

EDV-PROGRAMME für MATHEMATIK I auf dem HP48

1. Auflage

Herausgegeben vom Skriptenreferat der
Hochschülerschaft an der TU Graz GmbH.

1992

© 1992 Bernhard Heiden

INHALTSVERZEICHNIS

Aufschlüsselung verschiedener Vereinbarungen		3
I:	REELLE UND KOMPLEXE FUNKTIONEN	S.S.69
HORNER	Hornerschemaberechnung	5
[HOR1]		6
LAGR	Lagrang'sches Interpolationspolynom	7
LAGB		8
NEWT	Newton'sches Interpolationspolynom	9
NEWH		10
PARTB	Partialbruchzerlegung	11
[EINSETZ]		13
[KOEFGVL]		14
[VGL]		15
[VGLG]		16
[VGLV]		17
II:	DER VEKTORRAUM R^N	S.S.101
GS2E	Gerade als Schnitt zweier Ebenen	18
III:	DAS SKALARPRODUKT	S.S.121
ABPE	Abstand Punkt Ebene	19
GRAM	Gramschmidt'sches Orthog. Verfahren	20
FOURIER	Fourierkoeffizienten	21
IV:	LINEARE GLEICHUNGSSYSTEME	S.S.162
GAUSS	Gauss-Jordan Elimination	22
GAUE	Erweiterte Gauss-J. Elimination	24
V:	KURVEN ZWEITER ORDNUNG	S.S.184
KEGELS	Kegelschnitte	25
VI:	NÄHERUNGSWEISES LÖSEN VON NICHT LINEAREN GLEICHUNGEN	S.S.295
NÄHER	Nullstellennäherungsverfahren	26
[NEWV]	Newtonverfahren	28
[REGF]	Regula-Falsi	29
[KONV]	Konvergenztest	30
[LAGU]	Methode von LAGUERRE	31
VII:	MULTIFUNKTIONALE PROGRAMME	
[MAR1]	Mache eindimensionalen ARRAY	32
[MAR2]	Mache zweidimensionalen ARRAY	33
[QADE]	Quadratische Ergänzung	34
[BILD]	Erstellen eines 'Stackbildes'	35
[NUST]	Nullstellenbestimmung von Polynomen	36
Programmübersicht		37
Programmcode		38 ff.

AUFSCHLÜSSELUNG

verschiedener

VEREINBARUNGEN

dieses Skriptums

I

-
- * Die erste Zeile der nachfolgenden Seiten bezeichnet den Programmnamen wie er auf dem HP48 aufgerufen und abgespeichert wird. Namen in Klammern bedeuten, daß das Programm ein Hilfsprogramm für ein anderes darstellt. Falls es mehrere Seiten gibt so wird dies rechts durch römische Ziffern angezeigt.

 - * Die zweite Zeile gibt eine Kurzbeschreibung und evtl. direkte Abhängigkeiten zu anderen Programmen an.

 - * Die dritte Zeile enthält das Flußdiagramm.

 - * Die vierte Zeile gibt Erklärungen, zu beachten sind auf jeden Fall die **HINWEISE**, diese können im entsprechenden Fall die entscheidenden Einschränkungen aufzeigen, oder für das Verständnis unumgänglich sein.

 - * Die letzten zwei Zeilen enthalten **BEZÜGE**, das heißt, daß Zusammenhänge *notwendiger, logischer* oder *thematischer* Art mit anderen Programmen bestehen, und **BEISPIELE**, das sind Anwendungsbeispiele oder Verweise auf Beispiele.

 - * Die Abkürzung S.S. bedeutet Siehe Skriptum **MATHEMATIK 1** (Seite), Prof. Burkhard. Skriptum: 4987 ?

 - * Falls es Verzweigungen im Flußdiagramm gibt, so wird dies durch drei Punkte und einen Buchstaben angezeigt. Drei Punkte allein bezeichnen den Hauptzweig. Die Fortsetzung der Verzweigung erfolgt jeweils bei einem neuen Kästchen, wobei der jeweilige Buchstabe links über diesem steht.

 - * **Bezüge:**
notwendig: --> Pfeile vor den Programmnamen bedeuten, daß dies ein Unterprogramm für das aktuelle Programm ist
--> Pfeile nach den Programmnamen bedeuten, daß dies ein übergeordnetes Programm für das aktuelle Programm ist.
logische oder thematische Verbindung: keine Pfeile

 - * Wenn Vektoren oder andere Ausdrücke in {} eingegeben werden so sind in diesem Programm Umrechnungen mit allgemeinen Koeffizienten möglich (Algebraische Objekte).
-

-
- * Muß zu Beginn des Programms eine Stackbelegung vorliegen, so ist diese mit einer fett gedruckten **FORMAT** Anweisung angegeben.

 - * Die grau untermalten Flächen im Flußdiagramm stellen den jeweiligen Befehl dar, der einzugeben ist, bei Menüs kann jeweils einer davon gewählt werden.

 - * Werden in einem Programm Variable belegt, so dürfen diese vorher nicht immer mit einem Wert etc. belegt gewesen sein dürfen. Falls es daher Fehlermeldungen gibt und ein Programm nicht ordnungsgemäß funktioniert, so sind alle störenden Variablen (Teilmenge der verwendeten Variablen), zu entfernen.
-

WICHTIG !

Eine Diskette mit diesen Programmen für einen IBM-kompatiblen PC ist unter Einsendung einer Postkarte an bernhard heiden Sackstraße 34 8010 GRAZ, mit Angabe der Zahlungs- und Diskettenart gegen einen Unkostenbeitrag von 300 ÖS.- erhältlich.
Ktnr.:Steiermärkische Sparkasse 00201203312

HORNER

Vereinfachte Berechnung von Polynomwerten nach dem
HORNER-Schema
S.S.77

HORNER	
Geben Sie den Grad 'n' des Polynoms ein	Der Grad des Polynoms ist gleich der höchsten Potenz.
ENTER	
Geben Sie die Koeffizienten $a_0..a_N$ ein	a_N ist der Koeffizient der mit ' x^N ' multipliziert wird.
CONT	
Geben Sie den STARTWERT ein	Der Wert für den das Polynom berechnet wird
ENTER	
	Berechnung und Anzeige von $b_0..b_N$ und $c_0..c_N$
Menu:	Auswahl:
1:WEIT	1:Zurück und neuen 'STARTWERT' eingeben
2:ENDE	2:Ende, zurück ins Stackregister

ERGÄNZUNGEN:

Hinweis: Die Reihenfolge der Eingabe der Koeffizienten
ist wichtig.

Ausgabe: Es werden die Koeffizienten $a_0..a_N$,
 $b_N..b_0, c_N..c_0$ als 'markierte Objekte' in das
Stackregister ausgegeben.

Zuordnung der Koeffizienten zu HORNER SCHEMA:

$$\begin{array}{l} a_N \ a_{N-1}..a_0 \\ b_N \ b_{N-1}..b_0 \\ \hline b_N \ b_{N-1}..c_0 \end{array}$$

BEZÜGE: --> HOR1
BEISPIEL: S.S.79

[HOR1]

Berechnung von Polynomwerten nach dem HORNER-Schema mit vorgegebenen Koeffizienten $a_0..a_N$ und N (Hilfsprogramm zu HORNER)
S.S.77

HOR1

Es sind $a_0..a_N$, Grad des Polynoms und zu berechnender Wert der Reihe nach einzugeben.

ERGÄNZUNGEN:

FORMAT: $a_0..a_N$ Grad STARTWERT HOR1

Hinweis: HORNER ist das komfortablere Programm, HOR1 ist eigentlich nur ein Hilfsprogramm für HORNER.

Ausgabe: Es werden die Koeffizienten $a_0..a_N$, $b_N..b_0$, $c_N..c_0$ als 'markierte Objekte' in das Stackregister ausgegeben (wie bei HORNER)

Zuordnung der Koeffizienten zu HORNER SCHEMA:

$a_N a_{N-1}..a_0$

$b_N b_{N-1}..b_0$

$b_N b_{N-1}..c_0$

BEZÜGE: HORNER -->

BEISPIEL: S.S.79

LAGR

Berechnen eines Interpolationspolynoms mit dem Lagrang'schen Verfahren.

LAGR
Gib n-ten Grad an
CONT
Gib x(i)'s ein
CONT
Gib f(i)'s ein
CONT ...
...
LAGB ...
...
Löschen der Variablen
CONT

z.B. für ein quadrat. Interpolationspolynom: n=2
Variable werden in x gespeichert
Variable werden in fx gespeichert.
Eingabe des zu interpolierenden Wertes

ERGÄNZUNGEN:

Variable: Es existieren die Variablen fx und x im
FORMAT MAR1.

BEZÜGE: --> MAR1, --> LAGB, NEWT

BEISPIEL: S.S.85 ff.

LAGB

Berechnen eines *POLYNOMWERTES* aus einem mit *LAGR* erzeugten Interpolationspolynom.

