

UNIVERSITÄT OSNABRÜCK

Rechenzentrum

GRAPHISCHER OUTPUT – Eine Einführung –

Thomas Lassahn

Schriften des Rechenzentrums der Universität Osnabrück

Herausgeber: Dipl. - Math. Klaus Brauer -Nr. 1/79- März 1979

Einleitung

Einleitung

Am Rechenzentrum der Universitaet Osnabrueck steht seit Installation des TR440 (1977) ein PLOTTER zur Ausgabe GRAFISCHER INFORMATION zur Verfuegung. Somit ist dem Benutzer zunaechst eine Grundausstattung zur Verfuegung gestellt, um am TR440 in Osnabrueck (bzw. ueber die angeschlossenen Datenstationen) Grafik zu betreiben.

Um dem Benutzer einen Einstieg in die Moeglichkeiten dieses Geraetes und der damit in Verbindung stehenden Software zu geben, wurde diese Einfuehrung geschrieben.

Diese Einfuehrung soll nicht - und kann auch gar nicht - alle grafischen Moeglichkeiten mit Computern ueberhaupt erschoenfen, das wuerde den Rahmen einer EINFUEHRUNG sprengen. Dazu dient auch eher ein ausfuehrliches HANDBUCH. Ein solches HANDBUCH hatte zunaechst die Firma CGK (COMPUTER GESELLSCHAFT KONSTANZ) erarbeitet. Dieses Manual entspricht jedoch nicht mehr dem heutigen Stand der Software und ist somit hinfaelig geworden. Es ist jedoch eine neue DOKUMENTATION (ein HANDBUCH) in Arbeit, die ueber ERSTELLE verfuegbar gemacht werden wird. Diese DOKUMENTATION wird nicht nur alle augenblicklichen HARDWARE- und SOFTWARE - Moeglichkeiten enthalten, sondern darueberhinaus stets neuen Situationen angepasst werden. Diese DOKUMENTATION ist wie gesagt in Arbeit, die Kapitel I (Aufbau der geraeteneutralen graphischen Information),

Kapitel IV (Basissoftware), VI (FUPLOT) so wie die "Vorbemerkungen zu Kapitel VI" sind bereits zu grossen Teilen fertig und stehen dem Benutzer zum Teil schon zur Verfuegung.

Die nunmehr hier vorliegende EINFUEHRUNG soll in keiner Weise diesem HANDBUCH vorgreifen oder es gar ersetzen. Vielmehr sollen sich beide Schriften ergaenzen: das Handbuch, als Nachschlagewerk und die "Einfuehrung" eben als Einfuehrung. Dabei sei jedoch an dieser Stelle betont, dass die Einfuehrung sich auf einen ganz speziellen Teil der Grafik, naemlich den grafischen OUTPUT beschraenkt, genau genommen sogar nur den auf Teil des grafischen Outputs, der durch das System BU&ZEICHE versorgt wird

Im folgenden steht das Zeichen # fuer das jeweils codeabhaengige Fluchtsymbol, waehrend das Zeichen § fuer das codeunabhaengige Fluchtsymbol steht.

Inhalt

Inhalt

		SEITE
I	Einleitung	1
II	Vorhandene Hardware	4
III a	Datenfluss	7
III b	Dateien	11
IV	Die Grafik - Unterprogramme	13
IV a	Das UP PLOT	14
IV b	Die UPs SYMBOL und NUMBER	18
IV c	Die UPs LINE, AXIS und SCALE	22
IV d	Das UP NEWPEN	27
IV e	Das UP NEXT	30
IV f	Die UPs INIT, DEVICE und FINIT	31
IV g	Das FUPLOT - Programmpaket	33
V	ZEICHNE - Kommandobeschreibung	36
VI a	Anhang 1: Hinweise fuer ALG60 Benutzer und Beispielprogramme in ALG60	58
VI b	Anhang 2: Hinweise fuer PL/1 Benutzer: Das UP S&PL1F	62
VII	Stichwortregister	68

Im Rechenzentrum der Universitaet Osnabrueck stehen fuer den grafischen Output zur Zeit folgende Geraete zur Verfuegung:

1. ein PLOTTER vom Typ BENSON 132
2. ein TEKTRONIX Sichtgeraet 4014 mit Hardcopy - Option 4631
3. es besteht die Moeglichkeit, grafischen Output in stilisierter Form ueber einen Schnelldrucker oder ein Sichtgeraet auszugeben.

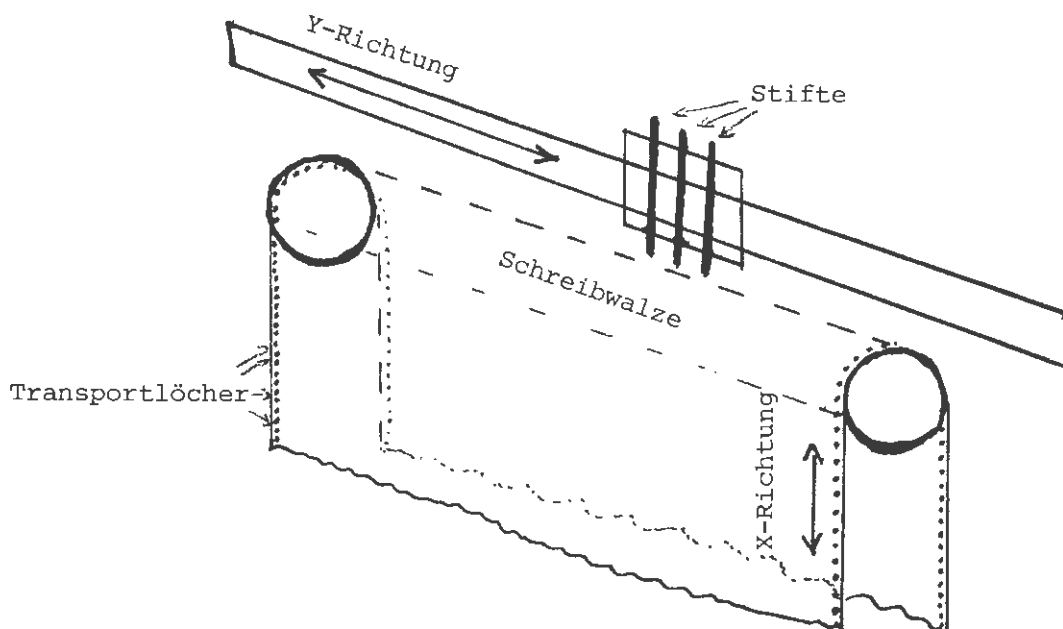
Diese dritte Moeglichkeit bereitet im allgemeinen keine brauchbare Zeichnung. Das Aufloesungsvermoegen, auch auf 132 Anschlaege breitem Drucker ist im allgemeinen viel zu gering, um brauchbare Linien darzustellen, Texte und Spezialsymbole werden als unidentifizierbares Buchstabengewirr dargestellt. Die Auffuehrung dieser Moeglichkeit erfolgt hier nur aus Gruenden der Vollstaendigkeit.

Der Plotter:

Der im RZ der Universitaet Osnabrueck zur Verfuegung stehende Plotter (BENSON 132) hat (schematisch) in etwa folgenden Aufbau:

Auf einer Vorratsrolle liegt das unbeschriebene Papier. Dieses ist an beiden Seiten mit Transportloechern versehen. Das zu bezeichnende Papier laeuft durch eine Unterdruckkammer.

ueber die Schreibwalze, durch eine weitere Unterdruckkammer und schliesslich ueber die sog. Stachelwalze in den Auffangkorb. Transportiert wird das Papier im wesentlichen durch die Schreibwalze, wobei die Unterdruckkammern in Zusammenarbeit mit der Vorratsrolle und der Stachelwalze das Papier stets gespannt halten. Direkt ueber der Schreibwalze befindet sich ein Schlitten mit drei Stiften. Dieser Schlitten bewegt sich auf einer Geraden parallel zur Schreibwalze. Wird nun einer der Stifte abgesenkt (Spitze beruehrt das Papier) so kann durch eine Bewegung der Schreibwalze ein Strich in X -Richtung, durch eine Bewegung des Schreibschlittens ein Strich in Y -Richtung und durch eine Bewegung beider Komponenten ein Strich in jede beliebige Richtung erzeugt werden.



Der Plotter, der im Rechenzentrum der Universitaet Osnabrueck installiert ist, kann mit zwei verschiedenen Papierbreiten betrieben werden:

‘breites’ Papier - 94 cm breit und

‘schmales’ Papier - 32 cm breit.

Auf breitem Papier lassen sich Zeichnungen bis zum Format DIN A 0 - quer (118,8*84 cm) anfertigen, auf schmalem Papier Zeichnungen bis zum Format DIN A 4 - hoch (21 * 29,7 cm) bzw. Zeichnungen bis zum Format DIN A 3 - quer (42 * 29,7 cm). (Siehe auch Spezifikation DIFORMAT im ZEICHNE Kommando)

Der Plotter verfuegt ueber drei verschiedene Stifte. Diese sind ueber das ZEICHNE Kommando auszuwaehlen. Die Farben, die zur Zeit zur Verfuegung stehen, sind schwarz, rot und gruen. Wird im ZEICHNE Kommando unter STIFTE nichts angegeben, so wird mit dem Standardstift (zur Zeit ist dies schwarz) gezeichnet.

