

Anwenderprogramme zum HP-28C/S Pocket Computer



ISBN 3-89374-029-5

Robert Hübner

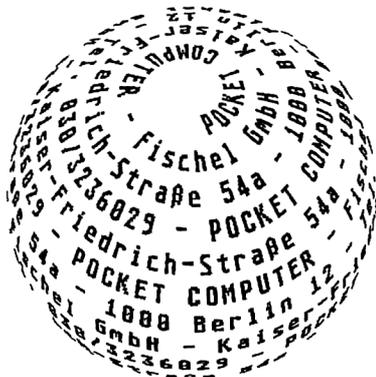
Fischel GmbH

POCKET COMPUTER

FISCHEL GmbH
Zeitschrift für Taschencomputer

durch Information vorn

Kaiser-Friedrich-Straße 54a
1000 Berlin 12
Telefon (030)3236029
HRB 19396 Amtsgericht Charlottenburg



=====
C FISCHEL GMBH

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder teile daraus auf fotomechanischem (Fotokopie, Mikrokopie) oder sonstigem Wege zu vervielfältigen. Es kann keine Haftung für die Richtigkeit der Programme übernommen werden, obwohl sie ausgetestet wurden.

Bankverbindung: Postgiroamt Berlin-West, Bankleitzahl 10010010, Kontonummer 461533-103

Öffnungszeiten: Montag-Freitag 10.00-18.00 Uhr, Samstag 10.00-14.00

Nur wenige Benutzer des Taschencomputersystems HP 28 S/C wissen, welche Möglichkeiten dieses Gerät für sie bereithält. Der HP 28 setzt durch seine Leistung und seine Vielseitigkeit neue Maßstäbe in der Micro-computerentwicklung. Nicht nur die 128 kByte mathematischer Funktionen, sondern auch das problemlose Verarbeiten von Variablen sind einzigartig auf dem Taschencomputersektor.

Dieses Buch soll einen Einblick in die professionelle Nutzung des HP 28 geben. Es bietet mit seinen vielen Programmen, Beispielen und Tips einem erfahrenen Benutzer, als auch einem Einsteiger, wichtige Hinweise zum schnellen Umgang mit diesem Taschencomputer und seiner Programmierung. Ich möchte Ihnen auch meine Erfahrung, die ich durch intensive Nutzung des HP 28 S/C gesammelt habe, weitergeben.

Zu Beginn des Buches gebe ich Ihnen einen kurzen Überblick über die Funktionen und die Benutzung des HP 28. Im weiteren Verlauf erlernen Sie das Programmieren und bekommen, anhand von 40 Programmen, einen Einblick in die Bearbeitung von komplexen Aufgaben und Problemstellungen.

An dieser Stelle möchte ich auch Thomas Hainzl für seine bereitwillige Hilfe und seiner zur Verfügung gestellten Zeit danken.

Nun wünsche ich Ihnen viel Erfolg beim Arbeiten mit diesem Buch und Ihrem Taschencomputer HP 28 S/C.

Robert Hübner

Inhaltsverzeichnis

1.	Der schnelle Umgang mit dem HP 28 S/C	8
1.1.	Zur Beachtung	8
1.2.	Die wichtigsten Befehle des HP 28 S/C	9
1.3.	Tips und Übungen für Anfänger und Fortgeschrittene	18
1.3.1.	Allgemeine Funktionen	18
1.3.2.	Die Verwendung von Variablen	23
1.3.3.	Erstellen von Abbildungen	24
1.3.3.1.	Kurvendiskussion	25
1.3.3.2.	Gleichzeitige graphische Darstellung von Funktionen	30
1.3.3.3.	Abbilden statistischer Daten	33
1.3.3.4.	Erstellen von Bildern	34
1.3.4.	Nullstellensuche bei Funktionen	39
1.3.5.	Lösen von Gleichungssystemen	43
1.3.6.	Integrieren und Differenzieren	44
1.3.7.	Speicher zu klein, was tun?	51
2.	Die Programmierung des HP 28 S/C	53
2.2.	Anzeigen von Strings in Programmen, Meldungen	55
2.3.	Programmschleifen	57
2.4.	Variable in Programmen	60
3.	Programme für den HP 28 S/C	61
3.1.	Graphische Abbildungen	61
3.1.1.	POLAR Kurven in Polarkoordinaten zeichnen	61
3.1.2.	PMTR Kurven in Parameterdarstellung zeichnen	62
3.1.3.	ST.P Neuer Abbildungsmaßstab, doppelte Geschwindigkeit	63
3.1.4.	EundK Ellipsen und Kreise zeichnen	64
3.1.5.	AFvF Allgemeine Form von Funktionen	65
3.2.	Algebraische Programme	72
3.2.1.	Repeat Definierte Wiederholung von Programmen	72
3.2.2.	ZPASS Vollständiges Zusammenfassen eines Terms	72
3.2.3a.	KÜN Binomische Reihe k über n	73
3.2.3b.	SQ R.i Wurzel aus reeller oder imaginärer Zahl	73
3.2.4.	HORS1 Horner Schema für Complex, Re und Parameter	74
3.2.5.	GAUS1 Gaußscher Algorithmus	76
3.2.6.	ROT Rotation eines Vektorfeldes	79
3.2.7.	DIV Divergenz eines Vektorfeldes	81
3.3.	Kurvendiskussion	82
3.3.1.	FZZK Fläche zwischen zwei Kurven	82
3.3.2.	LÄEKU Länge einer Kurve	83
3.3.3.	OFRKÖ Oberfläche eines Rotationskörpers	84
3.3.4.	LKPAR Länge einer Kurve in Parameterdarstellung	84
3.3.5.	ORPAR Oberfläche eines Rotationskörpers	85
3.3.6.	VROT Volumen eines Rotationskörpers	86
3.3.7.	VROT2 Volumen eines Rotationskörpers 2	86

Inhaltsverzeichnis

3.3.8.	NOR1	Normale an Kurve $f(x,y)$	88
3.3.9.	NOR2	Normale an Kurve $f(x)$	89
3.3.10.	TAN1	Tangente an Kurve $f(x,y)$	89
3.3.11.	TAN2	Tangente an Funktion $f(x)$	90
3.3.12.	WIN1	Schnittwinkel zweier Geraden	91
3.3.13.	WIN2	Schnittwinkel zweier Kurven	91
3.3.14.	root	Bestimmung von Nullstellen	94
3.3.15.	SKG	Schnittpunkt Kreis-Gerade	95
3.3.16.	DREH	Drehung des Koordinatensystems	97
3.3.17.	PDIV	Polynomdivision	98
3.4.		Komplexe Rechnung	100
3.4.1.	PADD	Polare Addition	100
3.4.2.	PSUB	Polare Subtraktion	100
3.4.3.	PMUL	Polare Multiplikation	101
3.4.4.	Pdiv	Polare Division	101
3.5.		Matritzen, Felder und Vektoren	101
3.5.1.	VinM	Vektoren in Matrix geben	101
3.5.2.	MinV	Matrix in Vektoren	102
3.5.3.	GLL	Gleichungssystem lösen (inhomogen)	102
3.6.		Fourier Reihe	103
3.7.		Statistik	105
3.7.1.	INIT	Größe einer Statistik-Matrix	105
3.7.2.	FIN	Ende-Programm	106
3.7.3.	GEOM	Geometrisches Mittel	106
3.7.4.	HARM	Harmonisches Mittel	107
3.7.5.	GHARM	Gewichtetes harmonisches Mittel	108
3.7.6.	PQMi	Quadratisches und Potenzmittel	109
3.7.7.	RANGE	Range	110
3.7.8.	GINI	Gini-Maß	111
3.7.9.	VSA	Verschiebungssatz	112
3.7.10.	STAW	Standartabweichung	113
4.		Fehlersuche bei Programmen	114
.....			
5.		Menüorganisation	116
.....			
Anhang			119

EINLEITUNG

Dieses Buch versucht Ihnen die Möglichkeiten des HP 28 S und des HP 28 C zur Bearbeitung komplexer Probleme aus der Mathematik und Physik darzulegen. Es ist jedoch nicht möglich, mit dem Inhalt dieses Buches, dem gesamten Leistungsspektrum des HP 28 zu entsprechen.

Differentiation und Integration numerischer oder symbolischer Funktionen, die Untersuchung eines Terms auf Nullstellen und Extremwerte oder die Berechnung von Ableitungen, die graphische Darstellung auch komplexer Funktionen und das vollständige Lösen von Differentialgleichungen sind nur einige Beispiele der Leistungsfähigkeit dieses Computers.

Das Buch gliedert sich im Wesentlichen in zwei Teile. Der erste Teil umfaßt eine Wiederholung der Funktionen und Befehle des HP 28 und gibt wichtige Hinweise und Tips über ihre Verwendung. Der zweite Teil führt Sie in die Programmierung Ihres Taschencomputers ein und vermittelt anhand von 40 Programmen einen tieferen Einblick in das systematische Bearbeiten von einfachen bis zu sehr komplexen mathematischen oder physikalischen Problemen.

Der Benutzer dieses Buches sollte mit der Bedienung und der Programmierung des HP 28 S/C grundsätzlich vertraut sein. Die einzelnen Programmbefehle werden nur dort erklärt, wo es notwendig erscheint. Vergleiche dazu das Referenz- und Benutzerhandbuch zum HP 28.

Die Schreibweise der einzelnen Befehle erklärt sich wie folgt:

SOLV	oder	<u>SOLV-</u>	Menueanwahl des SOLV-Menues
ON			Tastatureingabe von ON
EXP			Aufruf der Exponent-Funktion
POLI			Aufruf eines Programms oder einer Prozedur aus einem
.....			Menue

Die Basis für das Bearbeiten der im Buch enthaltenen Beispiele und Programme ist die Grundeinstellung des HP 28. Abweichungen werden jeweils angegeben. Vergleiche die Seiten 25 und 26 des Referenzhandbuches.

1.1. Zur Beachtung

Grundeinstellung

Die normale Stackanzeige und die Reaktivierung des Tastenfeldes erreichen Sie durch das Drücken von |ON|. Gegebenenfalls wiederholen Sie diese Prozedur.

Systemstopp

Wenn Sie ein Programm anhalten möchten, das nicht auf das Drücken von |ON| reagiert, führen Sie einen Systemstopp durch.

Drücken und halten Sie |ON|, drücken Sie |▲| und geben Sie |ON| wieder frei.

ACHTUNG: Dieser Befehl löscht: Alle unterbrochenen Programme
Alle lokalen Variablen
Das CUSTOM-Menü
Den Stack und alle rückgesicherten Elemente

Memory Reset

Um den gesamten Speicher zurückzusetzen und die Voreinstellung des HP 28 aufzurufen drücken und halten Sie |ON|, drücken und halten Sie |INS| und |▶|, geben Sie |INS| und |▶| frei, geben Sie |ON| frei.

ACHTUNG: Dieser Befehl führt einen Systemstopp durch.

Dieser Befehl löscht: Alle Verzeichnisse und Benutzervariablen
Alle geänderten Modi
Ein Tonsignal und die Anzeige "Memory Lost" werden ausgegeben.

Falsches Anwenden des "SOLVE-Befehls"

Wenn Sie "SOLVE" in einem Programm verwenden, welches die Stacktiefe nicht vergrößert, kann der Stackinhalt zerstört werden. Dies wird durch die Anzeige von "System Objekt" kenntlich gemacht. In diesem Fall führen Sie sofort einen Systemstopp durch.

ACHTUNG: Drücken Sie nicht |DROP| oder ■|CLEAR|, denn dies kann zu einem Memory Reset führen.

Löschen einer leeren Reihe innerhalb einer Befehlszeile

Zum Löschen einer leeren Reihe bringen Sie den Cursor in die betreffende Reihe und drücken Sie |DEL|.

ACHTUNG: Benutzen Sie nicht den Befehl █ |INS|, da dies einen Memory Reset zur Folge haben kann. Sollte dies doch geschehen, drücken Sie sofort |ON| oder |ENTER|.

Zeichnen gebrochen rationaler Funktionen

Bei einigen Bauversionen des HP 28 ist es nur über Umwege möglich, bestimmte gebrochen rationale Funktionen zeichnerisch darzustellen. Solche Funktionen besitzen Unendlichkeitsstellen (senkrechte Asymptoten), über die der HP 28, in der entsprechenden Version, nicht hinwegrechnen kann. Sie umgehen dies, wenn Sie zuerst sicherstellen, daß die Flags 57, 58 und 59 gelöscht sind und Sie folgende Anweisung an Ihren Rechner weitergeben:

█ |PLOT| |NEXT| 0.0000001 *W .

Vergleichen Sie dazu auch Seite 30 des Referenzhandbuches.

1.2. Die wichtigsten Befehle des HP 28 S/C

Algebra

COLCT	Terme zusammenfassen	'X*X*X'	→ 'X^3'
EXPAN	Ausdruck erweitern	'X^3*X'	→ 'X*(X*X)*X'
SIZE	Dimension eines Objekts	'X*X*X'	→ 5
TAYLR	Taylorreihe	$\sum_{i=0}^n \frac{x^i}{i!} \left(\frac{\partial^i}{\partial x^i} f(x) \right) \Big _{x=0}$	
ISOL	Variable isolieren (zB X)	'X^2-Y' 'X'	→ '√Y'
QUAD	Quadratische Form	'X^2-Y' 'X'	→ 's1*√((-4*(-Y)))/2'

Array

→ARRY	Stack in Feld übertragen	a ...a nm	→ [[Matrix]]
POT	Element in Feld	[F1] nm a	→ [F2]

GET	Element aus Feld holen	[Feld] nm	→ a
PUTI	Element in Feld, Index erhöhen	[F1] I1	a → [F2] I2
GETI	Element holen, Index erhöhen		
SIZE	Größe eines Feldes, einer Liste		
TRN	Transponieren einer Matrix	A_{ji}	→ A'_{ij}
RDM	Neuordnung einer Matrix	[F1] nm	→ [F2]
CROSS	Kreuzprodukt	A X B	
DOT	Skalarprodukt	A ° B	
DET	Determinante einer quadrat. Matrix		

Binary

DEC	Dezimaler Modus	, löscht Flag 43 und 44	#9471d
HEX	Hexadezimaler Modus	, setzt Flag 43 und 44	#24FFh
OCT	Oktaler Modus	, setzt Flag 43, löscht 44	#22377o
BIN	Binärer Modus	, löscht Flag 43, setzt 44	#1001...b
R→B	Reell- in Binärwert		
B→R	Binär- in Reellwert		

Complex

R→C	Reell in komplex	a,b → a+bj	a b → (a,bj)
C→R	Komplex in reell		
RE	Reeller Anteil		
IM	Imaginärer Anteil		
CONJ	Konjugiert-komplex		
SIGN	Einheitsvektor in Richtung	(x1,y1)	
R→P	Rechteck- in Polarnotation	(X,Y)	→ (r,φ)
ABS	Absolutbetrag = $\sqrt{X^2 + Y^2}$	(X,Y)	→ r
ARG	Winkel der komplexen Zahl	(X,Y)	→ φ

List

→LIST	Stack in Liste	a,b,...,m n	→ {a,b,...,n}
LIST→	Liste in Stack		
PUT	Element in Liste	L1 nm a	→ L2

GET	Element holen	
SUB	Teilmenge aus Liste holen	{L1} n1 n2 → {L2}

Logs

LOG	Dekadischer Logarithmus	$\log_{10} b = x$
ALOG	Dekadischer Antilogarithmus	$10^x = b$
LN	Natürlicher Logarithmus	$\log_e b = x$
EXP	Exponent	$e^x = b$

Memory²

MEM	Verfügbarer Speicherplatz	
MENU ³	Erzeugt CUSTOM Menü	{ LN,SUB,... → Menü
ORDER	Variablen neu ordnen	{ AX,SYS,... → geordnet
PATH ³	Aktuellen Pfad anzeigen (Programmebene)	
HOME ³	HOME Verzeichnis wählen (oberste Programmebene)	
CRDIR ³	Menüverzeichnis erzeugen	
VAR ³	Nimmt alle Variablen des aktuellen Verzeichnisses in Liste	
CLUSR	Aktuelles Verzeichnis löschen	CLUSR ENTER

²) Beim HP 28 C USER Menü ³) Nur beim HP 28 S

Mode

STD	Standart Format	
FIX	Anzahl der Nachkommestellen	2 FIX → 1.12 ...
SCI	Wissenschaftliches Anzeigeformat	n SCI → $a \cdot 10^n$...
ENG	Technisches Anzeigeformat	n ENG → $a \cdot 10^3 \cdot n$...
DEC	Grad-Winkel	
RAD	Radiant-Winkel	
CMD	COMMAND → speichert letzte Befehlszeile	
UNDO	UNDO → speichert gesamten Stack nach ENTER	
LAST	LAST → speichert letztes Argument	
ML	MULTILINE → Mehrzeilenformat für Ebene 1:	
RDX,	"," als Dezimalzeichen definieren	
PRMD	Modi drucken	

Plot

STEQ	Gleichung speichern	
RCEQ	Gleichung zurückrufen	
PMIN	Minimum speichern und abbilden	(X,Y) PMIN
PMAX	Maximum speichern und abbilden	(X,Y) PMAX
INDEP	Unabhängige Variable speichern	'X' INDEP
DRAW	Abbilden einer Funktion	
PPAR	Abbildungsparameter aufrufen	
RES	Schrittweite festlegen	
AXES	Koordinatenursprung festlegen	(X,Y) AXES
	Voreinstellung ist (0,0)	
CENTR	Abbildungszentrum speichern	(X,Y) CENTR
	Voreinstellung ist (68,15)	
*W	Breite multiplizieren	$X_e' = n \cdot X_e$
*H	Höhe multiplizieren	$Y_e' = n \cdot Y_e$
SCL \sum	Statistisches Streudiagramm in Anzeige einpassen	
DRW \sum	Führt DRAX aus und erzeugt ein statistisches Streudiagramm aus den Daten in \sum DAT	
CLLCD	Anzeige löschen	
DGTIZ ³	Cursor zurückrufen	
DRAX	Achsen zeichnen	
CLMF	Meldungsflag löschen (Grundanzeige)	

³) Nur beim HP 28 C

Print

PR1	Ebene 1: drucken	
PRST	Stackinhalt drucken	
PRVAR	Variable drucken	'X' PVAR
PRLCD	LCD drucken	
CR	Druckkopf rechts stellen	
TRAC	Protokoll drucken	
PRSTC	Stack kompakt drucken	
PRUSR	Benutzervariable drucken	

PRMD Modi drucken
 33 SF Sammeln von Druckdaten im Druckpuffer
 ..
 52 SF Erhöhte Druckgeschwindigkeit (nur mit Druckernetzteil!)
 ..
 47 SF Doppelter Zeilenabstand
 ..
 |ON| |L| Anzeigehalt drucken (Systemoperation)

Programm Branch

IF Beginn der Wenn-Bedingung
 IFERR Beginn der Wenn-Fehler-Bedingung IFERR X 0 == THEN
 THEN Dann-Bedingung
 ELSE Sonst-Bedingung
 END Ende der Programmstruktur
 START Anfang der bestimmten Schleife a b START ...
 FOR Anfang der bestimmten Schleife a b FOR x ...
 NEXT Ende der bestimmten Schleife und nächster Schritt
 STEP Ende der bestimmten Schleife und nächster definierter Schritt
 IFT IF-THEN-Befehl X 0 ==IFT
 IFTE IF-THEN-ELSE-Befehl 'IFTE(X=0,.....,1)'
 DO Unbestimmte Schleife DO X Y + ...
 UNTIL Bis-Bedingung DO ... UNTIL 0 ==
 WHILE Beginn der Solange-Bedingung WHILE X 0 <
 REPEAT Wiederhole-Bedingung WHILE X 0 < REPEAT ...

Programm Control

SST Programmablauf in Einzelschritten
 HALT Programmunterbrechung
 ABORT Programmabbruch
 KILL Unterbrochenes Programm abbrechen
 WAIT Definierte Programmunterbrechung zB. 2 Sekunden 2 WAIT
 KEY Gibt String für Tastencode zurück
 BEEP Definiertes Akustiksignal z.B. 200 Hz 3 Sek. 200 3 BEEP
 ERRN Fehlernummer

Programm Test

SF	Setze Flag
CF	Lösche Flag
FS?	Flag gesetzt?
FC?	Flag gelöscht?
SAME	Übereinstimmung zweier Objekte prüfen
==	Gleichheit
STOF	Flags speichern
RCLF	Status aller Flags aufrufen
TYPE	Objekttyp feststellen

Real

FACT	Fakultät
RAND	Zufallszahl aufrufen
RDZ	Anfangswert für Zufallsgenerator
SIGN	Vorzeichen einer Zahl
RND	Runden einer Zahl auf n Stellen bei Modus n-FIX
MAX	Gibt den größeren Wert zweier Argumente zurück
MIN	Gibt den kleineren Wert zweier Argumente zurück
%T	Prozent von Total

Solve

STEQ	Speichern einer Gleichung
RCEQ	Zurückrufen einer Gleichung
SOLVR	Menü für Lösungsvariablen
QUAD	Quadratische Lösung einer Gleichung
ROOT	Routine für Nullstellensuche

Stack

DUP	Objekt duplizieren
OVER	Objekt in Ebene 2 duplizieren
DUP2	Zwei Objekte duplizieren
DROP2	Zwei Objekte löschen

ROT	Den Inhalt von Ebene 3 in Ebene 1 ($\hat{=}$ 3 ROLL)
ROLLD	Den Inhalt von Ebene 1 in Ebene n rollen
PICK	n-tes Objekt duplizieren
DUPN	n Objekte duplizieren
DROPN	n Objekte löschen
DEPTH	Anzahl der Objekte im Stack feststellen

Stat

$\Sigma+$	Datenpunkte in Statistikmatrix geben
$\Sigma-$	Letzten Datenpunkt löschen
$N\Sigma$	Anzahl der Datenpunkte in der Statistikmatrix
$CL\Sigma$	Statistikmatrix löschen
$STO\Sigma$	Statistikmatrix speichern
$RCL\Sigma$	Statistikmatrix zurückrufen
TOT	Total
MEAN	Mittelwert der einzelnen Spalten der Statistikmatrix
SDEV	Standardabweichung
VAR	Varianz
$MAX\Sigma$	Maximum aller Koordinatenwerte der Spalten in ΣDAT
$MIN\Sigma$	Minimum aller Koordinatenwerte der Spalten in ΣDAT
$COL\Sigma$	Spalten wählen (für LR, $SCL\Sigma$, CORR und COV)
CORR	Correlation
COV	Covarianz
LR	Lineare Regression
PREDV	Linearer Vorhersagewert
UTPC	Rechtsseitige Chi-Quadrat-Verteilung
UTPF	Rechtsseitige F-Verteilung
UTPN	Rechtsseitige Normalverteilung
UTPT	Rechtsseitige t-Verteilung
COMB ³	Combination
PERM ³	Permutation

³) Nur beim HP 28 S

Store

STO+	Objekt addieren und speichern
STO-	Objekt subtrahieren und speichern
STO*	Multiplizieren und speichern
STO/	Dividieren und speichern
SNEG	Negieren und speichern
SINV	Invertieren und speichern
SCONJ	Konjugieren und speichern

String

↔STR	Objekt in String
STR↔	String in Objekt
CHR	Ein-Zeichen-String ausgeben, welcher das entsprechende HP 28 Zeichen enthält
NUM	HP Zeichen in Zeichencode zurückverwandeln
↔LCD ³	String in Anzeige
LCD↔ ³	Anzeige in String
POS	Position eines Substrings (1. Zeichen)
SUB	Substring bilden
DISP	Anzeigen von Strings in der n-ten Zeile

³) Nur im HP 28 S

Trig

SIN	Sinus
ASIN	Arcussinus
↔HMS	Dezimalwert in Stunden, Minuten, Sekunden
HMS+	Stunden, Minuten, Sekunden Addition
D↔R	Grad in Radiant
R↔D	Radiant in Grad

Units

NEXT	Nächsten Katalogeintrag anzeigen
PREV	Vorherigen Katalogeintrag anzeigen
SCAN*	Blättert automatisch vorwärts, jeder Eintrag wird kurz angezeigt.
FETCH	Beendet Katalog und übernimmt die Einheit in den Stack
QUIT	Beendet Katalog
A bis Z	Ruft den ersten Katalogeintrag mit dem entsprechenden Anfangsbuchstaben auf
#	Ruft den ersten Katalogeintrag auf, der nicht mit einem Buchstaben beginnt

Catalog

siehe Units

USE	Usage-Anzeige, zeigt die Verwendung eines Befehls
█ QUIT	Von der Usage-Anzeige zur Kataloganzeige des momentanen Befehls zurückkehren
█ ON	Aus dem Usage-Modus Kataloganzeige beenden
ON	Beendet die Kataloganzeige und löscht die Befehlszeile

Cursor

INS	Auf Einfügensmodus umschalten
DEL	Löscht das Zeichen an der Cursorposition
█ INS	Löscht alle Zeichen links des Cursors
█ DEL	Löscht alle Zeichen rechts ab der Cursorposition
▲	Bewegt den Cursor nach oben
█ ▲	Bewegt den Cursor ganz nach oben
▼	Bewegt den Cursor nach unten
█ ▼	Bewegt den Cursor ganz nach unten
█ ▶	Bewegt den Cursor ganz nach rechts
█ ◀	Bewegt den Cursor ganz nach links

1.3. Tips und Übungen für Anfänger und Fortgeschrittene

1.3.1. Allgemeine Funktionen

Einige Tastenfunktionen

←	Rückschritt-Taste
LC	Kleinschreibungsmodus
■ MENUS	Menü-Direktwahl, für die Wahl der <u>Buchstaben</u> muß nun die Umschalttaste ■ gedrückt werden
DROP	Löschen der Befehlszeile bzw. Ebene 1
CLEAR	Löscht den gesamten Stack
SWAP	Vertauscht die Stack-Ebene 1 mit Ebene 2
NEWLINE	Neue Zeile, der Befehl benötigt keinen Speicherplatz

Die Eingabemodi des HP 28 S/C

Ihr Taschencomputer stellt Ihnen drei Eingabemodi zur Verfügung, den unmittelbaren-, den algebraischen- und den Alpha-Eingabemodus.

Der unmittelbare Eingabemodus dient zum Eingeben von Zahlen, Namen, Listen oder Feldern. Die Berechnung von $1+(3-1)$ wird in diesem Modus wie folgt durchgeführt:

1 |ENTER| 3 |ENTER| 1 |ENTER| |-| |+| Ergebnis: 3

Eine Rechenoperation wird nach der Eingabe der zugehörigen Objekte ausgeführt. Zur Eingabe der Objekte kann auch ein Trennungszeichen verwendet werden:

1,3,1 |-| |+| oder 1 |SPACE| 3 |SPACE| 1 |-| |+|

Der algebraische Eingabemodus dient zur Eingabe von Namen, Variablen und algebraischen Ausdrücken. Führen Sie die oben angegebene Berechnung wie folgt aus:

Sie aktivieren den algebraischen Modus mit '|'|.

'|' 1 |+| |(| 3 |-| 1 |)| |ENTER|

Das Ergebnis erhalten Sie durch eine Auswertung der Befehlszeile mit dem Befehl |EVAL| oder |→NUM|.