LAGB
Gib Wert für Interpolation ein
Menu:
CONT
1:ENDE
2:NEU

Wert der in Interpolationspolynom eingesetzt wird. Berechnen des Wertes
--

Auswahl: Berechnen des Wertes 1:Ende oder zurück zu LAGR 2:Neuberechnung eines Interpolationswertes

ERGÄNZUNGEN:

Variable: Es existieren die Variablen *fx* und *x* im **FORMAT MARI** aus *LAGR*.

BEZÜGE: *LAGR* -->

BEISPIEL: S.S.85 ff.

NEWT

Newtoninterpolationsverfahren

NEWT	
Geben Sie den Grad 'n' des Polynoms an	Der Grad des Polynoms ist Anzahl Werte - 1
ENTER	
Geben Sie Koeffizient xN..x0 an	x - Werte der Reihe nach eingeben (auch allgemein möglich)
CONT	
Geben Sie Koeffizient fN..f0 an	f(x) - Werte der Reihe nach eingeben (w.o.)
CONT	
	Berechnung der Interpolation und Ausgabe mit BILD (scrollen möglich)
ATTN	
Variablen löschen?	Nach dem Löschen der Variablen kann NEWH nicht mehr verwendet werden
CONT	

ERGÄNZUNGEN:

Variable: Das Newton Interpolationspolynom wird nach dem Differenzenschema berechnet und folgendermaßen auf dem Graphikbildschirm ausgegeben,

$$\begin{pmatrix} x_1 & f_{11} & f_{12} & \dots & f_{1,N+1} \\ : & f_{21} & \vdots & \dots & \\ : & \vdots & f_{N,2} & & \\ x_{N+1} & f_{N+1,1} & & & \end{pmatrix}$$

sowie im STACK in N+1 Listen die den jeweils N+1 Zeilen zugeordnet sind. Dabei ist N der Grad des Interpolationspolynoms.

Außerdem existieren die Variablen x und f im FORMAT MAR1 und MAR2. Diese werden unter Umständen für eine weiterführende Berechnung eines Polynomwertes mit NEWH verwendet.

BEZÜGE: --> BILD, --> NEHH, --> MAR1, --> MAR2

BEISPIEL: S.S.87 ff.

NEWH

Berechnen eines POLYNOMWERTES aus einem mit **NEWT** erzeugten Interpolationspolynom.

NEWH	
GEBEN SIE AUSZUWERTENDE ZAHL EIN	Wert der in Interpolationspolynom eingesetzt wird
ENTER	
	Berechnung des Interpolationswertes und Ausgabe mit BILD (scrollen möglich)
ATPN	

ERGÄNZUNGEN:

Variable: Der Interpolationswert wird nach einem HORNERSCHEMA-artigen Verfahren berechnet und folgendermaßen im Graphikbildschirm ausgegeben,

$$\begin{pmatrix} f_{1,N+1} & \dots & f_{12} & f_{11} \\ g_{N+1} & \dots & g_2 & g_1 \\ h_{N+1} & \dots & h_2 & h_1 \end{pmatrix}$$

sowie im STACK in 3 Listen mit jeweils N+1 Elementen. Dabei ist N der Grad des Interpolationspolynoms.

Außerdem existieren die Variablen f im **FORMAT MAR2** und g bzw. h im **FORMAT MAR1**. Dabei wird f aus **NEWH** benötigt.

BEZÜGE: **NEWT** -->, --> **BILD**, --> **MAR1**, --> **MAR2**

BEISPIEL: S.S.88 ff.

Partialbruchzerlegung nach zwei Methoden.

1: Koeffizientenvergleich

2: Einsetzmethode

PARTB
Eingabe 1: Anzahl 'n' Brüche 2: Anzahl 'n+m' Koeff.
ENTER
Geben Sie Zähler ein f(X)
CONT
Geben Sie n Nenner ... oder 0 ein ... A
A
CONT
Geben Sie Vergleichsterm ein
...
CONT
Menu: 1: KOEF
...
2: EINS ... B
B
Geben Sie i - ten Wert ein i = 1..n
ENTER

Wahlweise Fall 1: Anzahl der Nenner='n' 2: (Falls die Nenner nicht gleiche Terme haben) Anzahl der Variablen='n+m'
Zähler des Partialbruchs in '' eingeben.
Fall siehe oben 1: n Nenner des Partialbruchs in '' eingeben 2: wenn wie oben Fall 2

Dieser Vergleichsterm wird in '' eingegeben. Er ist die multiplizierte rechte Seite (nicht ausmultip.) und ist $f(X, T_i) !$.

Auswahl: 1: Koeffizientenvergleich vereinfachen, Koefgl, Gleichung lösen (Bitte Warten) ... ENDE 2: Einsetzmethode
--

Der einzusetzende Wert ist einzugeben.

ERGÄNZUNGEN:

Eingabe: Die Zähler und Nenner müssen ein Funktion von X sein ($f(X)$).

Ausgabe: Der Reihe nach von oben nach unten, wobei sich die [] Klammern auf eine möglichen Weg lassen beziehen:
 [n-Brüche], Zähler, multiplizierte rechte Seite (G_0), [ausmultiplizierte rechte Seite (G)], n+m Gleichungen ($f(T_N)$), n+m Lösungen des Gleichungssystems ($T_0 \dots T_{N-1}$).

PARTIALBRUCH:

$$r(x) = \frac{\text{Zähler}}{N_1 * N_2 * \dots * N_N} = \frac{T_1}{N_1} + \frac{T_2}{N_2} + \frac{T_3 * X + T_4}{N_3} + \dots + \frac{T_{N+M-1} * X + T_{N+M}}{N_N}$$

dabei bezeichnen N_i die Nenner, 1..N die Anzahl der Brüche, 1..N+M die Anzahl der Variablen.

METHODE A: (siehe Flußdiagramm) ist anzuwenden, wenn ein oder mehrere Nenner in einem anderen enthalten sind.
 Z.B.: $N_1 = X - 2$ und $N_2 = (X - 2)^2$. Die Eingabe ist dann ident mit dem Zwischenergebnis G_0 der 'Standard-Methode'. Man erhält G_0 indem man alle Partialbrüche auf einen Nenner bringt --> $\text{Zähler}(X) = \text{Zähler}(X, T_i) = G_0$

Variablen (temporär): x im **FORMAT MARI** und m
BEZÜGE: --> VGLG, --> KOEFVGL, --> EINSETZ, --> MARI
BEISPIEL: S.S.90 ff.

[EINSETZ]

Hilfsprogramm für PARTB. Einsetzmethode

EINSETZ

Vergleichsterm, Zähler
und Anzahl der
einzusetzenden Zahlen
sind der Reihe nach
einzugeben.

ERGÄNZUNGEN:

FORMAT: 'a₀*X⁰ +..+ a_n*Xⁿ', 'b₀*X⁰ +..+ b_n*Xⁿ', n EINSETZ

Ausgabe: zwei Listen der Koeffizientenmatrixen a b
(Siehe auch KOEFVGL)

Lösung: direkt mit VGLG (Hinweise beachten!)

BEZÜGE: PARTB -->, VGLG, KOEFVGL

[K O E F V G L]

Hilfsprogramm für PARTB. Erstellt mit Hilfe von VGL das Gleichungssystem der Koeffizienten für PARTB.

KOEFVGL

Polynome a, b (Listen)
sowie Grad der Reihe
nach eingeben.

ERGÄNZUNGEN:

FORMAT: a b n

Ausgabe: 2 Listen der Koeffizientenmatrixen a und b .

$$\text{Matrizen: } \begin{bmatrix} a_{00} & \dots & a_{0n} \\ \vdots & & \vdots \\ a_{0n} & \dots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x^0 \\ \vdots \\ x^n \end{bmatrix} = \begin{bmatrix} b_0 \\ \vdots \\ b_n \end{bmatrix}$$

Ausgabeform:

('a₀₀..a_{0N}' 'a₁₀..a_{1N}' .. 'a_{NO}..a_{NN}') ('b₀'..'b_N')

BEZÜGE: PARTB -->, VGL -->

[VGL]

Hilfsprogramm für KOEFVGL. Vergleich von Polynomen die Funktion von (groß) 'X' sind. (f(X))

VGL

(Algebraischen Objekt des) Polynoms und dessen Vergleichs-Grad der Reihe nach eingeben.

ERGÄNZUNGEN:

FORMAT: 'a₁*X^N+...+a₀*X⁰', n VGL Variable vergleiche unten.

Ausgabe: Liste der Koeffizienten von f(X)

BEZÜGE: KOEFVGL -->

BEISPIEL: Eingabe: 'X³+X⁴+3*X²+2' 4 VGL
Ausgabe: Die Koeffizienten von X⁰..X^N in eine Liste geordnet also {2 0 3 1 1}

[VGLG]

Hilfsprogramm für PARTB. Lösung von Gleichungssystemen nicht allgemeiner Natur (Es dürfen keine als reelle Koeffizienten vorkommen !)

VGLG

Liste der algebraischen Objekte $a_1..a_i$ und $b_1..b_i$ der Reihe nach eingeben.