Am Anfang der Zeichnung wird, aehnlich wie beim Druckerprotokoll ein Benutzerstring erzeugt, ueber den dann die Operateure die Zeichnung dem Benutzer zuordnen koennen. Am Ende der Zeichnung wird eine Schnittlinie gezeichnet (wiederum im wesentlichen fuer die Operateure wichtig) Dies geschieht automatisch, der Benutzer braucht sich darum (wenn er im XBA - bzw. XBG - Kommando vernuenftige Strings angegeben hat) keine Sorgen zu machen. Die Zeichnung findet sich je nach betrieblicher Auslastung einige Zeit nach der Fertigstellung

im Fach fuer Plotterauftraege ein.

Das TEKTRONIX Sichtgeraet.

Das Rechenzentrum der Universitaet Osnabrueck verfuegt zur Zeit ueber ein grafisches Sichtgeraet vom Typ TEKTRONIX 4014. (SIG100 oder aehnliche Geraete sind nicht vorhanden). Da dies Geraet zur Zeit scheinbar nicht als Ausgabegeraet (wie z.B. Drucker/Kartenstanzer o.ae.) adressierbar ist, ist eine Ausgabe auf dieses Geraet zur Zeit nur im Gespraechsbetrieb vom TEKTRONIX aus moeglich. Deshalb wird an dieser Stelle auf eine Beschreibung des Geraetes verzichtet.

Kap. III a

Datenfluss

Um dem Anfaenger einen gewissen Einstieg zu geben, sei hier zunaechst der Datenfluss geschildert. Am einfachsten laesst sich der Weg der Daten an einem Vergleich mit "normalem" Output beschreiben.

Output auf dem Drucker kommt folgendermassen zustande:
Der Benutzer schreibt ein Programm, in dem Ausgabeanweisungen (in FTN WRITE, in ALG60 OUTPUT / PRINT ...) enthalten sind. Diese Anweisungen bewirken, dass beim Montieren UNTERPROGRAMME an das Hauptprogramm anmontiert werden, die in der Lage sind, Information in eine Datei zu schreiben. Beim Start des Programms werden die dazu benoetigten Dateien an das Programm angeschlossen. Hierbei sind grundsaeztlich zwei

verschiedene Faelle zu unterscheiden:

1. vom System erzeugte Dateien (Ausgabepuffer) und
 2. vom Benutzer erzeugte (und verwaltete) Dateien, Systemdateien
- werden automatisch (ueber die Kanalnummern 4,6 oder 9) an das Programm angeschlossen. Benutzerdateien muessen durch besondere Anweisungen dem System bekannt gemacht werden (z.B. #STARTE,...,DATEI=...).

Die dem Programm auf diese Art angeschlossenene Dateien werden dann vom Programm aus ueber die UPs gefuehrt. Anschliessend wird der Inhalt der Dateien ausgedruckt (bzw. auf den Bildschirm kopiert).

Auch hier ist wieder zu unterscheiden, ob es sich um Dateien handelt, die vom System verwaltet werden, oder um solche, die vom Benutzer deklariert und verwaltet werden. Systemdateien werden im allgemeinen automatisch vom System aus ausgestuert. Benutzerdateien muessen noch einmal explizit angesprochen werden. Dabei wird im allgemeinen ueber ein Kommando (z.B.#DRUCKE) ein Operator gestartet, der den Inhalt der Datei auf das entsprechende Ausgabemedium ausgibt.

Selbstverwaltete Outputdateien sind zwar in der Handhabung wesentlich aufwendiger, bieten dafuer aber, wie man sofort sieht, wesentlich mehr Moeglichkeiten, den Output den eigenen Anforderungen anzupassen (z.B. durch #EDIERE, durch mehrmalige Ausgabe oder durch Teilausgaben).

Bei grafischem Output sieht der Datenfluss ganz aehnlich aus. Zunaechst wird auch hier ein Benutzerprogramm geschrieben,

welches Output - Anweisungen enthaelt. Hier aber handelt es sich um spezielle GRAFIK - Unterprogramme. Da diese UPs jedoch nicht sprachspezifisch sein koennen, muessen sie im Programm wie echte UPs verwendet werden. (in FTN und PL/1 z.B. mit CALL, in ALG60 muessen sie deklariert werden) Auch diese UPs werden (durch #MONTIERE,...) an das Hauptprogramm angeschlossen und arbeiten auf einer Datei. Fuer diese Datei ist vom Grafik-System zunaechst die Kanalnummer 87 vorgesehen. Die Datei muss explizit deklariert werden. Sie wird im #STARTE Kommando an das Programm angeschlossen. (#STARTE,... ,DATEI=87-...) Die Datei, die hierbei gebraucht wird, ist also eine vom Benutzer zu erzeugende und zu verwaltende Datei. Dementsprechend wird der Inhalt auch nicht automatisch ausgestuelpt (wie das frueher der Fall war). Es muss ein Operator gestartet werden, der den Inhalt der Datei auf das entsprechende Ausgabemedium leitet.

Dieser Operator heisst BO&ZEICHNE und wird ueber das Kommando ZEICHNE gestartet. (BO&ZEICHNE startet seinerseits einen Sohnoperator, der die Information bearbeitet und ausgibt. (PLOT&ZCH123 / PLOT&DRUCK / PLOT&TEKTRON) Da dieser Sohnoperator vom BO&ZEICHNE verwaltet wird, braucht der Benutzer sich hierum jedoch nicht zu kuemmern.) Das ZEICHNE-Kommando (siehe Kapitel V) bewirkt, dass ein Teilauftrag auf das entsprechende Geraet ausgegeben wird.

Das Bemerkenswerte an diesem Verfahren ist, dass die Information, die von den Grafik-UPs in die Datei geschrieben

wird, unabhaengig davon ist, auf welches Geraet ausgegeben werden soll. Die Umschluesselung in geraetespezifische Information (und damit die Festlegung, auf welches Geraet ausgegeben werden soll) geschieht erst im #ZEICHNE, also auf SYSTEMEBENE. Unterschiedliche Programme (bzw. unterschiedliche Programmlaeufe) fuer die Ausgabe auf verschiedenen Geraeten werden dadurch ueberfluessig. Eine einmal mit Grafik-UPS gefuellte Datei kann beliebig oft auf beliebige Geraete ausgegeben werden.

Die Tatsache, dass die Information in der Datei nicht geraeteabhaengig ist, bietet noch einen weiteren Vorteil gegenueber frueheren Systemen: Es besteht die Moeglichkeit die zu zeichnende Datei "von Hand", also unabhaengig von Grafik - UPS zu fuellen. (z.B. mit WRITE / OUTPUT / PUT o. ae. auf Programmebene oder mit #TEINTRAGE / #EDIERE o. ae. auf Kommandosprachenebene) Dazu ist natuerlich eine genaue Kenntniss der Anweisungen, die in der Datei stehen sollen, noetig. Dies ist aber nicht so schwer, wie es zunaechst scheint; die Anweisungen erscheinen naemlich in der Datei in "Klarschrift" (Texthaltungsdatei). Auf den genauen Aufbau der Anweisungen soll jedoch an dieser Stelle nicht eingegangen werden. Gegebenenfalls lasse man sich einfach eine mit Grafik - UPS gefuellte Datei ausdrucken. Mit einem bisschen Geduld kommt man dann dahinter, wie die Information aussehen muss. (Siehe Kapitel I der HANDBUCHs: "Der Aufbau der geraeteneutralen Information".)

Die im Grafik - System gebrauchte Datei ist, wie schon im Kapitel Datenfluss erwahnt, eine vom Benutzer zu erstellende und verwaltende Datei. Der Benutzer muss also, bevor ein Programm gestartet wird, das grafischen Output liefert, eine Datei bereitstellen, in der dieser Output gesammelt wird. Diese Datei kann entweder eine temporaere Datei oder eine langfristige Datei sein. Das haengt im wesentlichen davon ab, ob die Datei noch im selben Auftrag ausgegeben ("gezeichnet") werden soll, oder erst in einem spaeteren Auftrag wieder Verwendung findet. Im allgemeinen soll die Datei direkt im Anschluss an den Lauf des Hauptprogramms ausgegeben werden. Hier ist es zweckmaessig, eine temporaere Datei zu kreieren. Dabei kann unter anderem auch die Satzzahl offengelassen werden, so dass an dieser Stelle keine Schwierigkeiten entstehen koennen. (z.B. #TDEKLARIERE,NAME= GRAFDATEI. Die Voreinstellung fuer die Satzzahl ist dann u10, es lassen sich aber ohne weiteres 1000 Saetze eintragen.)