Der Alpha-Eingabemodus dient zur Eingabe von Namen, Strings und Programmen. Dieser Modus wird durch die Tasten $|\alpha|$, $\blacksquare|"$ oder \ll aktiviert. Der α -Indikator erscheint in der Anzeige. Zum Verlassen des Alpha-Modus drücken Sie $|\alpha|$ einmal – Sie befinden sich nun im algebraischen Eingabemodus, durch nochmaliges Drücken von $|\alpha|$ kehren Sie zum unmittelbaren Eingabemodus zurück.

Der Umgang mit dem Stack

Der dynamische Stack des HP 28 hat den Vorteil, daß er keine feste Größe besitzt. Durch die Eingabe von Objekten werden immer neue Ebenen erzeugt; das heißt, daß Sie alle eingegebenen Daten immer für Berechnungen zur Verfügung haben. Eine obere Grenze wird nur durch den freien Speicherplatz gesetzt. Beachten Sie, daß Sie nicht mehr benötigte Objekte aus dem Stack löschen, um nicht einen beachtlichen Teil des Speichers damit zu belegen.

Anzeigen großer Objekte

Die Anzeige des HP 28 besitzt vier Zeilen und faßt 23 Zeichen pro Zeile. Wenn eine Zeile mehr als 23 Zeichen beinhaltet, werden die ersten Zeichen, bei Eingabe, nach links verschoben. Nach einer Übernahme des Objekts in den Stack erscheinen, falls der Multiline-Modus für Ebene 1 nicht aktiviert ist, drei Punkte am rechten Zeilen- bzw. Anzeigenrand.

Zur Ansicht der gesamten Zeile in Ebene 1 aktivieren Sie den Multiline-Modus oder betätigen die Tasten $\blacksquare|EDIT|$ oder $\blacksquare|VISIT|$.

Um ein Objekt in der n-ten Ebene anzusehen, kopieren Sie es mit $n \blacksquare|PICK|$ in die Ebene 1 und gehen wie oben vor oder betätigen $n \blacksquare|VISIT|$.

Wenn sich ein zu betrachtendes Objekt außerhalb des Anzeigefensters befindet, können Sie dieses Fenster mit den Befehlen $\blacksquare|VIEW\uparrow|$ und $\blacksquare|VIEW\downarrow|$ nach oben bzw. nach unten verschieben.

Editieren

Mit dem HP 28 haben Sie die Möglichkeit gespeicherte Objekte nachträglich

in die Befehlszeile zu übernehmen und abzuändern. Ihr Taschencomputer stellt Ihnen zwei Editierfunktionen zur Verfügung.

█ EDIT	Ebene 1 editieren
█ VISIT	Ebene 1 editieren
n █ VISIT	Ebene n editieren
'DIV' █ VISIT	Programm oder Variable editieren
ENTER nach EDIT	gibt das geänderte Objekt in die Befehlszeile zurück
ENTER nach VISIT	gibt das geänderte Objekt in die gespeicherte Variable zurück

Wenn der Editiervorgang unterbrochen werden soll, betätigen sie |ON|. Das Objekt wird unverändert übernommen.

Die Objekttypen des HP 28 S/C

0	Reelle Zahl	z.B. 0 1 1/3 -8 ...
1	Komplexe Zahl	(a,jb)
2	String	"Hallo"
3	Reeller Vektor oder Matrix	[[1 2 1/3 [-8 ...
4	Komplexer Vektor oder Matrix	[[(1,3) (6,2) [(4,4) ...
5	Liste	{
6	Name	'DIV'
7	Lokaler Name	nc
8	Programm	◀ DUP 3 ROLL SWAP ▶
9	Algebraischer Ausdruck	'2*X^2-1'
10	Binärwert	#100101100000b

Einheiten-Transformation

Der Speicher des HP 28 enthält 120 Einheiten, welche auf dem SI-Einheitensystem aufgebaut sind. Dieses internationale Einheitensystem enthält sieben Basiseinheiten, welche die Grundlage für die Einheiten-Umrechnung bilden.

Der CONVERT-Befehl erkennt jede Kombination dieser Einheiten mit Zahlenwerten, Konstanten oder anderen Einheiten. Der UNITS-Katalog enthält die installierten Einheiten und Ihren Umrechnungsfaktor gegenüber den Basis-

einheiten. Die sieben Basiseinheiten sind:

l,s	Länge in Meter	m
t	Zeit in Sekunden	s
I	Stromstärke in Ampere	A
m	Masse in Kilogramm	kg
T,	Temperatur in Grad Kelvin	°K
J,I	Lichtstärke in Candela	cd
v	Stoffmenge in Mol	mol

Sie können zusätzlich zu den sieben Basiseinheiten auch eine neue Basiseinheit definieren, welche vom HP 28 als solche erkannt wird. Dieses geschieht im Rahmen der benutzerdefinierten Einheiten.

Beispiel 1:

Sie wollen die Einheit 1 m in "international foot" umrechnen.

Geben Sie 1 m in den Stack: 1 , |LC| |M| |ENTER|

Rufen Sie den Units-Katalog auf: █ |UNITS|

Wählen Sie den ersten Eintrag mit dem Anfangsbuchstaben f: |F|

Blättern Sie mit NEXT solange bis die Einheit ft (int'l food) in der Anzeige erscheint
.....

Drücken Sie FETCH zum Übernehmen der Einheit in den Stack
.....

Betätigen Sie █ |CONVERT|

Sie erhalten als Ergebnis 3.28 'ft', dabei erscheint der neue Wert in Ebene 2: und die Benennung in Ebene 1:

Beispiel 2:

10 °Celsius in °Fahrenheit ohne den Units-Katalog

10 █ |°| |C| |ENTER| █ |°| |F| █ |CONVERT| Ergebnis: 50 °F

Umrechnen von Einheiten in Stringform

Wenn Sie Einheiten in Strings verwenden, können Sie ganze Ausdrücke von Einheiten definieren. Die Ausdrücke können dann folgendermaßen aussehen:

Einheit mit Exponent	"m^2"
Produkt zweier Einheiten	"A*s"
Quotient zweier Einheiten	"m/s"
Potenzen von Einheiten	"dl" für Dezi-Liter

Beachten Sie, daß bei einem Quotienten von Einheiten das Symbol / nur einmal im String vorkommen darf! Stellen Sie Liter pro Meter pro Meter pro Meter folgendermaßen dar: "l/m^3".

Rechnen Sie 1 ga (Fallbeschleunigung) in dm/s^2 (Dezimeter pro Sekundenquadrat) um.

Lösung: 1,ga,"dm/s^2" |ENTER| █ |CONVERT| Ergebnis: 98.0665 dm/s^2

Benutzerfunktion zum Umrechnen von Einheiten

Von Ihnen definierte Funktionen zum Umrechnen von Einheiten müssen folgende Kriterien erfüllen:

1. Das Argument muß genau festgelegt sein.
2. Es darf nur ein Ergebnis zurückgegeben werden.

Beispiel: Umrechnung von Knoten in Kilometer pro Stunde

```

<<          Programmanfang
→ x         Stack-Argument in die lokale Variable x benennen
<<         Anfang Konvertierungs-Routine
x          Argument in den Stack übernehmen
'knot' 'kph' Einheiten in den Stack übernehmen
CONVERT    Konvertieren
>>        Programmende

'kn-km' |STO| Programm unter dem Namen 'kn-km' speichern

```

Starten Sie Ihr Programm zur Umrechnung von 2 Knoten in Kilometer pro Stunde:

```
2 |USER| kn→km      Ergebnis:      3.704 kph
   .....          
```

Benutzerdefinierte Einheiten

Sie können Einheiten, welche Sie häufig zur Umrechnung benötigen, mit Ihrem Taschencomputer selbst definieren. Dazu erzeugen Sie eine Liste, die eine reelle Zahl als Umrechnungsfaktor und einen Einheitenstring enthält. Diese Liste speichern Sie in einer Variablen mit dem neuen Einheitenamen.

Wenn Sie z.B. die alte französische Mengeneinheit "Gros" (1 Gros = 144 Stück) häufig verwenden, können Sie diese wie folgt definieren.

```
{144 "St"} 'Gros' |STO|           St ≙ Stück
```

Beispiel: Umrechnung von 60 Stück in Gros

```
60,'St','Gros' |ENTER| █|CONVERT|           Ergebnis: 0.416̄ Gros
```

Neue Basiseinheit definieren

Um eine neue (achte) Basiseinheit zu definieren, muß diese dimensionslos sein, d.h. ihr Umrechnungsfaktor ist 1. Zur Kenntlichmachung der neuen, nicht im SI-Einheitenkatalog verzeichneten Einheit, nehmen sie den String "?" als zweiten Eintrag in die Einheiten-Liste auf. Speichern Sie diese dann unter dem Einheitenamen.

Beispiel: Sie wollen DM als neue Basiseinheit definieren, um dann in ÖS und SFR umzurechnen.

```
{1 "?"} 'DM' |STO|                 Basiseinheit
{.144 "?"} 'ÖS' |STO|              Österreichische Schilling
{1.17 "?"} 'SFR' |STO|            Schweizer Franken
```

Konvertieren Sie 50 DM in die anderen Währungen:

```
50,'DM','ÖS' |ENTER| █|CONVERT| █|STACK| DUP2 'SFR' █|CONVERT|
Ergebnis: 50 DM ≙ 347.2̄ SFR ≙ 42.74 ÖS
```

1.3.2. Die Verwendung von Variablen

Im HP 28 S/C finden drei Variablentypen Verwendung:

Globale Variablen

Globale oder allgemeine Variablen werden mit dem Befehl |STO| erzeugt. Sie sind eine Kombination von Buchstaben und anderen Objekten zur Benennung beliebiger Objekte.

Lokale Variablen

Lokale Variablen werden in Programmen mit der Sequenz → variable er-

zeugt. Vergleichen Sie dazu auch den Abschnitt Benutzerfunktionen. Lokale Variablen finden nur in Programmen Verwendung und existieren auch nur in dem Programm, in dem Sie erzeugt wurden. Um lokale Variablen von globalen Variablen zu unterscheiden, werden sie kleingeschrieben.

Namen
.....

Namen werden mit |STO| gespeichert und bestehen aus einer Verkettung beliebiger Zeichen. Namen benennen Programme, Ausdrücke, Listen und Felder.

ACHTUNG
.....

Variablen können bis zu 127 Zeichen beinhalten, sollten aber nicht länger als fünf Zeichen sein. Dies ist zweckmäßig, da ein Menüfeld nur fünf Zeichen darstellen kann.

Beachten Sie, daß das erste Zeichen in einer Variablen keine Zahl sein darf! Namen von Befehlen können nicht als globale Variablen benutzt werden. Die Konstanten e, i, , s1,s2,..., n1,n2,... können ebenfalls nicht verwendet werden. Außerdem sind folgende Zeichen auszuschließen:

(# [] ' " () { } <> * / + - ^ √ = ≤ ≥ ≠ ∂ ∫ . ,

Namen, welche mit "der" beginnen sind für benutzerdefinierte Ableitungen reserviert.

1.3.3. Erstellen von Abbildungen

Die Anzeige des HP 28 ist in 32 x 137 Pixel gegliedert. Sie ist normalerweise in vier Zeilen mit je 23 Zeichen aufgeteilt. Ein Zeichen ist 6 x 8 Pixel groß, wobei die angezeigten Zeichen selbst und die Felder am rechten Anzeigenrand nur 5 Pixel breit sind.

Zum Anzeigen von Strings mit dem Befehl DISP muß jede der vier Zeilen grundsätzlich als einzelne Einheit betrachtet werden. Dabei ist die oberste Zeile Zeile 1, die unterste Zeile 4.

Sie können zum Anzeigen von Graphiken auch jedes Pixel einzeln ansteuern. Dazu wird ein Pixel durch seine Koordinaten festgelegt, welche in komplexer Form (x,y) geschrieben werden. Dabei ist x die waagrechte- und y

die senkrechte Koordinate.

Das Verhältnis von Koordinate zu Pixel wird durch die Koordinaten der Anzeigen-Eckpunkte und durch die Position des Mittelpunktes festgelegt.

Die Anzeigen-Eckpunkte werden durch die Befehle PMIN und PMAX, der Koordinatenursprung durch AXES (Mittelpunkt) oder CENTR (Zentrieren eines bestimmten Punktes) festgelegt.

Die Grundparameter zum Abbilden eines Graphen werden in einer Liste mit dem Namen PPAR (mathematische Funktionen) oder \int PAR (statistische Streudiagramme) gespeichert.

PPAR: { PMIN PMAX unabhängige Variable Auflösung(RES) Mittelpunkt }
 \int PAR: { Spalte 1 (x) Spalte 2 (y) Achsenabschnitt Steigung }

Die Grundeinstellung ist:

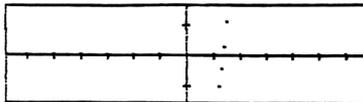
PPAR: { (-6.8,-1.5) (6.8,1.6) x 1 (0,0) }
 \int PAR: { 1 2 LR (Lineare Regression) }

Die Koordinatenachsen werden bei Erstellung des Graphen mit einem Maßstab versehen. Der Abstand der Markierungen ist 10 Pixel, die Einheit ist 1.

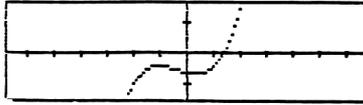
1.3.3.1. Kurvendiskussion

Stellen Sie die Funktion ' X^3+X^2-X-3 ' graphisch dar und bestimmen Sie Extremwerte und Nullstellen der Funktion.

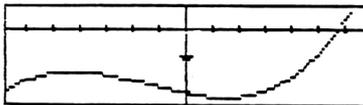
- 1) Geben Sie die Gleichung ein: ' X^3+X^2-x-3 ' |ENTER| |STEQ|
- 2) Definieren Sie die unabhängige Variable: entfällt, da $x =$ Voreinstellung
- 3) Legen Sie die Auflösung der Zeichnung fest: 1 RES (=Schrittweite)
- 4) Bilden Sie den Graphen ab: DRAW
.....



- 12) Ändern Sie vorläufig nur die Höhe der Abbildung und lassen Sie den Graphen neu zeichnen: `▣|PLOT| |NEXT| 5 *H ▣|PREV| DRAW`
 13) Der Graph wird nun in der Höhe fünf mal kleiner abgebildet.



- 14) Sie können nun ein lokales Maximum und ein lokales Minimum erkennen. Der "Löser" des HP 28 ist nur in der Lage, das lokale Maximum der Funktion zu bestimmen, da es sich in der Nähe des Punktes befindet, welcher der X-Achse am Nächsten liegt. Das lokale Minimum kann aber auf anderem Weg bestimmt werden. Um die Anzeige noch etwas zu verbessern, können Sie den Abbildungsbereich neu definieren. Bewegen Sie den Cursor auf den gewünschten linken unteren Eckpunkt der neuen Abbildung und betätigen Sie `|INS|`. Dann bewegen Sie den Cursor zum oberen rechten Eckpunkt und digitalisieren ihn mit `|INS|`. Rufen Sie die normale Anzeige zurück und definieren Sie die neuen Abbildungsparameter: `|ON| PMAX PMIN .` Führen Sie `DRAW` aus. Abbildungsbeispiel:



- 15) Suchen Sie nun den Wert des lokalen Maximums:
 Stellen Sie den Cursor auf das ungefähre Maximum und digitalisieren sie diesen Abbildungspunkt. Digitalisieren Sie dann die Punkte links und rechts des Maximums und betätigen sie `|ON|`.
 Fassen Sie diese drei Punkte in einer Liste zusammen und übergeben Sie sie in die Variable X:
`3 ▣|LIST| →LIST |SOLV| SOLVR X`
 Nun aktivieren Sie den Löser zum Bestimmen des genauen Ergebnisses. Das Ergebnis ist: `EXTREMUM -.99999`
- 16) Kehren Sie mit `|ON|` zur normalen Anzeige zurück.

- 17) Mit `EXPR=` rufen Sie den Y-Wert des lokalen Maximums ab. Die vollständigen Koordinaten lauten dann: $x = -.99999$ $y = -2$.
- 18) Kehren Sie mit `|ON|` zur normalen Anzeige zurück.

Um nun das lokale Minimum zu berechnen haben Sie zwei Möglichkeiten:

1. Bilden Sie die Ableitung der Funktion ' X^3+X^2-3 ' und berechnen deren Nullstellen mit dem Löser.
2. Bilden Sie die Ableitung der Funktion und stellen Sie sie graphisch dar. Dann ermitteln Sie die Nullstelle wie in Punkt 5.

Sie könnten die Berechnung des lokalen Maximums natürlich auch über die 1. Ableitung durchführen, denn die Bedingungen für Extremwerte sind:

Nullstelle : $f(x) = 0$

Monotonie : lokal monoton wachsend
für $f'(x) > 0$, $f''(x) < 0$

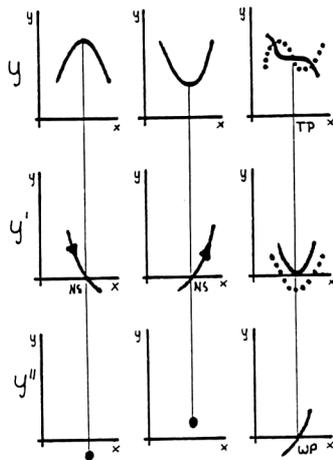
lokal monoton fallend
für $f'(x) < 0$, $f''(x) > 0$

Extrema : Maximum
 $f'(x) = 0$, $f''(x) < 0$

Minimum
 $f'(x) = 0$, $f''(x) > 0$

Wendepunkte : $f''(x) = 0$, $f'''(x) \neq 0$

Terassenpunkt
 $f'(x) = 0$



Lesen Sie dazu bitte auch das Kapitel "Nullstellensuche bei Funktionen und Gleichungen".

Um das lokale Minimum zu berechnen verfolgen wir hier den zweiten Lösungsweg.

- 19) Löschen Sie eine evtl. gespeicherte Variable X und rufen Sie die in EQ gespeicherte Funktion mit `RCEQ` zurück.

- 20) Bestimmen Sie die erste Ableitung der Funktion. Vergleichen Sie dazu auch das Kapitel "Integrieren und Differenzieren".

'X |ENTER| \blacksquare |d/dx| Ergebnis: '3*X^2+2*X-1'

- 21) Rufen Sie die Ausgangsgleichung noch einmal in den Stack.

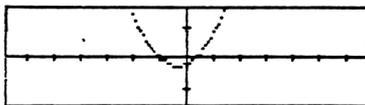
|SOLV| RCEQ

- 22) Speichern Sie die Funktion $f(x)$ zur späteren Verwendung in einer Variablen (z.B. GL1) und geben Sie die Ableitung in EQ.

GL1 |STO| STEQ

- 23) Bilden Sie die Gleichung ab. Vorher sollten Sie aber die Variable PPAR löschen und die Höhe der Abbildung um 5 erweitern, um wieder den gleichen Maßstab wie in Punkt 12) dieses Kapitels zu erhalten.

|USER| 'PPAR \blacksquare |PURGE| \blacksquare |PLOT| |NEXT| 5 *H \blacksquare |PREV| DRAW



- 24) Wie Sie jetzt erkennen können, hat der Graph zwei Nullstellen. Wie wir uns erinnern (oder wenn Sie den Graphen der Funktion $f(x)$ ausgedruckt vor sich haben), ist die erste Nullstelle die Stelle des lokalen Maximums der Ausgangsfunktion $f(x)$.

Die zweite Nullstelle muß also das lokale Minimum von $f(x)$ spezifizieren.

Gehen Sie mit dem Cursor an diese zweite Nullstelle und digitalisieren Sie sie. Dann rufen Sie die normale Anzeige zurück und starten die Lösungsroutine zum genauen Bestimmen des Wertes.

|INS| |ON| |SOLV| SOLVR X \blacksquare X Ergebnis: x= .33333

- 25) Bestimmen Sie jetzt den Y-Wert dieses Punktes.

Dazu rufen Sie die Funktion $f(x)$ zurück in den Stack:

|USER| GL1

Dann speichern Sie diese in EQ und gehen in das Solve-Menü (Der

X-Wert ist noch in der Variablen 'X' gespeichert): |SOLV| STEQ

Abschließend betätigen Sie SOLVR und EXPR= zum Berechnen des Y-Wertes. Als Ergebnis erhalten Sie: y= -3.185185.

Die Koordinaten des lokalen Minimums sind: X= .33333 y=-3.185185

Zur weiteren Bestimmung expliziter Werte vergleichen Sie das Kapitel "Integrieren und Differenzieren".

1.3.3.2. Gleichzeitige graphische Darstellung von Funktionen

Beim HP 28 können Sie zwei Wege gehen, um mehrere Graphen gleichzeitig darzustellen.

- 1) Wenn Sie zwei zu zeichnende Funktionen durch ein "=" verbinden, liest dies Ihr Taschencomputer als Gleichung, wobei jeder Seite einzeln ein Graph zugeordnet wird. Diese zwei Graphen werden gleichzeitig dargestellt.
- 2) Sie können aber auch eine oder mehrere Abbildungen digitalisieren, d.h. mehrere komplette Bilder in Variablen speichern, später wieder aufrufen und überlagern. Diese Art der Darstellung geht aber nur mit dem HP 28 S.

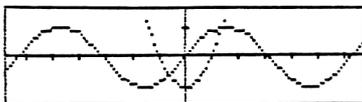
Beispiel 1:

Die Funktionen $y=x^2-1$ und $y=\text{SIN}(x)$ sollen gleichzeitig graphisch dargestellt werden. Gehen Sie folgendermaßen vor:

- 1) Geben Sie beide Funktionen in den Stack:
'X^2-1' , |TRIG| 'SIN(X) |ENTER|
- 2) Verbinden Sie diese durch ein "=" zu einer Gleichung:
|=| |ENTER|
- 3) Speichern Sie diese in EQ und bilden Sie sie ab:

■ |MODE| RAD ■ |PLOT| STEQ DRAW
... ..

Gehen Sie dabei sicher, daß keine Variable X gespeichert ist und die Voreinstellung der Abbildungsparameter gilt.



- 4) Jetzt sind Sie im Interaktiven Modus, d.h. Sie können ganz normal beliebige Punkte digitalisieren und abrufen.

Wenn Sie den Löser zum Bestimmen expliziter Werte benutzen, zeigt Ihr Taschencomputer als Lösung die Schnittpunkte beider Graphen an, da Sie eine Gleichung ($a=b$) gespeichert haben.

Beispiel 2:

Die Funktionen $y=X^2$, $y=x$ und $y=\text{SIN}(x)$ sollen in Variablen gespeichert, beliebig miteinander verknüpft und gleichzeitig dargestellt werden.

Beim HP 28 S können Sie eine ganze Abbildung durch das Betätigen der Taste $|\text{DEL}|$ digitalisieren, wenn sich der Rechner im interaktiven Modus befindet.

In den Stack wird nach $|\text{ON}|$ ein String zurückgegeben, dessen Inhalt der gespeicherten Abbildung entspricht. Dieser String kann in einer beliebigen Variablen gespeichert werden. Danach läßt er sich jederzeit wieder aufrufen und mit dem Befehl $\rightarrow\text{LCD}$ in die ursprüngliche Anzeige zurückverwandeln.

Beachten Sie, daß sich der Rechner nach erneutem Abbilden des Strings nicht mehr im interaktiven Modus befindet. Sie können aber den Cursor bzw. den interaktiven Modus durch den Befehl DGTIZ zurückrufen.

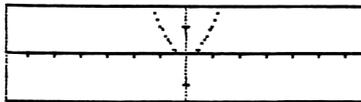
1) Geben Sie die Funktionen in den Stack:

' X^2 ' , 'X' , 'SIN(X) $|\text{ENTER}|$

2) Löschen Sie eine evtl. vorhandene Variable X und rufen Sie zum Zeichnen der Funktion den Radiant-Modus auf. Dann wählen Sie das Plot-Menü und übergeben die erste Funktion in die Variable EQ:

$|\text{USER}|$ 'X' $|\text{PURGE}|$ $|\text{MODE}|$ RAD $|\text{PLOT}|$ STEQ

3) Lassen Sie die Funktion mit DRAW zeichnen.



4) Digitalisieren Sie die gesamte Abbildung mit $|\text{DEL}|$.

5) Rufen Sie mit $|\text{ON}|$ die normale Anzeige zurück. Im Stack steht jetzt ein String, der der Abbildung entspricht.



- 6) Speichern Sie diesen String zum Beispiel in der Variablen A1. Gehen Sie bei den anderen Funktionen auf die gleiche Weise vor und speichern diese in den Variablen A2 und A3.

Um sicherzustellen, daß Sie den interaktiven Modus, nach einer Rücktransformation der Variablen A1 bis A3 in eine Abbildung, wieder aktivieren können legen Sie die Befehle →LCD, LCD→ und DGTIZ auf Menütasten. Dazu steht Ihnen das Custom-Menü zur Verfügung.

```
7) { |α| █ |STRING| →LCD LCD→ █ |PLOT| |NEXT| |NEXT| |NEXT| DGTIZ
    |ENTER| █ |MEMORY| MENU
```

Damit haben Sie die drei oben genannten Befehle im Custom-Menü gespeichert.

- 8) Rufen Sie die Abbildungen A1 bis A3 als String in den Stack zurück und verknüpfen Sie diese Strings mit OR, AND, XOR oder NOT.

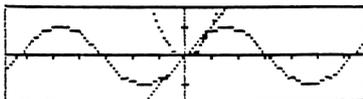
Die verschiedenen Arten der logischen Verknüpfung haben folgenden Einfluß auf die Darstellung der Funktionen:

- AND: Zeigt nur die Schnittpunkte der Graphen.
- OR : Zeigt alle Graphen vollständig und gleichzeitig.
- XOR: Zeigt alle Graphen gleichzeitig, jedoch ohne die Schnittpunkte.
- NOT: Stellt die Graphen invers dar.

```
|USER| A1 A2 A3 █ |BINARY| |NEXT| |NEXT| |NEXT| OR OR
```

- 9) Rufen Sie das Custom-Menü auf und wandeln Sie den String in eine Anzeige: █ |CUSTOM| →LCD

- 10) Sie können nach dem Zeichnen eines Graphen die unsichtbare Menütaste DGTIZ betätigen, wenn Sie dieses Menü vorher aufgerufen haben.



1.3.3.3. Abbilden statistischer Daten

Bilden Sie folgende Statistikpaare ab:

-0.8	-1.6
-0.6	-1.2
-0.4	-0.8
-0.2	-0.4
0	0
0.2	0.4
0.4	0.8
0.6	1.2
1	1.2
1.4	1.2
1.8	1.2
2.2	1.2

- 1) Löschen Sie eine evtl. vorhandene Statistikmatrix \sum DAT und die Liste der zugehörigen Abbildungsparameter \sum PAR.

|USER| | α | { \sum DAT, \sum PAR | α | ■ |PURGE|

- 2) Geben Sie die Statistikpaare in die Statistikmatrix ein:

■ |STAT| [-0.8,-1.6 \sum + [-0.6,-1.2 \sum + [-0.4,-0.8 \sum + ...

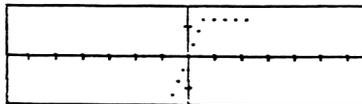
- 3) Spezifizieren Sie die zwei Spalten für die abhängigen und unabhängigen Statistikwerte mit COL \sum . Hier ist dies unnötig, da es nur zwei Statistikspalten gibt (die Voreinstellung ist Spalte 1 und 2).

- 4) Legen Sie eventuell den Achsenschnittpunkt und den Abbildungsmaßstab neu fest (AXES,*H,*W).