ERGÄNZUNGEN:

FORMAT: {'a₁'.. 'a_i'} {'b₁'.. 'b_i'} VGLG Variable
vergleiche unten

HINWEIS: Es müssen die Variablen in '' (der algebraischen Objekte) T₁..T_i heißen !

Ausgabe: Lösung des Gleichungssystems im Vektorformat ([]), aber nur wenn die Variablen T₁..T_i heißen und Vektor b nur reelle Zahlen erhält !

BEZÜGE: PARTB -->, --> VGLV

BEISPIEL: Eingabe:('T1*2+T2*3''T2') (0 2) VGLG
Ausgabe:Lösung der Gleichungen 2*T1+3*T2=0 und T2=2 also [-3 2]

[VGLV]

Hilfsprogramm für VGLG. Auflösen eines 'algebraischen Objekts', nach seinen Koeffizienten, speziell zur Lösung von Gleichungssystemen.

VGLV

<a>lgebraisches Objekt,
<v>variable und a<n>zahl
der möglichen Variablen
der Reihe nach eingeben.

ERGÄNZUNGEN:

FORMAT: 'a' "v" n VGLV Variable siehe oben

Form der Variablen v: v_1, v_2, \dots, v_N

BEZÜGE: VGLG -->

BEISPIEL: Eingabe: 'T1*X+T2*Y' "T" 3 VGLV
Ausgabe: 'X' 'Y' 0

GS2E

Gerade als Schnitt zweier Ebenen

GS2E
Geben Sie Punkt und Richtungsvektor in {} ein
CONT

Berechnung der zwei Ebenen aus den zwei eingegebenen Listen

ERGÄNZUNGEN:

Hinweis: Auch allgemeine Koeffizienten können verwendet werden, das heißt, daß die Liste nicht nur aus Zahlen, sondern auch aus Ausdrücken in '' (algebraischen Objekten) bestehen kann.

Ausgabe: Drei Gleichungen $f(\lambda)$, und zwei Ebenen (E1, E2)

BEISPIEL: S.S.110

ABPE

Abstand Punkt-Ebene

ABPE
Menu: ABSTAND PUNKT EBENE:
1: PARA ... A
2: KOOR ... B

A

Gib Vektor U,V,P,S ein
CONT

B

Gib aus $\langle a, x \rangle = c$ c, a sowie p an
CONT

Auswahl:
1: Parameterform
2: Koordinatendarstellung

Berechnung mittels der Parameterdarstellung Vektoren in []
--

Berechnung mittels Koordinatendarstellung. Vektoren in []

ERGÄNZUNGEN:

Ausgabe: Abstand Punkt-Ebene in leserlichem Format.

Form: A: $E: P+a*U+b*V \dots$ P,U,V sind VEKTOREN, S ist der PUNKT. (Reihenfolge beachten).
B: c ist ein Skalar, a der Normalenvektor der Ebene und p der PUNKT.

BEISPIEL: S.S.131,144.

GRAM

Gramschmidt'sches Orthogonalisierungsverfahren

GRAM	
Geben Sie den Hyperraum an	Die Dimension der zu orthonormierenden Vektoren ist anzugeben
ENTER	
Geben Sie die Dimension k an	Die Anzahl der zu orthonormierenden Vektoren ist anzugeben
ENTER	
Geben Sie die Vektoren 1..k ein	Die Vektoren sind in [] einzugeben, mit SPC als Zwischenraum (Vektorformat)
CONT	Die Vektoren werden erzeugt
Variablen löschen ?	Die erzeugten Variablen x,y,z,ya werden gelöscht
CONT	

ERGÄNZUNGEN:

Hinweis: Die Variablen x,y,z,ya werden mit dem Format **MAR1** erzeugt.

Ausgabe: Die Variablen werden im Stack in der leicht leserlichen Form y(1)..y(k) und z(1)..z(k) ausgegeben. Ebenso werden sie in der leicht verarbeitbaren Form im Vektorformat den oben erwähnten Variablen gespeichert. Es ist z.B. y(1) mit der Eingabe von y1 in den Stack zu holen (Siehe MAR1).

Bedeutung der Variablen: x(1)..x(k) sind die Ausgangsvektoren, y(1)..y(k) sind die orthonormierten Vektoren, z(1)..z(k) sind die Zwischenergebnisse, S.S. und k bezeichnet den Unterraum.

Löschen der Variablen: Entweder mit CONT als letztem Schritt, wobei dieser nach beliebiger Zeit und auch nach Ablauf anderer Programme gemacht werden kann. Es ist zu beachten, daß die Taste ATTN, außer in Programmen nicht gedrückt werden darf. Die zweite Möglichkeit Variablen zu löschen ist 'VARIABLENNAME+0' z.B.x0 einzugeben. Mit PURGE werden alle von MAR1 erzeugten Variablen gelöscht. Drittens genügt der Befehl 'VARIABLENNAME+D' z.B. xD.

BEZÜGE: --> FOURIER

BEISPIEL: S.S.127

FOURIER

Berechnung der Fourierkoeffizienten zu einem mit GRAM erzeugten Orthonormalsystem.

FOURIER	
Geben Sie den Vektor für Fourierkoeffizienten ein	Der Fourierrektor wird im Vektorformat [] eingegeben
CONT	Berechnung der k-Fourierkoeffizienten
Variablen löschen?	
CONT	Die erzeugte Variable 'Lamda' wird gelöscht

ERGÄNZUNGEN:

Hinweis: Die Variable 'Lamda' wird mit dem Format von MARI erzeugt.

Ausgabe: Die Variablen werden im Stack in der leicht leserlichen Form Lamda(1)..Lamda(k) ausgegeben. Ebenso werden sie in der leicht verarbeitbaren Form in der oben erwähnten Variablen gespeichert. Es ist zum Bsp. Lamda(1) mit der Eingabe von Lamda1 in den Stack zu holen. (Siehe MARI)

Bedeutung der Variablen: Lamda(1)..Lamda(k) sind die Fourierkoeffizienten aus dem eingegebenen Vektor und dem mit GRAM erzeugten Orthonormalsystem.

Löschen der Variablen: Entweder mit CONT als letztem Schritt, wobei dieser nach beliebiger Zeit und auch nach Ablauf anderer Programme gemacht werden kann. Es ist zu beachten, daß die Taste ATTN, außer in Programmen nicht gedrückt werden darf. Die zweite Möglichkeit Variablen zu löschen ist VARIABLENNAME+0 z.B.:x0 einzugeben. Mit PURGE werden alle von MARI erzeugten Variablen gelöscht. Drittens genügt der Befehl 'VARIABLENNAME+D' z.B. xD.

BEZÜGE: GRAM -->

BEISPIEL: S.S. Übung 66.

Dieses Programm führt das Gauss-Jordan Verfahren , auch mit allgemeinen Koeffizienten durch.

```

GAUSS
GIB DIE LISTE EIN

Menu:
1:  CONT      ... A
  ←  CONT      ... B
  →  CONT      ... C

2: BILD
3: GAUE      ... D

4: MATR

5: LISTE

6: COLL
7: EVALL
8: EXPAL
    
```

```

Eingabe der Koeffizientenmatrix in einer doppelten Liste siehe u.

Auswahl:
1: Gauss-Jordan Elimination

2: siehe Programm BILD
3: siehe Programm GAUE (Erweiterte Gausselimination (erst nach Fertigstellung von GAUSS))
4: Erstellung einer Matrize aus einer 'doppelten Liste', beachte HINWEIS
5: Erstellung einer 'doppelten Liste' aus einer Matrize.
6: COLCT f.normale Liste
7: EVAL  f.normale Liste
8: EXPAND f.normale Liste
    
```

```

A
bis ENDE
CONT
    
```

```

Schrittweise Ausführung der G.J. Elimination ...

ENDE
    
```

```

B
... ENDE
    
```

```

Wie A nur, daß alle Schritte ohne Unterbrechung durchgeführt werden.
    
```

```

C
bis ENDE
ATTN
    
```

```

Anzeige der Matrize mit Hilfe von Bild im übersichtlichen GRAPHIK-DISPLAY (bei Bedarf kann gescrollt werden)
    
```

```

D
... GAUE
    
```

```

Siehe Programm GAUE
    
```

ERGÄNZUNGEN:

Hinweis:

- * Doppelte Liste: Die Zeilen werden jeweils in eine Liste gegeben, diese werden wieder zu einer Liste zusammengefaßt. z.B.: Matrix --> doppelte Liste

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \quad \text{--> } \{(a \ b \ c) \ (d \ e \ f) \ (g \ h \ i)\}$$
- * Die Berechnung versagt wenn die jeweils erste Zeile eine Nullzeile ist, bei der Ergänzung mit GAUE ist dies die jeweils letzte Zeile.
- * Um eine doppelte Liste in eine Matrize umzuwandeln dürfen in dieser nur Zahlen sein, *negative Zahlen* dürfen nicht in '' stehen (Algebraischer Ausdruck)
- * Falls bei der *Rundung* mit der Funktion --> Q Brüche mit großen Zahlen entstehen, dann kann man vor Ablauf des Programms mit der eingebauten Funktion **FIX** (zb 2 **FIX**) die Kommastelle herabsetzen und diese großen Brüche bei der Berechnung vermeiden.