Wenn zudem die Datei nicht weiter mit Texthaltungskommandos bearbeitet werden soll, empfiehlt es sich, die Datei als sequentielle Datei zu kreieren. Das spart Speicherplatz. (z.B. #DATEI, NAME=GRADAT, TYP=SEQ, SATZZAHL=NN, SATZBAU=NN, TRAEGER=P) Die Daten, die die Groesse der Datei ausmachen (Satzzahl / Satzbau) koennen auch hier wieder dem Grafik System ueberlassen werden. Natuerlich ist in jedem Fall

fuer ausreichend TSP / PSP zu sorgen. (z.B. PSB=200, TSB=200)

Schwieriger wird die Deklaration der Datei, wenn nach Beendigung des Auftrags die grafische Information noch zur Verfuegung stehen soll. Dies kann z.B. sinnvoll sein, wenn die Zeichnung auf Magnetband archiviert, wiederholt ausgegeben oder gar auf ein Geraet ausgegeben werden soll, das zur Zeit nicht zur Verfuegung steht. In diesem Fall kann die Grafik - Datei auch eine LF - Datei sein. Hier treten aber im allgemeinen Schwierigkeiten auf, da man oft nicht im voraus weiss, wie gross die Datei wird.

Die Groesse der Datei richtet sich natuerlich nach der Menge des Outputs, so dass man eigentlich keinen festen Richtwert angeben kann. Im Kapitel IV f: Die UPS INIT, DEVICE und FINIT sind aber zumindest einige Hinweise, wieviel Platz in der Datei fuer einen Aufruf benoetigt wird. (Mehr als Ausprobieren war am Ort nicht moeglich.) Wichtig ist in diesem Zusammenhang die Beschreibung der UPS INIT, DEVICE und FINIT auch, weil sich durch den fleissigen Gebrauch dieser UPS nicht nur Speicherplatz, sondern vor allem auch Rechenzeit einsparen laesst.

In diesem Kapitel sollen nun die einzelnen UPs vorgestellt werden, die zur Ausgabe grafischer Information zur Verfügung stehen. Alle diese Unterprogramme werden durch #ERSTELLE, SYSTEM, GRAFIK zur Verfügung gestellt.

Die UPs gliedern sich nach ihrer Funktion, ihrer Bedeutung und ihrer Handhabung in drei Gruppen:

1. Die Basissoftware
2. Erweiterungen zur Basissoftware
3. Das FUPLOT-Programmpaket.

Zur Basissoftware gehören zunächst das UP PLOT, zum Zeichnen von Strichen, zur Ursprungsverschiebung und zum Vordefinieren von Massstabsfaktoren, und die UPs SYMBOL und NUMBER zur Beschriftung von Zeichnungen. Weiterhin gehören zur Basissoftware das UP NEWPEN zur Auswahl eines neuen Stiftes, das UP NEXT zur Definition einer neuen Zeichenfläche, die UPs INIT, DEVICE und FINIT, die zur Optimierung von Rechenzeit und Speicherplatz dienen, sowie die UPs LINE, AXIS und SCALE, zum Zeichnen einer einfachen Kurve mit einem Koordinatenkreuz.

Zur Erweiterung der Basissoftware gibt es eine ganze Reihe von UPs, z.B. zum Zeichnen von gestrichelten Linien, zum Zeichnen von Kreisen usw.. Diese UPs werden hier allerdings

nicht aufgefuehrt, gegebenenfalls erkundige man sich in der Benutzerberatung des Rechenzentrums.

Zum FUPLOT - Programmpaket gehoeren eine Reihe von Unterprogrammen, die mit einem einzigen Aufruf spezielle Funktionen mit Achsenkreuz, Beschriftung, Rahmen und allem, was sonst noch so dazugehoert, zeichnen. Ausserdem gehoeren zum FUPLOT - Programmpaket noch folgende Hilfsprogramme:

- CONFUM zum Aendern globaler Voreinstellungen aller FUPLOT-Programme.
- RASTER zum Ein- und Ausschalten eines Linienrasters im Diagramm
- SELFSC und AUTOSC zum Ein- und Ausschalten einer automatischen Skalierung und
- PFRAME zum Zeichnen eines Bildrahmens, der Achsen und der Bildbeschriftung.

Das UP PLOT ist das wichtigste UP im gesamten Grafik - System. Fast alle anderen Unterprogramme greifen auf dieses UP zurueck. PLOT hat drei verschiedene Grundaufgaben:

1. Das Zeichnen eines Striches von der augenblicklichen Stiftposition aus zu den angegebenen Koordinaten.
2. das Anfahren eines neuen Anfangspunktes mit gehobenem Stift und
3. eine Verschiebung des Koordinatenursprungspunktes.

Dementsprechend hat PLOT die drei Parameter X, Y, und IPEN. (X und Y REAL, IPEN INTEGER) X und Y geben die Koordinaten

des neu anzufahrenden Punktes an, IPEN bestimmt, ob mit gehobenem Stift (kein Strich) oder mit gesenktem Stift (Strich wird gezeichnet) gefahren werden soll und ob der neue Punkt [X,Y] gleichzeitig neuer Koordinatenursprung sein soll. Dabei wird folgendermassen verfahren:

Ist $ABS(IPEN) = 2$, so wird mit gesenktem Stift gefahren
(gezeichnet)

Ist $ARS(IPEN) = 3$, so wird mit gehobenem Stift gefahren
(nicht gezeichnet)

Ist $IPEN < 0$ so wird der neu anzufahrende Punkt als neuer
Koordinatenursprungspunkt definiert

Ist $IPEN > 0$ so findet keine Verschiebung des
Koordinatenursprungspunktes statt.

Achtung! Alle Koordinatenangaben in allen Grafik - UPs beziehen sich immer auf den zuletzt definierten Koordinatenursprungspunkt. Dies gilt natuerlich auch fuer PLOT - Aufrufe, auch, wenn $IPEN < 0$ ist. Hat man also z.B. mit :

```
CALL PLOT (2.5, 2.5, -3)
```

den Ursprungspunkt verschoben, so ist diese Verschiebung im wesentlichen nur mit:

```
CALL PLOT (-2.5, -2.5, -3)
```

rueckgaengig zu machen. Ein Aufruf in der Art:

```
CALL PLOT (0.0, 0.0, -3)
```

bewirkt **k e i n e** Ursprungsverschiebung!

Vor dem ersten Aufruf von PLOT steht der Zeichenstift definiert im Punkt [0,0]. Das ist der erste Koordinatenursprungspunkt.

er liegt in der linken unteren Ecke der Zeichenflaeche.
(Ein gewisser Abstand zum Benutzerstring bzw. zum Papierrand
wird automatisch eingehalten.) Kommt z.B. als erster PLOT
Aufruf :

```
CALL PLOT (2.5, 3.0, 2)
```

so wird zuerst ein Strich von [0,0] nach [2.5,3.0] gezeichnet.
Anschliessend steht der Stift im Punkt [2.5,3.0] (d.h.
2,5 cm rechts und 3 cm oberhalb der linken unteren Ecke)

Gleich noch ein Beispiel: mit

```
CALL PLOT (0.0, 29.7, 2)
```

```
CALL PLOT (21.0, 29.7, 2)
```

```
CALL PLOT (21.0, 0.0, 2)
```

```
CALL PLOT (0.0, 0.0, 2)
```

kann man einen Rahmen um eine DIN A 4 Zeichnung legen.

Der Zeichenstift steht anschliessend wieder im Punkt [0,0].

Und noch ein Beispiel:

```
CALL PLOT (10.0, 10.0, 2)
```

```
CALL PLOT (5.0, 15.0, 2)
```

```
CALL PLOT (0.0, 10.0, 2)
```

```
CALL PLOT (10.0, 10.0, 2)
```

```
CALL PLOT (10.0, 0.0, 2)
```

```
CALL PLOT (0.0, 0.0, 2)
```

```
CALL PLOT (0.0, 10.0, 2)
```

```
CALL PLOT (10.0, 0.0, 2)
```

zeichnet das beruehmte "Nikolaus-Haus", das ja bekanntlich
gezeichnet werden kann, ohne den Stift zwischendurch abzuheben.

An dieser Stelle sei gleich ein Wort zur Optimierung gesagt:

Um Plotzeit einzusparen programmiere man moeglichst so, dass moeglichst wenig und moeglichst kurze "Leerfahrten" (Fahren mit gehobenem Stift) entstehen.

Um dies im Programm dynamisch optimal zu gestalten, ist es oft erforderlich, zu wissen, wo der Zeichenstift gerade steht. Ausserdem ist die augenblickliche Stiftposition von Bedeutung, weil sie ja zugleich der Anfang der als naechstes zu zeichnenden Linie ist. Um diese Position zu erfragen, gibt es einen weiteren Eingang in das UP PLOT. Dieser Eingang heisst WHERE und liefert ueber die drei Ausgabeparameter die Koordinaten der Stiftposition sowie den im Augenblick gueltigen Massstabsfaktor (s.u.). Formal sieht der Aufruf (in FTN) so aus:

```
CALL WHERE (RX, RY, RFACT)
```

(alle drei Parameter REAL.) Nach dem Aufruf von WHERE steht in RX der letzte X - Wert, in RY der letzte Y - Wert und in RFACT der zur Zeit gueltige Massstabsfaktor.