- 5) Stellen Sie die Werte graphisch dar:

■ |PLOT| |NEXT| |NEXT| DRW \sum

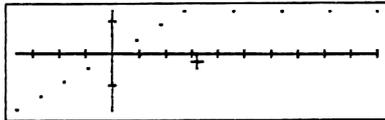
Der interaktive Modus wird aktiviert.



Sie können mit dem Befehl SCL \sum die Abbildung auch anders darstellen.

SCL \int bewirkt ein genaues Einpassen der Statistikpunkte in die Anzeige. Das heißt, daß die waagrechten und senkrechten Koordinaten von P_{MAX} und P_{MIN}, entsprechend der Daten in den Spalten für die unabhängigen und abhängigen Werte, neu festgelegt weden.

Nach dem Betätigen von SCL \int sieht die Anzeige folgendermaßen aus:



1.3.3.4. Erstellen von Bildern

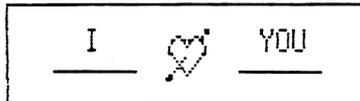
Mit Ihrem Taschencomputer ist es möglich, beliebige Bilder zu erzeugen. Dazu ist es nötig, jeden Abbildungspunkt einzeln zu definieren und darzustellen.

Die Befehlsprozedur ist: (x,y) PIXEL

Sie können gewisse Punktfolgen auch durch Programme erzeugen.

Leider erfordert das Erstellen von Bildern einen größeren Arbeitsaufwand und benötigt sehr viel Speicherplatz. Daher ist es sinnvoll, den freien Speicherplatz zwischendurch zu überprüfen.

Nehmen wir an, Sie wollen folgende Abbildung erstellen:



Fassen wir die einzelnen Zeichensequenzen in einem Programm zusammen:

Beachten Sie, daß sie Wörter oder Buchstaben, welche mit dem Befehl DISP in die Anzeige gebracht werden, immer zuerst aufrufen! Wenn Sie zuerst bestimmte Zeichen mit PIXEL darstellen und dann eine Zeile mit DISP hineinschreiben, werden alle anderen Zeichen in dieser Zeile gelöscht.

1) Die unterstrichene Schrift

Dazu gestalten wir ein kleines Programm, das die Schrift mit dem Befehl DISP darstellt und dann mit PIXEL eine Linie einfügt. Dieses Programm

speichern wir unter der Variablen PIX4.

```

« CLLCD                Löschen der Anzeige
"      I              YOU      Schrift, positioniert mit Leerzeichen
"                                als String
2 DISP -5 -2          Linke "Unterstreichungs-Linie"
  FOR n n -.5 R+C
PIXEL .1
  STEP 2 5
  FOR n n -.5 R+C      Rechte "Unterstreichungs-Linie"
PIXEL .1
  STEP DGTIZ          Aufrufen des interaktiven Modus, Programm-
>                                ende

```

'PIX4 |STO| Geben Sie DGTIZ nur beim HP 28 S ein!

2) Der Rand

Das nachfolgende Programm stellt mit PIXEL eine Umrandung des Anzeigefeldes dar. Speichern Sie dieses Programm unter der Variablen PIX5.

```

« CLLCD                Löschen der Anzeige
  -6.8 6.8            Oberer Rand
  FOR n n 1.6 R+C
PIXEL .1
  STEP -6.8 6.8
  FOR n n -1.5 R+C    Unterer Rand
PIXEL .1
  STEP -1.5 1.6
  FOR n n -6.8 SWAP   Linker Rand
R+C PIXEL .1
  STEP -1.5 1.6
  FOR n n 6.8 SWAP    Rechter Rand
R+C PIXEL .1
  STEP DGTIZ          Interaktiver Modus, Programmende
>

```

'PIX5 |STO|

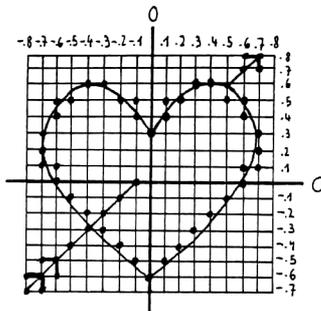
Achtung: Der Befehl DGTIZ darf nur in den HP 28 S eingegeben werden!

3) Die Erstellung des Bildes "Herz"

Das Darstellen von Bildern ist eine relativ aufwendige Angelegenheit und erfordert ein wenig Übung. Hier wird nun ein Herz dargestellt, welches zuerst graphisch entworfen und dann in die Anzeige eingepaßt werden muß. Dies geschieht Punkt für Punkt. Der Arbeitsablauf ist typisch für die Darstellung eines Bildes auf dem HP 28.

Am Anfang zeichnen Sie sich vergrößert ein Raster, das der Anzeige Ihres Taschencomputers entspricht (32 x 137 PIXEL). Das Raster können Sie dann für den Entwurf anderer Bilder kopieren.

Dann malen Sie in dieses Raster ein beliebiges Bild (im Beispiel ein Herz) hinein und markieren die Punkte oder PIXEL welche den gezeichneten Linien am nächsten sind. Sollte dieses Bild etwas verschoben oder verformt wirken, können Sie es etwas kleiner darstellen, um die Schärfe zu erhöhen.



Als nächstes übertragen wir jeden markierten Pixel in eine Liste, die alle Abbildungspunkte in komplexer Form enthalten soll z.B.:

$(-1,4)$ $(2,3)$ $(-2,5)$...

Dabei ist zu beachten, daß der Anzeigenmittelpunkt in der Grundstellung des Rechners die Koordinaten $(0,0)$ besitzt und die Koordinaten für PMIN und PMAX $(-6.8,-1.5)$ und $(6.8,1.6)$ lauten.

Wir gehen von dieser Voreinstellung aus und bezeichnen die Punkte des Rasters entsprechend.

Fassen Sie bestimmte Pixelfolgen, wenn möglich, mit bestimmten Programmsequenzen zusammen.

Im Beispiel geschieht dies mit der Programmschleife:

a b FOR n ... R→C PIXEL .1 STEP.

Das zugehörige Programm nennen wir PIX6 und lautet wie folgt:

PIXEL-Liste

(rechte Herzhälfte)

```
(0,.3)
(.1,.4)
(.2,.5)
(.3,.6) zusammenfassen
(.4,.6)
(.5,.5)
(.6,.5)
(.6,.4)
(.7,.3)
(.7,.2)
(.6,.1)
(.7,.1)
(.6,0)
(.5,-.1)
(.4,-.2)
(.3,-.3)
(.2,-.4) zusammenfassen
(.1,-.5)
(0,-.6)
```

```
< CLLCD
  0 .3
  FOR n n 0 + n .3 +
  R→C PIXEL .1
  STEP (.4,.6) PIXEL
  (.5,.5) PIXEL
  (.6,.5) PIXEL
  (.6,.4) PIXEL
  (.7,.3) PIXEL
  (.7,.2) PIXEL
  (.6,.1) PIXEL
  (.7,.1) PIXEL 0 .6
  FOR n .6 n - 0 n -
  R→C PIXEL .1
  STEP (.1,.5) PIXEL
```

Programmlisting

```
0 .3
  FOR n 0 n - n .3 +
  R→C PIXEL .1
  STEP (-.4,.6)
  PIXEL (-.5,.5) PIXEL
  (-.6,.5) PIXEL
  (-.6,.4) PIXEL
  (-.7,.3) PIXEL
  (-.7,.2) PIXEL
  (-.6,.1) PIXEL
  (-.7,.1) PIXEL 0 .6
  FOR n -.6 n + 0 n
  - R→C PIXEL .1
  STEP (-.1,.5)
  PIXEL -.8 -.1
  FOR n n DUP R→C
  PIXEL .1
  STEP (-.8,-.7)
  PIXEL (-.7,-.8)
  PIXEL (-.7,-.6)
  PIXEL (-.6,-.7)
  PIXEL .5 .8
  FOR n n DUP R→C
  PIXEL .1
  STEP (.7,.8) PIXEL
  (.8,.7) PIXEL DGTIZ
  >
```

Beachten Sie dabei, daß der Befehl DGTIZ nur im HP 28 S existent ist. Speichern Sie dieses Programm unter dem Namen PIX6 ab: 'PIX6 |STO|.

Nun haben Sie alle Voraussetzungen für die Darstellung des gewünschten Bildes auf Ihrem Rechner geschaffen. Sie können die entsprechenden Programme beliebig miteinander kombinieren und immer neue Bilder entstehen lassen. Dabei ist die Art der Darstellung jedoch von Ihrem Rechnertyp abhängig:

Darstellung mit dem HP 28 C

Sind Sie im Besitz des HP 28 C verknüpfen Sie die einzelnen Zeichenprogramme in einem Neuen Programm, in beliebiger Kombination. Mit diesem Taschencomputer müssen aufeinanderfolgende Bilder leider immer neu gezeichnet werden. Dies ist eine relativ zeitaufwendige Prozedur.

Das folgende Programm erstellt das komplette Herzbild von Seite 34. Vorher ist es jedoch nötig, den Befehl CLLCD aus den Programmen PIX5 und PIX6 zu löschen, da es nur im ersten Programm notwendig ist die Anzeige des HP 28 C zu löschen. Außerdem stellen Sie bitte sicher, daß der Befehl DGTIZ nicht in den Programmen vorkommt. Anschließend geben Sie das Programm ein:

```
<<| |USER| PIX4 PIX5 PIX6 |ENTER|
      . . . . .
```

Speichern Sie das Programm in der Variablen HERZ: 'HERZ |STO|.

Jetzt können Sie durch das Betätigen der Menütaste HERZ die Abbildung erstellen.

Darstellung mit dem HP 28 S

Mit dem HP 28 S ist eine Darstellung aufeinanderfolgender Bilder viel einfacher möglich. Lassen Sie die Programme PIX4 bis PIX6 einzeln zeichnen und digitalisieren Sie die Abbildungen wie in Kapitel 1.3.3.2. auf S.31, Beispiel 2. Speichern Sie die so gewonnenen Strings in den Variablen A4 bis A6. Danach können Sie die Programme PIX4 bis PIX6 wieder löschen um Speicherplatz zu sparen.

Nun können Sie in einem einfachen Programm beliebige Bildsequenzen erzeugen.

Wenn Sie das gesamte Herzbild wie auf Seite 34 darstellen wollen, verbind-

den Sie die Abbildungsstrings A4 bis A6 mit der logischen Verknüpfung OR und bilden den Ergebnisstring mit →LCD ab:

```
|USER| A4 A5 A6 █ |BINARY| |NEXT| |NEXT| |NEXT| OR OR →LCD
|..| |..| |..| |..| |..| |..| |..| |..|
```

Sie haben aber auch die Möglichkeit, bewegte Bilder darzustellen.

Mit dem Programm HBLK (Herzblinken) stellen Sie die Bilder A4 bis A6 in schneller abwechselnder Folge dar:

Unterprogramm PX:

```
« CLLCD A2 A1 A3 OR
OR →LCD .3 WAIT
CLLCD A2 A1 OR →LCD
.3 WAIT CLLCD A2 A3
OR →LCD .3 WAIT A2
A1 A3 OR OR →LCD
»
```

'PX |STO|

Hauptprogramm HBLK:

```
« 1 5
START PX
NEXT
»
```

'HBLK |STO|

Sie können das Programm HBLK starten, wenn Sie das User-Menü anwählen und die Menü-Taste HBLK betätigen.

Weitere Beispiele der Abbildung bewegter Bilder sind ein Hase, der mit den Ohren wackelt, ein Wurm der über die Anzeige kriecht, Laufschrift, sogar kleine Spielsequenzen sind mit dem HP 28 S möglich. Ihrer Fantasie sind keine Grenzen gesetzt.

1.3.4. Nullstellensuche bei Funktionen

Der HP 28 S/C bietet Ihnen im Solve-Menü drei Routinen zum Aufsuchen von Nullstellen an: QUAD, ROOT und den "Löser", der mit █ Variable gestartet wird.

Das Auffinden von Nullstellen mit QUAD

Die Prozedur QUAD gibt einen symbolischen Ausdruck für die zwei Nullstellen einer quadratischen Funktion zurück. Dabei wird eine Näherung der Nullstelle über eine Taylorreihe zweiten Grades berechnet.

Beispiel: $X^2 - 4X - 1$

1) Geben Sie die Funktion und die Lösungsvariable (hier X) ein.

'X^2-4*X-1 |ENTER| 'X |ENTER|

2) Lösen Sie den Ausdruck mit: |SOLV| QUAD
.....

Als Ergebnis erhalten Sie: '(4+s1*4.472135955)/2'

Die Variable s1 steht für ein beliebiges Vorzeichen. Sollten Sie als Ergebnis nur die Zahl 4.2360679775 erhalten ist der Flag 34 gesetzt. Das heißt, diese Lösung ist der numerische Hauptwert der gesuchten Nullstelle, jedoch ohne Berücksichtigung des Vorzeichens.

3) Um die vollständige Lösung des symbolischen Hauptwertes zu erhalten, speichern wir zuerst -1 und dann +1 in der Variablen s1 und werten den Ausdruck jeweils aus.

1 |CHS|, |LC| 's1 |STO| |EVAL| Ergebnis: 4.2360679775

Holen Sie die symbol. Nullstelle wieder in den Stack: █ |UNDO|

1,'s1 |STO| |EVAL| Ergebnis: -.2360679775

Die Koordinaten der Nullstellen sind somit (4.236,0) und (-.236).

Die Lösung mit ROOT

Die ROOT-Lösungsroutine benötigt im Stack drei Objekte zum Bestimmen einer Nullstelle:

Ebene 3: Programm, Ausdruck oder Funktion

Ebene 2: Variablenname

Ebene 1: Anfangsnäherung

Beachten Sie, daß das Objekt in Ebene 3 nur eine Variable beinhaltet!

Als Anfangsnäherung können Sie einen Schätzwert für eine Nullstelle, eine Liste mit zwei Zahlen als Intervall a bis b oder eine Liste mit drei Zahlen, wobei die erste die Näherung für die Nullstelle und die anderen zwei Zahlen das Intervall darstellen, vorgeben.

Eine Anfangsnäherung erhalten Sie entweder durch eine Abschätzung des Objekts in Ebene 3 oder durch eine graphische Darstellung wie in Kapitel 1.3.3.1. und der Übernahme markanter Punkte in den Stack.

Beispiel: $X^4 + 3X^3 + 2X^2 + X - 1$

Nehmen wir an, daß Sie über den Verlauf der Funktion bzw. deren Graphen noch keinerlei Aussage machen können.

Man kann sagen, daß sich die meisten Funktionen in Schule und Studium im Bereich von $-5 \leq x \leq 5$ bewegen und von einer Maßeinheit zur anderen nicht mehr als eine Nullstelle besitzen.

Wenn Sie dies berücksichtigen, können Sie ganz leicht ein einfaches Programm gestalten, welches Ihnen alle Nullstellen einer Funktion im Intervall $[-5 \ 5]$ berechnet. Dazu unterteilen Sie das Intervall in zehn Abschnitte (Teilintervalle) und aktivieren für jeden Abschnitt die Routine ROOT. Bedenken Sie aber, daß ROOT auf ein anderes Intervall übergreift, wenn im vorgegebenen Intervall keine Nullstelle existiert. Sie erhalten also bei dem nachfolgendem Programm zehn Werte für Nullstellen, wovon aber nur einige voneinander verschieden sind. Die anderen wurden von ROOT doppelt berechnet, da im vorgegebenen Intervall keine Nullstelle existierte und zur Berechnung das Intervall erweitert wurde.

Programm "Lösen":

```
<<                               Programmanfang
-5 4 FOR n                         n im Intervall [-5 4]
EQ                               Funktion aufrufen
'X'                               X in den Stack
n n 1 + 2 →LIST                   Anfangsnäherung: Intervall [n n+1]
ROOT                             Aktivieren des Lösers
NEXT                              nächstes n
8 FIX >>                          8 Nachkommastellen und Programmende

|ENTER| 'Lösen |STO|
```

Wenn Sie nun die Funktion von Seite 40 berechnen wollen, gehen Sie so vor:

```
'X^4+3*X^3+2*X^2+X-1 |ENTER| |USER| Lösen
.....
```

```
Ergebnis:                      -2.41421356
                                -2.41421356
                                -2.41421356
                                0.41421356
```

```

0.41421356
0.41421356
0.41421356
0.41421356
0.41421356
0.41421356

```

Wie Sie erkennen können, sind nur zwei Werte relevant: 0.41421356 und -2.41421356. Das sind die zwei Nullstellen der Funktion.

Mit dem Programm 3.3.14 'root' wird das Programm "Lösen" noch erweitert, indem alle überflüssigen Werte gelöscht werden.

Der "Löser"

Mit dem Löser haben Sie die Möglichkeit, Gleichungen, Funktionen oder beliebige Ausdrücke neu anzuordnen bzw. nach einer Variablen aufzulösen. Danach werden bestimmte Werte des Ausdrucks exakt berechenbar. Beziehen Sie sich auch auf das Kapitel 1.3.3.1. Punkt 6 bis 11.

Für die Berechnung einer Nullstelle mit dem Löser muß der Ausdruck mit STEQ in der Variablen EQ gespeichert werden und, zum schnelleren Auffinden des Ergebnisses, eine Anfangsnäherung in X übergeben werden.

Beispiel:

Berechnen Sie für 'Z*X-S+U*Y-N*X=Z-X' eine Lösung für X, wenn Z=1, S=2, U=3, Y=4 und N=5 ist.

```
'Z*X-S+U*Y-N*X=Z-X |ENTER| |SOLV| STEQ SOLVR
      .... ..
```

Im Löser-Menü sind nun alle Variablen verzeichnet, die in der Gleichung vorkommen. Übergeben Sie die oben genannten Werte in die Variablen Z bis N.

```
1 Z 2 S 3 U 4 Y 5 N
```

lösen Sie den Ausdruck nach X: \blacksquare X Ergebnis: 3.0

Dieser Wert erfüllt beide Seiten der Gleichung EQ.

Mit |USER| { | α | N Y U S Z | α | \blacksquare |PURGE| löschen Sie wieder alle Variablen.

Beispiel:

Berechnen Sie alle Nullstellen von $f(x,y) = x^4 + 3x^3 + 2x^2 + x - 1 - y$.

' $x^4 + 3x^3 + 2x^2 + x - 1 - y$ |ENTER|

Isolieren Sie Y, um einen Ausdruck zu erhalten der nur X als Variable enthält: 'Y |SOLV| ISOL

Speichern Sie diese Funktion und aktivieren Sie den Löser nach Eingabe einer Näherung für die Nullstelle.

Eine Näherung erhalten Sie wie in Kapitel 1.3.3.1. durch eine zeichnerische Darstellung oder durch abschätzen des Terms.

STEQ SOLVR { 0.4,0,1 X ■ X

Ergebnis: 0.41421356 ZERO d.h. die Nullstelle ist exakt

-2,-3,-1 X ■ X

Ergebnis: -2.4142135 SIGN REVERSAL d.h. die Nullstelle konnte nicht exakt ermittelt werden. Stattdessen wurden zwei Punkte mit unterschiedlichem Vorzeichen ermittelt in denen die Nullstelle enthalten ist. Der zurückgegebene Wert stellt den Punkt dar, dessen Y-Wert näher bei 0 liegt.

In Ausnahmefällen kann bei der Aktivierung des Löasers ein Fehler auftreten. Dies führt zu einer Fehlermeldung.

BAD GUESS(ES) Eine oder beide Anfangsnäherungen liegen außerhalb des Wertebereiches der Funktion.

CONSTANT? Für jeden berechneten Punkt ergibt sich der gleiche Wert für die abhängige Variable.

1.3.5. Lösen von Gleichungssystemen

Lineares System

$$2x + y + 3z = 1$$

$$5x + 6y - z = 2$$

$$-2x + y + 3z = 3$$

1) Konstantenvektor eingeben:

[1,2,3 |ENTER|

2) Koeffizientenmatrix eingeben:

[[2,1,3[5,6,-1[-2,1,3 |ENTER|

3) Lösung berechnen:

|÷|

Ergebnis: [-.5 .82 .4] x=-0.5, y=0.82, z=0.4

Komplexe lineare Gleichungssysteme werden nach dem gleichen Verfahren gelöst.

Zur Verbesserung der Genauigkeit geben Sie das Programm im Referenzhandbuch Seite 112 ein.

Nichtlineare Systeme

Beziehen Sie sich bei der Berechnung auf Kapitel 2.5.4. Programm zum inhomogenen Gleichungssystem.

1.3.6. Integrieren und Differenzieren

Differenzieren

1) Schrittweises Differenzieren

Differenzieren Sie $3\sqrt{\cos(1-x^2)}$ nach x:

|'| █ |d/dx|X|(| 3*√cos(1-x^2 |ENTER| → '∂X(3*√cos(1-X^2))'

|EVAL| - erste Auswertung/Differentiation, der konstante Faktor wird vor das Differential geholt. → '3*∂X(√cos(1-X^2))'

|EVAL| - zweite Auswertung, hier wird die Anwendung der Kettenregel deutlich.

Die Ableitung einer Funktion $y=f|u(x)|$ ist gleich dem Produkt der Ableitungen von äußerer und innerer Funktion.

$\frac{d}{dx} 3\sqrt{\cos(1-x^2)} = 3 \cdot \frac{d}{dx} \cos(1-x^2) \cdot \frac{1}{2\sqrt{\cos(1-x^2)}}$; hier wurde nur die Wurzel ausgewertet.

→ '3*(∂X(cos(1-X^2))/(2*√cos(1-X^2)))'

|EVAL| → dritte Auswertung, in diesem Schritt wurde auch die Cosinus-

Funktion, also die innere Funktion der Wurzel differenziert.

$$\frac{d}{dx} 3 \sqrt{\cos(1-X^2)} = 3 * (-\sin(1-X^2)) * \frac{\frac{d}{dx}(1-X^2)}{2 * \sqrt{\cos(1-X^2)}})$$

$$\rightarrow '3 * (-\sin(1-X^2)) * \partial X(1-X^2) / (2 * \sqrt{\cos(1-X^2)})''$$

[EVAL] - die 1 im letzten Term ist eine Konstante und verschwindet.

$$\rightarrow '3 * (-(-\sin(1-X^2)) * (-\partial X(X^2))) / (2 * \sqrt{\cos(1-X^2)})''$$

[EVAL] - fünfte Auswertung, die innere Funktion vom Sinus wurde differenziert.

$$\rightarrow '3 * (-(\sin(1-X^2)) * (-(\partial X(X) * 2 * X^{(2-1))})) / (2 * \sqrt{\cos(1-X^2)})''$$

[EVAL] - diese Auswertung gibt das vollständige Ergebnis zurück.

$$\frac{d}{dx} 3 \sqrt{\cos(1-X^2)} = \frac{3 * X * \sin(1-X^2)}{\sqrt{\cos(1-X^2)}} = y' = f' |u(x)|$$

$$\rightarrow '3 * (-(\sin(1-X^2)) * (-2 * X)) / (2 * \sqrt{\cos(1-X^2)})''$$

2) Vollständiges Differenzieren

Sie können die am Anfang genannte Funktion auch sofort vollständig differenzieren, wenn Sie die Funktion in den Stack eingeben, die Variable definieren nach der differenziert werden soll und dann die Tasten \blacksquare und $|d/dx|$ betätigen.

'3 * $\sqrt{\cos(1-X^2)}$ |ENTER| 'X |ENTER| \blacksquare |d/dx| - Sie erhalten sofort die vollständige Ableitung. Diese stimmt mit dem obigen Ergebnis überein.

3) Benutzerdefinierte Funktionen und ihre Ableitung

Benutzerdefinierte Funktionen und Ihre Ableitungen haben große Bedeutung für die Programmierung des HP 28 S/C. Wenn Sie in einem Programm solche Funktionen verwenden hat das den Vorteil, daß das eigentliche Programm sehr übersichtlich bleibt und gegebenenfalls sehr schnell durchlaufen werden kann. Außerdem können benutzerdefinierte Funktionen in mehreren verschiedenen Programmen Verwendung finden.

Beispiel: Eine Funktion $F(x,y,z)$ soll lauten: $F(x,y,z) = x*y*z$

◀▶ x y z 'x*y*z' |ENTER| 'F |STO|

Differenzieren Sie dann $F(X, X^2, X^3 - X)$ nach X.

Dabei wird X für x, X^2 für y und $X^3 - X$ für z substituiert. Anschließend wird der neue Term differenziert und in den Stack zurückgegeben.

'F(x,y,z)', 'X |ENTER| $\frac{d}{dx}$ |

Ergebnis: ' $(X^2 + X * (2 * X)) * (X^3 - X) + X * X^2 * (3 * X^2 - 1)$ '

Dieses Ergebnis kann noch zusammengefaßt werden: $\frac{d}{dx}$ |ALGEBRA| COLCT COLCT

Ergebnis: ' $3 * (X^3 - X) * X^2 + (-1 + 3 * X^2) * X^3$ '

Dieser Ausdruck kann mit dem Programm 3.2.2. Vollständiges Zusammenfassen noch verkürzt werden. Sie können aber auch durch mehrfaches Erweitern des Terms mit EXPAN und anschließendem Zusammenfassen mit COLCT einen vereinfachten Ausdruck generieren. Benutzen Sie dabei die Funktion EXPAN und COLCT solange, bis sich am Ausdruck nichts mehr verändert.

Ergebnis: ' $-(4 * X^3) + 6 * X^5$ '

ACHTUNG!

Das Differenzieren größerer oder komplizierterer Ausdrücke und das Erweitern bzw. Zusammenfassen der Ergebnisse benötigt oft sehr viel Speicherplatz! Versuchen Sie deshalb, vor allem mit dem HP 28 C, den Speicher Ihres Rechners von nicht mehr benötigten Objekten freizuhalten. Unter Umständen ist es sinnvoll die Funktionen COMMAND (CMD), UNDO und LAST zu deaktivieren.

Falls Sie eine "Insufficient Memory" Meldung erhalten bedenken Sie, daß bei deaktiviertem LAST die Argumente der letzten Operation verlorengehen.

Beziehen Sie sich auch auf das Kapitel 1.3.7. Speicher zu klein, was tun?.

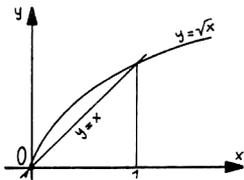
Integrieren

1) Numerische Integration

Bei der numerischen Integration sollte der Stack folgendermaßen aufge-

4) Symbolische Berechnung von Mehrfachintegralen

Beispiel 1:



Berechnen Sie die Fläche, die von den Kurven $y=x$ und $y=\sqrt{x}$ begrenzt wird.

Dazu ist das in die Fläche gelegte Flächendifferential $dx dy$ zuerst in Richtung der y -Achse von $y=x$ bis $y=\sqrt{x}$ zu integrieren. Das

so erhaltene Fadenintegral ist dann noch in den Grenzen $x=0$ bis $x=1$ zu berechnen. Es ergibt sich somit ein Doppelintegral:

$$A = \int_{x=0}^1 \int_{y=x}^{\sqrt{x}} dy dx$$

Gehen Sie bei der Berechnung so vor:

Zwischen den Integralzeichen und dem Integranden steht eine nicht geschriebene Konstante mit dem Wert 1. Sie können das Integral deshalb auch etwas anders schreiben:

$$A = \int_{x=0}^1 \int_{y=x}^{\sqrt{x}} c dy dx \quad c=1$$

Für die Berechnung des Integranden $c dx dy$ geben sie die Konstante, die Integrationsvariable und den Grad des Polynoms in den Stack und integrieren diese Konstante.