Methoden

- A: Die Einzelschrittausführung ermöglicht eine schrittweises verfolgen der Gaussjordan-elimination.
- B: Die Gesamtberechnung
- C: Die übersichtliche Darstellung mit sofortiger Darstellung 'komplizierter' Elemente (Scrollen des Bildschirms !)
- D: Gesamtlösung des linearen Gleichungssystems mit GAUE

Zweck:

- * Die Berechnung von linearen Gleichungssystemen mit Eingabe der Koeffizientenmatrix, wobei die Koeffizienten allgemeine Buchstaben sind
 - a: anschließendes Rückeinsetzen (gestaffeltes Gleichungssystem)
 - b: vollständige Lösung mit GAUE
- * Die Berechnung der INVERSEN durch rechtsseitige Eingabe der Einheitsmatrix und anschließender Verwendung von GAUE

BEZÜGE: --> BILD, --> GAUE**BEISPIEL:** S.S.162 ff ... lineare Gleichungssysteme

GAUE

Erweiterte Gausselimination, anschließend an Programm
GAUSS

GAUE	
Erweiterte Gaussversion	Das Listenergebnis aus GAUSS steht an erster Stelle im STACK
Menu:	
CST	Benutzerdefiniertes Menü, in diesm Fall dasselbe wie bei GAUSS, auch die Funktionen sind ident.

ERGÄNZUNGEN:

Erklärung: Das Programm bringt eine Matrize die unterhalb der Diagonale auf Null gebracht worden ist (keine Nullzeile !) auch oberhalb der Diagonale auf lauter Nullen. Damit ist im allgemeinen die Lösung eines linearen Gleichungssystems erreicht. Bei allgemeinen Koeffizienten ist zu beachten das $RG(A) = RG(A|B)$. Eine 'Fallunterscheidung' muß schon nach GAUSS gemacht werden, dabei kann man diverse Werte einsetzen und mit der Menüfunktion **EVALL** jeweils eine Liste auswerten. Details, Siehe S.S. 162 ff.

BEZÜGE: --> BILD,GAUSS -->
BEISPIEL: S.S.165

KEGELS

Dieses Programm klassifiziert allgemeine Kegelschnitte, berechnet einzelne Spezifikationen wie Determinanten, SPUR A usf.

KEGELS	
Gib a,b,c,d,e,f der Reihe nach ein	Die Koeffizienten aus untenstehender Gleichung sind einzugeben.
CONT	
PARABEL ... A ELLIPSE,HYPERBEL ... B ANDERE SCHNITTE ...	Ausgabe der Klassifikation je nach Typ ...
A	
Quadratische Ergänzung	Die Parabel wird im gedrehten System quadratisch ergänzt ...Bitte Warten !
CONT	
... B	
B	
Zeichnen	Zeichnen der Ellipse oder Hyperbel zentriert, oder der Parabel.
CONT	
...	
Variablen löschen?	Löschen der Variablen
CONT	

ERGÄNZUNGEN:

Variable: da,db,A,sa, lamda1, lamda2,r,s,phi, dabei bedeuten sie der Reihe nach DET(A),DET(B),SP(A),A,Eigenwert1,Eigenwert2,Halbachsel,Halbachse2,Drehwinkel.

Ausgabe:

- ALLGEMEIN:DET(A),SP(A),DET(B),"Type"
- ELLIPSE,HYPERBEL:Mittelpunkt, Eigenwerte(lamda1,lamda2), Halbachsen(r,s)
- PARABEL:Drehwinkel, Eigenwerte (lamda1,lamda2), Drehmatrix, transformierter Vektor d, Gleichung im gedrehten System, quadratisch ergänzte Gleichung, Parameter p, Scheitel(s1,s2). FORM der Parabel S.S.

FORM: $Q(x,y)=ax^2+2bxy+cy^2+2dx+2ey+f=0$

$$d = \begin{bmatrix} d & e \end{bmatrix}$$

$$B = \begin{bmatrix} a & b & d \\ b & c & e \\ d & e & f \end{bmatrix} \quad A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$$

BEZÜGE: --> ELHY,--> PARQ
BEISPIEL: S.S.187 ff.

Berechnen von Näherungswerten von nicht algebraisch lösbaren Gleichungen der Form $f(x)=0$.

NÄHER	
Gib Funktion $f(x)$ aus Gleichung $f(x)=0$ ein	
Menu:	I
1:CONT	
2:WIED	
Gib Anfangswert an	
ENTER	
Menu:	II
1:ENDE	...
2:NEWV	... A
3:REGF	... B
4:KONV	... C
5:LAGU	... D

A

NEWV	... Menu II
------	-------------

B

REGF	... Menu II
------	-------------

C

KONV	... Menu II
------	-------------

D

LAGU	... Menu II
------	-------------

...

Variablen löschen?
CONT

Es ist die Funktion ($f(x)!$) einzugeben, sie wird in f gespeichert.	
Auswahl:	I
1:Die Funktion wird neu eingegeben 2:Die Funktion wird wiederholt.(f ist vorhanden)	
Der Wert wo die Näherung begonnen wird (er ist nicht immer wichtig !)	
Auswahl:	II
1: ... ENDE 2:Newtonverfahren Normal,Doppelschritt 3:REGF Regula Falsi 4:KONV Konvergenztest (Fixpunktsatz 5:LAGU Intervallbestimmung nach Laguerre	

Programm NEWV

Programm REGF

Programm KONV

Programm LAGU

Die Variablen f,ff werden gelöscht.

ERGÄNZUNGEN:

Hinweis:

- * Die Funktion muß unbedingt Funktion von 'klein x' sein ($f(x)$) !
- * In den jeweiligen Unterprogrammen werden Näherungen berechnet, mit Nummern versehen und in den STACK ausgegeben. Mit dem Programm BILD können diese sehr praktisch und übersichtlich dargestellt werden.

Variable: Es existieren die Variablen f und ff. Diese werden für die Unterprogramme von A,B,C,D benötigt, wobei letztere mit dem Vohandensein dieser Variablen auch unabhängig von NÄHER verwendet werden können.

Unterprogrammvariable:

fx bezeichnet jeweils die Funktion des
Näherungsverfahrens **FORMAT: reelle_Zahl fx**
--> neuer Näherungswert

BEZÜGE: --> NEWV, --> REGF, --> KONV, --> LAGU, BILD

BEISPIEL: S.S.295 ff.

[NEWV]

Berechnen von Nullstellen mit verschiedenen Newton-Verfahren.

NEWV	
Menu:	I
Gib neuen Anfangswert ein	
1: ENDE	
2: NEW	
3: DNEW	
4: NEMA	... A

Auswahl:	I
Anfangswert ist bei NEW und DNEW einzugeben	
1: ... ENDE oder zurück in NÄHER	
2: Das Newton-Näherungsverfahren wird ausgeführt.	
3: Das Doppelschrittnewtonverfahren wird ausgeführt.	
4: Newton Maehly-Iteration zur Bestimmung aller reellen Nullstellen eines Polynoms	

A

N-1 Nullstellen, Schätzwert, (N)r.	
Menu:	II
1: CONT	... Menü I

* N-1 Nullstellen sind die schon ermittelten Nullstellen. Die erste Nullstelle kann nicht hiemit erstellt werden.	
* Der Schätzwert liegt z.B. in der Nähe der letzten Nullstelle.	
* N bezeichnet die N-te zu ermittelnde Nullstelle.	
Rückkehr zu Menü I	

ERGÄNZUNGEN:

Variable: Es existieren die Variablen f und ff aus NÄHER. sowie fx, f1, f12 wobei diese nicht alle gleichzeitig vorhanden sein müssen.

BEZÜGE: NÄHER -->, BILD

BEISPIEL: S.S.296-300.

[REGF]

Berechnen von Nullstellen mit Hilfe der *REGULA FALSI*.

REGF
Gib Intervall für Regula Falsi ein
Menu:
1: ENDE
2: CONT

Zwei reelle Zahlen sind einzugeben
Auswahl:
1: ... ENDE oder zurück in NÄHER
2: Nach Eingabe der reellen Zahlen erfolgt die Berechnung der Nullstelle mit der Regula Falsi.

ERGÄNZUNGEN:

Variable: Es existieren die Variablen f und ff aus NÄHER. sowie fx , $f12$.

BEZÜGE: NÄHER -->, BILD

BEISPIEL: S.S.295.

[KONV]

Feststellen des Konvergenzverhaltens zwecks Bestimmung der Nullstellen.

KONV	
Gib zu untersuchenden Punkt ein	Zwei reelle Zahlen sind einzugeben
Menu:	Auswahl: Anfangswert ist bei NEW und DNEW einzugeben
1: ENDE	1: ... ENDE oder zurück in NÄHER
2: NEWT	2: Konvergenzbedingung für Newtonverfahren
3: FIXP	3: Konvergenzbedingung mit Fixpunktsatz

ERGÄNZUNGEN:

Hinweis: Es ist zu beachten, daß die Konvergenzbedingung nur hinreichend für Konvergenz ist, das heißt, daß ein Versagen des Tests trotzdem Konvergenz bedeuten kann.