Sofern nichts anderes vereinbart wurde, gelten alle Angaben zunaechst in cm. Das heisst aber nicht, dass man nun alle Daten in cm umrechnen muss. Es genuegt, wenn man einen Massstabsfaktor angibt, mit dem alle folgenden Groessenangaben in allen Grafik - UP Aufrufen (mit Ausnahme von FACTOR!) multipliziert werden. Dies kann durch einen weiteren Eingang in das UP PLOT voreingestellt werden. Dieser Eingang heisst FACTOR. Formal sieht der Aufruf etwa folgendermassen aus (in FTN):

CALL FACTOR (FACT)

(FACT ist REAL). Ein Aufruf von FACTOR bewirkt, dass alle Werte in darauffolgenden Grafik - UP - Aufrufen mit FACT multipliziert werden. Zum Beispiel CALL FACTOR (0,5) bewirkt, dass alle nachfolgenden Werte halbiert werden, der anschliessende Aufruf CALL FACTOR (1,0) bewirkt, dass wieder der urspruengliche Massstabsfaktor (also in cm) gilt. FACTOR wirkt also nicht accumulativ.

Eine halbwegs vernuenftige Zeichnung muss in irgendeiner Weise beschriftet werden. Dazu dienen die UPs SYMBOL und NUMBER.

Mit SYMBOL ist es moeglich, an beliebiger Stelle in einer Zeichnung Text anzubringen. Dazu ist natuerlich erforderlich zu wissen, wo in der Zeichnung der Text anfangen soll, in welchem Winkel der Text verlaufen soll, wie lang der Text wird und schliesslich wie hoch die einzelnen Zeichen werden sollen. Dies ist alles dem System in jedem SYMBOL - Aufruf mitzuteilen.

Formal sieht das folgendermassen aus (in FTN):

CALL SYMBOL (X, Y, HEIGHT, ITEXT, ANGLE, NCHAR)

(X, Y, HEIGHT und ANGLE REAL, NCHAR INTEGER, ITEXT H - Konstante oder ein Feld zur Aufnahme einer H - Konstanten)
(siehe auch SYMBOL - Beschreibung im Grafik - Handbuch)

Als Parameter ITEXT wird eine H = Konstante (oder ein Feld, das eine H = Konstante beinhaltet) nach FTN = Konventionen erwartet. ALGOL60-Benutzer muessen diesen Text entsprechend aufarbeiten. (siehe Anhang 1: Hinweise fuer ALG60 Benutzer)

Die Parameter X und Y bezeichnen die Koordinaten des linken unteren Eckpunktes des ersten zu zeichnenden Zeichens. Ist einer dieser Parameter = 999.0, so wird statt dessen das Ende des zuletzt mit SYMBOL oder NUMBER ausgegebenen Strings als Anfangskoordinate berechnet. Auf diese Art ist es moeglich, mehrere Strings in verschiedenen SYMBOL und NUMBER Aufrufen "hintereinander" zu zeichnen.

HEIGHT bezeichnet die Hoehe des auszugebenen Strings in cm, ANGLE bezeichnet den Winkel zwischen der Schriftgrundlinie und der positiven X = Richtung in Grad.

Als NCHAR ist die Zahl der auszugebenen Zeichen anzugeben. Wird NCHAR = 0 gewaehlt, so wird ein Zeichen ausgegeben. Dieses Zeichen ist dann das im ersten Wort ganz rechts stehende Zeichen.

Ein kleines Beispiel:

```
CALL SYMBOL(3.0, 2.0, 0.5, 20HDIES IST DIE X-ACHSE, 0.0, 20)
CALL SYMBOL(2.0, 3.0, 0.5, 20HDIES IST DIE Y-ACHSE, 90.0, 20)
```

Diese beiden Aufrufe wuerden die Achsen eines (geeigneten) Achsenkreuzes beschriften.

Das UP SYMBOL kann auch benutzt werden, um einzelne Punkte durch ein spezielles Zeichen, ein sogenanntes Spezialsymbol















zu kennzeichnen. Der Aufruf von SYMBOL sieht dann etwas anders aus:

```
CALL SYMBOL(X, Y, HEIGHT, INTEQ, ANGLE, ICODE)
```

Die Parameter HEIGHT und ANGLE haben dieselbe Bedeutung wie in dem oben beschriebenen normalen Aufruf von SYMBOL.

X und Y bezeichnen den Mittelpunkt des zu zeichnenden Spezialsymbols.

Die Parameter INTEQ und ICODE (beide INTEGER) zeichnen den Spezialaufruf aus: ICODE ist eine negative Konstante die angibt, ob mit gehobenem oder mit gesenktem Stift zum zu markierenden Punkt gefahren werden soll. Ist dieser Parameter = -1, so wird mit gehobenem Stift (also ohne Verbindungslinie) zum angegebenen Punkt [X,Y] gefahren, ist der Parameter < -1, so wird mit gesenktem Stift zum angegebenen Punkt [X,Y] gefahren. INTEQ ist schliesslich die Nummer des zu zeichnenden Spezialsymbols demaess folgender Zuordnung: S P E Z I A L S Y M B O L E-----

NR. 0	NR. 1	NR. 2	NR. 3
			
NR. 4	NR. 5	NR. 6	NR. 7
			
NR. 8	NR. 9	NR. 10	NR. 11
			
NR. 12	NR. 13		
			

Will man nun an eine bestimmte Stelle in der Zeichnung eine Zahl eintragen, die man erst waehrend des Programmlaufes berechnet, so muss diese Zahl zunaechst in eine Zeichenkette gewandelt werden. Dies leistet das UP NUMBER. Der Aufruf sieht folgendermassen aus:

```
CALL NUMBER(X, Y, HEIGHT, FPN, ANGLE, NDEC)
```

Die Parameter X, Y, HEIGHT und ANGLE sind die gleichen wie in den bereits beschriebenen SYMBOL Aufrufen. FPN ist die zu wandelnde und zu zeichnende Zahl, NDEC (INTEGER) ist ein Steuerparameter, der das Ausgabeformat der Zahl angibt. Dabei wird wie folgt verfahren:

NDEC > 0; NDEC bezeichnet die Zahl der Dezimalstellen hinter dem Dezimalpunkt.

NDEC = 0; nur der ganzzahlige Teil von FPN wird mit Dezimalpunkt gezeichnet.

NDEC = -1; wie NDEC = 0, jedoch ohne Dezimalpunkt.

NDEC < -1; Es werden die letzten |NDEC| - 1 Ziffern vom gerundeten ganzzahligen Teil der Zahl abgeschnitten, der verbleibende Rest wird gezeichnet.

Ein kleines Beispiel koennte dies vielleicht verdeutlichen:

```
X=123456789.987654321
```

```
CALL NUMBER(1.0, 1.0, 1.0, X, 0.0, 4)
```

```
CALL NUMBER(999.0, 999.0, 0.5, X, 45.0, 0)
```

```
CALL NUMBER(999.0, 999.0, 0.75, X, 90.0, -3)
```

```
CALL NUMBER(999.0, 999.0, 0.25, X, 180.0, 9)
```

```
CALL NUMBER(999.0, 999.0, 0.25, X, 315.0, -5)
```

erzeugt folgenden Output:

123456789.9873123456790.1234567

123456789.9873123456790.1234567

Fuer viele Probleme genuegt es einen Polygonzug mit dazugehoerigem Achsenkreuz zu zeichnen. Natuerlich laesst sich das durch die bereits beschriebenen UPs PLOT und SYMBOL leisten. Dies fuehrt jedoch gerade bei der Darstellung der Achsen zu aufwendiger Kleinarbeit. Ausserdem treten Schwierigkeiten auf, wenn es darum geht, fuer eine noch nie gesehene Zeichnung Skalierungsparameter aufzustellen. Um hier unnoetige Doppelarbeit zu ersparen, wurden die UPs LINE, AXIS und SCALE geschaffen. Sie gestatten es mit verhaeltnissmaessig wenig Aufwand Polygonzuege gemuess

einer Menge von Wertepaaren zu zeichnen. Die Werte werden in zwei eindimensionalen Feldern erwartet.

Die Arbeitsweise sieht folgendermassen aus: Zunaechst werden mit je einem Aufruf von SCALE Skalierungsparameter fuer die X - bzw. Y - Werte berechnet. Dazu ist erforderlich, dass die Laenge der vorgesehenen Achse angegeben wird.

Die Werte, die berechnet werden sind:

1. FIRSTV: Der Achsenanfangspunkt und
2. DELTAV: Der Zuwachs pro cm Achse

Diese Werte werden in den Feldern, in denen schon die Koordinaten stehen, abgespeichert, und zwar jeweils hinter die letzten Koordinatenwerte. Danach werden mit je einem Aufruf von AXIS die X - und Y - Achse gezeichnet. Die dafuer erforderlichen Parameter FIRSTV und DELTAV koennen (und sollen auch ruhig) aus dem entsprechenden Koordinatenfeld herausgezogen werden. Das stellt sicher, dass die Achsen die richtige Laenge und Skalierung haben. Zum Schluss wird dann mit Hilfe des UPS LINE ein Polygonzug in das bereits erstellte Achsenkreuz gezeichnet. Da in dem UP AXIS Parameter fuer die Achsenbeschriftung vorgesehen sind, laesst sich so schon ein vollstaendiges Bild erzeugen.