'C', 'Y', 1 |ENTER| $\int \int$

Sie erhalten: 'C*Y'

Das Integral sieht nun so aus: $A = \int_{x=0}^1 c*y \Big|_x^{\sqrt{x}} dx$

Definieren Sie C als 1 und berechnen Sie das innere Integral.

1 'C |STO| |EVAL| Ergebnis: 'Y'

Obere Grenze eingeben und auswerten: ' \sqrt{X} ', 'Y |STO| Ergebnis: ' \sqrt{X} '

Untere Grenze eingeben u. auswerten: |LAST| 'X', 'Y |STO| \rightarrow 'X'

Obere Grenze minus untere Grenze: |-| Ergebnis: ' $\sqrt{X}-X$ '

Das Integral sieht nun so aus: $A = \int_{x=0}^1 [\sqrt{X}-X] dx$. Dies entspricht der Lösung des inneren Integrals.

Bestimmen Sie das äußere Integral. Diese Berechnung können Sie auf numerischem Weg durchführen, da die Grenzen keine Variablen enthalten.

{X 1 0 |ENTER| .0001 |ENTER| █ |}

Ergebnis: 2: -.166666746919

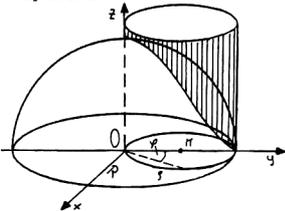
1: 1.67317346614E-4

Wenn Sie wissen wollen, welcher Bruch dem Zahlenwert $-.16\bar{6}$ entspricht löschen Sie den Fehler und betätigen █ |1/x|.

Sie erhalten $-5.9\bar{9}$, das entspricht einem sechstel.

Die Lösung lautet also: $A = \frac{1}{6}$

Beispiel 2:



Berechnen Sie das den Körpern gemeinsame Volumen, wenn eine Halbkugel mit Radius R von einem Zylinder mit Radius $R/2$ durchdrungen wird. Der Kugelmittelpunkt liegt auf dem Zylindermantel.

Arbeiten Sie mit Zylinderkoordinaten.

Es ergibt sich das Dreifachintegral oder Volumenintegral:

$$\frac{V}{2} = \int_{\varphi=0}^{\pi/2} \int_{\varphi=0}^{\varphi} \int_{z=0}^{\sqrt{R^2-\varphi^2}} \varphi \, dz \, d\varphi \, d\psi.$$

Da wir auf dem HP 28 nur μ als griechisches Zeichen zur Verfügung haben, definieren wir φ als phi und φ als ro.

Lösen Sie das Integral:

1. Innere Integration:

Geben Sie den Integranden, die Variable und den Grad in den Stack und berechnen Sie den ersten Ausdruck:

'ro','Z',1 |ENTER| █ |}

Ergebnis: 'ro*Z'

Setzen Sie die Grenzen ein und werten Sie den Ausdruck aus.

Beachten Sie, daß Sie die Grenzen in Potenzform eingeben, wenn Wurzelaus-

drücke darin vorkommen! Wurzeln werden bei der Integration normalerweise nicht verarbeitet.

'(R^2-ro^2)^(1/2)', 'Z |STO| |EVAL|

Ergebnis: 'ro*(R^2-ro^2)^(1/2)'

$$\frac{V}{2} = \int_{\varphi=0}^{\pi/2} \int_{\rho=0}^{R \cdot \cos \varphi} \rho \sqrt{R^2 - \rho^2} \, d\rho \, d\varphi$$

Dieses Integral läßt sich nicht mit einfachen Mitteln lösen. Deshalb führen wir eine Substitution durch, um den Integranden zu vereinfachen.

Definieren Sie: $R^2 - \rho^2 = T$.

Auch die Variable $d\rho$ muß gegen einen Ausdruck mit T ausgetauscht werden.

Dies erreicht man durch eine Differentiation des Ausdrucks T nach $d\rho$.

Sie erhalten für $\frac{dT}{d\rho}$ den Term: $-2 \cdot \rho$ d.h. $d\rho = -\frac{dT}{2\rho}$.

Passen Sie auch die Grenzen von ρ an den neuen Integranden an. Dazu setzen Sie die alten Grenzen in den Ausdruck für T ein. Das neue Integral lautet dann:

$$V = \int_{\varphi=0}^{\pi/2} \int_{t=R^2}^{R^2 \cdot \sin^2 \varphi} \frac{\sqrt{t}}{2} \, dt \, d\varphi \longrightarrow V = \int_{\varphi=0}^{\pi/2} \left. \frac{t \cdot \sqrt{t}}{3} \right|_{R^2}^{R^2 \cdot \sin^2 \varphi} \, d\varphi$$

Geben Sie das neue Integral ein und berechnen Sie es innerhalb der neuen Integrationsgrenzen.

'T^(1/2)/3 |ENTER| 'R^2*SIN(phi)^2 |ENTER| 'T |STO| |ENTER| |EVAL|

█ |SWAP| 'R^2 |ENTER| 'T |STO| |EVAL| |-|

Kürzen Sie den Term mit dem Programm ZFASS oder durch mehrmaliges Erweitern und anschließendem wiederholten Zusammenfassen. Ändern Sie den Modus auf zwei Nachkommastellen.

|USER| ZFASS 2 █ |MODE| FIX
.....

Ergebnis: '0.33*SIN(phi)^3*R^3-(0.33*R^3)' $V = \int_0^{\pi/2} \left(\frac{R^3 \cdot \sin^3 \varphi}{3} - \frac{R^3}{3} \right) d\varphi$

Für die letzte Berechnung können wir ein numerisches Verfahren anwenden. Dazu ist es sinnvoll $\frac{R^3}{3}$ auszuklammern.

'R^3*(1/3)^-1 |ENTER| |*| |USER| ZFASS
.....

die Befehlszeile gespeichert. Befanden Sie sich im Editier-Modus und COMMAND ist aktiv, löschen Sie das ursprüngliche Objekt und betätigen **|COMMAND|**.

LOW MEMORY! = Zu kleiner Speicher

Es sind weniger als 128 Bytes freier Speicherplatz vorhanden. Löschen Sie alle nicht mehr benötigten Objekte.

NO ROOM TO SHOW STACK = Kein Platz für Stackanzeige

Es werden nur noch Objekttypen angezeigt, welche die im Stack gespeicherten Objekte darstellen. Löschen Sie einige Objekte oder speichern Sie sie in Variablen.

OUT OF MEMORY = Kein Speicherplatz mehr vorhanden

Es ist kein freier Speicher mehr vorhanden, um selbst Systemoperationen durchzuführen.

Der Taschencomputer stellt Ihnen eine Prozedur zum Löschen verschiedener Objekte oder Argumente zur Verfügung:

Ihr Rechner fragt Sie der Reihe nach, ob folgendes gelöscht werden soll:

COMMAND-Stack falls aktiv

UNDO-Stack falls aktiv

LAST-Funktion falls aktiv

STACK

Jede einzelne Benutzervariable

Falls Sie Elemente löschen wollen, betätigen Sie die Menü-Taste YES, wollen Sie sie behalten, drücken Sie NO.

Nach dem Löschen von mindestens einer Funktion können Sie versuchen, die Prozedur durch betätigen von **|ATTN|** abubrechen.

2. Die Programmierung des HP 28 S/C

2.1. Der Aufbau von Programmen

Ein Programm im HP 28 beginnt immer mit dem Zeichen « .

Ein Programm kann mehrere Unterprogramme enthalten, welche entweder auch mit dem Zeichen« beginnen oder in einem Namen enthalten sind.

Die sehr objektorientierte Programmiersprache des HP 28 stellt ein nahezu ideales Werkzeug zum Bearbeiten komplexer mathematischer Aufgaben dar. Allerdings verlangt die Gestaltung von Programmen etwas mehr Aufmerksamkeit und Logik als bei der Programmierung in BASIC, welche ja in fast allen anderen Taschencomputern zu finden ist. So gibt es im HP 28 keine Befehle, welche ein "wildes Herumspringen" in Programmen erlauben, wie z.B. die Befehle GOTO oder GOSUB in der Programmiersprache BASIC. Stattdessen existiert eine Reihe von bedingten Verzweigungen, wie IF...THEN.. ELSE...END oder IFERR...THEN...END, die nacheinander durchlaufen werden müssen. Außerdem hat jedes Programm im HP 28 nur einen Ein- und Ausgang. So bleiben die Programme logisch, sind relativ einfach zu schreiben und zu testen. Wenn Sie am Anfang eines Programms alle darin vorkommenden Funktionen in Variablen definieren, bleibt das Programm sehr klein und übersichtlich.

Belohnt wird Ihre Mühe durch eine höhere Verarbeitungs- und Rechengeschwindigkeit und durch eine größere Übersichtlichkeit von Programmen gegenüber einem Rechner, der mit BASIC arbeitet.

Beachten Sie also bei der Programmierung folgende Grundregeln:

1. Programmieren Sie strukturiert, d.h. gestalten Sie ein Programm so, daß die Programmanweisungen nacheinander durchlaufen werden.
2. Vermeiden Sie Sprungstellen, wie z.B. geschachtelte Verzweigungen.
3. Definieren Sie zu Beginn des Programms alle benötigten Variablen und Funktionen.
4. Schreiben Sie Unterprogramme nach Möglichkeit nicht ins Hauptpro-

gramm, sonder extern in Variable.

5. Definieren Sie keine Endlosschleifen.
6. Löschen Sie am Programmende alle nicht mehr benötigten Variablen und Funktionen.
7. Rufen Sie am Programmende alle Modi wieder auf, in denen sich der Rechner vor Beginn des Programms befand.
8. Wenn Sie Unterprogramme mit \ll innerhalb des Hauptprogrammes aufrufen und Variablen benutzen, die außerhalb dieses Unterprogramms als lokale Variablen definiert wurden, werden diese nicht erkannt. Verwenden Sie in diesem Fall nur normale Variablen.

Sie haben zwei Möglichkeiten, ein Programm in Ihrem Rechner ablaufen zu lassen:

1. Sie geben die für das Programm benötigten Werte in den Stack und rufen das Programm auf.
2. Sie rufen ein Programm auf, welches eine qualifizierte Meldung in den Stack gibt und dann die Eingabe der benötigten Werte verlangt. Mit \blacksquare |CONT| (CONTINUE) starten Sie das Programm erneut. Beziehen Sie sich auch auf das Kapitel 2.2. Anzeigen von Strings in Programmen, Meldungen.

Beispiel 1:

Dieses schematische Programm verlangt die Eingabe zweier Werte in den Stack, bevor es gestartet wird. Diese Werte werden entweder in lokale Variablen oder mit |STO| in normalen Variablen gespeichert und am Programmende wieder gelöscht.

A.

```

« 2 FIX + a b           Programmbeginn, 2 Kommast., zwei Werte in a,b
« PROGRAMM             Programm als separate Einheit da lokale Varia-
»                       blen existieren
»
»

```

Die lokalen Variablen verschwinden am Programmende von selbst, da sie ja immer nur in dem Programm existieren, in dem sie aufgerufen wurden.

B.

```

< 2 FIX 'A' STO 'B'      Programmbeginn, 2 Kommastellen, Werte in A und B
STO PROGRAMM            Hauptprogramm
  < UNTERPROGRAMM       Unterprogramm
    > { A B } PURGE     Variablen löschen, Programmende
  >

```

Beispiel 2:

Dieses Programm (schematisch) erzeugt zu Programmbeginn eine Meldung und unterbricht die Programmausführung.

```

< CLLCD                 Programmbeginn, Löschen der Anzeige
"Enter a und b" 2      Meldung für Eingabe in Zeile 2
DISP HALT → a b       Programmunterbrechung, dann Werte in a und b
  < PROGRAMM           Programm
    >
  >                   Programmende

```

Die Anzeige würde nach einem Programmstart wie folgt aussehen.

Enter a und b

Geben Sie nach der Meldung die entsprechenden Werte in den Stack und setzen Sie das Programm mit **▣** |CONT| fort.

Sobald Sie nach Anzeigen der Meldung eine beliebige Taste drücken, wird die Meldungsanzeige gelöscht. Dies hat keinen Einfluß auf die Funktion des Programms.

2.2. Anzeigen von Strings in Programmen, Meldungen

Im Prinzip genügt es, die Eingangsparameter eines Programmes vor Programmbeginn in den Stack zu übergeben und dann das Programm zu starten. Dies ist vor allem beim HP 28 sinnvoll, da eine Meldung im Programm nur unnötige Speicherplatzverschwendung wäre.

Wenn Sie jedoch eine Meldung benötigen, finden Sie nachfolgend einige

grundlegende Beispiele:

1. Bei Programmstart soll der Name des Programms einige Zeit im Display erscheinen und dann durch eine Anzeige über die Eingangsparameter ersetzt werden. Nach der Eingabe dieser Werte und dem Betätigen von **|CONT|** soll das Programm normal durchlaufen werden.

« CLLCD	Löschen der Anzeige
" DEMO ONE " 2	Programmname in Zeile 2
DISP 4 WAIT CLLCD	4 Sekunden Programmunterbrechung
"Geben Sie die	Eingabemeldung Teil 1 in Zeile 1
Grundparameter ein"	
1 DISP "Enter a,b,c"	Eingabemeldung Teil 2 in Zeile 4
4 DISP HALT → a b c	Programmstop, 3 Werte in a,b und c
« 'a+b+c' EVAL	Berechnen von a+b+c
»	
»	Programmende

2. Bei Programmstart werden die Eingangsparameter verlangt und dann im Programm gefragt, ob diese Werte multipliziert oder dividiert werden sollen. Betätigen Sie dazu die Taste M oder D.

« CLLCD	Löschen der Anzeige
"Geben Sie die Grund-"	
1 DISP	Meldung Teil 1 in Zeile 1
'parameter a und b ein"	
2 DISP "Enter a,b" 4	Meldung Teil 2 in Zeile 2
DISP HALT → a b	Meldung Teil 3 in Zeile 4
«	Programmstop, 2 Werte in a und b
DO CLLCD	Beginn äußere Schleife:
"Multiplizieren oder"	Auswahlmeldungen anzeigen
1 DISP "Dividieren?"	
2 DISP	
"Druecke M fuer Multipl.	
"	
3 DISP	
"oder D fuer Dividieren"	
4 DISP	
DO	Beginn innere Schleife, solange wiederholen
UNTIL KEY	bis eine Taste gedrückt wird. Falls Taste
END	gedrückt gibt KEY einen String u. 1 zurück
UNTIL < "M" "D"	Beginn äußere Tetanweisung: Taste gedrückt?
» SWAP POS DUP	Taste definiert? und Liste mit definierten

Tasten. Vergleichen des Tasten-Strings mit den in der Liste definierten. POS gibt 1 für M und 2 für D und 0 für andere Taste zurück. DUP erstellt eine Kopie der Position.

```

    IF 1 ==
    THEN 'a*b'
EVAL
    ELSE 'a/b'
EVAL
    END SWAP
    END CLMF
  >
  »

```

Wenn gedrückte Taste = M Wahr-Flag zurück dann berechnen von a*b
Ausdruck auswerten
sonst berechnen von a/b
Ende IF-THEN-Struktur, falls Taste definiert ist Wahr-Flag im Stack. Ende äußere Schleife falls die gedrückte Taste definiert war endet die Anzeigeschleife .
CLMF aktiviert die normale Stackanzeige.

3. Nach der Berechnung der Summe zweier Zahlen soll das Ergebnis in der Anzeige so dargestellt werden: "Summe = 4".

Dabei wird das Ergebnis in einen String umgewandelt und mit dem String "Summe=" verknüpft. Anschließend wird der Ergebnisstring mit DISP angezeigt.

```

« → a b
« CLLCD "Summe= "
'a+b' EVAL →STR + 2
DISP
  >
  »

```

Die Werte a und b vom Stack in a,b
Anzeige löschen und Ergebnisstring in Stack a+b berechnen, in String umwandeln, mit 1. Teilstring verbinden und in Zeile 2 anzeigen
Programmende

2.3. Programmschleifen

Im HP 28 S/C werden Ihnen mehrere bedingte Verzweigungen und Schleifenprozeduren angeboten, deren Befehle im Programm Branch-Menü verzeichnet sind. Viele dieser Programmstrukturen basieren auf Test-Anweisungen, wobei ein Flag vom Stack genommen und auf seinen Status geprüft wird. Hat der Flag den Wert 0, ist er "falsch" oder "gelöscht", hat er einen anderen Wert, ist er "wahr" oder "gesetzt".

Jede Programmverzweigung ist davon abhängig, ob ein Flag den Wert 0 oder 1 besitzt.

Gibt z.B. das Auswerten der Test-Anweisung einer IF Test-Anweisung THEN

Wahr-Anweisung ELSE Falsch-Anweisung END einen "falsch"-Flag (0) zurück, so wird die Falsch-Anweisung ausgeführt.

Es gibt verschiedene Testbefehle, die in Test-Anweisungen Verwendung finden können. Der Befehl $a < b$ prüft z.B., ob das Argument a kleiner dem Argument b ist und gibt, falls dies zutrifft, einen "wahr"-Flag in den Stack zurück.

Weitere Testbefehle sind:

$>$, \leq , \geq , $==$, \neq , FS?, FC?, FS?C, FC?C .

Diese Befehle sind im Referenzhandbuch Seite 252 ausführlich beschrieben.

In diesem Buch will ich nur ganz kurz die wichtigsten Merkmale der verschiedenen Verzweigungs- und Schleifenprozeduren herausstellen.

```
IF Test-Anweisung THEN Wahr-Anweisung END
.....
```

```
IF X 0 < THEN NEG END
```

Diese Prozedur ändert das Vorzeichen eines Objekts, wenn X kleiner 0 ist.

Der Befehl THEN prüft den Flag auf seinen Status und führt die Wahr-Anweisung aus, falls der Flag den Wert 1 besitzt.

```
IF Test-Anweisung THEN Wahr-Anweisung ELSE Falsch-Anweisung END
.....
```

```
IF X 0 == THEN "Nullstelle" ELSE "Extremum" END
```

Diese Prozedur gibt den String "Nullstelle" in den Stack, wenn X gleich 0 ist. Ansonsten wird der String "Extremum" angezeigt.

```
IFT
...

```

IFT ist die abgekürzte Form von IF...THEN...END. IFT nimmt einen Flag von Ebene 2: und ein Objekt von Ebene 1:.. Ist der Flag "wahr", wird das Objekt ausgewertet; wenn nicht, wird es verworfen.

```
X 0 == "Nullstelle" IFT
```

Der String Nullstelle wird angezeigt, wenn X gleich 0 ist.

```
IFTE (IF...THEN...ELSE...END)
.....
```

IFTE nimmt einen Flag von Ebene 3 und zwei Objekte aus Ebene 1: und 2:.. Ist der Flag 1, wird die THEN-Prozedur ausgeführt. Ist der Flag 0, wird die Else Prozedur gestartet.

```
X 0 == "Nullstelle" "Extremum" IFTE      Erklärung siehe oben.
```

```
IFERR Prüf-Anweisung THEN Fehler-Anweisung END
.....
```

```
IFERR '1/X' EVAL THEN "Unendliches Ergebnis" 2 DISP END
```

Falls beim Auswerten von 1/X ein Fehler auftritt wird der String angezeigt.

Falls LAST aktiv ist, wird im Fehlerfall das Argument in den Stack zurückgegeben.

```
IFERR Prüf-Anweisung THEN Fehler-Anweisung ELSE Normal-Anweisung END
.....
```

```
IFERR '1/X' EVAL THEN "Unendliches Ergebnis" 2 DISP ELSE "OK" END
```

Der Unterschied zu obiger Prozedur liegt darin, daß der String "OK" ausgegeben wird, wenn bei der Berechnung von 1/X kein Fehler auftritt.

```
Anfang Ende START Schleifen-Anweisung NEXT
.....
```

```
1 5 START + NEXT
```

Fünf Werte werden vom Stack genommen und addiert.

```
Anfang Ende START Schleifen-Anweisung Schrittweite STEP
.....
```

```
1 6 START + 2 STEP
```

Der Befehl + wird dreimal ausgeführt.

```
Anfang Ende FOR Name Schleifen-Anweisung NEXT
.....
```

```
1 5 FOR n '√n' EVAL NEXT
```

In der Variable n werden die Werte 1 bis 5 gespeichert und nachfolgende Schleifen-Anweisung ausgeführt. Als Ergebnis erhalten Sie die Wurzeln von 1 bis 5.

```
Anfang Ende FOR Name Schleifen-Anweisung Schrittweite STEP
.....
```

```
1 7 FOR x '√x' EVAL 2 STEP
```

Sie erhalten die Wurzeln der Zahlen 1, 3, 5 und 7.

```
DO Schleifen-Anweisung UNTIL Test-Anweisung END
.....
```

```
DO X + UNTIL 0 == END
```

Die X-Werte werden solange addiert, bis X gleich 0 ist.

```
WHILE Test-Anweisung REPEAT Schleifen-Anweisung END
.....
```

```
WHILE n 0 ≠ REPEAT '1/n' EVAL END
```

Die Berechnung von $1/n$ wird solange wiederholt, bis die Variable n den Wert 0 annimmt.

2.4. Variable in Programmen

Mit dem HP 28 ist es möglich Programme zu schreiben, welche Variablen verarbeiten. Dies ist wohl der größte Vorteil des HP 28 Taschencomputers gegenüber anderen Rechnern.

Eine Berechnung algebraischer Objekte mit Variablen ist in Ihrem Taschencomputer mittels des ALGEBRA-Menüs möglich. Es gibt auch einige Befehle außerhalb dieses Menüs, die in der Lage sind symbolische Argumente zu verarbeiten, wie z.B. ∂ oder \int .

Wollen Sie aber normale Funktionen auf Variable in Programmen anwenden, müssen Sie einen etwas anderen Weg gehen.

Übergeben Sie die Variablen immer in eine Liste, da, wie wir uns erinnern, eine Liste aus einer Reihe willkürlicher Objekte bestehen kann. Mit den Befehlen LIST \rightarrow , \rightarrow LIST, PUT, GET, PUTI, GETI, SUB und SIZE können Sie dann willkürlich auf die einzelnen Objekte einer Liste zugreifen, neue Listen erstellen oder ändern.

Wenn Sie z.B. eine Variable aus einer Liste in den Stack übergeben, können Sie normale Berechnungen mit dieser Variablen durchführen und Sie anschließend wieder in einer Liste speichern.

Vergleichen Sie bitte das Programm 3.2.7. DIV (Divergenz).

In diesem Programm werden zwei Vektoren $a(r)$ und r in zwei Listen gespeichert:

$$| x y z | = a(r) \text{ in } \{ x y z \} \text{ und } | x y z | = r \text{ in } \{ x y z \}$$

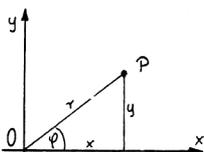
Dann werden die Objekte in der Liste a aus der Liste entnommen, in Variablen gespeichert (D,C,B) und diese dann einzeln differenziert. Die Ergebnisse werden addiert und wiederum in einer Variablen (di) abgelegt.

Anschließend werden die Objekte der Liste r entnommen und in Variablen gespeichert, die im Objekt 'di' vorkommen. Das Objekt 'di' wird abschließend ausgewertet und als Ergebnis in den Stack geschrieben.

3. Programme für den HP 28 S/C3.1. Graphische Abbildungen3.1.1. POLAR -- Kurven in Polarkoordinaten zeichnen

Im Cartesischen Koordinatensystem wird die Lage eines Punktes P durch seine Abstände zu den beiden Achsen (Abszisse und Ordinate) bestimmt. Funktionen höheren Grades, welche eine Drehbewegung beschreiben, können durch die Einführung eines neuen Koordinatensystems in Ihrer Darstellung oft wesentlich vereinfacht werden.

Dieses Koordinatensystem nennen wir Polarkoordinatensystem. Im Gegensatz zum rechtwinkligen System soll P jetzt durch seine Entfernung r von einem festen Punkt O und einem Winkel φ , den eine durch O und P gelegte Gerade mit einer festen Achse bildet, definiert werden.



r und φ heißen Polarkoordinaten des Punktes P.

Die feste Achse nennt man Polarachse.

r ist immer positiv zu nehmen.

Die Beziehungen zwischen cartesischen und Polarkoordinaten ist folgende:

$$X = r \cos \varphi \quad Y = r \sin \varphi \quad r = \sqrt{x^2 + y^2} \quad \tan \varphi = \frac{Y}{X} \quad 0^\circ \leq \varphi \leq 360^\circ$$

Das folgende Programm verlangt die Eingabe einer Funktion von r und dem Intervall in dem die Funktion dargestellt werden soll.

Anschließend setzt es die φ -Werte in r ein und stellt die so ermittelten Punkte graphisch dar.

```
« CLLCD
```

```
"ENTER R(phi),<INT>"
```

Eingabe von $r(\varphi)$ und {Intervall von bis}

```
2 DISP HALT → r 1
```

```
« DEG CLLCD DRAX 1
```

```
LIST→ DROP
```

```
FOR i i 'phi'
```

```
STO r EVAL phi R+C
```

```
P+R PIXEL 5
```

```
STEP r 1 DGTIZ {
```

Eingangsparameter in den Stack, Starten des interaktiven Abbildungsmodus

```
FPAR phi } PURGE
```

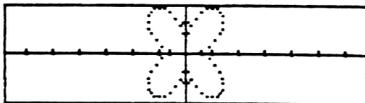
```
»
```

```
»
```

Speichern Sie dieses Programm unter dem Namen POLAR ab.

Beispiel: $r=2*\sin(2\varphi)$ im Intervall $[0^\circ \ 360^\circ]$

|USER| POLAR '2*SIN(2*phi)',{0 360}|ENTER| ■ |CONT|



Sie können jetzt jeden Punkt der Anzeige oder beim HP 28 S auch die gesamte Anzeige digitalisieren und abrufen. Sollte Ihnen die Anzeige ungenügend sein, ändern Sie den Maßstab mit *H, *W, PMIN oder PMAX.

Falls Sie keine Anzeige erhalten, prüfen Sie, ob Sie den Parameter φ genau wie oben in die Funktion geschrieben haben – φ muß klein geschrieben werden.

3.1.2. PMTR – – – Kurven in Parameterdarstellung zeichnen

Bei vielen technisch wichtigen Kurven treten die Veränderlichen oft in hohen Potenzen auf und bilden einen komplizierten algebraischen Ausdruck. Deshalb ist eine Darstellung sehr schwierig oder unmöglich, wenn man sie in rechtwinkligen Koordinaten angibt.

Deshalb kann man einen Kurvenverlauf auch durch zwei Gleichungen, abhängig von einem Hilfsparameter z.B. t ausdrücken. Diese Darstellung heißt PARAMETERDARSTELLUNG.

Vorteilhaft ist diese Darstellungsart bei der Bewegung von Körpern, wenn diese von der Zeit t abhängt. Die Parameterdarstellung von Kurven kann durch Elimination der Veränderlichen in eine Darstellung in rechtwinkligen Koordinaten umgewandelt werden.

Das Programm PMTR berechnet für zwei Gleichungen $x(t)$ und $y(t)$ Kurvenpunkte im Intervall von t .