BEZÜGE: NÄHER-->, BILD

BEISPIEL: S.S. 299.

[LAGU]

Dieses Programm berechnet mit der Methode von LAGUERRE ein Intervall in dem die Nullstellen des Polynoms liegen!

Gib n,a(n-1),a(n-2) ein
Menu:
1: ENDE
2: CONT

siehe unten
Auswahl: Anfangswert ist bei NEW und DNEW einzugeben
1: ... ENDE oder zurück in NÄHER
2: Berechnung des Intervalls indem die Nullstellen liegen

ERGÄNZUNGEN:

Form des Polynoms:

$$P(X) = x^n + a_{n-1}x^{n-1} + \dots + a_0$$

NÄHER-->, BILD

BEISPIEL: S.S.300.

[MAR1]

Erstellung eines eindimensionalen ARRAY auf dem HP48 mit zusätzlichen Hilfestellungen.

MAR1

Anzahl und Name der Variablen in ""
(Zeichenkette - type 2)
der Reihe nach eingeben

ERGÄNZUNGEN:

FORMAT: ANZAHL "VNAME" MAR1

Hinweis: Einfachheitshalber wird in den Anwenderprogrammen jede mit MAR1 erzeugte Variable als Variable mit **FORMAT MAR1** bezeichnet. Das heißt, daß alle mit MAR1 erzeugten Untervariablen ebenso vorhanden sind.

Untervariable: Der Variablenname wird hier kurz mit v abgekürzt.

v: **FORMAT:** n v ergibt den Variablennamen, in dem das n-te Element gespeichert ist.
z.B.:1 v ergibt 'v1'

vN: Größe der eindimensionalen Liste

v0: Liste aller Variablen die mit MAR1 erzeugt wurden ((Liste - Typ 5) - diese ist vor allem beim Löschen sehr nützlich (v0 PURGE).

VR: RCL-Option.
FORMAT: Elementnummer vR.
Auf diese Weise wird das Objekt aus v(Elementnummer) in den aktuellen Stack geholt.

VS: STO-Option
FORMAT: Elementnummer Speicherobjekt vS.
Auf diese Weise wird das Objekt in v(Elementnummer) gespeichert.

VD: DEL - oder Löschoption
FORMAT: vD .. löscht alle Variablen.

v1..vN: Variablennamen für den eindimensionalen ARRAY

BEZÜGE: GRAM -->,FOURIER -->, etc.

[MAR2]

Erstellung eines zweidimensionalen ARRAY auf dem HP48 mit zusätzlichen Hilfestellungen.

MAR2

Spalte, Reihe und Name der Variablen in ""
(Zeichenkette - type 2)
der Reihe nach eingeben

ERGÄNZUNGEN:

FORMAT: ZEILEN(anzahl) SPALTEN(anzahl) "VNAME" MAR2

Hinweis: Einfachheitshalber wird in den Anwenderprogrammen jede mit MAR2 erzeugte Variable als Variable mit **FORMAT MAR2** bezeichnet. Das heißt, daß alle mit MAR2 erzeugten Untervariablen ebenso vorhanden sind.

Untervariable: Der Variablenname wird hier kurz mit v abgekürzt.

v: **FORMAT:** m n v ergibt den Variablennamen in dem das (m,n)-te Element gespeichert ist.
z.B.: 1 2 v ergibt 'v12'

VNM: Größe der zweidimensionalen Liste
(Liste - type 5)

v0: Liste aller Variablen die mit MAR2 erzeugt wurden (Liste - type 5) - diese ist vor allem beim Löschen sehr nützlich (**FORMAT: v0 PURGE**).

VR: RCL-Option.
FORMAT: Zeilennummer Spaltennummer VR.
Auf diese Weise wird das Objekt aus v(Elementnummer) in den aktuellen Stack geholt.

VS: STO-Option
FORMAT: Zeilennr. Spaltennr. Speicherobjekt VS.
Auf diese Weise wird das Objekt in v(Spaltennummer,Reihennummer) gespeichert.

VD: DEL - oder Löschoption
FORMAT: VD .. löscht alle Variablen.

v11..vNM: Variablenamen für den zweidimensionalen ARRAY

BEZÜGE: NEWT -->, NEWH -->

[QADE]

Quadratische Ergänzung auf dem HP48

QADE

Es ist a,b,c,x der Reihe nach einzugeben.

ERGÄNZUNGEN:

FORMAT: a b c 'x' QADE a,b,c,x siehe unten

Zuweisung der Variablen:

$$ax^2+bx+c$$

.. ist Ausgangsgleichung

$$\left[x + \frac{b}{2a} \right]^2 = -\frac{c}{a} + \frac{b^2}{4a^2}$$

.. ist Ergebnisgleichung

BEZÜGE: KEGELS -->

BEISPIEL: Eingabe :1 -4 4 'x' QADE

Ausgabe : '(x-2)^2'

BILD

Erstellen einer Darstellung von n Stackdaten auf dem Graphikbildschirm.

BILD
ATTN

Eingabe der Anzahl darzustellender Stackelemente
--

Anzeige der Stackdaten

Abbruch

ERGÄNZUNGEN:

FORMAT: n BILD

Anzeige: Ein Bild der n Stackdaten in kleinen Zeichen, wobei diese bei Überschreitung der Bildgröße mit den Pfeiltasten gescrollt werden können.

Zweck: Die übersichtliche Darstellung von Stackdaten, zum Beispiel bei KEGELS,FOURIER,GAUSS usf.

BEZÜGE: KEGELS -->,FOURIER -->,GAUSS -->,GAUSSE --> etc.

[NUST]

Dieses Programm berechnet sukzessive die *n* Nullstellen eines reellen Polynoms *n*-ten Grades.

NUST

Eingabe des Polynoms in ' ' als Funktion von X (f(X)) und dem Grad n.

... Bitte Warten

ERGÄNZUNGEN:

FORMAT: 'Polynom n-ten Grades' n NUST

HINWEIS: - 'Groß X' darf keine verwendete Variable sein.
- Die Nullstellenberechnung kann z.B. beim Berechnen von Eigenwerten aus der Eigenwertgleichung nützlich sein.

AUSGABE: Die n - Nullstellen werden der Reihe nach ausgegeben. Es wird mit der Nullstelle am weitesten links auf der x-Achse begonnen.

Form des Polynoms:

$$P(X) = X^n + a_{n-1}X^{n-1} + \dots + a_0$$

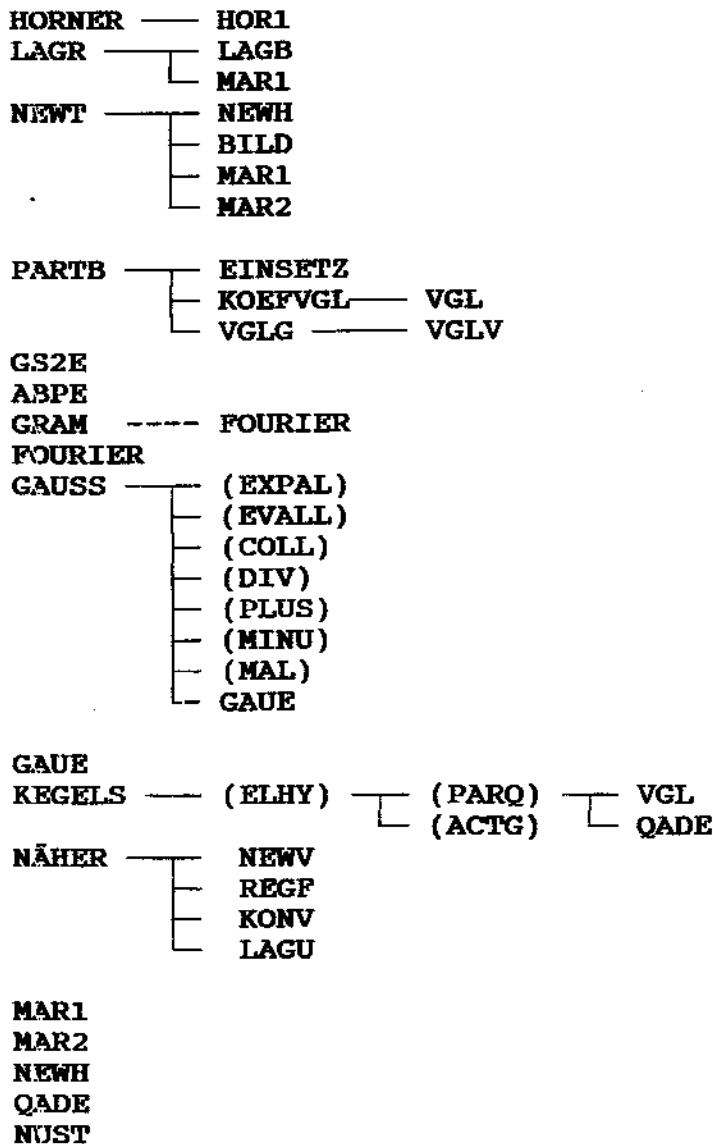
BEZÜGE: NEWV, LAGU

BEISPIEL: Eingabe: 'X^4+2*X^3-13*X^2-14*X+24' 4 NUST
... Bitte Warten
Ausgabe: -4 -2 .999999999999 3

Programmübersicht

Die folgende Liste zeigt alle Programme in der selben Reihenfolge wie im Inhaltsverzeichnis, sowie die Abhängigkeiten von anderen Programmen. Dies ist vor allem für das einzelne Verwenden der 'Unter'-Programme von Nutzen. Programme die in diesem Skriptum nicht näher beschrieben werden stehen in (). Strichlinien bedeuten, daß die Programme nicht unbedingt zusammen verwendet werden müssen. Die Programme EXCO und MULTI sind dem HP Handbuch zu entnehmen, und werden vorausgesetzt.