Formal sehen die Aufrufe (in FTN) etwa folgendermassen aus:

```
CALL SCALE (ARRAY, AXLEN, NPTS, INC)
```

```
CALL AXIS (X, Y, ITEXT, NCHAR, AXLEN, ANGLE, FIRSTV, DELTAV)
```

```
CALL LINE (XARRAY, YARRAY, NPTS, INC, LINTYP, INTEQ)
```

Dabei sind ARRAY, XARRAY und YARRAY eindimensionale Felder

vom Typ REAL, NPTS, INC, NCHAR, LINTYP und INTEQ sind einfache INTEGER, HTEXT ist eine H-Textkonstante oder eine Variable bzw. ein Feld, dem ein Text zugewiesen wurde. (siehe auch Anhang 1: Hinweise fuer ALG60 Benutzer) Die genaue Bedeutung der Parameter entnehme man dem Grafik - Handbuch

Dazu schell ein kleines Beispiel:

C---MINIPROGRAMM ZUM ZEICHNEN VON $F(X) := \sin(X)$

```
DIMENSION XWERT(52), YWERT(52)
```

```
DELTA=(8.0*ATAN(1.0))/49.0
```

```
X=-DELTA
```

```
DO 10 I=1,50
```

```
  X=X+DELTA
```

```
  XWERT(I)=X
```

```
  YWERT(I)=SIN(X)
```

```
10 CONTINUE
```

```
CALL SCALE(XWERT, 15.0, 50, 1)
```

```
CALL SCALE(YWERT, 10.0, 50, 1)
```

```
CALL AXIS(0.0, 5.0, 21HX-ACHSE F(X)=SIN(X), +21, 15.0, 0.0,
```

```
F      XWERT(51),XWERT(52))
```

```
CALL AXIS(0.0, 0.0, 7HY-ACHSE, -7, 10.0, 90.0, YWERT(51),
```

```
F      YWERT(52))
```

```
CALL LINE(XWERT, YWERT, 50, 1, 0, 2)
```

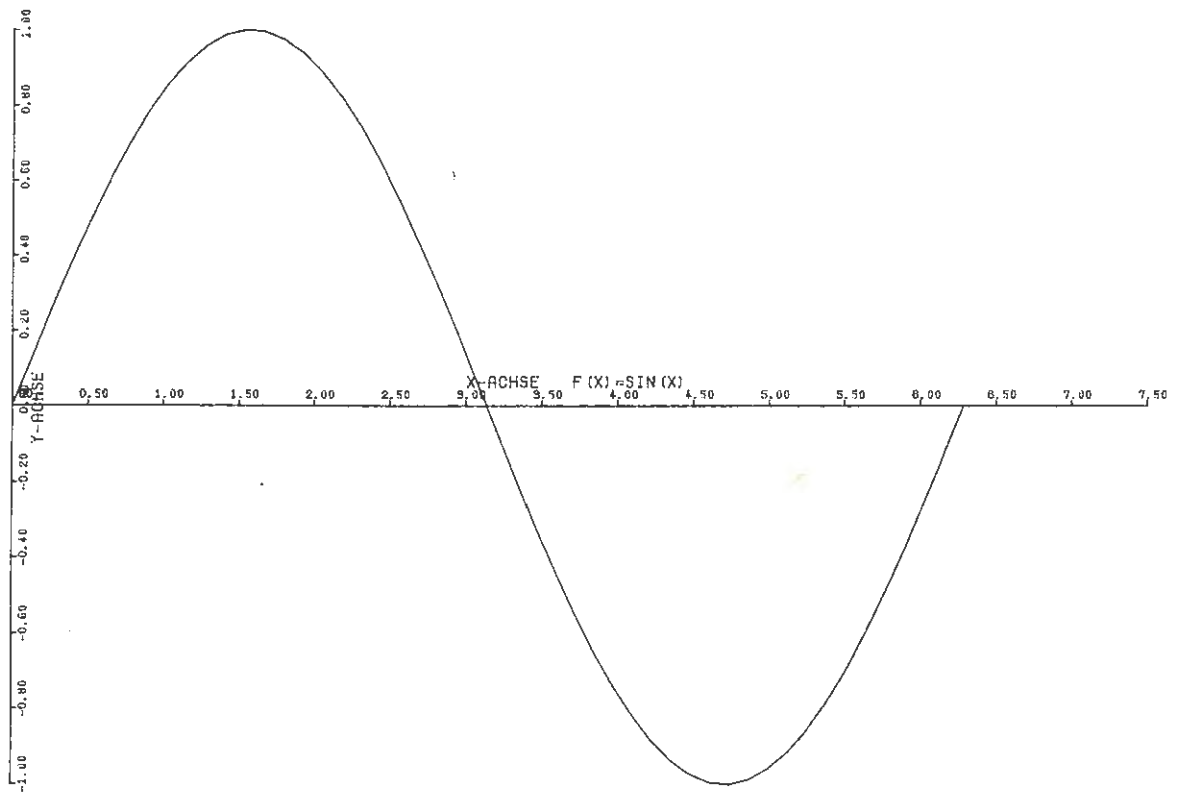
```
STOP
```

```
END
```

Dieses Programm leistet folgendes:

Nachdem die Werte fuer X und SIN(X) berechnet worden

sind, werden in SCALE die Werte FIRSTV und DELTAV berechnet und in das Feld XWERT bzw, in das Feld YWERT (jeweils in das 51. bzw 52. Feldelement) eingetragen. Mit AXIS werden dann die Achsen gezeichnet und beschriftet. Der Aufruf von LINE schliesslich bewirkt, dass die "Kurve" in das Achsenkreuz gezeichnet wird. Der Output der auf diese Weise entsteht, sieht folgendermassen aus:



Will man nun mehrere Polygonzuege in dasselbe Achsenkreuz zeichnen, so empfiehlt es sich, alle Daten in insgesamt nur zwei SCALE Aufrufen zu skalieren. Sinnvollerweise schreibt man dazu alle Daten in zwei grosse Felder, auf die dann SCALE angewendet wird. Die beiden Zahlen FIRSTV und DELTAV, die in den beiden AXIS Aufrufen und in den einzelnen LINE Aufrufen (LINE erwartet die Parameter FIRSTV und DELTAV in den beiden Koordinatenfeldern jeweils als n+1. bzw. n+2. Feldelement) gebraucht werden, entnimmt man dann den beiden grossen Feldern. Sie muessen natuerlich, wenn ein einheitlicher Massstab beibehalten werden soll, in allen LINE Aufrufen identisch sein.

Der Parameter LINTYP im UP LINE bietet weiterhin die Moeglichkeit, zusaetzlich zur "Kurve" auch noch einzelne Punkte durch Spezialsymbole (siehe SYMBOL - Beschreibung) zu kennzeichnen. LINTYP ist der fuenfte Parameter. Ist LINTYP ungleich 0, so wird auf jeden n-ten Punkt der "Kurve" ein bestimmtes Spezialsymbol gesetzt, wobei $n = \text{ABS}(\text{LINTYP})$ ist. Die Nummer des gewuenschten Spezialsymbols ist als sechster Parameter (INTEQ) anzugeben. Es ist aber darauf zu achten, dass eine Linie durch alle Datenpunkte nur dann gezeichnet wird, wenn $\text{LINTYP} \leq 0$ ist. Ist $\text{LINTYP} > 0$, so wird nur das mit INTEQ bezeichnete Spezialsymbol gezeichnet. Beispiel: $\text{LINTYP} = -12$ bewirkt, dass eine Linie durch alle Datenpunkte gezeichnet wird und zusaetzlich jeder 12-te Wert mit einem Spezialsymbol gekennzeichnet

wird. Dagegen bewirkt LINTYP=20 dass nur jeder 20-te Punkt durch ein Spezialsymbol gekennzeichnet wird und keine Linie gezeichnet wird.

Im allgemeinen erwarten die UPS SCALE und LINE eindimensionale Felder, in denen die Daten kompakt angeordnet sind. Deshalb ist der Parameter, der angibt, um wieviel Feldelemente pro Datenwert weiter zu zaehlen ist (INC, jeweils der vierte Parameter) im allgemeinen = 1 zu waehlen. Handelt es sich aber bei den Feldern um mehrdimensionale Felder oder stehen zwischen den Koordinatenwerten noch andere Werte, so ist INC entsprechend anders zu waehlen.

Kap. IV d

Das UP NEWPEN

Der Plotter verfuegt, wie bereits eingangs erwaeht, ueber drei Stifte. Auch andere Ausgabegeraete koennen ueber mehr als einen Stift verfuegen. Es kann deshalb sinnvoll sein, einen neuen Stift zu waehlen. Dazu gibt es das UP NEWPEN. NEWPEN hat einen Parameter NPEN. Dieser Parameter ist INTEGER.