Die Eingangsparameter werden am Programmende wieder in den Stack geschrieben, um sie vielleicht später weiterverwenden zu können.

```

< CLLCD DEG
"ENTER x(t),y(t),(t,INT)
"
2 DISP HALT → x y 1
  < 1 LIST→ DROP
SWAP DUP ROT - -99 /
0 99 CLLCD DRAX
  START SWAP ROT
STO x →NUM y →NUM
R→C PIXEL 1 1 GET
DUP →NUM ROT + LAST
SWAP DROP
  NEXT DROP2 PURGE
x y 1 250 .07 BEEP
DGTIZ
  >
>

```

Eingabe von x(t), y(t),
 {Parameter, Intervall von bis}
 Programmstop (weiter mit CONT)

Eingangparameter in den Stack, Akustiksig-
 nal, interaktiver Modus

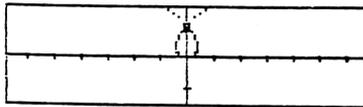
Beispiel:

Stellen Sie die Kurve, gegeben durch $x(t)=t-t^3$ und $y(t)=t^2$ im Intervall von -3.14 bis 3.14 dar.

```

|USER| PMTR 'T-T^3','T^2',{T,-3.14,3.14} |ENTER| █ |CONT|
.....

```



3.1.3. ST.P Neuer Abbildungsmaßstab, doppelte Geschwindigkeit

Das folgende kleine Programm verdoppelt die Höhe und Weite einer Abbildung und ändert die Auflösung des Graphen auf zwei PIXEL.

Die erforderlichen Parameter werden in PPAR gespeichert.

```

< (-13.6,-3) PMIN
(13.6,3.2) PMAX 2
RES
>

```

Um die neuen Abbildungswerte in PPAR zu speichern, betätigen Sie die Tasten |USER| und ST.P .

3.1.4. EundK Ellipsen und Kreise zeichnen

Das Programm EundK zeichnet Ellipsen oder Kreise auf der Basis der allgemeinen Kegelschnitt-Gleichung: $\frac{(X-X_m)^2}{a^2} + \frac{(Y-Y_m)^2}{b^2} = 1$

X_m und Y_m stellen den Mittelpunkt des Kegelschnitts dar, a und b seine Halbachsen. Das Programm nimmt die Werte für X_m , Y_m , a und b vom Stack, setzt diese in die Kegelschnitt-Gleichung ein und stellt das Ergebnis mit PIXEL dar.

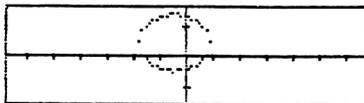
```

« CLLCD
"ENTER xn,ym,a,b" 2          Geben Sie Xm, Ym, a, b nacheinander in den
DISP HALT CLLCD             Stack, weiter mit CONT
" 2 *H / W" 2                Meldung
DISP 3 WAIT → c d a         3 Sekunden Programmpause, Werte in c,d,a,b
b
  « CLLCD 2 *H 2 *W
DRAX '(X-c)^2/a^2+(Y       Achsenkreuz zeichnen
-d)^2/b^2=1' 'Y'
ISOL 'Y' STO -1 1           Y isolieren
  FOR n n 's1' STO
c a - c a +
  FOR m m 'X'
STO X Y EVAL R→C
PIXEL .3                     Graphen zeichnen
  STEP 2
  STEP c d a b               Eingangswerte in den Stack
DGTIZ ( PPAR X Y s1        interaktiver Modus, Variablen löschen
) PURGE
  »
  »

```

Dieses Programm setzt voraus, daß vor Programmbeginn keine geänderten Abbildungsparameter in PPAR gespeichert sind.

Beispiel: $X_m = -1, Y_m = 1, a = 3, b = 2$



Sind die Werte für a und b gleich, würde ein Kreis gezeichnet werden.

3.1.5. AFvF Allgemeine Form von Funktionen

Funktionen der Form $ax^2+by^2+cx+dy+f$ werden berechnet, ihre wichtigsten Zeichenparameter angezeigt und graphisch dargestellt.

Sollte die Funktion im Reellen nicht definiert sein, wird ihr Realteil gezeichnet. Bei der Funktion \sqrt{x} wird auch der unter der x-Achse liegende Ast gezeichnet, obwohl der Graph einer negativen Wurzel normalerweise nicht berechnet wird. Dies geschieht durch "Spiegelung" der definierten Funktion $+\sqrt{x}$.

Das Programm AFvF besteht im Wesentlichen aus verschiedenen Abfrageschleifen, welche die Funktion $ax^2+by^2+cx+dy+f$ auf ihren Inhalt hin überprüfen und dann die entsprechenden Gleichungen zur Berechnung dieses Terms anwenden.

Im Hauptprogramm können je nach Bedarf sieben Unterprogramme aufgerufen werden:

1. zeige
.....

gibt ein Akustiksignal von 150 Hz und 0.01 s Dauer aus und unterbricht die Programmausführung für 3 s.

Wollen Sie dieses Programm zur Anwendung bringen, löschen Sie bitte Flag 51 zum Aufrufen des Akustiksignals im HP 28.

2. topx
.....

berechnet den Scheitel einer Funktion $by^2+cx+dy+f$.

3. topy
.....

berechnet den Scheitel der Funktion $ax^2+cx+dy+f$.

4. consty

.....

berechnet und speichert die Gleichung einer für y konstanten Funktion.

5. constx

.....

berechnet und speichert die Gleichung einer für x konstanten Funktion.

6. creq

.....

bildet eine Gleichung aus dem negativen und positiven Realteil einer nicht in \mathbb{R} definierten Funktion.

7. gpar

.....

ermittelt die Grundparameter zum Zeichnen eines Kegelschnittes und erstellt daraus die allgemeine Kegelschnittgleichung. Die Grundparameter werden dann im Display angezeigt.

Berechnet werden Mittelpunkt

Halbachsen

Asymptoten

des Kegelschnitts. Mit der Anzeige wird ein Akustiksignal ausgegeben.

Nach dem Anzeigen einer Meldung wird das Programm unterbrochen (der Indikator "o" erscheint in der linken oberen Ecke der Anzeige) und kann mit **|** |CONT| fortgesetzt werden. Anschließend wird der Graph gezeichnet.

Sollte der Graph nicht in der Anzeige erscheinen, ändern Sie den Abbildungsmaßstab und zeichnen den Graphen mit DRAW erneut. Dies ist auch beim HP 28 C nötig, da sich der Rechner nach dem Ablauf des Programms nicht mehr im interaktiven Modus befindet. Mit DRAW wird dieser Modus aktiviert. Dazu wird die zur Zeichnung benötigte Funktion von AFvF automatisch in EQ gespeichert und kann auch für andere Berechnungen weiterverwendet werden.

Das Programm AFvF verlangt die Eingabe der Koeffizienten in Form einer Liste {a,b,c,d,f}. Nicht existierende Parameter müssen in Form einer 0 in die Liste aufgenommen werden.

Beispiel: Die Funktion $x^2 - y^2 - 1$ soll eingegeben werden.

| { | 1,-1,0,0,-1 |ENTER|

Sollte ein Fehler im Programmablauf auftreten, lesen Sie Kapitel 4.

Listing AFvF (Hauptprogramm)

```

.....
« 51 CF CLLCD 2 FIX           Akustiksignal aktivieren, 2 Kommastellen
"Funktionen zeichnen "      Meldung Zeile 1
1 DISP
"→ a*X^2+b*Y^2+c*X+d*Y+f    Meldung Zeile 2
"
2 DISP
"Enter ( a,b,c,d,f )"      Meldung Zeile 4
4 DISP HALT LIST+          Programmunterbrechung, Liste auflösen
DROP f STO d STO c        Anzahl der Objekte in Liste löschen und Ob-
STO b STO a STO           jekte in f,d,c,b,a speichern
CLLCD 2 *N 2 *H           Meldung
" 2 * H , 2 * W"
2 DISP zeige CLLCD        Unterprogramm 1
  IF a 0 ≠
  THEN
    IF b 0 ≠
    THEN gpar HALT        Für Funktion  $ax^2+by^2+cx+dy+f$ 
                           Unterprogramm 7
    ELSE
      IF d 0 ==
      THEN                 Für Funktion  $ax^2+cx+f$ 
    "Gleichung enth. kein Y" Meldung
  1 DISP
    IF a 0 == c
    0 == OR
    THEN
      " x = constant" 3
    DISP
    "X-Y-Achse vertauscht" Meldung
  2 DISP constx           Unterprogramm 5
    ELSE
      END
      ELSE 'a*X^2+c*
X+d*Y+f' EVAL 'Y'
ISOL creaq STEQ          Unterprogramm 6
"Scheitel= " topy       Unterprogramm 3
+STR + 3 DISP           Meldung
  END zeige HALT
  END ( GG HRA HAB
SX sy ASy MPx MPy e1
e2 ) PURGE

```

Fortsetzung Listing AFvF
.....

ELSE	
IF c 0 ==	Für Funktion by^2+dy+f
THEN	
"Gleichung enth. kein X"	Meldung
1 DISP	
IF b 0 == d 0	Für Funktion by^2+f oder $dy+f$
== OR	
THEN	
" y = constant " 3	Meldung
DISP consty	Unterprogramm 4
ELSE	
END zeige HALT	
ELSE	
IF b 0 ≠	Für Funktion $by^2+cx+dy+f$
THEN	
"Scheitel =" topx	Unterprogramm 2
+STR + 3 DISP	Meldung
IF d 0 ≠	Für Funktion $by^2+cx+dy+f$
THEN	
"x - y Achse vertauscht"	
1 DISP 'b*Y^2+c*X+d#	
Y+' EVAL 'X' ISOL	
'Y' INDEP creq STEQ	Unterprogramm 6
ELSE 'b*Y^2+	Für Funktion by^2+cx+f
c*X+' EVAL 'Y' ISOL	
creq STEQ	Unterprogramm 6
END HALT	
ELSE '-(c*X+f)	Für Funktion $cx+dy+f$
/d' EVAL creq STEQ	Unterprogramm 6
END	
END	
END CLLCD DRAW	Ende aller Schleifen, Funktion zeichnen
DGTIZ { a b c d f	Interaktiver Modus (nur HP 28 S)
ASY Y ≤1 PPAR }	Löschen aller Variablen
PURGE 51 SF	Löschen Akustiksignal
»	

Listing zeige (Unterprogramm 1)

```
.....
< 150 .01 BEEP 3
WAIT
>
```

Listing topx (Unterprogramm 2)

```
.....
< 'RND(-d/(2*b))'
EVAL 'RND(-((f-d^2/(
4*b))/c))' EVAL R+C
>
```

Listing consty (Unterprogr. 4)

```
.....
< " y= " 'b*Y^2+d*
Y+f' EVAL 'Y' ISOL
DUP 1 's1' STO EVAL
SWAP -1 's1' STO
EVAL
IF f 0 >
THEN IM SWAP IM
DUP 'j' * →STR
ELSE SWAP DUP →STR
END 3 ROLLD = STEQ
+ 4 DISP
>
```

Listing creq (Unterprogramm 6)

```
.....
< DUP 1 's1' STO
EVAL SWAP -1 's1'
STO EVAL = RE
>
```

Listing topy (Unterprogramm 3)

```
.....
< 'RND(-c/(2*a))'
EVAL 'RND(-((f-c^2/(
4*a))/d))' EVAL R+C
>
```

Listing constx (Unterprogr. 5)

```
.....
< " x= " 'a*X^2+c*
X+f' EVAL 'X' ISOL
DUP 1 's1' STO EVAL
SWAP -1 's1' STO
EVAL
IF f 0 >
THEN IM SWAP IM
DUP 'j' * →STR
ELSE SWAP DUP →STR
END 3 ROLLD = STEQ
+ 4 DISP
>
```

Listing gpar (Unterprogramm 7)

```

.....
< "Mittelpkt= " RND
(-c/(2*ABS(a)))'
EVAL DUP 'mpx' STO '
RND(-d/(2*ABS(b)))'
EVAL DUP 'mpy' STO
R+C →STR + 1 DISP '
mpx^2+mpy^2-f' EVAL
'e1' STO
IF e1 0 ==
THEN 1 'HAa' STO 1
'HAb' STO 'a*X' EVAL
DUP -1 * = STEQ
ELSE
"H-Achsen (f" 'e1/
ABS(a*b)' EVAL 'e2'
STO b ABS e2 * RND
DUP ABS DUP ↓ 'HAa'
STO SWAP a * SIGN
'sx' STO →STR + ",√"
+ a ABS e2 * RND DUP
ABS DUP ↓ 'HAb' STO
SWAP b * SIGN 'sy'
STO →STR + ")" + 2
DISP
IF sx sy ==
THEN

```

Fortsetzung gpar

```

.....
IF sx -1 ==
THEN 1 1 'sx'
STO 'sy' STO
ELSE
END
ELSE
END "Asymt.= "
IF HAa 0 *
THEN 'ABS(HAb/
HAa)' EVAL DUP 'ASY'
STO RND 'GG' STO 'y=
GG*x' EVAL →STR + 3
DISP "Asymt.= " ASY
-1 * RND 'GG' STO 'y
=GG*x' EVAL →STR + 4
DISP
ELSE DROP
'Asymt.= y - Achse"
4 DISP
END 'sx*SQ((X-
mpx)/HAa)+sy*SQ((Y-
mpy)/HAb)=1' EVAL
'Y' ISOL creq STEQ
END zeige
>

```

Die hier gelisteten Unterprogramme lassen sich auch einzeln aufrufen und getrennt vom Programm AFvF benutzen. Vor allem für Benutzer des HP 28 C, und wenn Sie das doch relativ lange Programm AFvF aus Gründen von zu knapp werdendem Speicherplatz nicht eingeben wollen oder können, mag das interessant sein.

Sie brauchen nur die entsprechenden Koeffizienten in den verwendeten Variablen speichern (mit z.B. 'a' |STO| 'c' |STO| usw.) und das Programm starten. Die ermittelten Werte werden dann angezeigt und die Gleichung des Kegelschnitts in EQ gespeichert. Diese kann mit DRAW gezeichnet werden.

Die einzelnen Unterprogramme benötigen aus der Funktion $ax^2+by^2+cx+dy+f$ folgende Koeffizienten:

topx b,c,d,f diese werden mit ' ' |STO| gespeichert

topy a,c,d,f mit ' ' |STO| speichern

consty b,d,f mit ' ' |STO| speichern

constx a,c,f mit ' ' |STO| speichern

creq verlangt eine Funktion, welche z.B. im Negativen nicht berechnet wird. Ein Beispiel ist die Funktion $f(x) = \sqrt{x}$.
Geben Sie die Funktion mit ' ' |ENTER| in den Stack.

Beispiel für AFvF:

Zeichnen Sie die Funktion $x^2+y^2-2=0$.

|USER| AFvF → Anzeige:

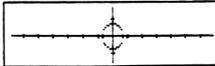
Funktionen zeichnen + a*X^2+b*Y^2+c*X+d*Y+f Enter (a,b,c,d,f)

{| 1,1,0,0,-2 → Anzeige:

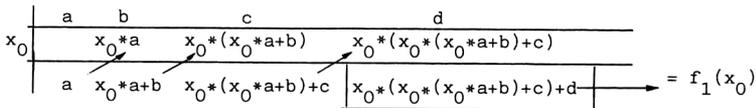
Mittelpkt= (0.00,0.00) H-Achsen (√2.00,√2.00) Asymt.= 'y=x' Asymt.= 'y=-x'

2 * H , 2 * W

█ |CONT| → Anzeige:



Haben Sie eine Funktion $y = \sqrt{x^2+x-2}$ vorliegen, können Sie nach einem Quadrieren des Ausdrucks und nachfolgendem Umstellen der Gleichung in die Form $x^2-y^2+x-2=0$, diese für das Programm AFvF benutzen.



Der letzte Ausdruck stellt den Funktionswert an der Stelle x_0 dar. Ist dieser Ausdruck Null, handelt es sich bei x_0 um eine richtige Nullstelle. Ist x_0 eine Nullstelle so gilt:

$$ax^3+bx^2+cx+d:(x-x_0) = ax^2+(x_0*a+b)x+(x_0*(x_0*a+b)+c)+ \frac{x_0*(x_0*(x_0*a+b)+c)+d}{(x-x_0)}$$

Entwickelt man das Schema weiter, indem man den x_0 -Wert und den a-Wert wieder eine Zeile tiefer schreibt, kann man mit x_0 und a genau so verfahren wie oben. Das Ergebnis von x_0*a in der zweiten Rechnung wird dann zum Term x_0*a+b addiert. Für die letzte Stelle, d.h. für den Funktionswert der ersten Berechnung, findet keine Auswertung mehr statt. Die letzte Stelle der zweiten Berechnung steht also unter dem oberen Koeffizienten c. So erhalten wir ein Dreieck, dessen untere Ecke beim a endet. Die so ermittelten Funktionswerte $f_1(x_0)$ bis $f_4(x_0)$ können wir in eine neue Gleichung schreiben: Aus ax^3+bx^2+cx+d wird:

$a*(x-x_0)^3+f_3*(x-x_0)^2+f_2*(x-x_0)+f_1$. Dieser Term ist oft einfacher, da einige $f_n(x_0)$ -Terme 0 sind und wegfallen.

Das Programm HORS1 simuliert nun jeweils eine solche Prozedur, wobei die Koeffizienten der Ausgangsgleichung in Ebene 1: und das Ergebnis in Ebene 2: dargestellt wird. Die nächsten Ergebniszeilen stehen darüber.

Mit DROP oder einer entsprechenden Funktion können Sie dann Zeile für Zeile abrufen.

Eingabe: {Koeffizientenliste}, x_0 -Wert (Entwicklungsstelle)

Ausgabe: vollständige Entwicklung nach Horner

Beispiel: $z^4-8z^3+22z^2-24z+12$ soll an der Stelle $x_0=2$ entwickelt werden

Eingabe: HORS1 {1,-8,22,-24,12 |ENTER| 2 |ENTER| |||CONT|

```

Ausgabe: 6:{
          1.00
5:{      1.00 0.00
4:{      1.00 -2.00 -2.00
3:{      1.00 -4.00 2.00 0.00
2:{ 1.00 -6.00 10.00 -4.00 4.00
1:{ 1.00 -8.00 22.00 -24.00 12.00
    
```

Nimmt man jetzt die jeweils letzten Werte einer Liste, so kann man die Funktion neu schreiben:

$$(z-2)^4+0*(z-2)^3-2*(z-2)^2+0*(z-2)+4 \longrightarrow (z-2)^4-2*(z-2)^2+4$$

Listing:

```

« CLLCD
"ENTER ( ),NS" 2           Meldung
DISP HALT → a b
  « 2 'c' STO a b a
SIZE 1 SWAP a LIST→       Größe der Koeffizientenliste
Ø SWAP 1 + →LIST 'a'
STO
  START a 1 GET           Schleife für Anzahl der Durchläufe
  DUP 2 a SIZE
  FOR i b * a i          Schleife für Berechnung der einzelnen Terme
  GET +
    « EXPAN              Zusammenfassen des Ergebnisses
    » Repeat
COLCT DUP
  NEXT DROP DROP         nächster Term
a SIZE 1 - →LIST DUP     Ergebnis in Liste
'a' STO c 1 + DUP
'c' STO ROLLD           Liste nach oben rollen
  NEXT DRUP 'c'         nächster Durchlauf
PURGE
  »
»

```

3.2.5. GAUS1 Gaußscher Algorithmus

.....

Der Gaußsche Algorithmus ist eine effektive Methode Gleichungssysteme aufzulösen.

Dazu werden die Koeffizienten der einzelnen Gleichungen untereinander in ein Schema geschrieben (siehe Beispiel). Dann wird die erste Zeile noch einmal darunter gesetzt und versucht, einzelne Koeffizienten zu eliminieren. Das geschieht, indem man eine beliebige Zeile so umformt, daß ihr erster Koeffizient dem des darüber liegenden entspricht. Nur das Vorzeichen muß verschieden sein.

Anschließend werden die Zeilen 1 und n addiert und neu darunter geschrieben. Das wird so oft wiederholt, bis sich eine dreieckige Form der restlichen Koeffizienten einstellt. Diese Form nennt man Gauß-Jordan Form.

Beispiel:

$$\begin{array}{l}
 x_1 - x_2 - x_3 = -2 \\
 2x_1 + ux_3 = -4 \\
 -x_1 + 2x_2 + x_3 = v
 \end{array}
 \longrightarrow
 \begin{array}{ccc|c}
 1 & -1 & -1 & -2 \\
 2 & 0 & u & -4 \\
 -1 & 2 & 1 & v
 \end{array}$$

$$\begin{array}{l}
 1. \text{Zeile} \cdot (-2) + 2. \text{Zeile} \\
 1. \text{Zeile} - 3. \text{Zeile}
 \end{array}
 \longrightarrow
 \begin{array}{ccc|c}
 1 & -1 & -1 & -2 \\
 0 & 2 & u+2 & 0 \\
 0 & 1 & 0 & v-2
 \end{array}$$

$$\begin{array}{l}
 \text{usw.} \\
 \swarrow
 \end{array}
 \begin{array}{ccc|c}
 1 & -1 & -1 & -2 \\
 & 2 & 2+u & 0 \\
 & & -1-u & -2+v
 \end{array}$$

Das Programm GS.1 nimmt die Koeffizienten in Listenform vom Stack und simuliert das Gauß Schema schrittweise. Sie können sich also jedes Teilergebnis in Ruhe ansehen. Für den nächsten Rechenschritt setzen Sie das Programm mit █|CONT| fort. Das Programm wird automatisch abgebrochen, wenn die letzte Berechnung abgeschlossen ist.

Das Programm GS.1 ruft das Unterprogramme ug und rolle auf. ug formt eine Entwicklungszeile um und legt das Ergebnis als neue Zeile in den Stack. rolle dreht die Objektreihenfolge im Stack um.

Eingabe für oberes Beispiel:

```

GS.1 {1,-1,-1,-2 |ENTER| {2,0,'u',-4 |ENTER| {-1,2,1,'v' |ENTER|
...
█|CONT|
    
```

Ausgabe: 1. Ergebnismatrix, wobei jede Zeile als Liste geschrieben wird. Weiter mit █|CONT| bis das Programm endet (der Programmstopp-Indikator "o" erlischt). Ergebnis siehe oben.

Deutung des Ergebnisses:

Eine Deutung des Ergebnisses findet immer in Bezug auf die letzte Zeile des Ergebnisses statt.

Existiert der Term auf der linken Seite der Gleichung, d.h. $u+2 \neq 0 \rightarrow u \neq -2$, dann gibt es eine eindeutige Lösung. Der Lösungsvektor kann bestimmt werden.

Sind beide Terme 0, d.h. $u=-2$ und $v=2$, dann existieren unendlich viele Lösungen.

Ist der Term auf der linken Seite 0 und der Term auf der rechten Seite ungleich 0, dann existiert keine Lösung.

Listing GS.1 (Hauptprogramm)
.....

```

« CLLCD
"Homogene Form AX=B "
1 DISP
"Enter (h,i,...)" 2
DISP
"      (l,m,...)" 3
DISP
"      (v,w,...)" 4
DISP HALT DEG 0 FIX
'w' PURGE 'v' PURGE
1 5
  START DUP SIZE 1 -
  'w' STO 0 'q' STO
  DUP DUP SIZE 'z1'
  STO LIST→ 'z' STO
rolle 1 z
  START
  IFERR DUP
  IF 0 ==
  THEN DROP 1
  'q' STO+
  ELSE
  END

```

Fortsetzung GS.1
.....

```

  THEN DROP
  END
  NEXT z1 q -
+LIST SIZE
  IF 2 >
  THEN w ROLL w 1
- 1 SWAP
  START w ROLL
u9
  NEXT w ROLLD
CLLCD 250 .1 BEEP
"Weiter? = CONT" 2
DISP 3 WAIT CLMF
HALT
  ELSE
  END { w q z1 z }
PURGE
  NEXT
»

```

Listing rolle (Unterprogramm)
.....

```

« z1 2
  FOR i i ROLLD -1
  STEP
»

```

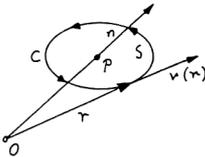
Listing ug (Unterprogramm)

```

« → a b
  « b 1 GET a 1 GET
  / NEG 'v' STO a SIZE
  2 SWAP
    FOR i a i GET v
  * b i GET + EXPAN
  COLCT
    NEXT a SIZE 1 -
  +LIST a 'v' PURGE
  »
»
    
```

3.2.6. ROT Rotation eines Vektorfeldes

Die Rotation $\text{rot } u$ eines Vektorfeldes $a(r)$ gibt an, ob in einem Punkt p lokal betrachtet eine Rotationsbewegung stattfindet.



Dabei liegt p in einem kleinen Flächenstück mit dem Flächeninhalt S und der Randkurve C , die den Einheitsvektor n in p entgegen dem Uhrzeigersinn umlaufe. n ist der Normalenvektor in p . S lasse man gegen 0 streben.

Es ergibt sich ein Ausdruck $n \text{ rot } u$ als Projektion von $\text{rot } u(p)$ auf die Richtung von n .

Ist nun z.B. v ein Geschwindigkeitsfeld der Form $v(r) = \omega(n \times r)$, dann gilt $\text{rot } v(r) = 2\omega n$.

Ist der Wert von $\text{rot } u$ gleich 0 , so ist das Vektorfeld u wirbelfrei.

Die Rotation u in cartesianischen Koordinaten wird als Determinante bzw. Matrix dargestellt:

$$\text{rot } u = \begin{vmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ v_x & v_y & v_z \end{vmatrix}$$

Die Berechnung erfolgt nach der Regel von Sarrus.

Das Programm ROT berechnet diese Determinante für ein Vektorfeld $a(r)$ und einen Punkt p , der durch den Vektor \vec{r} dargestellt wird.

Dafür gilt:

$$a(\vec{r}) = \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix} = \begin{pmatrix} B \\ C \\ D \end{pmatrix} \quad \vec{r} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Listing ROT
.....

« CLLCD	
"ENTER {a(r)},{r}" 2	Meldung für Eingabe
DISP HALT → a r	Programmstop, a(r) und r in a und r
« a LIST→ DROP 'D'	v_x, v_y, v_z in D,C,B
STO 'C' STO 'B' STO	
'ΔY(D)'	$\frac{dD}{dy}$ berechnen
« EVAL	
» Repeat 'ΔZ(C)'	$\frac{dC}{dz}$ berechnen
« EVAL	
» Repeat - COLCT	subtrahieren und zusammenfassen
'ΔZ(B)'	$\frac{dB}{dz}$ berechnen
« EVAL	
» Repeat 'ΔX(D)'	$\frac{dD}{dx}$ berechnen
« EVAL	
» Repeat - COLCT	subtrahieren und zusammenfassen
'ΔX(C)'	$\frac{dC}{dx}$ berechnen
« EVAL	
» Repeat 'ΔY(B)'	$\frac{dB}{dy}$ berechnen
« EVAL	
» Repeat - COLCT	subtrahieren und zusammenfassen
3 DUPN 3 →LIST 4	Ergebnisse duplizieren und in Liste, Liste
ROLLD 'W' STO 'V'	nach oben rollen Ergebnisse in w,v,u spei-
STO 'U' STO r LIST→	chern, x,y,z aus Liste r holen
DROP 'Z' STO 'Y' STO	Anzahl löschen und x,y,z in X,Y,Z
'X' STO U EVAL COLCT	u auswerten und zusammenfassen
V EVAL COLCT W EVAL	v und w auswerten und zusammenfassen
COLCT 3 →LIST	Ergebnis in Liste
» { a r B C D U V	Variablen löschen
W X Y Z } PURGE	
»	

Das Ergebnis wird in Listenform ausgegeben.