S. II 5616
5618
1. Ausgabe
April 1950



ABPE	ACTG	BILD	COLL ()
<pre> < "ABSTAND PUNKT-EBENE" { { "PARA" < "GIB VETOR U,V,P,S EIN" PROMPT 4 DUPN 4 DUPN - 3 ROLLD CROSS DOT ABS 5 ROLLD DROP DROP CROSS ABS SQ 7 RND "/√" SWAP + + "ABSTAND=" SWAP + CONT > } { "KOOR" < "GIB AUS <a,x>=c c,a SOWIE p AN" PROMPT 3 DUPN DOT SWAP - ABS 3 PICK ABS SQ 7 RND "/√" SWAP + + "ABSTAND=" SWAP + CONT > } } MENU PROMPT 0 MENU > LAGU < { { "ENDE" < "E" CONT > } { "CONT" < CONT > } } MENU "Gib n,a(n-1),a(n-2) ein" PROMPT IF DUP "E" SAME THEN DROP ELSE → n a1 a2 < n 1 - a1 * SQ 2 n * n 1 - * a2 * - DUP → d < "D" →TAG a1 NEG d √ - n INV * a1 NEG d √ + n INV * { 1 2 } →ARRY "INTERVALL" →TAG > > END > </pre>	<pre> < → 0 < IF '0==0' THEN 'π/2' ELSE 0 INV ATAN END > > </pre>	<pre> < → n < 64 STWS HEX n DUPN 130 1 n FOR i n 1 + ROLL 1 →GROB DUP SIZE DROP B→R 3 ROLL MAX NEXT R→B n 6 * R→B PDIM n 1 FOR i i ROLL PICT SWAP # 0h n i - 6 * R→B 2 →LIST SWAP REPL -1 STEP { } PVIEW > > </pre>	<pre> < OBJ+ → d < 1 d FOR i d ROLL < COLCT > MULTI (→Q) NEXT d →LIST > > </pre>

DIV	EINSETZ	ELHY	
<pre> < INV MAL > COLL <i>Siehe Vorseite</i> < OBJ → → d < 1 d FOR i d ROLL < COLCT > MULTI →Q NEXT d →LIST > > </pre>	<pre> < → a z n < 1 n FOR i "Geben Sie " i + " - ten Wert ein" + "" INPUT OBJ → → u < z X u 2 →LIST +MATCH DROP EXCO a X u 2 →LIST +MATCH DROP EXCO = > NEXT 1 n FOR i n PICK EQ → DROP NEXT n →LIST 1 n FOR i n 1 + PICK EQ → SWAP DROP NEXT n →LIST SWAP > > </pre>	<pre> < → a b c d e f < (0,0) → m < DUP DUP CASE "ELLIPSE" SAME SWAP "HYPERBEL" SAME OR THEN A INV NEG d e { 2 } →ARRY * DUP OBJ → DROP R → C 'm' STO "Mittelpunkt" →TAG IF b 0 ≠ THEN a c - b 2 * / ACTG 2 / DUP 'phi' STO "Drehwinkel" →TAG END a c + DUP a c - SQ b SQ 4 * + √ DUP 3 ROLL + 2 / 'λ1' STO - 2 / 'λ2' STO IF b 0 > λ1 λ2 > XOR THEN λ1 λ2 'λ1' STO 'λ2' STO END λ1 "λ1" →TAG λ2 "λ2" →TAG db NEG λ1 da * / 'r' STO db NEG λ2 da * / 's' STO r "r" →TAG s "s" →TAG END DUP "PARABEL" SAME THEN IF b 0 ≠ THEN a c - b 2 * / ACTG 2 / DUP 'phi' STO "Drehwinkel" →TAG END IF b 0 > THEN a c + 'λ1' STO 0 'λ2' STO ELSE a c + 'λ2' STO 0 'λ1' STO END λ1 "λ1" →TAG λ2 "λ2" →TAG "1/√" a SQ b </pre>	<pre> SQ + + "*" + DUP IF b 0 > THEN a b NEG b a ELSE b NEG a NEG a b NEG END { 2 2 } →ARRY DUP 3 ROLLD + "Drehmatrix" →TAG 3 ROLLD d e { 1 2 } →ARRY SWAP * DUP 3 ROLLD + "d" →TAG SWAP OBJ → DROP IF λ1 0 SAME THEN λ2 ' y^2' * ELSE λ1 ' x^2' * END 3 PICK 2 * 'x' * 'a' * + 2 PICK 2 * 'y' * 'â' * + SWAP DROP SWAP DROP DUP → z < 'a' '1/ √A' 'A' a SQ b SQ + 2 →LIST +MATCH DROP 2 →LIST +MATCH DROP "Transf.Gleich." →TAG HALT z > IF λ1 0 == THEN 'y' ELSE 'x' END a SQ b SQ + PARQ END END HALT 'a*x ^2+2*b*x*y+c*y^2+2* d*x+2*e*y+f=0' EVAL STEQ CONIC 'x' INDEP 'y' DEPND 1 1 SCALE m CENTR ERASE DRAX LABEL DRAW > > > </pre>

EVALL ()	EXPAL	FOURIER	GAUE
<pre> < OBJ→ → d < 1 d FOR i d ROLL EVAL (→Q) NEXT d →LIST > > </pre>	<pre> < OBJ→ → d < 1 d FOR i d ROLL < EXPAN > MULTI →Q NEXT d →LIST > > </pre>	<pre> < STD yN → k < "Geben Sie Vektor für Fourierkoeffizienten ein" PROMPT → v < v λ0 PURGE k "λ" MARI 1 k FOR i i y RCL v DOT DUP i λ STO i yaR √ * 6 RND "/√" i yaR + + "λ" i + -TAG NEXT "Variablen löschen?" PROMPT λD > > > </pre>	<pre> < "Erweiterte Gaussversion" 1 MENU PROMPT DUP OBJ→ → d < d 2 FOR j j 2 FOR i d j - 1 + PICK d i - 3 + ROLL DUP j GET 3 ROLL SWAP MAL MINU d i - 2 + ROLLD IF 1 FS? THEN IF 2 FC? THEN d ELSE HALT END END -1 STEP -1 STEP d →LIST > > </pre>

GAUSS		GRAM	
<pre> < { { "CONT" { < 1 SF 2 SF CONT > < 1 CF CONT > < 1 SF 2 CF CONT > } } { "BILD" < BILD > } { "GAUE" < GAUE > } { "MATR" < OBJ→ 0 → i j < OBJ→ DUP 'j' STO →LIST 1 i FOR k i j 1 - k 1 - * + ROLL OBJ→ DROP NEXT i j 2 →LIST →ARRY > > } { "LISTE" < OBJ→ OBJ→ DROP → i j < i 1 FOR k j →LIST j k 1 - * 1 + ROLLD -1 STEP 1 i FOR k i k - 1 + ROLLD NEXT i →LIST > > } { "COLL" < COLL > } { "EVALL" < EVALL > } { "EXPAL" < EXPAL > } } MENU "GIB DIE LISTE EIN" PROMPT DUP OBJ→ → d < d 2 FOR j j 2 FOR i j ROLL DUP d j - 1 + GET DUP IF 0 SAME THEN DROP ELSE DIV END j ROLLD i 1 - ROLL DUP d j </pre>	<pre> - 1 + GET j 1 + PICK SWAP MAL MINU i 1 - ROLLD IF 1 FS? THEN IF 2 FC? THEN d ELSE HALT END END -1 STEP -1 STEP DUP d GET DUP IF 0 SAME THEN DROP ELSE DIV END IF 1 FS? THEN IF 2 FC? THEN d BILD ELSE HALT END END d →LIST > > </pre>	<pre> < STD "Geben Sie den Hyperraum an" "" INPUT OBJ→ → h < "Geben Sie die Dimension von U an" "" INPUT OBJ→ → k < "Geben Sie die Vektoren 1..k ein" PROMPT x0 y0 z0 ya0 1 4 FOR i DUP IF TYPE 5 == THEN PURGE ELSE DROP END NEXT k DUPN k "x" MAR1 k "y" MAR1 k "z" MAR1 k "ya" MAR1 k 1 FOR i i x STO -1 STEP 1 x RCL DUP DUP DUP ABS / 1 y STO ABS SQ 7 RND DUP 1 ya STO "1/√" SWAP + "+" + SWAP + "y(1)=" SWAP + 2 k FOR i l h FOR j 0 NEXT h →ARRY i z STO i i l - 0 → zw < FOR j j y RCL i x RCL DOT j y RCL * i z RCL + i z STO j ya RCL INV zw + 'zw' STO NEXT zw →Q DUP IF TYPE 9 == THEN OBJ→ DROP DROP SWAP DROP END DUP i z RCL * 7 RND SWAP ")=1/√" "z(" i + </pre>	<pre> SWAP + SWAP + "*" + SWAP + i x RCL i z RCL - DUP ABS / i y STO i x RCL i z RCL - zw →Q DUP IF TYPE 9 == THEN OBJ→ DROP DROP SWAP DROP END * 7 RND DUP ABS SQ 6 RND DUP i ya STO ")=1/√" "y(" i + SWAP + SWAP + "*" + SWAP + > NEXT "Variablen löschen ?" PROMPT xD yD zD yaD > > </pre>

