0.7183
:	:	:

Sachregister

Ableitung einer Funktion 96
Ändern eines Programms 37, 39, 219
Alphaketten 231
Alpha-Register 206
APPEND 206, 231
Arbeitsbereich 23
ARCL 207
ASN 214
ASTO 207
Aufruf von Programmen 92, 223, 224
Aufzeichnen von Programmen 78

Beschriften von Magnetkarten 79
Bruttopreise 41

Cassinische Kurven 65
CATALOG 1 217
CATALOG 3 209
CLP 221

Decrement and skip if zero (DSZ) 147
DEL 220
Determinante 54, 93
Diodenkennlinie 72
Diodensperrströme 29
Dreiecksrechnung 102
DSE 226

Einwiegen 63
Ellipsengleichung 52
Elliptisches Integral 151
END 216
Enter 16
Erreichen von Programmzeilen 35
Extremwertefinder 136

Flags 118

Ganzzahligkeit 141
Geradeausprogramm 31, 41
Gesamtwiderstand 62
Gleichungen mit einer Unbekannten 45
Gleichungen mit zwei Unbekannten 52
Gleichungen mit drei Unbekannten 54
Gleichungen mit vier Unbekannten 93
Go to (GTO) 101, 216, 219
Go to subroutine (GSB) 92

Halbleiterwiderstand 73
Histogramm 161
Hyperbelfunktionen 43

Increment and skip if zero (ISZ) 147
Indirekte Adressierung 158, 228
Innere Funktionen 211, 212
Integration nach Simpson 149
Interne Stellenzahl 25
I-Register 91, 147, 158, 162
ISG 226

Klasseneinteilung 161
Klassenstärken 28
Kompakte Dateneingabe 45
Kontaktwiderstände 28

Label 31, 223
Last-X-Register 47
Löschen von Programmen 220, 221
Logarithmische Darstellung 42

Maßstabsänderung 50
MERGE 81, 91
Mittelwert 26
Mondlandespiel 68

Nachtgefecht 161
Newtonsches Näherungsverfahren 132
Nullstellenfinder 132
Numerische Differentiation 152, 155
Numerische Integration 149

Optimale Gerade 70

Plancksche Strahlungsformel 89, 109
Plasmaphysik 83
Primfaktorenzerleger 139
Primzahlenfinder 142
Programm-Marken 31, 223
Programm-Schleifen 126, 129, 226
Programmschritt 34
Programmspeicher 33
PROMPT 231

Ratensparen 56
Raumdiagonale 44
Registerstapel 16
Reißfestigkeit 27
Relative Änderung 48
Runge-Kutta-Verfahren 152, 155
Run/Stop (R/S) 44

Schleifen mit Vergleichsprüfungen 129
Schleifen ohne Bedingung 126
Schrittweises Erhöhen einer Veränderlichen 64
Sekundärregister 69
SIZE 216, 230
Speicherarithmetik 60
Speicherplatz 34
Speicherregister 48
Spirale 64
Spirale die eine Spindel umwikkelt 66
Sprung zu Programmen 101, 225
Stack 16
Stackbelegung 20
Stack-Lift 17, 32, 56, 109
Standardabweichung 26
Standardfunktionen 14, 15, 209
Statistik 26
Steckmodul 204
Strahlungsphysik 87, 112

Tastenaufdruck 14, 15
Tastenfeld 12
TRACE 168

Überlauf 23
Überprüfen des Programms 75
Umgekehrte Polnische Notation (UPN) 16
Umrechnung Celsius-Fahrenheit 42
Umrechnung Gramm-Unzen 51
Umrechnung Inch-Millimeter 50
Unterlauf 23
Unterprogramme 92
USER-Modus 214

Verarbeitbare Stellenzahl 24
Vergleichsprüfungen 106
Verkehrszählung 61
Verschwinden des Nenners 109, 110
VIEW 208, 231

Wertpapierrendite 58
Write/Data (W/DATA) 90

XEQ 213, 231

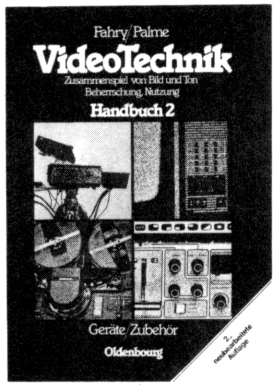
Zahlenmengen 28
Zahlenumwandlung 143
Zinseszinstabelle 55