Beispiel:

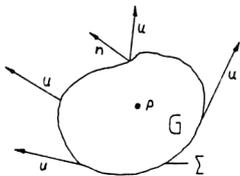
$$a(r) = \begin{pmatrix} y^2+z^2 \\ 0 \\ 0 \end{pmatrix} \quad r = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

Eing: ROT {'y^2+z^2',0,0 |ENTER| {1,2,2, |ENTER| █ |CONT|

Ausg: {0,2*z,-2*y}(symbolisches Ergebnis von ROT)
 {0,4,-4}(Lösungsvektor)

3.2.7. DIV Divergenz eines Vektorfeldes

Die Divergenz $\text{div } u$ ist ein Maß für die Quellen eines Feldes $u(p)$. Ist



$\text{div } u$ für ein Gebiet G gleich 0, dann ist das Vektorfeld $u(p)$ in diesem Gebiet "quellenfrei".

Befinden sich elektrische Ladungen mit der Ladungsdichte ρ im Vakuum, dann läßt sich die elektrische Feldstärke E in folgender

Gleichung ausdrücken: $\epsilon_0 \text{div } E(p) = \rho(p)$. Ist nun $\text{div } E$ gleich 0, so befinden sich in G keine Ladungen. Das Feld ist "quellenfrei". Das bedeutet der Gesamtvektorenfluß von G ist 0.

Die dabei zur Anwendung kommende Formel lautet:

$$\text{div } v = \frac{dv_x}{dx} + \frac{dv_y}{dy} + \frac{dv_z}{dz}$$

Das Programm verwendet diese Formel zur Berechnung des Vektorfeldes.

Eingabe: $a(r)$ in Liste, r in Liste

Ausgabe: symbolisches und numerisches Ergebnis

Listing:

Listing Fortsetzung


```

< CLLCD
"ENTER (a(r)),(r)" 2
DISP HALT + a r
  < a LIST+ DROP '0'
STO 'C' STO 'B' STO
'X(B)'
  < EVAL
  > Repeat 'X(C)'
  < EVAL
  > Repeat + 'X(D)
    
```

```

< EVAL
  > Repeat + DUP
'di' STO r LIST+
DROP 'Z' STO 'Y' STO
'X' STO di EVAL
COLCT ( X Y Z B C D
di ) PURGE
  >
  >
    
```

Beispiel:

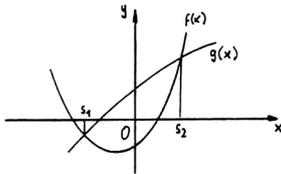
$$a(r) = \begin{pmatrix} 2xy^2 \\ 2x^2y \\ 4z^3 \end{pmatrix} \quad r = \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$$

Eingabe: DIV { '2*x*y^2', '2*x^2*y', '4*z^3' |ENTER| { 1,2,2 |ENTER| █ |CONT|

Ausgabe: 2*y^2+2*x^2+4*(3*z^2) , 58

3.3. Kurvendiskussion

3.3.1. FZ2K Fläche zwischen zwei Kurven



Dieses Programm berechnet den Schnittpunkt zweier Kurven $f(x)$ und $g(x)$ nach der Vorgabe einer Näherung und bildet dann das Integral über die von $f(x)$ und $g(x)$ eingeschlossene Fläche A .

Als Ergebnis erhalten Sie den Flächeninhalt und den absoluten Fehler dieses Wertes. Grundlage der Berechnung ist das Integral:

$$A = \int_{s_1}^{s_2} |f(x) - g(x)| dx$$

Listing:
.....

< CLLCD

"ENTER F(X),G(X),Genauigkeit"

Eingabe der Funktionen $f(x)$ und $g(x)$ und der Genauigkeit der Berechnung

2 DISP HALT → fx gx

Programmstop, Werte holen

9

< 2 FIX CLLCD

"Graph wird gezeichnet"

Meldung

1 DISP

"Bitte Schnittpunkte "

3 DISP

"digitalisieren → CONT"

4 DISP 3 WAIT CLLCD

3 Sekunden Programmunterbrechung

'fx=gx' EVAL STEQ

$f(x)=g(x)$ in EQ

'PPAR' PURGE 5 #H

PPAR löschen

DRAW DGTIZ HALT 'a'

Graph zeichnen mit 5-facher Höhe

STO 'b' STO 'X' RCEQ

digitalisierte Schnittpunkte eingeben

Listing Fortsetzung

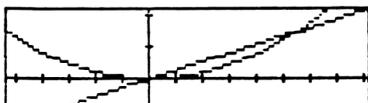
```
'X' a ROOT RCEQ 'X'           Lösungsroutine für s2 und s1
b ROOT 3 →LIST 'ABS(        X,s2 und s1 in Liste
fx-gx)' SWAP g f           Integranden bilden und Integral berechnen
  > ( a b X EQ PPAR        Variable löschen
) PURGE
>
```

Beispiel: f(x)= x² g(x)= x Genauigkeit= 0.005

Eingabe: FZ2K 'X^2','X ',0.005 |ENTER| █ |CONT|

Ausgabe: Graph

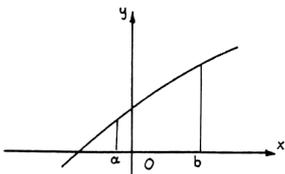
Dieser kann durch Ändern des Abbildungsmaßstabes und neues Zeichnen mit DRAW verschoben werden.



Eingabe: Digitalisieren Sie die Schnittpunkte und setzen Sie das Programm mit █ |CONT| fort.

Ausgabe: A= 7.81 Fehler= 0.04

3.3.2. LÄEKGU Länge einer Kurve im Intervall a bis b



Das Programm berechnet die Länge einer Kurve im Intervall von a bis b. Grundlage der Berechnung ist das Linienintegral:

$$L = \int_b^a \sqrt{1 + \left(\frac{df}{dx}\right)^2}$$

Listing:

```
< CLLCD
"ENTER F (X),MIN,MAX"
2 DISP HALT → x a b
< 2 FIX '(1+8X(x)
```

Listing Fortsetzung

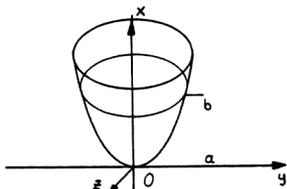
```
.....
^2)' 'X' a b 3 →LIST
.005 f
>
>
```

Beispiel: $f(x) = x^3 - x^2 + 4x + 2$ $a=0$ $b=2$

Eingabe: LÄEKO 'X^3-X^2+4*X+2',0,2 |ENTER| █ |CONT|

Ausgabe: Länge: 12.19 Fehler: 0.06

3.3.3. OFRKÖ Oberfläche eines Rotationskörpers



Die Oberfläche eines rotationssymmetrischen Körpers über einer Achse im Intervall seiner Höhe, wird mit Hilfe des Integrals:

$$O = \int_a^b 2\pi f(x) \sqrt{1 + \left(\frac{d}{dx}f(x)\right)^2} \text{ berechnet.}$$

Listing:
.....

```

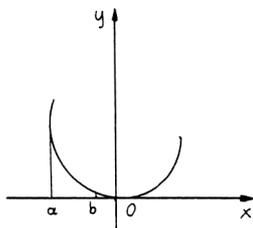
< CLLCD
"ENTER F(X),(VAR,a,b)"
2 DISP HALT → x 1
  < 2 FIX 'f(1+dx(x)
^2)' 'ffx' STO '2*pi*
x*ffx' 1 .005 ↓
  > 'ffx' PURGE
>
    
```

Beispiel: $f(x) = x^2$ im Intervall $[0, 1]$

Eingabe: OFRKÖ 'X^2', { X,0,1 |ENTER| █ |CONT|

Ausgabe: Oberfläche: 3.81 Fehler: 0.02

3.3.4. LKPAR Länge einer Kurve in Parameterdarstellung



Das Integral
$$L = \int_a^b \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}$$

stellt einen Ausdruck für die Länge einer Kurve in Parameterdarstellung dar. Das Programm nimmt die Funktionen $x(t)$ und $y(t)$ und einen Anfangs- und Endwert für die Berechnung des Integrals vom Stack. Mit diesen Werten wird das Integral L berechnet.

Listing:
.....

```

< CLLCD
"ENTER X(T),Y(T),a,b"
2 DISP HALT → x y a
b
  < 2 FIX '∫(SQ(∂T(x
)))+SQ(∂T(y)))' 'T' a
b 3 →LIST .005 ∫
  >
>
    
```

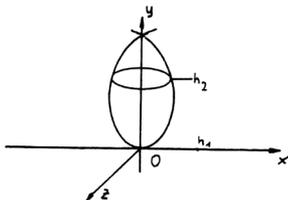
Beispiel: $x(t) = t - t^3$ $y(t) = t^2$ im Intervall $[0, 3]$

Eingabe: LKPAR 'T-T^3','T^2',0,3 |ENTER| █ |CONT|

Ausgabe: 26.79 Fehler: 0.13

3.3.5. ORPAR Oberfläche eines Rotationskörpers in Parameterdarstellung

lung
.....



Das Programm ORPAR berechnet die Oberfläche eines rotationssymmetrischen Körpers über einer Achse im Intervall seiner Höhe.

Dabei findet das Integral:

$$O = \int_a^b 2\pi y(t) \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \quad \text{Verwendung.}$$

Eingegeben werden die Funktionen $x(t)$, $y(t)$ und die Werte a und b für das Intervall $[h_1, h_2]$.

Listing:
.....

```

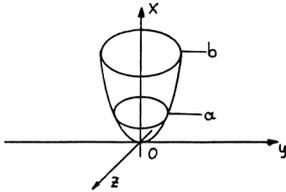
< CLLCD
"ENTER X(T),Y(T),a,b"
2 DISP HALT → x y a
b
  < 2 FIX '∫(SQ(∂T(x
)))+SQ(∂T(y)))' 'FFX'
STD '2*π*FFX*y' EVAL
'T' a b 3 →LIST .005 ∫
  > 'FFX' PURGE
>
    
```

Beispiel: $x(t) = t - t^3$ $y(t) = t^2$ $a=0$ $b=0.8$

Eingabe: ORPAR 'T-T^3','T^2',0,0.8 |ENTER| █ |CONT|

Ausgabe: 1.41 Fehler: 0.01

3.3.6. VROT Volumen eines Rotationskörpers



Das Volumen eines Rotationskörpers über einer Achse im Bereich einer bestimmten Höhe soll berechnet werden. Dazu können wir das folgende Integral verwenden:

$$V = \int_a^b 2\pi x f(x) dx$$

Für die Berechnung ist die Eingabe der Funktion $f(x)$ und die Angabe des Intervalls von h_1 bis h_2 notwendig.

Listing:

.....

```

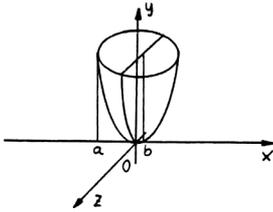
« CLLCD
"Vol.eines Rot.Koerpers"
1 DISP
"ueber d. x-Achse" 2
DISP
"F(X),x1(h1),x2(h2) ?"
4 DISP HALT → y a b
  « 3 FIX 'π*y^2'
'X' a b 3 →LIST .005
↓
  »
»
    
```

Beispiel: $f(x) = x^2$ $h_1 = 0$ $h_2 = 2$

Eingabe: VROT 'X^2',0,2 |ENTER| █ |CONT|

Ausgabe: 20.105 Fehler: 0.1

3.3.7. VROT2 Volumen eines Rotationskörpers 2



Das Programm berechnet das Volumen eines Rotationskörpers über einer Achse, mit den Grenzen auf der anderen Achse.

Das Volumen wird mit diesem Integral berechnet:

$$V = \int_a^b \pi f(x)^2 dx$$

Wenn sich der Körper um die y-Achse dreht und die Grenzen auf der x-Achse liegen, geben Sie bitte folgende Parameter ein: $f(x), x_1, x_2$.

Das Programm berechnet dann das Integral mit einer Genauigkeit von 0.005.

Listing:
.....

```

« CLLCD
"Vol ueber d. y-Achse "
1 DISP
"Grenzen auf x-Achse"
2 DISP "F(X),a,b ?"
4 DISP HALT + y a b
« 3 FIX '2*π*X*y'
'X' a b 3 →LIST .005
↓
»
»

```

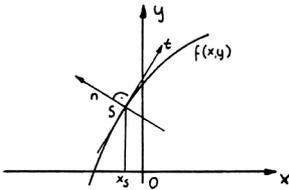
Beispiel: $f(x) = x^2$ $x_1 = 0$ $x_2 = 2$

Eingabe: VROT2 'X^2',0,2 |ENTER| █ |CONT|

Ausgabe: 25.132 bei einem Fehler von 0.125

Die Programme auf den letzten Seiten, welche Längen, Flächen oder Volumen berechnen, lassen sich auch auf andere Koordinatenachsen beziehen. Die entsprechenden Berechnungsparameter sind abzuändern.

3.3.8. NOR1 Normale an Kurve f(x,y)



NOR1 berechnet die Gleichung einer Normalen n an eine Kurve f(x,y) im gegebenen Schnittpunkt s.

Dazu bestimmt das Programm die Steigung der Normalen mit der Formel:

$$m = -\frac{\frac{df}{dy}}{-\frac{df}{dx}}$$

Dann wird der Wert der Steigung in die allgemeine Geradengleichung eingesetzt. Der x-Wert und der y-Wert des Schnittpunktes werden ebenfalls in die Gleichung geschrieben und diese dann ausgewertet. Die vollständige Normalengleichung wird anschließend angezeigt.

Listing:
.....

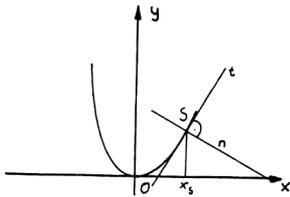
<pre> < 3 FIX CLLCD " F(X,Y),xs" 2 DISP HALT → f p < 'INV(-∂X(f)/∂Y (f))' < EVAL > Repeat 'm' STO P 'X' STO f 'Y' p ROOT 'Y' STO 'y=m*(x -X)+Y' < EVAL > Repeat ZFRSS > (X Y m EQ) PURGE > </pre>	<p>Eingabemeldung Programmstop, f(x,y) und x_s in f und p</p> <p>Steigung berechnen</p> <p>Auswerten</p> <p>in m speichern</p> <p>x_s in X speichern</p> <p>Lösungsroutine für y-Wert, y-Wert in Y</p> <p>Geradengleichung auswerten und zusammenfassen</p> <p>Variablen löschen</p>
---	--

Beispiel: f(x,y)= x²-y x_s= 1

Eingabe: NOR1 'X²-Y',1 |ENTER| █ |CONT|

Ausgabe: 'y= 1.5+0.5*x'

3.3.9. NOR2 Normale an Kurve f(x)



NOR2 berechnet die Normale n an eine Kurve f(x) im Schnittpunkt s.

Die Formel für die Steigung einer Normalen an eine Kurve f(x) lautet:

$$m = - \frac{1}{\frac{df}{dx}}$$

Der Programmablauf ist vergleichbar mit dem von NOR1. Ausgegeben wird am Ende die vollständige Geradengleichung für n.

Listing:
.....

```
< CLLCD 2 FIX
"Y= ?,Xs" 2 DISP
HALT → f p
  < '-INV(∂X(f))'
  < EVAL
  > Repeat 'm' STO
p 'X' STO 'f=Y' EVAL
'Y' p ROOT 'Y' STO '
y=m*(x-X)+Y'
  < EVAL
  > Repeat ZFASS
  > { Y X m EQ }
PURGE
>
```

Beispiel: f(x)= x² x_s = 1

Eingabe: NOR2 'X^2',1 |ENTER| █ |CONT|

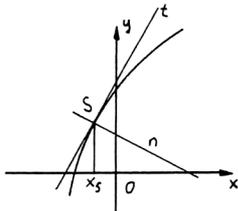
Ausgabe: 'y=1.5+0.5*x'

3.3.10. TAN1 Tangente an Kurve f(x,y)

Das Programm TAN1 berechnet die Gleichung einer Tangente t an eine Kurve f(x,y) im Schnittpunkt s.

Dabei wird die Formel: $m = \frac{df}{dx} / \frac{df}{dy}$ zur Berechnung der Tangenten-Steigung

angewendet. Der Programmablauf ist vergleichbar mit dem von NOR1 und NOR2.



Listing:
.....

```
TAN1
< HALT 3 FIX CLLCD
"F(X,Y),xs" 2 DISP
HALT → f p
  < '-∂X(f)'
  < EVAL
  > Repeat '∂Y(f)'
  < EVAL
  > Repeat / 'm'
STO p 'X' STO f 'Y'
p ROOT 'Y' STO 'y=m*'
(x-X)+Y'
  < EVAL
  > Repeat ZFRSS
  > ( X Y m EQ )
PURGE
>
```

Beispiel: $f(x,y) = x^2 + x*y - 3 \quad x = 2$

Eingabe: TAN1 'X^2+X*Y-3',2 |ENTER| █ |CONT|
.....

Ausgabe: 'y= 3-1.75*x'

3.3.11.TAN2 Tangente an Funktion f(x)
.....

Berechnet wird die Gleichung einer Tangente t an eine Funktion f(x) im Schnittpunkt s. Erklärung und Programmablauf siehe TAN1.

Angewendete Formel für m: $m = f'(x)$

Beispiel: $f(x) = \frac{x^2-3}{x} \quad x = 2$

Eingabe: TAN2 '-(X^2-3)/X',2 |ENTER| █ |CONT|
.....

Ausgabe: 'y= 3-1.75*x'

Listing:
.....

```

< 2 FIX CLLCD
"Y=?,xS" 2 DISP HALT
→ f p
  < 'ΔX(f)'
    < EVAL
      > Repeat 'm' STO
p 'X' STO f EVAL 'Y'
STO 'y=m*(x-X)+Y'
  < EVAL
    > Repeat ZFASS
  > { Y X m EQ }
PURGE
>
    
```

3.3.12. WIN1 Schnittwinkel zweier Geraden
.....



WIN1 berechnet den Schnittwinkel zweier Geraden anhand ihrer Steigungen mit der Funktion:

$$\varphi = \text{ATAN} \frac{m_2 - m_1}{1 + m_1 * m_2}$$

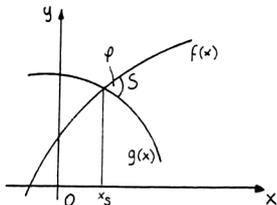
Listing:
.....

```

< DEG → a b 'ATAN((b
-a)/(1+a*b))'
>
    
```

Beispiel: $y = x$ $y = 0$
 Eingabe: 1,0 |ENTER| WIN1
 Ausgabe: 45°

3.3.13. WIN2 Schnittwinkel zweier Kurven
.....



WIN2 berechnet den Schnittwinkel zweier Kurven im Schnittpunkt s.
 Dazu nimmt das Programm zwei Funktionen \$f(x)\$ und \$g(x)\$ vom Stack, verbindet diese und zeichnet den Graphen. Mehrere aus dem Graphen ermittelte Anfangsnäherungen werden gespei-

chert und mit einer Lösungsroutine genau bestimmt. Anschließend werden die zugehörigen y -Werte ermittelt und die einzelnen Schnittwinkel berechnet.

Abschließend werden die Koordinaten der Schnittpunkte und die einzelnen Schnittwinkel angezeigt.

Die Schnittwinkel werden mit dem im Hauptprogramm aufgerufenen Programm WIN1 berechnet.

Außerdem wird im Hauptprogramm das Unterprogramm XSP, zum Ermitteln der x -Koordinaten der Schnittpunkte aufgerufen.

Listing WIN2 (Hauptprogramm)

.....

« CLLCD 'EQ' PURGE	
'PPAR' PURGE	PPAR löschen
"Winkel zw. 2 Kurven"	Meldung 1
2 DISP 2 WAIT CLLCD	2 Sekunden Programmpause
"Enter f(x),g(x)" 2	Meldung 2
DISP HALT → k1 k2	f(x) in k1 und g(x) in k2
« CLLCD	
"Graph wird gezeichnet"	Meldung 3
1 DISP	
"Schnittpkte digitalis."	
3 DISP	
"und Anzahl angeben"	
4 DISP 3 WAIT CLLCD	
'k1=k2' EVAL STER 5	f(x) und g(x) verbinden und in EQ speichern
*H DRAW DGTIZ HALT	Graph zeichnen, Programmpause
'N' STO N →LIST XSP	Anzahl Nullstellen in N, unterprogramm XSP
1 N	
START 'X' STO X	Schleife 1 zum Ermitteln der den x -Werten
k1 EVAL R→C N ROLL	der Schnittpunkte zugehörigen y -Werte
NEXT 'X' PURGE N	
DUPN 1 N	Stack duplizieren
START C→R DROP N	Schleife 2 zum Abrufen der x_s -Werte aus den
ROLL	komplexen Koordinaten.
NEXT 0 N	
START N ROLL	Schleife 3 zum Wiederherstellen der alte Rei-
NEXT 1 N	folge im Stack
FOR k '78+k'	Schleife 4 zum Erzeugen der Variablen, in
EVAL CHR STR→ STO	denen die x -Werte gespeichert werden, x -Werte
NEXT	speichern

Listing WIN2 Fortsetzung

```

1 N
  FOR k '78+k'
EVAL CHR STR→ 'X'
STO 'dX(k1)' →NUM 'd
X(k2)' →NUM WIN1
  NEXT
  » 'R' 'Q' 'P' '0'
  'X' 'PPAR' z k '5+N'
EVAL →LIST PURGE '8-
(5+N)' EVAL DROPN
  'N' PURGE
  »

```

Schleife 5 zum Abrufen der x-Werte aus den Variablen, speichern in X und Auswerten der Formeln für die Steigungen Unterprogramm WIN1

Variablen löschen

alle nicht mehr benötigten Stack-Objekte löschen

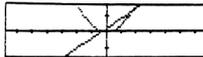
Beispiel: $f(x) = 3x + 1$ $g(x) = 2x^2$

Eingabe: WIN2 '3*X+1','2*X^2 |ENTER| █ |CONT|

Ausgabe: Meldung 3

Graph wird gezeichnet
Schnittpunkte digitalisi-
ert und Anzahl eingegeben

Graph



Schnittpunkte digitalisieren. Beim HP 28 C muß an dieser Stelle das Programm mit |ON| unterbrochen werden, da sich der Rechner nicht im interaktiven Modus befindet. Die Gleichung des Graphen wurde jedoch in EQ gespeichert und der Graph kann mit DRAW neu gezeichnet werden. Vorher läßt sich natürlich der Zeichenbereich ändern. Digitalisieren Sie die Schnittpunkte und setzen Sie das Programm mit █ |CONT| fort.

Ausgabe: Schnittpunkte: (1.781,6.342) (-0.281,0.158)
Schnittwinkel: 10.444° 60.116°

Listing XSP (Unteprogramm)

.....

```

< → 1
  < 1 LIST→ 'n' STO
1 n
  FOR i i ROLL
RCEQ SWAP X SWAP
ROOT 'X' PURGE
  NEXT
  > 'n' PURGE
>

```

3.3.14. root Bestimmung von Nullstellen

Das Programm root sucht selbstständig die Nullstellen einer beliebigen Funktion $f(x)$ im Intervall $[-5 \ 5]$. Vergleichen Sie dazu bitte auch Seite 41.

Dazu speichern Sie bitte die Funktion $f(x)$ mit |SOLV| STEQ in EQ. Falls Sie eine Funktion schon graphisch dargestellt haben, ist die dazu erforderliche Funktionsgleichung schon in EQ abgelegt.

Dann starten Sie das Programm mit |USER| root.

Als Ergebnis erhalten Sie alle Nullstellen der Funktion.

Listing:

.....

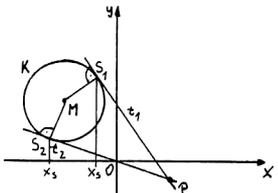
```

< 1 'z' STO -5 4
  FOR n EQ 'X' n n 1
+ 2 →LIST ROOT
  NEXT 3 FIX 0 3
  START RND SWAP RND
== LAST 3 ROLL
  IF 1 ==
  THEN DROP
  ELSE SWAP "NS" z
1 + 'z' STO z →STR +
STR→ STO
  END
  NEXT "NS" z 1 +
→STR + STR→ STO 'z'
PURGE
>

```

Die Nullstellen werden in den Variablen NO bis Nx abgelegt und können durch Betätigen der entsprechenden Variablen-Taste abgerufen werden.

3.3.15 SKG Schnittpunkt und Gleichung einer Tangente an Kreis



Das Programm SKG berechnet die Schnittpunkte und die Gleichungen zweier Tangenten t an einen Kreis, wenn diese durch einen bestimmten Punkt p außerhalb des Kreises laufen.

SKG nimmt die Parameter für den Mittelpunkt und Radius des Kreises und die Koordinaten von p vom Stack und berechnet als erstes die Strecken $\overline{x_m x_p}$ und $\overline{y_m y_p}$ als Koordinatendifferenz des Punktes p bezüglich des Kreises K.

Dann werden die eingegebenen Werte in eine Gleichung, welche der Tangente und dem Kreis im Schnittpunkt von t und K genügt, eingesetzt und der x-Wert der beiden Schnittpunkte (einer davon gespiegelt) berechnet. Der y-Wert der Schnittpunkte wird ebenfalls ermittelt und die Koordinaten von S_1 und S_2 in den Stack geschrieben.

Nach dieser Prozedur wird eine Meldung angezeigt, welche Sie vor die Wahl stellt, entweder das Programm abzubrechen oder die Tangenten-Gleichungen berechnen zu lassen. Wenn Sie die Taste |J| betätigen erscheinen im Stack die Gleichungen der Tangenten an K. Dazu wird zuerst die Steigung der Tangente im Schnittpunkt ermittelt und mit den Koordinaten von S in die allgemeine Geradengleichung eingesetzt.

Listing:


```

* CLLCD 2 FIX
"Schnittpkt Kreis - Ge -
"
1 DISP
"rade und Gleichung"
3 DISP 5 WAIT CLLCD
"Enter Xmk, Ymk, R, Xp, Yp"
1 DISP HALT + xM yM
r xp yp
* yp yM - 'B' STO
xp xM - 'A' STO
    
```

Programmname Meldung 2

Meldung 2

Koordinatendifferenz von p zu M

Listing Fortsetzung:

.....

```

-1 1
  FOR s1 '(f(-(2*(
2+2*A^2*B^(-2))*(
INV(B)*r^2)^2-r^2))+
(-(2*A*B^(-2)*r^2))^
2)*s1+2*A*B^(-2)*r^2
)/(2+2*A^2*B^(-2))'
EVAL DUP 'X' STO xm
+ '(r^2-A*X)/B' EVAL
ym + R>C 2
  STEP
"Geradengleichung ? "
1 DISP
"druecke J " 2
DISP
"nur Schnittpkte" 3
DISP "andere Taste"
4 DISP
  DO
  UNTIL KEY
  END "J" SAME
  IF 1 ==
  THEN CLMF 0 1
  START DUP C>R
'YS' STO 'XS' STO '(
YS-yp)/(XS-xp)' EVAL
'M' STO '-M*XS+YS'
EVAL 'Y' STO 'y=M*x+
Y' EVAL 3 ROLL
  NEXT SWAP
  ELSE
  END CLMF
  » { X R A B YS XS M
Y } PURGE
  »

```

Ausdruck für $f_{\text{Kreis}}=f_{\text{Tangente}}$ im Schnittpunkt S \longrightarrow X-Wert

X_s und Y_s berechnen
als Koordinate darstellen

Meldung 3

Schleife für Tastaturabfrage

Prozedur für Taste J

Geradengleichung berechnen

Prozedur für andere Taste

Variable löschen

Beispiel: $X_m = -3$ $Y_m = 2$ $R = 2$ $X_p = 2$ $Y_p = -2$

Eingabe: SKG -3,2,2,2,-2 |ENTER| █ |CONT|

Ausgabe: 1. Meldung P.-Name u.
Eingabe

Schnittpunkt Kreis - Gerade und Gleichung

Enter X_m, Y_m, R, X_p, Y_p

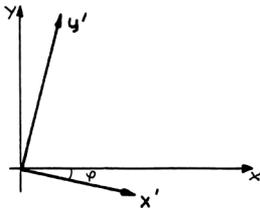
2. Meldung 3

Geradengleichung?
drücke J
nur Schnittpunkte
andere Taste

Schnittpunkte: (-1.33,3.09) (-3.7,0.13)

Tangenten: 'y=-(1.53*x)+1.06' 'y=-(0.37*x)-1.25

3.3.16. DREH Drehung des Koordinatensystems



Es kommt sehr häufig vor, daß die Gleichung eines Kegelschnitts diesen nicht in achsenparalleler Lage wiedergibt.