GS2E		HOR1	HORNER
<pre> « "Geben Sie Punkt und Richtungsv. in {} ein" PROMPT → p v « p v p OBJ→ → n « n DROPN 1 n FOR i "X" i + OBJ→ NEXT n →LIST p v 'λ' MAL PLUS DUP OBJ→ → n « n DROPN OBJ→ DROP n 1 + ROLL OBJ→ DROP 1 n FOR i n 2 + i - ROLL = n i - 2 * 1 + ROLLD NEXT » n DUPN 'λ' ISOL EQ→ → x « DROP EQ→ 'λ' 'x' EVAL 2 →LIST DUP 4 ROLLD +MATCH DROP = 3 ROLLD SWAP EQ→ 3 ROLL +MATCH DROP = EXCO →Q E1 →TAG SWAP EXCO →Q E2 →TAG » » » » » </pre>		<pre> « → n x « n DUPN 0 "b" n 1 - + →TAG n 1 + ROLLD n 2 FOR i x * DUP "b" i 2 - + →TAG i 1 + ROLLD + -1 STEP DROP 1 n FOR i n 2 * PICK NEXT DUP DTAG "c" n 1 - + →TAG n 1 + ROLLD n 2 FOR i x * + "c" i 2 - + →TAG DUP i ROLLD -1 STEP DROP » » </pre>	<pre> « STD "Geben Sie den Grad 'n' des Polynoms ein" "" INPUT OBJ→ 1 + → n « "Gib Koeffizienten a0..a_n ein" PROMPT 1 n FOR i DUP IF TYPE 12 == THEN DTAG END "a" n i - + →TAG n ROLLD NEXT DO 0 MENU "Geben Sie den STARTWERT ein " "" INPUT OBJ→ → x « n x HOR1 n 2 * DUPN CLLCD 1 n FOR i n 1 + PICK 2 →LIST i DISP NEXT n DROPN 3 FREEZE » { { WEIT « n 2 * DROPN CONT » } { ENDE « "E" CONT » } } TMENU HALT UNTIL DUP "E" SAME END DROP 0 MENU » » </pre>

KEGELS		KOEFLVGL	KONV
<pre> < "Gib a b c d e f der Reihe nach ein" PROMPT → a b c d e f < a b b c { 2 2 } →ARRY DUP 'A' STO DET DUP 'da' STO "DET(A)" →TAG a c + DUP 'sa' STO " SP(A)" →TAG a b d b c e d e f { 3 3 } →ARRY DET DUP 'db' STO "DET(B)" →TAG IF 'da≠0 AND db ≠0' THEN IF 'da<0' THEN "HYPERBEL" ELSE IF 'db*sa<0' , THEN "ELLIPSE" END IF 'db*sa>0' , THEN "LEERE MENGE" END END IF 'db≠0 AND da ==0' THEN "PARABEL" END IF 'db==0 AND da≠0' THEN "(DOPPEL-)PUNKT, PAAR SCHNEIDENDER GERADEN ODER L.M" END IF 'da==0 AND db==0' THEN "PARALLELE GERADEN ODER (DOPPEL-)GERADE" </pre>	<pre> END a b c d e f ELHY HALT 'da' 'db' 'A' 'sa' 'λ1' 'λ2' 'r' 's' 'phi' 9 →LIST PURGE > > </pre>	<pre> < → a b n < a n VGL b n VGL > > --- </pre>	<pre> < DO { { "ENDE" < "ENDE" CONT > } { "NEWT" < → y < ff 'x' Δ 'fff' STO 'fx' PURGE 'x' PURGE 'fx(x)=ABS(f*fff/ fff^2)' EVAL DEFINE y fx STD "fx"(" y + ") + →TAG > CONT > } { "FIXP" < 'fx' PURGE DUP "fx"(" SWAP + ")" + 'fx(x)=ABS(ff)' EVAL DEFINE SWAP fx SWAP →TAG CONT > } } MENU "Gib zu untersuch enden Punkt ein" PROMPT DUP UNTIL "ENDE" SAME END DROP > </pre>

LAGB	LAGR	LAGU	MAL
<pre> « RCLMENU → m « DO { { "ENDE" « "E" CONT » } { "NEU" « CONT » } } MENU "Gib Wert für Interpolation ein" PROMPT DUP IF "E" ≠ THEN → xn « 0 → zw2 « 1 xN FOR k k fxR 1 → zw1 « 1 xN FOR i IF i k ≠ THEN xn i xR - k xR i xR - / zw1 * 'zw1' STO END NEXT zw1 * zw2 + 'zw2' STO » NEXT zw2 » » END UNTIL DUP "E" SAME END DROP m MENU » » </pre>	<pre> « "Gib n-ten Grad an" PROMPT 1 + → n « "Gib x(i)'s ein" PROMPT "Gib fx(i)'s ein" PROMPT n "x" MAR1 n "fx" MAR1 n 1 FOR i i SWAP fxS -1 STEP n 1 FOR i i SWAP xS -1 STEP » LAGB "Löschen der Variablen" PROMPT xD fxD » » </pre>	<pre> « { { "ENDE" « "E" CONT » } { "CONT" « CONT » } } MENU "Gib n,a(n-1),a(n-2) ein" PROMPT IF DUP "E" SAME THEN DROP ELSE → n a1 a2 « n 1 - a1 * SQ 2 n * n 1 - * a2 * - DUP → d « "D" →TAG a1 NEG d √ - n INV * 1 a1 NEG d √ + n INV * { 1 2 } →ARRY "INTERVALL" →TAG » » END » » </pre>	<pre> « → l m « l OBJ → → d « l d FOR i m * d ROLL NEXT d →LIST COLL » » » » </pre>

MAR1	MAR2	MINU	NÄHER
<pre> < → n var < var 0 + OBJ → DUP IF TYPE 5 == THEN PURGE ELSE DROP END var OBJ → var "N" + OBJ → var 0 + OBJ → var "R" + OBJ → var "S" + OBJ → var "D" + OBJ → 1 n FOR i var i + OBJ → NEXT n 6 + →LIST var 0 + OBJ → STO n var "N" + OBJ → STO "< → n < " var 0 + + " n 6 + GET >>" + OBJ → var OBJ → STO "< → n < n '" var + "' EVAL RCL >>" + OBJ → var "R" + OBJ → STO "< → n v < v n '" var + "' EVAL STO >>" + OBJ → var "S" + OBJ → STO "<" var 0 + + " PURGE >" + OBJ → var "D" + OBJ → STO > > </pre>	<pre> < → m n var < var 0 + OBJ → DUP IF TYPE 5 == THEN PURGE ELSE DROP END var OBJ → var "MN" + OBJ → var 0 + OBJ → var "R" + OBJ → var "S" + OBJ → var "D" + OBJ → 1 m FOR i 1 n FOR j var i →STR j + + OBJ → NEXT NEXT m n * 6 + →LIST var 0 + OBJ → STO m n 2 →LIST var "MN" + OBJ → STO "< → m n < " var 0 + + " (DUP 2 GET RCL OBJ → DROP DROP m 1 - * n + 6 + GET >>" + OBJ → var OBJ → STO "< → m n < m n '" var + "' EVAL RCL >>" + OBJ → var "R" + OBJ → STO "< → m n v < v m n '" var + "' EVAL STO >>" + OBJ → var "S" + OBJ → STO "<" var 0 + + " PURGE >" + OBJ → var "D" + OBJ → STO > > </pre>	<pre> < → 11 12 < 11 OBJ → → d < 12 OBJ → DROP 1 d FOR i d 2 * ROLL d 1 + PICK - NEXT d →LIST COLL d 1 + ROLL d DROPN > > > </pre>	<pre> < { { "CONT" < CONT > } { "WIED" < f CONT > } } MENU "Gib Funktion f(x) au Gleichung f(x)=0 ein" PROMPT EQ → DROP 'f' STO "Gib Anfangswert an" "' INPUT OBJ → f 'x' δ 'ff' STO 'x' PURGE DO { { "ENDE" < "ENDE" CONT > } { "NEWV" < NEWV CONT > } { "REGF" < REGF CONT > } { "KONV" < KONV CONT > } { "LAGU" < LAGU CONT > } } MENU "Gib Auswahl an" PROMPT UNTIL DUP "ENDE" SAME END DROP 2 MENU "Variablen löschen" PROMPT { f ff fff fx f12 f1 } PURGE > </pre>