VideoTechnik

Dieter Fahry / Klaus Palme

Handbuch 1:
Grundlagen/Anwendung
2. neubearbeitete Auflage 1980.
366 S., 219 Abb. 33 Tabellen
DM 122,– ISBN 3-486-21902-2

Problemorientiert und systematisch behandelt dieses Standardwerk alle Fragen, die für den Anwender in der Praxis von Bedeutung sind. Der Leser erfährt, wie Videoaufnahmen entstehen, wie sie bearbeitet, gespeichert und wiedergegeben werden. Er kann die Technik fachgerecht bedienen.

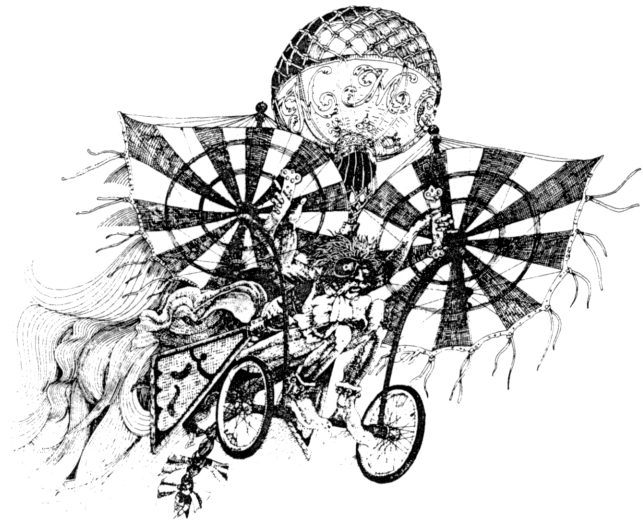


Dieter Fahry / Klaus Palme

Handbuch 2: Geräte/Zubehör
2. neubearbeitete Auflage 1980.
261 Seiten, 184 Datenblätter,
DM 84,– ISBN 3-486-21912-X

Handbuch 2 liefert ergänzend dazu einen sorgfältig klassifizierten und übersichtlichen Geräteteil – eine wichtige Entscheidungshilfe. In der Neuauflage sind alle auf der Funkausstellung 1979 präsentierten Videogeräte berücksichtigt, technische Neuerungen erläutert und in Planung befindliche Systeme verarbeitet, die das Jahr 1981 einschließen.

Oldenbourg



619 Probleme und Fragen

aus der physikalischen Alltagswelt –

zum Lesen, Nachdenken, Diskutieren, Knobeln.

Der fliegende Zirkus der Physik

von J. Walker bringt auf 193 Seiten, mit 222 Bildern, eine einmalige Sammlung physikalischer Phänomene. Teils lustig, teils tiefgründig wird über Blitz und Donner, Sanddüne und Seifenblasen, Sonnenbrillen und Wasserleitungen, Eier und Teetassen, Colaflaschen und Zucker berichtet.

Viel Spaß und viel Belehrung auf
193 Seiten für DM 19,80.
Dazu gibt es ein Lösungsbuch.
112 Seiten, DM 15,80.

Oldenbourg

Programmierbare Taschenrechner sind ein ausgezeichnetes Werkzeug, mit dem sich auch komplexe Rechnungen überraschend einfach lösen lassen. Das Buch wendet sich nun an alle diejenigen, die über einen programmierbaren Taschenrechner verfügen, aber trotz Studium des jeweiligen Bedienungshandbuches die praktische Anwendung weiterhin schwierig finden. Hier soll zum sicheren Bedienen sowie zum einfachen übersichtlichen Programmieren dieser Rechner Hilfestellung gegeben werden. Mit dem Schwerpunkt auf dem HP 67 und 97 werden der HP 25 (für die Rechner HP 33 bis 38) sowie – in einem eigenen Kapitel – der HP-41 C besprochen. Mit einer Vielzahl ausführlicher Programmbeispiele lernt der Anwender schrittweise einfaches und sicheres Programmieren.

Oldenbourg Verlag ISBN 3-486-23492-7