DREH transformiert eine Gleichung der Form: $ax^2+bxy+cy^2+dx+ey+f$ im gedrehten System in die Gleichung: $ax'^2+cy'^2+d'x'+e'y'+f$ im

normalen System. Dazu wird die Gleichung des Kegelschnitts in Form einer Koeffizientenliste eingegeben. Leere Stellen werden mit 0 aufgefüllt.

Das Programm berechnet dann den Drehwinkel und daraus mit Hilfe der

Transformationsgleichungen: $x = x' \cos \varphi - y' \sin \varphi$ und

$y = x' \sin \varphi + y' \cos \varphi$ die Funktion im

normalen System.

Der Drehwinkel ergibt sich aus: $\varphi = \frac{1}{2} * \text{ATAN} \frac{b}{a-c}$.

Der Drehwinkel und die Gleichung der Funktion im normalen System werden angezeigt. Diese Funktion können Sie dann z.B. mit dem Programm AFvF auf Seite 65 graphisch darstellen.

Beispiel: $f(x,y) = 5X^2 - 6XY + 5Y^2 - 32$ soll transformiert werden.

Eingabe: DREH { 5,-6,5,0,0,-32 |ENTER| █ |CONT|

Ausgabe: '2*X^2+8*Y^2-32'

Listing:

```

.....
< CLLCD
"DREHUNG KOORD.SYSTEM"           Programmname
1 DISP
"AX^2+BXY+CY^2+DX+EY+F"         Meldung 1
3 DISP
"ENTER (A,B,C,D,E,F)"
4 DISP 2 FIX HALT →
1
  < 1 LIST→ DROP → a
b c d e f
  < CLLCD
"DREHWINKEL = " '               Ergebnismeldung
IFTE(a*c,ATAN(b/(a-c
))/2,45)' EVAL DUP             Drehwinkel berechnen
'eta' STO →STR + ""
+ 2 DISP HALT CLLCD            Drehwinkel anzeigen und Programmstop
'a*x^2+b*y*x+c*y^2+d
*x+e+y+f' EVAL DUP             Funktionsgleichung bilden
'HG' STO 'x*cos(eta)
-y*sIN(eta)' EVAL 'x           x-Wert
*sIN(eta)+y*cos(eta)          y-Wert
' EVAL 'Y' STO 'X'
STO EVAL                       in Funktionsgleichung einsetzen und
  < EXPAN                       zusammenfassen
  > Repeat COLCT
DUP 'TG' STO CLLCD
→STR 3 DISP
"TRANSFORM. GLEICHUNG"         Ergebnismeldung
1 DISP HG TG eta (
eta X Y HG TG )               Variable löschen
PURGE
  >
  >
  >

```

3.3.17. PDIV Polynomdivision

Das Programm PDIV führt eine vollständige Polynomdivision durch. Dann wird das Ergebnispolynom und, falls vorhanden, ein Restpolynom angezeigt. Für die Berechnung wird das Nennerpolynom und das Zählerpolynom in Form

einer Koeffizienten-Liste eingegeben. Dabei ist darauf zu achte, daß das Zählerpolynom höhere oder gleich hohe Potenzen als das Nennerpolynom (Divisor) aufweist.

Das Programm PDIV ruft zwei Unterprogramme auf:

- UPDIV Das Programm UPDIV simuliert das vollständige Horner Schema für zwei Koeffizientenlisten
- UBEQ Das Programm generiert ein vollständiges Polynom aus einer Liste mit Koeffizienten.
Dieses Polynom wird in EQ gespeichert. Im Programm PDIV finden wir darin den ganzrationalen Term des Ergebnisses.

Listing PDIV (Hauptprogramm)
.....

```

< CLLCD
" Polynondivision"
1 DISP
"Enter(Zaehlerkoeffiz.)
3 DISP
" (Nennerkoeffiz.)"
4 DISP HALT UPDIV 1
3
START ROT ULF DROP
RCEQ
NEXT / + 'EQ'
PURGE
»
    
```

Listing ULF (Unterprogramm)
.....

```

< DUP LIST→ 1 - 0 0
3 ROLL
FOR k k 2 + ROLL
'X' k ^ * + -1
STEP STEQ
»
    
```

Listing UPDIV (Unterprogramm)
.....

```

< OVER SIZE OVER
SIZE - DUP 4 ROLL 0
IF ≥
THEN DUP 4 ROLL 3
PICK 0
IF >
THEN 1 4 PICK
START 0 +
NEXT
END 1 2
START SWAP LIST→
1 →LIST →ARRY
NEXT 0 4 PICK
START DUP 3
ROLLD DUP ( 1 ) GET
3 PICK ( 1 ) GET
SWAP / DUP 6 ROLLD *
- ARRY→ LIST→ - 1
→LIST →ARRY SWAP
DROP SWAP ARRY→ SWAP
DROP LIST→ - 1 →LIST
→ARRY
NEXT DROP ARRY→
LIST→ DROP →LIST
OVER 3 + ROLL 1 +
→LIST SWAP ROT
ELSE ROT DROP ( 0
) 3 ROLL
END
»
    
```

Das Programm UBEQ läßt sich natürlich auch sehr gut für andere Programme anwenden, welche Koeffizientenlisten zur Berechnung verwenden.

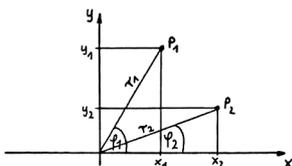
Beispiel: $3x^4-10x^3+22x^2-24x+10 : x^2-2x+3$

Eing.: PDIV { 3,-10,22,-24,10 |ENTER| { 1,-2,3 |ENTER| █ |CONT|

Ausg.: '3*X^2-4*X+5+(-2*X-5)/(X^2-2*X+3)' $\rightarrow 3x^2-4x+5 - \frac{2x-5}{x^2-2x+3}$

3.4. Komplexe Rechnung

3.4.1. PADD Polare Addition



Das Programm PADD addiert zwei komplexe Zahlen, die in Polarkoordinaten angegeben sind. Das Programm nimmt zwei Zahlen (r_1, φ_1) und (r_2, φ_2) vom Stack, rechnet diese in rechtwinklige Koordinaten um und addiert sie.

Anschließend wird das Ergebnis wieder in Polare Darstellung zurückverwandelt.

Listing:
.....

```
« 2 FIX → x y 'R→P(
P→R(x)+P→R(y)»
»
```

Beispiel: (2,45)+(4,60)

Eingabe: (2,45),(4,60) |ENTER| PADD

Ausgabe: (5.95,55.01)

Die folgenden drei Programme funktionieren nach dem gleichen Prinzip wie das Programm PADD. Es ändert sich im Programm nur das Rechenzeichen in der Funktion.

Listing PSUB Polare Subtraktion

```
« 2 FIX → x y 'R→P(
P→R(x)-P→R(y)»
»
```

Listing PMUL Polare Multiplikation

.....

```

« 2 FIX → x y 'R→P(
P→R(x)*P→R(y))'
»

```

Listing pdiv Polare Division

.....

```

« 2 FIX → x y 'R→P(
P→R(x)/P→R(y))'
»

```

3.5. Matritzen , Felder und Vektoren

3.5.1. VinM Vektoren in Matrix

.....

Das Programm VinM nimmt alle Vektoren vom Stack und übergibt sie in eine Matrix. Bedingung ist, daß alle Vektoren die gleiche Anzahl Elemente besitzen und der Stack leer ist.

Das Programm zählt die Stack-Objekte (Vektoren) und ihre Größe. Dann werden alle Vektoren aufgelöst, d.h. ihr Inhalt wird in den Stack geschrieben. Anschließend werden die einzelnen Elemente in eine Matrix geschrieben.

Listing:

.....

```

« HALT DUP DEPTH 1 -
SWAP SIZE LIST→ DROP
→ a b
« 0 a 1 -
FOR n n b # a n
- + ROLL ARRAY→ DROP
NEXT a b 2 →LIST
→ARRAY
»
»

```

Beispiel: [1,2,3] [2,3,4] [3,4,5] in Matrix

Eingabe: [1,2,3 |ENTER| [2,3,4 |ENTER| [3,4,5 |ENTER| VinM
.....

Ausgabe: [[1,2,3][2,3,4][3,4,5]]

3.5.2. MinV Matrix in Vektoren

Das Programm wandelt eine beliebig große Matrix in einzelne Vektoren um.
Dabei wird jede Matrixzeile ein Vektor.

Listing:
.....

```

< ARRY→ LIST→ DROP →
a b
  < 1 a
    FOR n b 1 →LIST
  →ARRY a n - b * n +
  ROLLD
  NEXT
  >
>

```

Beispiel siehe VinM.

3.5.3. GLL Gleichungssystem lösen

Das Programm GLL löst ein System von Gleichungen bzw. eine Matrix mit den
Koeffizienten und dem Lösungsvektor.

Die dabei zur Anwendung kommende Formel lautet: $L = \frac{A^t * B}{A^t * A}$

Listing:
.....

```

< CLLCD
  "Inhomog. Glg-System"
  i DISP
  "Enter [[AX]], [B] "
  3 DISP HALT → ma v
  < v ARRY→ LIST→ 2
  →LIST →ARRY ma TRN
  SWAP * ma TRN ma * /
  >
>

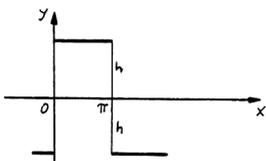
```

Beispiel: $x_1 + 2x_2 + 3x_3 + 4x_4 = 1$
 $2x_1 + 3x_2 + 4x_3 + 5x_4 = 2$
 $3x_1 + 4x_2 + 5x_3 + 6x_4 = 2$

Eing.: GLL [[1,2,3,4][2,3,4,5][3,4,5,6 |ENTER| [1,2,2 |ENTER| █ |CONT|

Ausg.: [[-0.12] [0.59] [0.01] [0.02]]

3.6. Fourier Reihe



Entwickelt und gezeichnet werden soll eine Fourierreihe der Form:

$$f(x) = \frac{4h}{\pi} (\sin x + \frac{1}{3} \sin 3x + \frac{1}{5} \sin 5x \dots)$$

Das ist eine Rechteckkurve s. Bild. Nehmen wir an, Sie wollen die ersten 5 Terme dieser Reihe graphisch darstellen. Dazu können Sie ein kleines Programm schreiben, das Ihnen die einzelnen Terme berechnet, sie addiert und in EQ speichert, um die Funktion zu zeichnen.

Die einzelnen Terme können Sie etwas umschreiben in: $\frac{4 \cdot h}{\pi} * \frac{\sin(i \cdot X)}{i}$.

Nun müssen Sie nur noch sicherstellen, daß die Variable i von 1 aus in Dreier-Schritten anwächst. Das geht mit einer kleinen Schleife, die in Zweier-Schritten ab 1 den Wert erhöht. Sie sieht so aus:

```
1 2*n FOR i ... 2 STEP.
```

Dann addieren sie die Terme, speichern sie in EQ und lassen die Funktion zeichnen. Die 0 am Anfang des Programms ist der erste Summand.

Listing:


```
< RAD CLLCD
"  Enter h,n" 2
DISP HALT → h n
< 0 1 n 2 *
  FOR i '4*h/π*SIN
(i#X)/i' EVAL + 2
  STEP CLLCD STEP
DRAW
»
»
```

Pocket + Laptop Computer

INHALT

I. Grundlagen

- 1) Variablentypen: Lokalität von Bezeichnern 3
- 2) Menüsteuerung: Programmbaum und MENU-Befehl 5
- 3) Die Flags und ihre praktische Anwendung 8

II. Mathematische Programme

- 1) Algebra und Numerik 10
- 2) Lineare Algebra 19
- 3) Komplexe Zahlen 23
- 4) Folgen und Reihen 24
- 5) Funktionen 29
- 6) Statistik 37

III. Elektrotechnik

- 1) Ortskurven 41
- 2) Netzwerke 42
- 3) Zweitore 43
- 4) Nachrichtentechnik 46
- 5) Digitale Technik 48

IV. Graphik und Spiele

- 1) HP-Graphik-Strings 52
- 2) 'Desktop-Publishing' 56
- 3) Bewegte Graphik 1: Lissajous-Figuren 58
- 4) Bewegte Graphik 2: TILT - ein schnelles Aktionsspiel 60
- 5) Fraktale Graphik: Die Mandelbrotmenge 62

V. Betriebssoftware

- 1) Drucker-Handling 64
- 2) Directory-Management 67

VI. Synthetische Programmierung

- 1) Der SYSEVAL-Befehl 71
- 2) Systemuhr im Hintergrund 73
- 3) Maschinensprache auf dem HP-28S? 76

VII. Ergänzungen zu den Rechner-Handbüchern

77

ANHANG: 250 SYSEVAL-Adressen

80

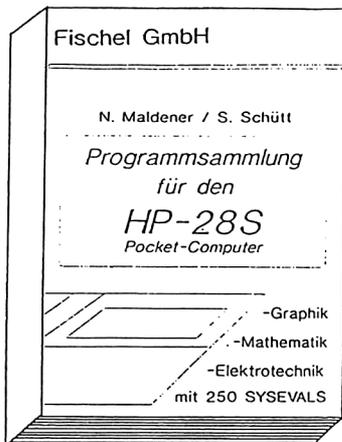
INDEX

86



**HEWLETT
PACKARD**

sofort lieferbar !



49,- DM
(inkl. 7% MwSt.)

POCKET COMPUTER

Fischel GmbH

Kaiser-Friedrich-Straße 54 a

1000 Berlin 12 - Tel. 030 / 323 60 29

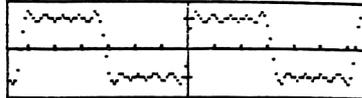
Durch Information vorn

h und n im Programm bedeuten: h ist die Höhe des Rechtecks über der x-Achse

n ist die Anzahl der gewünschten Berechnungen

Zurück zum Beispiel:

sie erhalten diese Grafik:



3.7. Statistik

Allgemeines:

In diesem Kapitel werden einige auserwählte Beispiele für die Anwendung statistischer Berechnungen auf dem HP 28 S/C beschrieben.

Dabei gilt:

$$x_i \quad \text{Beobachtungsreihe } x_1 \dots x_n$$

$$i, n \in \mathbb{N} \quad 1 \leq i \leq n$$

Die Eingabe der Daten $x_1 \dots x_n$ erfolgt über die Funktion $|\Sigma|$.

Werden weitere Eingaben benötigt, so werden sie explizit bei den einzelnen Programmen beschrieben.

Programme:

In den meisten Programmen, die in diesem Kapitel beschrieben sind, werden zwei Unterprogramme aufgerufen, die nachfolgend erklärt werden.

3.7.1. INIT Größe einer Stat-Matrix

INIT speichert die Größe einer Matrix in den Variablen x und y.

Listing:

.....

```

< RCL SIZE LIST+
DROP 'x' STO 'y' STO
>
    
```

3.7.2. FIN Ende-Programm

FIN beendet ein Programm, legt die Ausgabe der Werte fest und löscht alle Variablen.

Listing:
.....

```

« 2 FIX
  IF x 1 >
    THEN x 1 →LIST
  →ARRY
  END ( x y z )
PURGE
»

```

A. Lagemaßzahlen:

Die Lagemaßzahlen charakterisieren das allgemeine Niveau einer quadratischen Merkmalsausprägung. Eine Gruppe von Lagemaßzahlen sind die Mittelwerte, wobei das arithmetische Mittel (HP 28 Standardfunktion MEAN) den am häufigsten verwendeten Mittelwert darstellt.

Nachfolgend werden einige andere interessante Mittelwerte besprochen.

3.7.3. GEOM Geometrisches Mittel

Die Anwendung des arithmetischen Mittelwertes ist bei relativen Änderungen der Merkmalsausprägungen (z.B. Lohnerhöhungen oder Produktionssteigerung) nicht sinnvoll.

Deshalb bedient man sich dazu des geometrischen Mittels \bar{x}_g .

$$\bar{x}_g = \sqrt[n]{\prod_{i=1}^n x_i} \quad \text{wobei } x_i \geq 0$$

Eingabe: Statistikpaare

Ausgabe: \bar{x}_g oder $[\bar{x}_{g1}, \dots, \bar{x}_{gn}]$

Listing:
.....

```

« INIT 1 x
  FOR n 1 y
    FOR m RCLΣ m n 2
  →LIST GET

```

Listing Fortsetzung:


```

NEXT 1 y 1 -
FOR j *
NEXT NΣ INV ^
NEXT FIN
    
```

Beispiel:

Ein Unternehmen teilt sich in zwei Bereiche, wobei jeder Bereich die prozentuelle Produktionssteigerung getrennt errechnet wurden.

Bereich a: -3% , -1.5% , 4% , 3% , 5%

b: 1% , 2% ,1.5%,3.5%, 4%

Berechne die mittlere Änderungsrate \bar{x}_g der letzten 5 Jahre getrennt für jeden Unternehmensbereich.

Lösung:

Wegen $1 + \frac{x_i}{100}$ ergibt sich für

Bereich a: 0.97 0.985 1.04 1.03 1.05

b: 1.01 1.02 1.015 1.035 1.04

Diese Daten werden in Form von Statistikpaaren [0.97 1.01], [0.985 1.02], ... mit |Σ+| eingegeben.

Nach drücken von $\overline{\text{GEOM}}$ erhält man den Lösungsvektor [1.01 1.02], der auf zwei Stellen gerundete \bar{x}_g -Werte enthält.

Die mittlere Änderung des Bereichs a beträgt ca. 1% und die von b ca. 2%.

3.7.4. HARM Harmonisches Mittel

Ein weiterer oft benötigter Mittelwert ist das harmonische Mittel \bar{x}_h .

Das gewöhnliche harmonische Mittel hat die Form:

$$\bar{x}_h = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}} \quad \text{wobei } x_i \neq 0$$

Eingabe: $x_1 \dots x_n$

Ausgabe: \bar{x}_h oder $[\bar{x}_{h1}, \dots, \bar{x}_{hn}]$

Listing:

.....

```

« INIT 1 x
  FOR n 1 y
    FOR m RCLΣ m n 2
  →LIST GET
    NEXT INV 1 y 1 -
    FOR j SWAP INV +
    NEXT NΣ SWAP /
  NEXT FIN
»

```

3.7.5. GHARM Gewichtetes harmonisches Mittel

.....

In der Physik ist es oft erforderlich, eine Mittlung von Geschwindigkeiten über unterschiedliche Strecken zu berechnen. Dies geschieht mit dem gewogenen harmonischen Mittel \bar{x}_h .

Wir haben: $v_1 \dots v_n$ n- Geschwindigkeiten
 $s_1 \dots s_n$ n- Streckenlängen
 $t_1 \dots t_n$ n- Zeitabschnitte

Die Formel für den Weg s ist: $s = v * t$

Die Gesamtstreckenlänge ergibt sich aus: $S = \sum s_i$

Die Gesamtzeit erhält man aus $T = \sum t_i$, wobei $t_i = s_i / v_i$ ist

Daraus kann die Durchschnittsgeschwindigkeit ermittelt werden.

Es gilt: $S = \bar{v} * T \longrightarrow \bar{v} = \frac{\sum_{i=1}^n s_i}{\sum_{i=1}^n s_i / v_i}$, wobei $s_1 \dots s_n$ die allgemeine Gewichtung darstellt.

Listing:

.....

```

« 0 'x' STO 0 'y'
STO RCLΣ ARRAY→ DROP
1 NΣ
  FOR n OVER 'x'
STO+ / 'y' STO+
  NEXT x y / 2 FIX (
x y } PURGE
»

```

Beispiel:

Ein Autofahrer fährt die Strecke \overline{AB} mit folgenden Geschwindigkeiten (in km/h) : 140,60,100,110,180, wobei für jede Geschwindigkeit der Weg gemessen wurde: 50km,45km,80km,60km,100km für eine Zeit t.

Wie groß ist die Durchschnittsgeschwindigkeit?

Lösung:

Eingabe der Vektoren [Strecke Geschwindigkeit] in \sum^+ . Das Programm GHARM liefert dann 111.36 km/h als mittlere Geschwindigkeit.

3.7.6. PQMi Quadratisches und Potenzmittel

Das quadratische Mittel \overline{x}_q hat die Form:

$$\overline{x}_q = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}, \text{ wobei } x_i \in \mathbb{R}$$

und der Potenzmittelwert \overline{x}_p die Form:

$$\overline{x}_p = \sqrt[r]{\frac{1}{n} \sum_{i=1}^n x_i^r}, \text{ bei ungeraden } r \text{ muß } x_i > 0 \text{ sein}$$

Aus den beiden Formeln für \overline{x}_q und \overline{x}_p wird ersichtlich, daß das quadratische Mittel ein Spezialfall des Potenzmittelwertes ($r=2$) ist.

Listing:

```

«      DEPTH 0 == 2
IFT 'z' STO INIT 1 x
  FOR n 1 y
    FOR n RCLΣ m n 2
+LIST GET
    NEXT z ^ 1 y 1 -
    FOR j SWAP z ^ +
    NEXT NΣ INV * z
  INV ^
  NEXT FIN
»

```

Eingabe: r(=Potenz), default=2

Ausgabe: \overline{x}_p oder $[\overline{x}_{p1} \dots \overline{x}_{pn}]$

Beispiel:

Von den nachfolgenden Datenpaaren soll das quadratische Mittel berechnet werden.

a: 57 102 92

b: 366 621 250

Lösung:

Speichern der Stat.-Paare [57 366] in Σ^+ .

Für r ist beim quadratischen Mittel kein Wert einzugeben. Nach drücken der Menü-Taste PQMi erhalten wir den Lösungsvektor [85.86 440.49].

Bei einer genaueren Betrachtung des Potenzmittels wird ersichtlich, daß für bestimmte r-Werte das arithmetische, das geometrische und das harmonische Mittel weitere Spezialfälle des Potenzmittels darstellen.

Harmonisches Mittel: $r = -1$ ($\bar{x}_p = \bar{x}_h$)

Arithmetisches Mittel: $r = 1$ ($\bar{x}_p = \bar{x}$)

Quadratisches Mittel: $r = 2$ ($\bar{x}_p = \bar{x}_q$)

Geometrisches Mittel: r gegen 0 ($\bar{x}_p = \bar{x}_g$)

B. Streuungsmaßzahlen

Diese Maßzahlen stellen die Lage der Merkmalsausprägungen zueinander fest.

3.7.7. RANGE

Listing:

```
< MAXΣ MINΣ -
>
```

Beispiel:

Folgende Werte sind zu bestimmen: 5 29 17 6 -1 7. Wie groß ist der Range?

Lösung:

Der größte Wert ist 29, der kleinste -1. Daher beträgt der Range 30.

3.7.8. GINI Gini-Maß

Das Gini-Maß Δ_r ist eine Maßzahl, die auf den Abständen aller Merkmalsausprägungen voneinander beruht.

$$\text{Differenz aller Abstände} = \Delta_r = \frac{2}{n(n-1)} \sum_{i=1}^n \sum_{\substack{j=1 \\ i < j}}^n |x_i - x_j|$$

wobei $\frac{n(n-1)}{2}$ die Anzahl aller möglichen Abstände darstellt.

Listing:

```

« INIT 1 x
  FOR n 0 1 y 1 -
    FOR i 1 1 + y
      FOR j RCLΣ 1 n
2 +LIST GET RCLΣ j n
2 +LIST GET - ABS +
      NEXT
    NEXT 2 * y y 1 -
* /
  NEXT FIN
»

```

Beispiel:

Von folgenden Merkmalsausprägungen soll das Gini-Maß ermittelt werden:

78 19 63.

Lösung:

Es sind wegen $\frac{n(n-1)}{2}$ nur 3 mögliche Abstände zu berechnen:

$$|78-19| + |78-63| + |19-63| = 118$$

118 ist die Summe aller Abstände, die nur noch durch die Anzahl zu dividieren ist. Mit Hilfe von GINI erhält man nach Eingabe der Daten 39.33 als Ergebnis.

Berechnung der Varianz mittels Verschiebungssatz

Als eine der wichtigsten Streuungsmaßzahlen gilt die Varianz, die das arithmetische Mittel \bar{x} als Abstandsmittelung besitzt.

Wobei $\frac{1}{n} \sum (x_i - \bar{x})$ eine Minimumseigenschaft aller Maßzahlen vom Typ $\frac{1}{n} \sum (x_i - C)$

besitzt (C= Lagemaß).

Die allgemeine Formel zu Berechnung der Varianz ist.

$$s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

Die im Handbuch des HP 28 beschriebene Formel $s^2 = \frac{1}{1-n} \sum_{i=1}^n (x_i - \bar{x})^2$ bezeichnet man oft als Stichprobenvarianz einer unendlichen Grundgesamtheit und wird häufig in der schließenden Statistik verwendet.

Da sich bei numerischen Berechnungen oft Rundungsfehler bei der Berechnung des Mittelwertes ergeben, ist es oft vorteilhafter den sog. Verschiebungssatz zu verwenden.

$$s^2 = \frac{1}{n} \left[\sum_{i=1}^n x_i^2 - \frac{(\sum_{i=1}^n x_i)^2}{n} \right]$$

Da diese Formel \bar{x} nicht verwendet, ist der verschiebungssatz stabil gegen Rundungsfehler.

3.7.9. VSA Verschiebungssatz

Eingabe: Statistikpaare

Ausgabe: s^2 laut Verschiebungssatz

Listing:


```

« INIT 1 x
  FOR n 0 1 y
    FOR i RCLΣ i n 2
+LIST GET SQ +
      NEXT 0 1 y
      FOR i RCLΣ i n 2
+LIST GET +
        NEXT SQ y / - y
      /
    NEXT FIN
»
    
```

Beispiel siehe Potenzmittelwert

Lösung:

Nach Betätigen von VSA erhält man [372.22 24013.56] als Lösungsvektor.
 ...

Beachten Sie, daß schon bei diesen wenigen Zahlen eine kleine Differenz (im Bereich 10^{-8}) zur normalen Varianzberechnung auftritt.