NEWH	NEWT	NEWV	
<pre> < "GEBEN SIE AUSZUWERTENDE ZAHL EIN" "" INPUT OBJ → xN → y n < n "g" MAR1 n "h" MAR1 l n fR n h STO 0 n g STO n l - 1 FOR k k l + hR y k xR - * DUP k g STO l k fR + k h STO -1 STEP n l FOR i l i fR -1 STEP n →LIST n 1 FOR i i gR -1 STEP n →LIST n 1 FOR i i hR -1 STEP n →LIST 3 BILD "Variablen löschen?" PROMPT gD hD » » </pre>	<pre> < 20.01 MENU STD "Geben Sie den Grad 'n' des Polynoms an" "" INPUT OBJ → l + → n < n "x" MAR1 n n "f" MAR2 "Geben Sie Koeffizient xN..x0 ein" PROMPT l n FOR i i x STO NEXT "Geben Sie Koeffizient fN..f0 ein" PROMPT l n FOR i i l f STO NEXT 2 n FOR j l n FOR i IF i j + n 2 + < THEN i l + j l - fR i j l - fR - i j l - + xR i xR - / i j f STO END NEXT NEXT CLLCD l n FOR i i xR l n FOR j IF i j + n 2 + < THEN i j fR END NEXT n 2 + i - →LIST NEXT n BILD 0 MENU "Variablen löschen?" PROMPT fD xD » » </pre>	<pre> < STD DO { { "ENDE" < "E" CONT } { "NEW" < 'fx' PURGE ' fx(x)=x-f/ff' EVAL DEFINE "N" CONT } { "DNEW" < 'fx' PURGE ' fx(x)=x-2*f/ff' EVAL DEFINE DUP 'f1' PURGE 'f1(x)=f ' EVAL DEFINE f1 SIGN → s < 0 → z < DO DUP fx UNTIL DUP f1 SIGN s ≠ 'z' INCR 10 < OR END » » CONT } { "NEMA" < { { "CONT" < CONT » } } MENU "N-1 Nullstellen, Schätzwert,(N)r." PROMPT l - → xn k < 'fx' PURGE l k FOR i x SWAP - f SWAP / k ROLLD NEXT IF k l ≠ THEN l k l - FOR i + NEXT END ff SWAP - f SWAP / x SWAP - 'f1' STO 'fx(x)=f1' EVAL DEFINE xn "N" » CONT » } } MENU IF DUP "N" SAME THEN DROP DUP "x(0)" →TAG SWAP fx 0 + % </pre>	<pre> < WHILE 2 DUPN ≠ 'z' INCR 10 < AND REPEAT DUP "x(" z + ")" + →TAG SWAP fx END DROP » END "Gib neuen Anfangswert ein" PROMPT DUP UNTIL "E" SAME END DROP » </pre>

NUST	PARQ	PARTB	
<pre> < + a n < a 'f' STO a 'X' ð 'ff' STO a n VGL DUP n 1 + GET DIV → b < b n GET EVAL b n 1 - GET EVAL > → a1 a2 < n 1 - a1 * SQ 2 n * n 1 - * a2 * - √ a1 + NEG n / > 'FX' PURGE a 'X' ð → b < 'FX(X)=X-a/b' EVAL DEFINE WHILE DUP FX DUP FX ≠ REPEAT FX END FX 1 n 1 - FOR i DUP .1 + i → xn k < k DUPN 'FX' PURGE 1 k FOR i X SWAP - f SWAP / k ROLLD NEXT IF k 1 ≠ THEN i k 1 - FOR i + NEXT END ff SWAP - f SWAP / X SWAP - 'FX(X)' SWAP = EVAL DEFINE xn WHILE DUP FX DUP FX ≠ REPEAT FX END FX > NEXT > > > </pre>	<pre> < + au xy aa < au IF λ1 0 == THEN λ2 ELSE λ1 END / EXCO xy 'X' 2 →LIST 1 2 VGL OBJ→ DROP → c b a < IF xy 'x' SAME THEN 'y' ELSE 'x' END DUP c SWAP 1 2 →LIST +MATCH DROP EXCO '1 /√A' 'A' aa 2 →LIST +MATCH DROP → yx cc ab < 1 b c xy QADE EQ→ 2 PICK xy 0 2 →LIST 1 NEG â ab 2 →LIST +MATCH DROP →Q { '&a^2' &a } +MATCH DROP → s1 < cc IF cc { â 1 } 1 0 < THEN NEG END / EXCO DUP yx 0 2 →LIST 1 NEG â ab 2 →LIST +MATCH DROP →Q → s2 < cc IF cc { â 1 } 1 0 < THEN END DUP NEG 4 ROLLD * = â ab 2 →LIST +MATCH DROP →Q "Q.E." →TAG SWAP 2 / EXCO â ab 2 →LIST +MATCH DROP →Q "p" →TAG s1 "s1" →TAG s2 "s2" →TAG > > > > </pre>	<pre> < 0 'm' STO "Eingabe 1:Anzahl 'n' Brüche 2:Anzahl 'n+m' Koeff." "" INPUT OBJ→ → n < "Geben Sie Zähler ein f(X)" PROMPT → z < "Geben Sie n Nenner oder 0 ein" PROMPT IF DUP 0 SAME NOT THEN "Erweitern" 1 DISP n DUPN n "x" MARI n 1 FOR i i x STO -1 STEP z 0 1 n FOR i "T" i m + + OBJ→ IF i xR { 'X^2' 'X^2' } +MATCH SWAP DROP THEN 'm' INCR DROP 'X' * "T" i m + + OBJ→ + END 1 n FOR j IF i j < THEN j END NEXT + NEXT DUP "GO" →TAG SWAP x0 PURGE ELSE DROP z "Geben Sie Vergleichsterm ein" PROMPT DUP "GO" →TAG SWAP END { } "KOEFF" < </pre>	<pre> "vereinfachen" 1 DISP EXCO DUP "G" →TAG SWAP z n m + 1 - "Koeffvgl" 1 DISP KOEFVGL 2 DUPN n m + → n < OBJ→ DROP n 1 + ROLL OBJ→ DROP 1 n FOR i n 2 + i - ROLL = n i - 2 * 1 + ROLLD NEXT n 2 + ROLL n 2 + ROLL > CONT > } { "EINS" < z n m + EINSETZ CONT > } } TMENU "MENÜ" PROMPT 0 MENU "Gleichung lösen" 1 DISP VGLG OBJ→ DROP 1 n m + FOR i n m + ROLL 7 RND →Q "T" i + →TAG NEXT > > > </pre>

PLUS	QADE	REGF	VGL
<pre> « -1 MAL MINU » </pre>	<pre> « → a b c x « x b a 2 * / EXCO + 2 ^ c a / NEG b 2 ^ a 2 ^ 4 * / + = » » </pre>	<pre> « DO { { "ENDE" « "ENDE" CONT » } { "CONT" « { fx f12 } PURGE 'f12(x)=f' EVAL DEFINE 'fx(x1, x2)=x1+f12(x1)*(x1- x2)/(f12(x2)-f12(x1))' EVAL DEFINE 0 → xa xb z « xa "x1" →TAG xb WHILE 2 DUPN ≠ 'z' INCR 10 ≤ AND REPEAT DUP DUP IF f12 SIGN -1 == THEN 'xa' ELSE 'xb' STO END "x" z 1 + + →TAG xa xb fx END » DROP CONT » } } MENU "Gib Intervall für Regula Falsi ein" PROMPT UNTIL DUP "ENDE" SAME END DROP » </pre>	<pre> « → a n « 0 n FOR i CASE i 0 == THEN a { X 0 } →MATCH DROP END EXCO i 1 == THEN a 2 PICK - EXCO IF n 1 ≠ THEN 2 n FOR j ' X^j' EVAL 0 2 →LIST →MATCH DROP EXCO NEXT END { X 1 } →MATCH DROP EXCO END a 2 i 1 + FOR j j PICK 'X' i j - 1 + ^ * - NEXT EXCO IF i 1 + n ≤ THEN i 1 + n FOR j 'X^ j' EVAL 0 2 →LIST →MATCH DROP NEXT END 'X^i' EVAL 1 2 →LIST →MATCH DROP EXCO END NEXT n 1 + →LIST » » </pre>

VGLG	VGLV		
<pre> < + a b < b OBJ→ + n < n →ARRY l n FOR i a i GET "t" n VGLV NEXT { n n } →ARRY / > > > </pre>	<pre> < + a v n < l n FOR i a l n FOR j v j + OBJ→ IF i j == THEN 1 2 →LIST →MATCH DROP ELSE 0 2 →LIST →MATCH DROP END NEXT EXCO NEXT > > </pre>		