Solange nichts anderes vereinbart wurde, wird mit dem Standardstift gezeichnet. Will man nun einen anderen Stift auswaehlen, so rufe man NEWPEN auf:

```
CALL NEWPEN (NPEN)
```

NPEN muss dabei > 0 sein. In der Datei in der die Grafikinformatio n abgelegt wird, wird dann ein Vermerk eingetragen, so dass

spaeter ueber das ZEICHNE - Kommando von dieser Stelle an ein neuer Stift gewaehlt werden kann. Da die Information, die in der Datei abgelegt wird, unabhaengig vom Ausgabegeraet ist, kann als NPEN irgendeine natuerliche Zahl angegeben werden. Mit welchem Stift nachher tatsaechlich gezeichnet wird, wird erst im ZEICHNE - Kommando festgelegt. Das geht folgendermassen:

Unter der Spezifikation STIFTE im ZEICHNE - Kommando werden mehrere Teilwerte angegeben, die den einzelnen realen Stiften entsprechen. Dies ist vom Ausgabegeraet abhaengig, beim Plotter z.B. sind die Angaben 0 (kein Stift), 1 (schwarz), 2 (rot) bzw. 3 (gruen) sinnvoll. (Beim Tektronix - Sichtgeraet z.B. nur die Angaben 0 oder 1). Die Zuordnung der Stifte erfolgt dabei ueber die Position innerhalb des Spezifikationswertes. Der als i-ter Teilwert angegebene reale Stift entspricht dem mit NEWPEN(i) gewaehlten virtuellen Stift. Hat man also als Parameter bei NEWPEN die Zahlen 1, 2, 4, 6 und 7 angegeben, so muss, wenn man nicht mit dem Standardstift zeichnen will, der Spezifikationswert zu STIFTE aus 7 Teilwerten bestehen. Hierzu auch ein kleines Beispiel:

```
CALL SYMBOL (0.0, 4.0, 0.5, 28) MIT STANDARDSTIFT GEZEICHNET,
*           0.0, 28)
```

```
C DA BISHER NOCH NEWPEN - AUFRUF VORKAMM WIRD NOCH MIT DEM
C STANDARDSTIFT GEZEICHNET
```

```
CALL NEWPEN (2)
```

```
CALL SYMBOL (0.0, 3.0, 0.5, 22) MIT STIFT 2 GEZEICHNET, 0.0, 22)
```

```
CALL NEWPEN (3)
```

CALL SYMBOL (0.0, 2.0, 0.5, 22HMIT STIFT 3 GEZEICHNET, 0.0, 22)

CALL NEWPEN (4)

CALL SYMBOL (0.0, 1.0, 0.5, 22HMIT STIFT 4 GEZEICHNET, 0.0, 22)

Im ZEICHNE - Kommando sei die Stiftzuordnung :

#ZEICHNE,...,STIFTE=0'2'0'3

(4 Teilwerte). Der Output, der auf diese Weise entsteht, sieht folgendermassen aus: (Leider war es drucktechnisch nicht moeglich, den Output auch farbig erscheinen zu lassen)

MIT STANDARDSTIFT GEZEICHNET	(erscheint schwarz)
MIT STIFT 2 GEZEICHNET	(erscheint rot)
	(erscheint gar nicht)
MIT STIFT 4 GEZEICHNET	(erscheint grün)

Wie man aber schon sieht, ist es auf diese Weise auch moeglich, Teilzeichnungen auszugeben (bzw. nicht auszugeben, zu unterdruecken) (siehe auch Kap V ZEICHNE - Kommandobeschreibung)

Oft will man in einem Programm mehrere Zeichnungen zusammenstellen. Dazu gibt es das UP NEXT. NEXT hat einen Parameter. Dieser Parameter ist INTEGER und muss bei der hiesigen Gerätekombination = 1 sein. Der ZEICHNE - Operator berechnet, wenn er auf ein NEXT stößt, in optimaler Weise einen Papiervorschub, so dass dem Benutzer quasi eine neue Zeichenfläche zur Verfügung steht.

Der Aufruf von NEXT ist besonders sinnvoll zwischen je zwei FUPLOT Aufrufen.

An dieser Stelle sei gleich noch auf eine andere Möglichkeit hingewiesen: Oft ist es sinnvoll mehrere Zeichnungen in einem #ZEICHNE auszugeben. Dazu kopiere man alle "Zeichnungen" (Dateien in denen Grafikinformatio n steht) die man ausgeben will in eine einzige Datei, wobei man zwischen je zwei "Zeichnungen" die Zeile:

```
NEXT 1;
```

einfügt. Der Operator BO&ZEICHNE verteilt dann die Zeichnungen optimal auf der zur Verfügung stehenden Fläche. So können z.B. auf 94 cm breitem Papier bis zu vier DIN A 4 Zeichnungen nebeneinander gezeichnet werden.

Im Normalfall belegt jede grafische Operation (PLOT - Aufruf / SYMBOL - Aufruf / NEWPEN - Aufruf usw.) eine Zeile in der Grafikdatei. Dementsprechend genuegt es in aller Regel, wenn diese Datei mit einem Satzbau von U 55 0 deklariert wird. Da aber in jedem Fall der Satzschluessel jeder einzelnen Zeile innerhalb der Datei auch einen gewissen Speicherplatz belegt, ist es besonders in "grossen" Dateien immer wuensenswert, moeglichst wenig Saetze zu haben. Die einzelnen Saetze koennen bis zu 1024 Oktaden lang sein (!). Um nun diesen Satzbau auch ausnutzen zu koennen, gibt es die UPs INIT und DEVICE. Beide Unterprogramme bewirken, dass nach ihrem Aufruf der Output, der von den Grafik - UPs erzeugt wird, nicht direkt in die Datei geschrieben wird, sondern zunaechst in einem Puffer im Kernspeicher gesammelt wird. Das hat ausserdem noch den Vorteil, dass man erheblich Rechenzeit einspart, da der Ausgabevorgang dadurch optimiert wird.

Darueberhinaus ist es mit dem UP DEVICE auch noch moeglich, die Kanalnummer, ueber die ausgegeben wird, variabel zu halten. Es kann also auf diese Weise in einem einzigen Programmlauf auf mehrere Dateien ausgegeben werden.

Formal sehen die Aufrufe (in FTN) folgendermassen aus:

CALL INIT

CALL DEVICE (ISGNR, LNG)

CALL FINIT

(INIT und FINIT sind parameterlos, die beiden Parameter von DEVICE, ISGNR und LNG sind INTEGER)

Der Aufruf von INIT bewirkt, dass bis zu 9 Grafikoperationen in einen Satz der Grafikdatei geschrieben werden. Der Satzbau sollte in dem Fall U 200 0 betragen.

Der Aufruf von DEVICE bewirkt, dass

1. alle noch im Ausgabepuffer vorhandenen Grafikoperationen sofort ausgegeben werden,
2. die Geräetenummer auf ISGNR umgeschaltet wird und
3. von nun an bis zu LNG Grafikoperationen gesammelt und in einen Satz ausgegeben werden.

(Achtung: fuer LNG muss gelten: $1 \leq \text{LNG} \leq 63$, es koennen also maximal 63 Grafikoperationen in einen Satz ausgegeben werden) Bei Gebrauch von DEVICE mit $\text{LNG} = 63$ sollte der Satzbau U 1024 0 betragen. Wird LNG kleiner gewaehlt, so koennen die einzelnen Saetze der Grafikdatei natuerlich auch kuerzer sein (ausprobieren).

Achtung: In jedem Fall, ob nun INIT oder DEVICE verwendet wird, wird die Grafikinformatio n in einem internen Puffer gesammelt. Im Gegensatz zu allen anderen Faellen bedarf es eines SCHLUSSAUFRUFES. Dieser Schlus saufruf bewirkt, dass der letzte Pufferinhalt ausgegeben wird. (Ausgaben zwischendurch, wenn der Puffer gefue llt ist, erfolgen natuerlich automatisch) Dieser Schlus saufruf kann entweder sein:

CALL FINIT

(ohne Parameter) oder;

CALL DEVICE (I,J)

(wobei I und J nahezu beliebig sein koennen). Ein Schlusssaufruf in dieser Weise ist aber nur sinnvoll, wenn auch vorher ueber die UPS INIT oder DEVICE ein Ausgabepuffer organisiert wurde.

Mit den UPS des FUPLOT - Programmpakets ist es moeglich, ganze Funktionen oder sogar ganze Funktionsfamilien einschliesslich Achsenkreuz, Beschriftung, Rahmen usw. in einem Aufruf zu zeichnen. Die zu zeichnenden Funktionen muessen im allgemeinen als FUNKTION EXTERNAL vorliegen. Da die FUPLOT-UPS alle aehnlich aufgebaut sind und auch im Grafik - Handbuch ausfuehrlich beschrieben sind, soll hier nicht noch einmal eine ausfuehrliche Beschreibung aller FUPLOT UPS anfehrt werden. Auch die immer wiederkehrenden Parameter "TITEL", "TEXT1", "TEXT2" und "LAYOUT" sind inzwischen in den "Vorbemerkungen zu KAPITEL 6" im Grafik - Handbuch beschrieben.

Statt dessen soll an dieser Stelle ein kleiner, aber vollstaendiger Beispieljob angefuegt werden.