3.7.10. STAW Standartabweichung

Die hier verwendete Standartabweichung baut auf die vorher besprochene Varianz auf.

$$s = \sqrt{\frac{1}{n} \left[\sum_{i=1}^n x_i^2 - \frac{\left(\sum_{i=1}^n x_i \right)^2}{n} \right]}$$

Listing:
.....

```

* VSA INIT
  IF x 1 ==
    THEN ↓
  ELSE ARRAY→ DROP 1
*
  FOR i i ROLL ↓ i
  ROLLD
  NEXT
  END FIN
*
```

Beispiel: wie bei Potenzmittelwert und Varianz

Lösung:

Der Lösungsvektor für die Standartabweichung lautet: [19.29 154.96].

4. Fehlersuche in Programmen

Es kann nach Eingabe eines Programms vorkommen, daß das Programm nach seinem Starten plötzlich abgebrochen oder unterbrochen wird und eine Fehlermeldung in der Anzeige erscheint.

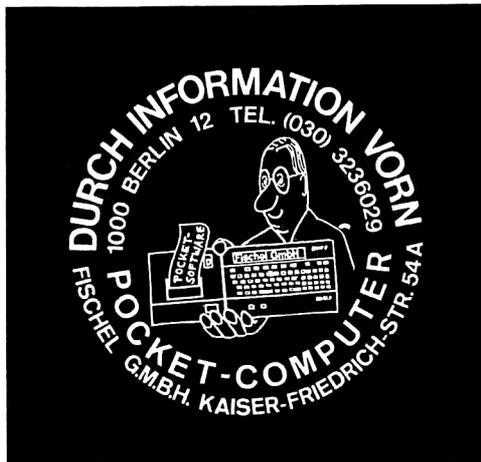
Gehen Sie dann bitte so vor:

1. Die Meldung des HP 28 gibt Auskunft über die Art des Fehlers. Lesen Sie dazu bitte im Referenzhandbuch die Seiten 349 bis 354 über Fehlermeldungen.
 2. Betätigen Sie eine beliebige Taste zum Anzeigen des normalen Stacks.
 3. Die Anordnung des Stacks in Verbindung mit der Fehlermeldung kann Aufschluß über die nicht durchführbare Operation geben. Suchen Sie in diesem Falle die Stelle im eingegebenen Programm:
 - 4a. Dies können Sie, indem Sie das entsprechende Programm mit 'Name |ENTER| █ |VISIT| aufrufen. Sie befinden sich jetzt im Editier-Modus. Stellen Sie den Cursor in die entsprechende Programmzeile. Bedienen Sie sich dazu der Cursor-Taste |▼|.
 - 5a. Vergleichen Sie das eingegebene Programm an dieser Stelle mit dem vorgegebenen Listing.
 - 6a. Haben Sie einen Eingabefehler entdeckt, so bessern Sie in z.B. mit DEL, INS oder durch Überschreiben aus und betätigen |ENTER|. Das geänderte Programm wird wieder in dem Programmnamen gespeichert.
 - 7a. Starten Sie das Programm erneut. Tritt wieder ein Fehler auf wiederholen Sie diese Prozedur.
- Sollte Ihre Eingabe richtig sein oder wissen Sie mit der ausgegebenen Meldung nichts anzufangen, können Sie das Programm schrittweise überprüfen.
- 4b. Editieren Sie das Programm mit █ |VISIT|.
 - 5b. Fügen Sie in das entsprechende Programm oder Unterprogramm am Anfang den Befehl HALT aus dem Programm Control-Menü ein. Beachten Sie, daß der Befehl nur in dem Programm oder Unterprogramm gilt, in dem er steht. Ist in diesem Programm ein Unterprogramm ent-

halten, so wird das Unterprogramm nicht schrittweise durchlaufen, wenn kein HALT-Befehl darin vorkommt.

- 6b. Geben Sie das Programm mit |ENTER| zurück in seinen Namen.
- 7b. Starten Sie das Programm neu. Es wird an der "Haltestelle" automatisch unterbrochen.
- 8b. Wählen Sie das Programm Control-Menü an : █ |Program Contrl|.
- 9b. Mit dem Menü-Befehl SST (Single Step) läßt sich das Programm debuggen. Verfolgen Sie schrittweise den Programmablauf, wobei jeder im Programm verwendete Befehl in der obersten Zeile angezeigt wird. So können Sie die Fehlerstelle genau lokalisieren und anhand der Stack-Konfiguration und der Fehlermeldung auf den Programmfehler schließen.
- 10b. Korrigieren Sie ihn und starten Sie das Programm nach Betätigen von |ENTER| erneut.

Sind alle Fehler verbessert, entfernen Sie den Befehl HALT wieder aus dem Programm.



5. Menüorganisation (nur HP 28 S)

Es ist sinnvoll, die einzelnen Programme in Oberbegriffen zusammenzufassen und die Oberbegriffe im Haupt oder HOME-Menü zu speichern.

Mit dieser Methode haben Sie einen viel Schnelleren Zugriff auf die entsprechenden Programme, da Sie das Haupt-Menü nicht ständig durchzublättern haben.

Definieren Sie schon vor der Eingabe von Programmen ihren Überbegriff und das Menü in das sie gespeichert werden sollen. Natürlich können Sie diese Organisation auch nachträglich durchführen.

Beispiel:

Sie wollen alle Programme, welche einen Graphen zeichnen unter dem Oberbegriff "GRAPH" speichern.

Geben Sie den Namen GRAPH in den Stack und betätigen Sie im Memory-Menü den Befehl CRDIR: 'GRAPH |ENTER| █ |MEMORY| CRDIR .

Der Name GRAPH steht nun im Home-Verzeichnis ihres Rechners.

Wählen Sie das Home-Verzeichnis mit |USER| an und Betätigen Sie GRAPH

Es erscheint ein leeres Menü, da Sie sich im GRAPH-Menü befinden. Nun können Sie alle Zeichenprogramme ganz normal in dieses Menü speichern.

Wollen Sie in das GRAPH-Menü keine Programme mehr eingeben, so kehren Sie mit dem Befehl HOME im Memory-Menü zum Hauptmenü zurück:

```
█ |MEMORY| HOME |USER|
  ....
```

Definieren Sie einen neuen Überbegriff und gehen Sie genauso vor wie beschrieben.

Befinden Sie sich in einem Untermenü wie z.B. in GRAPH können Sie natürlich wieder ein Untermenü im GRAPH-Menü mit CRDIR erzeugen. So erhalten Sie eine übersichtliche Menügestaltung.

Sehen Sie sich den Menü-Vorschlag für die in diesem Buch enthaltenen Programme an.

Haben Sie mehrere Untermenüs in geschachtelter Form erzeugt, so kommen Sie z.B. aus dem Menü ALG im ALGEB-Menü nur durch betätigen von HOME in das

Haupt-Menü zurück. Sie können auch nur in das Hauptmenü zurückkehren. Wollen Sie aber nur eine Ebene höher, z.B. vom Untermenü ALG ins Untermenü ALGEB, so müssen Sie erst ins Hauptmenü und dann ins ALGEB-Menü.

Ein kleines Programm simuliert diese Prozedur und blättert z.B. vom ALG-Menü ins ALGEB-Menü hoch.

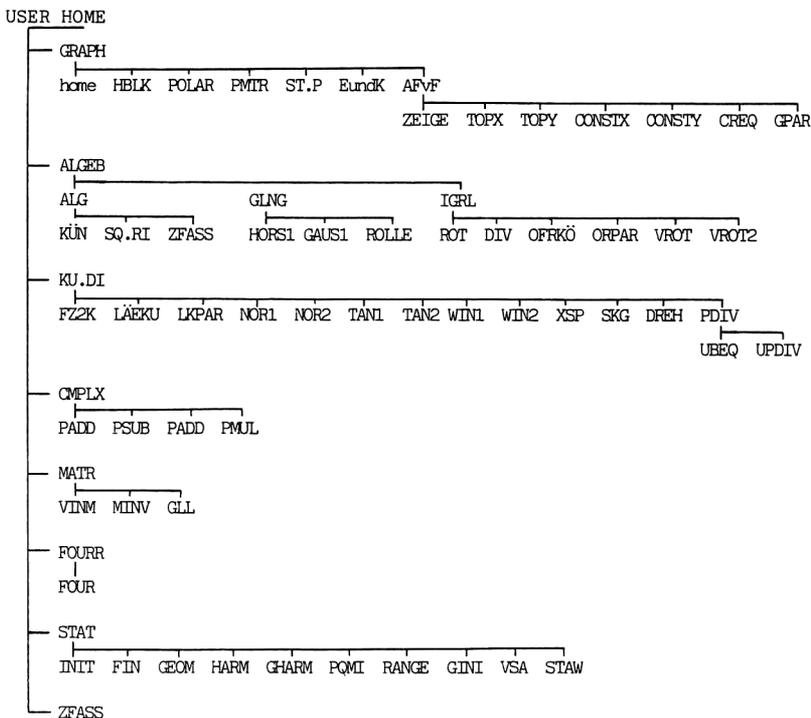
<<HOME ALGEB>>

Nennen Sie dieses Programm zweckmäßig algeb zum Definieren der Prozedur. Sie müssen aber den Namen anders schreiben, hier z.B. klein.

Es ist auch sinnvoll in die erste Unterprogrammebene ein Programm home zu speichern, das ihnen die Anwahl des Home-Menü gestattet: HOME .

Beispiel Menüorganisation:

.....



Haben Sie nun schon alle Programme gespeichert und wollen sie nachträglich ordnen gehen Sie so vor:

Rufen Sie das entsprechende Programm z.B. HBLK aus dem Home-Menü auf:

|USER| 'HBLK |ENTER| █ |RCL|

Bilden Sie das GRAPH Menü: 'GRAPH |ENTER| █ |MEMORY| CRDIR

Gehen Sie mit |USER| ins Home-Menü und mit GRAPH ins GRAPH-Menü.
.....

Speichern Sie das Programm normal mit 'HBLK |ENTER| |STO| ab.

Gehen Sie ins Home-Menü und löschen Sie dort HBLK.

Gehen Sie bei allen Programmen so vor.

Ein Unterverzeichnis können Sie mit dem Befehl: CLUSR löschen, wenn Sie sich im Unterverzeichnis befinden.

Ein andere Möglichkeit ist das Löschen aller Variablen mit:

HBLK POLAR ... █ |PURGE|

Literaturverzeichnis

Lehr und Übungsbuch Mathematik 1-3, Harri Deutsch Verlag, Leipzig 1983

w.P. Minorski: Aufgabensammlung der höheren Mathematik, Vieweg, Braunschweig 1986

Kusch: Mathematik 1-4, Girardet Verlag, Essen 1978

Bronstein-Semendjajew: Taschenbuch der Mathematik, Nauka Verlag, Moskau 1987

Bartsch: Taschenbuch mathematischer Formeln, Harri Deutsch Verlag, Leipzig 1986

Stoer-Bulirsch: Einführung in die numerische Mathematik 1 und 2, Berlin 1980

Vorlesungsauszüge

ATARI, CASIO, HEWLETT-PACKARD, SHARP

Super-Bestellschein

Hiermit bestelle ich:

Anzahl: _____ Buch: _____

Atari

PC Portfolio Anwendungshandbuch
ISBN 3-89374-046-5, VK = 49,- DM

Casio FX-850P

FX-850P Anwendungshandbuch
ISBN 3-89374-000-7, VK = 49,- DM
Der FX-850P in Deiner Hand
ISBN 3-89374-020-0, VK = 49,- DM

Casio PB-1000 / 2000

PB-1000 Tips und Tricks Programmierhandbuch
ISBN 3-89374-007-4, VK = 48,- DM
PB-1000 Anwendungshandbuch
ISBN 3-89374-028-7, VK = 49,- DM
PB-1000 Intern
ISBN 3-89374-028-7, VK = 59,- DM
PB-1000 Power-Software
ISBN 3-89374-044-9, VK = 49,- DM
PB-1000 Systemhandbuch
ISBN 3-89374-043-7, VK = 49,- DM
PB-2000 Anwendungshandbuch
ISBN 3-89374-042-2, VK = 46,- DM

Hewlett Packard

HP-28C/S Anwenderprogramme
ISBN 3-89374-029-5, VK = 49,- DM
HP-28C/S Programmierung
ISBN 3-89374-041-4, VK = 49,- DM

Sharp PC-1500(A)/PC-1600

PC-1500A/PC-1600 Anwendungshandbuch,
ISBN 3-89374-130-0, VK = 48,- DM
PC-1500A Tips und Tricks,
ISBN 3-89374-122-2, VK = 49,- DM
Ergänzungshft zum PC-1500A Maschinensprachehand-
buch, ISBN 3-89374-173-3, VK = 15,- DM
PC-1500 Intern von Schleifer
VK = 59,- DM
PC-1600 Systemhandbuch,
ISBN 3-89374-319-9, VK = 49,- DM
PC-1600 Anwendungshandbuch,
ISBN 3-89374-227-55-6, VK = 49,- DM
PC-1600 Maschinensprachehandbuch
ISBN 3-89374-401-5, VK = 49,- DM
PC-1600 Tips und Tricks Programmierhandbuch,
ISBN 3-89374-227-85-5, VK = 49,- DM
Die besten Programme für den PC-1600
ISBN 3-89374-040-8, VK = 49,- DM

Sharp PC-1401/02/03/21/50/75

PC-1401/02 Systemhandbuch,
ISBN 3-89374-227-31-9, VK = 39,- DM
PC-1401/02 Maschinensprachehandbuch,
ISBN 3-89374-227-11-4, VK = 49,- DM
PC-1403 Systemhandbuch,
ISBN 3-89374-227-59-4, VK = 39,- DM
PC-1403 Anwendungshandbuch,
ISBN 3-89374-227-65-3, VK = 49,- DM
PC-1403 Maschinensprachehandbuch,
ISBN 3-89374-227-73-4, VK = 49,- DM
Die besten Programme für den PC-1403
ISBN 3-89374-039-2, VK = 49,- DM
PC-1450 Anwendungshandbuch,
ISBN 3-89374-227-18-1, VK = 49,- DM
PC-1450 Maschinensprachehandbuch,
ISBN 3-89374-227-23-4, VK = 49,- DM
PC-1401/02/21/03 Tips und Tricks-Programmhandbuch,
ISBN 3-89374-227-33-5, VK = 49,- DM
PC-1401/02/21 Maschinenspracheprogrammiersammlung,
ISBN 3-89374-227-18-1, VK = 49,- DM
PC-1421 Begleitb.,
ISBN 3-89374-227-28-9, VK = 15,- DM
PC-1475/PC-1280 Anwendungshandbuch,
ISBN 3-89374-227-95-5, VK = 49,- DM
PC-1475/PC-1280 Maschinensprachehandbuch
ISBN 3-89374-008-2, VK = 49,- DM
PC-1475/PC-1280 Systemhandbuch
ISBN 3-89374-227-96-3, VK = 49,- DM
PC-1475/PC-1280 Tips + Tricks Programmierhandbuch
ISBN 3-89374-007-6, VK = 49,- DM
Der Sharp in Deiner Hand
VK = 48,- DM

Sharp PC-2500

PC-2500 Systemhandbuch,
ISBN 3-89374-220-3, VK = 49,- DM

Sharp PC-1350/60

Probheft der Zeitschrift "POCKET + LAPTOP COMPUTER" gegen 6,- DM in Briefmarken

(aus dem Ausland 7,- DM Euroschek oder Internationale Postwertzeichen)

ensenden an:

Pocket + Laptop Computer
Fischel GmbH
Kaiser-Friedrich-Str. 54a,
1000 Berlin 12,
Tel.: 030/323 60 29

PC-1350 Anwendungshandbuch,
ISBN 3-89374-227-15-7, VK = 49,- DM
PC-1350 Maschinensprachehandbuch,
ISBN 3-89374-227-10-6, VK = 49,- DM
PC-1350 Systemhandbuch
ISBN 3-89374-012-0, VK = 49,- DM
Die besten Programme für den PC-1350
ISBN 3-89374-038-4

Sharp PC-1100/1150/1245/1246S/

48/51/60/61/80

PC-1100 Anwendungshandbuch,
ISBN 3-89374-227-45-8, VK = 39,- DM
PC-1150/1246S/1248/1262/1270 Anwendungshandbuch
ISBN 3-89374-008-5, VK = 49,- DM
PC-1260/81 Maschinensprachehandbuch,
ISBN 3-89374-227-29-7, VK = 49,- DM
PC-1475/PC-1280 Anwendungshandbuch,
ISBN 3-89374-227-95-5, VK = 49,- DM
PC-1280/PC-1475 Systemhandbuch
ISBN 3-89374-227-96-3, VK = 49,- DM
PC-1280/PC-1475 Maschinensprachehandbuch
ISBN 3-89374-008-2, VK = 49,- DM
PC-1280/1475 Tips + Tricks Programmierhandbuch
ISBN 3-89374-037-8, VK = 49,- DM

Sharp MZ 700/800

MZ-700/800 Maschinensprachehandbuch,
ISBN 3-89374-027-6, VK = 48,- DM

Computertechnik allgemein

"Desktop Publishing (DTP)" mit Pocket Computern
ISBN 3-89374-030-9, VK = 49,- DM
Rechnetechnik mit Sharp Taschencomputern,
ISBN 3-89374-027-40-9, VK = 49,- DM
Umsetzungshandbuch für Sharp Taschencomputer,
ISBN 3-89374-227-77-7, VK = 49,- DM
Hardware-Entwicklung für Sharp Pocket - Computer
ISBN 3-89374-227-82-4, VK = 49,- DM
Entwicklung Intelligenter Hardware mit Sharp Taschen-
computern, ISBN 3-89374-043-0, VK = 49,- DM
Maschinensprachehandbuch für Sharp Taschencomputer,
ISBN 3-89374-227-74-2, VK = 49,- DM
Einführung in die Maschinensprache für Sharp Taschen-
computer, ISBN 3-89374-027-9, VK = 49,- DM
Datenübertragungshandbuch für Sharp Taschencomputer,
ISBN 3-89374-227-43-7, VK = 49,- DM
CAD- und Grafikprogrammierung für Sharp
Taschencomputer, ISBN 3-89374-227-44-0, VK = 49,- DM
Basic-Erweiterungen für Sharp Taschencomputer,
ISBN 3-89374-227-45-8, VK = 49,- DM
Hacker-Handbuch für Sharp Computer,
ISBN 3-89374-227-24-6, VK = 49,- DM
Computerlexikon und Recordshandbuch für Sharp
Computer, ISBN 3-89374-227-21-1, VK = 49,- DM
Schriftschrift und Textverarbeitung für Sharp Computer,
ISBN 3-89374-227-37-8, VK = 49,- DM
Lehr- und Übungshandbuch für Sharp Taschencomputer
ISBN 3-89374-006-8, VK = 49,- DM
Basic Lehrbuch für Sharp Computer,
ISBN 3-89374-227-69-2, VK = 49,- DM
Grafikhandbuch für Sharp Computer,
ISBN 3-89374-027-04-1, VK = 49,- DM

Wirtschaftswissenschaften

Hauswirtschaft mit Pocket Computern
ISBN 3-89374-031-7, VK = 49,- DM
Kaufmannsprache Programmierung für Sharp Taschen-
computer, ISBN 3-89374-227-75-0, VK = 49,- DM
Betriebswirtschaft mit Sharp Taschencomputern,
ISBN 3-89374-227-69-8, VK = 49,- DM
Finanz- und Wirtschaftsprogrammierung für Sharp
Computer, Band 1, ISBN 3-89374-227-30-0, VK = 49,- DM
Finanz- und Wirtschaftsprogrammierung für Sharp
Computer, Band 2, ISBN 3-89374-227-97-1, VK = 49,- DM
Wertpapierverwaltung mit Sharp Taschencomputern,
ISBN 3-89374-227-62-0, VK = 49,- DM
Pocket Computer Programmiersammlung für Geld- und
Bankgeschäfte, ISBN 3-89374-013-9, VK = 49,- DM
Pocket Computer Programmiersammlung für das Personalve-
sen (Lohn und Gehalt), ISBN 3-89374-032-5, VK = 49,- DM
Steuerverwaltungsprogrammierung für Sharp Taschen-
computer, ISBN 3-89374-227-51-3, VK = 49,- DM
Lohn- und Einkommensteuer mit Sharp Taschencomputern,
ISBN 3-89374-227-48-3, VK = 49,- DM
Pocket Computer Programmierung für Immobilien
ISBN 3-89374-227-39-4, VK = 49,- DM
Pocket Computer Programmiersammlung für Juristen
ISBN 3-89374-227-89-0, VK = 49,- DM

Naturwissenschaften

Mathematikprogrammiersammlung für Sharp Computer,
Band 1, ISBN 3-89374-227-25-4, VK = 49,- DM
Mathematikprogrammiersammlung für Sharp Computer,
Band 2, ISBN 3-89374-227-49-8, VK = 49,- DM
Mathematikprogrammiersammlung für Sharp Computer,
Band 3, ISBN 3-89374-227-90-4, VK = 49,- DM
Mathematikprogrammiersammlung für Sharp Pocket Computer,
Band 4, ISBN 3-89374-024-4, VK = 49,- DM
Statistikprogrammiersammlung für Sharp Computer,
ISBN 3-89374-34-3, VK = 49,- DM
Chemieprogrammiersammlung für Sharp Taschencomputer,
Band 1, ISBN 3-89374-227-79-3, VK = 49,- DM
Chemieprogrammiersammlung für Sharp Taschencomputer,
Band 2, ISBN 3-89374-227-81-5, VK = 49,- DM
Physikprogrammiersammlung für Sharp Taschencomputer,
ISBN 3-89374-342-2, VK = 49,- DM
Astronomie-Programmiersammlung für Pocket Computer
ISBN 3-89374-015-5, VK = 49,- DM
Pocket Computer Programmierung für Meteorologen,
ISBN 3-89374-036-8, VK = 49,- DM
Programmiersammlung für Mineralogen mit Sharp Taschen-
computern, ISBN 3-89374-002-3, VK = 49,- DM

Ingenieurwissenschaften

Pocket Computer Programmierung für Ingenieure
ISBN 3-89374-013-3, VK = 49,- DM
Vermessungswesen Programmierung für Sharp
Taschencomputer, Band 1
ISBN 3-89374-227-42-4, VK = 49,- DM
Vermessungswesen Programmierung für Sharp
Taschencomputer, Band 2
ISBN 3-89374-227-88-2, VK = 49,- DM
Elektrotechnik-Programmiersammlung für Sharp Taschen-
computer, ISBN 3-89374-227-46-7, VK = 49,- DM
Elektrotechnik für Sharp Taschencomputer, Band 2
ISBN 3-89374-003-1, VK = 49,- DM
Medienbearbeitung mit Sharp Taschencomputern,
ISBN 3-89374-227-72-6, VK = 49,- DM
Medienbearbeitung mit Pocket Computern, Band 2
ISBN 3-89374-018-6, VK = 49,- DM
Bauteilgenieur und Bauteilstatik Programmiersammlung für Sharp
Pocket Computer, ISBN 3-89374-227-41-8, VK = 49,- DM
Bauteilgenieur und Bauteilstatik Programmiersammg. für Pocket
Computer, Band 2, ISBN 3-89374-022-6, VK = 49,- DM
Programmierung für Heizungs-, Klima- und Sanitär-
technik mit Sharp Taschencomputern
ISBN 3-89374-227-99-8, VK = 49,- DM

Hobby und Freizeit

Fotoelektronik mit Sharp Taschencomputern,
ISBN 3-89374-227-86-6, VK = 49,- DM
Pocket Computer Programmierung für Radiolink-
amateure, ISBN 3-89374-034-1, VK = 49,- DM
Flugnavigation mit Sharp Taschencomputern
ISBN 3-89374-227-78-5, VK = 49,- DM
Pocket Computer Programmierung für Segelflieger
ISBN 3-89374-018-3, VK = 49,- DM
Navigationsprogrammiersammlung für Sharp Computer
(für die Schifffahrt), ISBN 3-89374-227-49-1, VK = 49,- DM
Navigationsprogrammiersammlung für Pocket Computer
Band 2, ISBN 3-89374-022-8, VK = 49,- DM
Ton- und Musikprogrammiersammlung für Sharp Taschen-
computer, ISBN 3-89374-227-83-1, VK = 39,- DM
Astronomie, Horoskop und Biorhythmik Programmierung
für Pocket Computer, ISBN 3-89374-025-2, VK = 49,- DM
Schachprogrammiersammlung für Sharp Taschencomputer,
ISBN 3-89374-227-64-5, VK = 49,- DM
Lotto- und Glücksspiel-Programmiersammlung für Sharp -
Taschencomputer, ISBN 3-89374-227-62-9, VK = 49,- DM
Lehr- und Schulprogrammiersammlung für Sharp Computer
ISBN 3-89374-227-53-0, VK = 39,- DM
Spiele für Sharp Taschencomputer, Band 2
ISBN 3-89374-227-87-4, VK = 49,- DM
Spiele für Pocket Computer, Band 3,
ISBN 3-89374-028-0, VK = 49,- DM
Fremdsprachenhandbuch für Sharp Taschencomputer,
ISBN 3-89374-227-78-9, VK = 49,- DM

Personal Computer

Software Recht,
ISBN 3-89374-023-0, VK = 38,- DM
PC-Fax Telefax mit dem Personal Computer
ISBN 3-89374-045-7, VK = 49,- DM

Gesamtpreis: DM

Datum, Unterschrift:

Verrechnungsscheck liegt bei:
Bel Nachnahme wld zzgl. 5,- DM Gebühr erhoben.

Lieferschrift:

An alle Auslandskunden

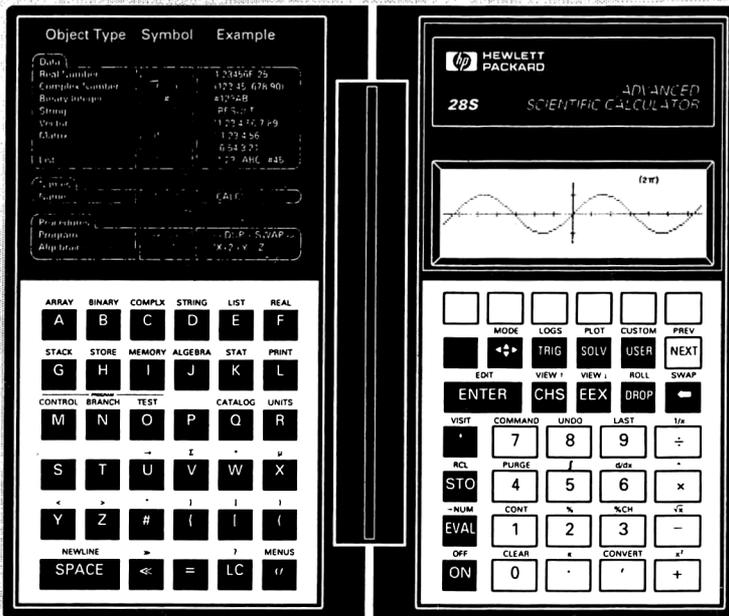
Wenn Sie best. bestellen, so fügen Sie bitte einen Vorrasscheck (Euro-
scheck) bei. Das gilt vor allem, wenn Sie in Österreich oder in der Schweiz
wohnen. Es werden keine Nachnahmeordnungen ins Ausland geschickt.
Bankverbindung: Postgirokonto 461533-103
Postgiroamt Berlin-West
Bankleitzahl: 10010010

Anwenderprogramme zum HP-28C/S Pocket Computer



HEWLETT
PACKARD

HP-28S



ISBN 3-89374-029-5

Robert Hübner

Fischel GmbH