Nach der XBA - Karte und dem Passwort geht es weiter mit

#TDEKLARIERE, QUELLDATEI

```
#TEINTRAGE, QUELLDATEI, INFORMATION=/  


```

```
REAL FUNCTION EXTSIN(X)
```

```
C EXTERNES UNTERPROGRAMM ZUM BERECHNEN DES SIN
```

```
EXTSIN=SIN(X)
```

```
RETURN
```

```
END
```

```
C HAUPTPROGRAMM
```

```
EXTERNAL EXTSIN
```

```
CALL PFUNC(EXTSIN, 0.0, 8*ATAN(1.0), 002, 13HF(X) = SIN(X),
```

```
F          9HX - ACHSE, 9HY - ACHSE)
```

```
STOP
```

```
END
```

```
#TDEKLARIERE, ZEICHDAT
```

ZEICHDAT soll spaeter die grafische Information aufnehmen,

SATZZAHL wird hier nicht angegeben (Voreinstellung = U

10).

```
#UEBERSETZE, QUELLDATEI
```

```
#ERSTELLE, SYSTEM . GRAFIK (Wichtig!)
```

```
#MONTIERE
```

Hier wird unter anderem das UP PFUNC an das Hauptprogramm

anmontiert.

```
#STARTE, DATEI=87-ZEICHDAT
```

Danach steht die grafische Information in der Datei ZEICHDAT

und kann nun mit:

```
#ZEICHNE, GERAET=PL, DATEI=ZEICHDAT
```

ausgegeben werden. Dabei wuerde auf breites Papier gezeichnet

werden. Da die Zeichnung jedoch DIN A 4 - Format hat, wuerde

man eine solche Zeichnung eher auf schmalem Papier ausgeben.

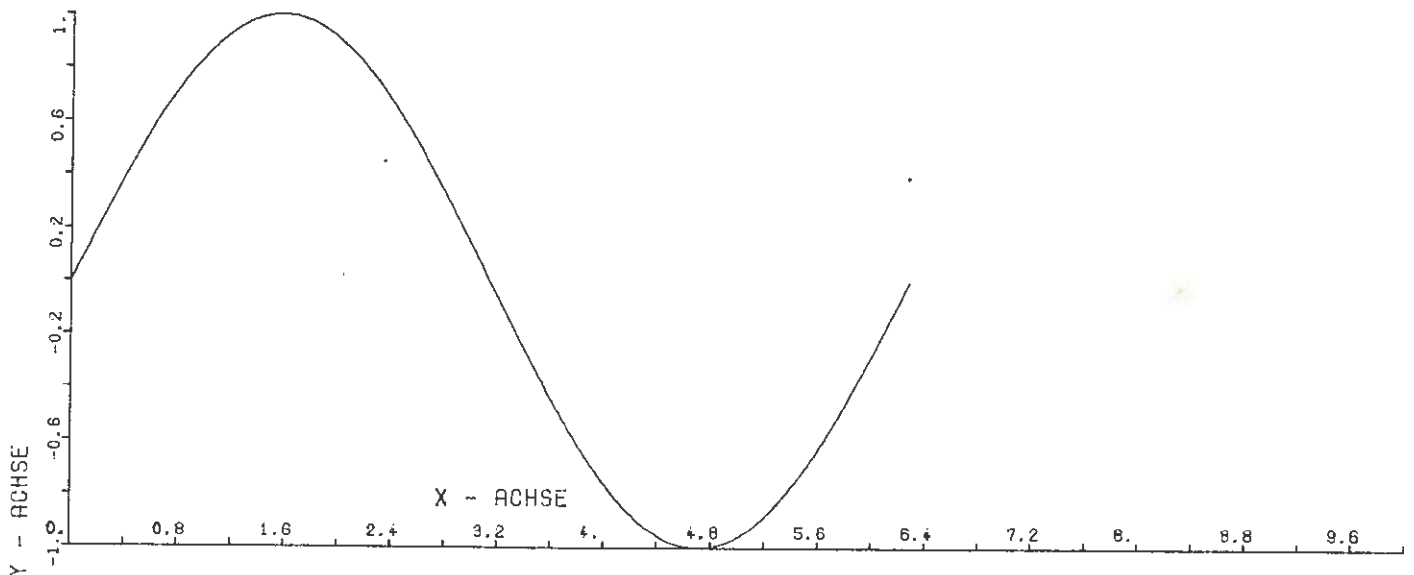
Das ZEICHNE - Kommando wuerde dann wie folgt aussehen:

```
#ZEICHNE, GERAET=PL, DATEI=ZEICHDAT, MODUS=DINA4F, DINAFORMAT=5,
```

```
AUSSCHNITT=0.0°0.0°29.8°21.1
```

Ausserdem koennte man an dieser Stelle natuerlich noch eine ANZAHL angeben, Stifte auswaehlen oder ein anderes Format waehlen, dies ist aber im Kapitel V "ZEICHNE - Kommandobeschreibung" beschrieben. Nach dem ZEICHNE - Kommando kommt dann noch die bekannte #4XEN# Karte. Der Output der dabei entsteht, sieht folgendermassen aus:

$$F(X) = \sin(X)$$



Z E I C H N E

Ausgabe graphischer Information auf ein Zeichengerät oder in einer Datei

Spezifikation:

1	GERAET	Angabe des Zeichengerätes oder der Zieldatei
2	DATEI	Angabe der auszugebenden Information
3	ANZAHL	Anzahl der Ausgabevorgänge
4	MODUS	Angabe des Ausgabemodus
5	PROTOKOLL	Angabe einer zusätzlichen Protokollierung
6	BEREICH	Spezifizierung eines zu bearbeitenden Dateibereichs
7	PARAMETER	Angaben zur Parametrisierung der Zeichnung
8	DINAFORMAT	Angabe der gewünschten Größe der Zeichnung
9	AUSSCHNITT	Definition des rechteckigen Ausschnitts aus der gesamten Zeichnung, der gezeichnet werden soll.
10	STIFTE	Angabe der gewünschten Zeichenstifte
Kommando für das Programmiersystem		anlagenspezifische Voreinstellung:

Einschränkung:

Das ZEICHNE - Kommando benutzt den Operator BO&ZEICHNE, dieser kann durch #ERSTELLE, SYSTEM . GRAFIK oder durch #ERSTELLE, KOMMANDO . ZEICHNE verfügbar gemacht werden.

Wirkung:

Das Kommando ZEICHNE erfüllt zwei Funktionen:

- Die vom Benutzer erstellte geräteneutrale Beschreibung einer Zeichnung kann in Fahrbefehle o.ä. für einen bestimmten Gerätetypgewandelt und in wählbarer Größe ganz oder teilweise ausgegeben werden.
- Eine Datei, die bereits gerätespezifische Information enthält, kann auf ein entsprechendes Gerät ausgegeben werden.

Die vom Benutzer mit einem Programm oder "von Hand" erstellte Beschreibung einer Zeichnung enthält keinerlei gerätespezifische Angaben; erst im ZEICHNE-Kommando wird festgelegt, welcher Teil der Zeichnung in welcher Größe mit welchen Stiften auf welches Gerät gezeichnet werden soll.

Außerdem lassen sich unter PARAMETER endgültige Zahlen für irgendwelche Koordinatenwerte, Faktoren o.ä. angeben, die in der Bildbeschreibung zunächst offengelassen wurden.

Z E I C H N E

formal:

```
<ZEICHNE-Kommando> ::= # ZEICHNE [ , [ <Spezifikationsname> = ]  
                                <Spezifikationswert> ]∞  
  
<Spezifikationsname> ::= GERAET | DATEI | ANZAHL | MODUS | PROTOKOLL  
                           BEREICH | PARAMETER | DINAFORMAT |  
                           AUSSCHNITT | STIFTE
```

Beispiel:

```
#ZEICHNE ,GER.=PL,DATEI=DBP.PLOTTERDATEI,ANZ.=2
```

Die Datei PLOTTERDATEI aus der Datenbasis DBP soll zweimal auf dem Zeichengerät PL ausgegeben werden.

```
#ZEICHNE ,KO,KO
```

Die Zeicheninformation soll im Dialog am Terminal angefordert und gerätespezifisch auch dort ausgegeben werden.

ZEICHNE

① G E R A E T

G E R A E T

Angabe des Zeichengerätes oder der Zieldatei

Spezifikationswert:

"undefiniert" es wird kein Bild erstellt

$g[(a[,b])][-[c][-m]]$ Ausgabe auf Gerät g mit Gerätenummer a und Stationsnummer b in Code c auf Material m

g: DR Drucker
(FS8 Fernschreiber, 8 Spuren) (keine Codeangaben)
SI50 SIG 50
WAEHL Wählgerät
PL Plotter (keine Codeangabe)
KS Kartenstanzer
SS5 Streifenstanzer, 5 Spuren
SS8 Streifenstanzer, 8 Spuren
m: Materialkennzeichen

c: KC1 Kartencode 1
KC2 Kartencode 2
KC3 Kartencode 3
KC4 Kartencode 4
SC1 Streifencode 1
SC2 Streifencode 2
SC4 Streifencode 4
DC1 Druckercode 1
DC2 Druckercode 2
BIN BINÄR

} nur wenn g=KS
} nur wenn g=SS5
} nur wenn g=SS8
} nur wenn g=DR
} nur wenn g=KS,SS5

KO Ausgabe ins Konsolprotokoll des laufendes Gespräches
DRP Ausgabe ins Ablaufprotokoll
[dbn.]dtn[(g.v)][-pw]? Ausgabe in Datei dtn in Datenbasis dbn mit der GV-Nr. g.v und dem Paßwort pw

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische Voreinstellung: —

Einschränkung:

Welche Gerätebezeichnungen sinnvoll sind, hängt von der jeweiligen Installationskonfiguration ab.

FS8 ist z.Zt. nicht als graphisches Ausgabegerät adressierbar.

Wirkung:

Die folgenden Angaben sind möglich:

-Gerätebezeichnung, z.B. PL

Die Zeichnung wird als Teilauftrag auf dem angegebenen Gerät ausgegeben.

Die möglichen Geräte sind im 1. Kapitel des Graphik-Handbuch beschrieben.

-KO:

Es wird ins Konsolprotokoll des laufenden Gesprächs gezeichnet.

#ZEICHNE,KO,... ist die übliche Angabe beim interaktiven Arbeiten am Sichtgerät.

-DRP:

Die Zeichnung wird ins Ablaufprotokoll ausgegeben.

Wichtig: Wenn sich aus dieser Geräteangabe nicht erkennen läßt, um was für eine Art von Gerät es sich handelt, muß dies unter MODUS ausgewählt werden.

Soll dieselbe Zeichnung sehr oft auf ein und demselben Gerät ausgegeben werden, auch bei Versorgung abgesetzter Geräte über externe Datenträger, kann es sich lohnen, die graphische Information in gerätespezifischer Form aufzubewahren. Zu diesem Zweck kann bei GERAET auch eine Dateibezeichnung angegeben werden. In diesem Fall muß der Gerätetyp unbedingt unter MODUS spezifiziert werden.

formal:

<Wertzuweisung GERAET> ::= [GERAET=] <Gerätebezeichnung> [<Identifizierung>]
[- [<Codeangabe>] [- <Materialangabe>]]

<Gerätebezeichnung> ::= DR | FS8 | SI50 | WAEHL | PL | KS | SS5 | SS8

<Identifizierung> ::= ([<Gerätenummer>] [, <Gerätestation>])

<Gerätenummer> ::= <natürliche Zahl zwischen 0 und 254>

<Gerätestation> ::= <natürliche Zahl zwischen 0 und 254>

<Codeangabe> ::= KC1 | KC2 | KC3 | KC4 | SC1 | SC2 | SC4 | DC1 | DC2 | BIN

<Materialangabe> ::= <natürliche Zahl zwischen 0 und 254>

Betreffs möglicher Kombinationen vergleiche Vorderseite.

Beispiel:

...,GERAET=PL,...

Es wird auf den Plotter gezeichnet.

...,GER.=KO,...

Es wird ins Konsolprotokoll des laufenden Gesprächs gezeichnet.

DATEI

Angaben der zu zeichnenden Information

Spezifikationswert:

/f f:Information als Fremdstring
KO Information wird im Dialog von der Konsole angefordert
[dbn.]dtn[(g.v)][-pw]Information liegt in der Datei dtn in der Datenbasis dbn und hat die GV-Nr. g.v und das Passwort pw

opitionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: -

Einschränkung:

Wirkung:

(Aus historischen Gründen trägt diese Spezifikation den Namen DATEI, obwohl die zu zeichnende Information auch als Fremdstring eingegeben werden kann.)

Die folgenden Angaben sind möglich:

Dateibezeichnung : Die Beschreibung des gewünschten Bilds wird aus der Datei gelesen. Die Datei muß, falls sie nicht vom Satzbau W ist, geräteneutrale Information enthalten. Ist die Datei vom Satzbau W, wird sie ohne Berücksichtigung der Spezifikationen MODUS, BEREICH, PROTOKOLL, PARAMETER, STIFTE, DINAFORMAT und AUSSCHNITT direkt auf das Gerät ausgegeben. In diesem Fall muß die Datei gerätespezifische (!) Information enthalten.

/ Fremdstring : Die Bildbeschreibung wird direkt angegeben. Beispiel:

```
# ZEICHNE,KO,/clear
```

löscht die Graphik auf der Konsole.

KO : Nach dem ZEICHNE-Kommando werden wiederholt Eingaben graphischer Befehle (Konventionen vgl. Kap.3 des Graphik-Handbuchs) angefordert. Die eingegebenen Befehle werden sofort ausgeführt. Diese Angabe ist nützlich in der Verbindung OZEICHNE, KO, KO, zum interaktiven Arbeiten. Im folgenden kann man graphische Befehle eingegeben und deren Wirkung sofort erkennen.

formal:

<Wertzuweisung DATEI> ::= [DATEI=] { /<Fremdstring>[#/] }
 { <Dateibezeichnung> }
 KO

<Dateibezeichnung> ::= [<Katalogname>] <Dateiname> [- <Paßwort>]

<Dateiname> ::= <Name von Standardlänge> [(<Generationsnummer> .
 <Versionsnummer>)]

Beispiel:

#ZEICHNE, PL, STDPLD

Die in der Datei STDPLD enthaltene Information wird auf dem Plötter ausgegeben.

..., DATEI=PRIV.BILD(3.0),...

Die Information steht in der Datei BILD der Datenbasis PRIV mit der GV-Nr. 3.0.

#ZEICHNE, KO, KO, MODUS=TEKTRONIX

Die Information wird von der Konsole angefordert und auf dem Bildschirm nach Konventionen des TEKTRONIX-Sichtgeräts ausgegeben.

ANZAHL

Anzahl der Ausgabevorgänge

Spezifikationswert:

"undefiniert" : Die Zeichnung soll 1-mal ausgegeben werden.

n : Die Zeichnung soll n-mal ausgegeben werden.

opitonale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: 1

Einschränkung:

Ist GERÄT=<Dateibezeichnung> wird die Spezifikation ANZAHL nicht ausgewertet.

Wirkung:

In der Spezifikation ANZAHL läßt sich eine Zahl zwischen 1 und 16 angeben. Im Falle GERAET= <Geräteangabe> kann so eine Zeichnung ohne den für die Ausführung mehrerer ZEICHNE-Kommandos erforderlichen Mehraufwand an Rechenzeit bis zu 16-mal ausgegeben werden. Ist GERAET=KO oder GERAET=DRP vervielfacht sich der Zeitbedarf des ZEICHNE-Kommandos mit der gewünschten Zahl.

formal:

$$\langle \text{Wertzuweisung ANZAHL} \rangle ::= [\text{ANZAHL} =] \left\{ \begin{array}{l} \text{natürliche Zahl zwischen} \\ 1 \text{ und } 16 \end{array} \right\}$$

Beispiel:

..., ANZAHL=2, ...

MODUS

Angabe der Ausgabekonventionen

Spezifikationswert:

"undefiniert"	:	Die Ausgabekonventionen richten sich nach der Angabe unter GERAET.
Gerätetyp	:	ZCH123 DINA4F TEKTRONIX DRUCK
		Angabe, nach welchen Gerätekonventionen die Information umgeschlüsselt werden soll
Zusatzangabe	:	KURZ SYMBOLKURZ OHNE SYMBOL OHNE TEXT
		Beeinflußung der Zeichnung

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische Voreinstellung:"undefiniert"

Einschränkung:

Es darf höchstens ein Teilwert zum Gerätetyp vorhanden sein.

Wirkung:

Unter dieser Spezifikation lassen sich zwei gänzlich verschiedene Arten von Angaben machen:

1. Gerätetyp

In den meisten Fällen spezifiziert die Angabe unter GERAET zwar die "Adresse" nicht aber den Typ des gewünschten Ausgabegerätes. Zur Klarstellung ist hier eine zusätzliche Angabe zu machen, z.B.

#ZEICHNE, KO, BILDDATEI, MODUS=TEKTRONIX

falls es sich bei der benutzten Konsole um ein Datensichtgerät mit Bildspeicherröhre handelt.

Welche Angabe erforderlich ist, kann der ZEICHNE-Beschreibung des Graphik-Handbuchs entnommen werden.

2. Zusatzangabe

Das Aussehen der Zeichnung kann mit einer Reihe von Angaben beeinflußt werden. Mehrere Angaben dieser Art sind zulässig.

KURZ : Eine stark vergrößerte Version der Zeichnung wird ausgegeben. Wo "gekürzt" wird, richtet sich nach den Eigenschaften des gewünschten Gerätes.

SYMBOL KURZ : Falls das Gerät einen Zeichengenerator besitzt, werden Texte und Zentralsymbole mit diesem erzeugt und nicht durch einzelne Striche wiedergegeben - auch wenn dies nicht genau in der gewünschten Größe und Lage möglich ist.

OHNE SYMBOL : Alle Zeichensymbole (SYMBOL mit ICODE <0) werden weggelassen

OHNE TEXT : Alle Texte (SYMBOL mit NCHAR >0) werden weggelassen.

formal:

$\langle \text{Wertzuweisung MODUS} \rangle ::= [\text{MODUS} :=] \{ \langle \bar{m} \rangle \}$

$\langle m \rangle ::= \left\{ \langle \text{gerätetyp} \rangle [\langle \text{zusatzangabe} \rangle]^\infty \right\}$

$\langle \text{gerätetyp} \rangle ::= \text{ZCH123} \mid \text{DINA4F} \mid \text{TEKTRONIX} \mid \text{DRUCK}$

$\langle \text{zusatzangabe} \rangle ::= \text{KURZ} \mid \text{SYMBOLKURZ} \mid \text{OHNESYMBOL} \mid \text{OHNETEXT}$

Beispiel:

..., MODUS=TEKTRONIX'KURZ,...

Die Zeichnung wird nach TEKTRONIX-Konventionen umgeschlüsselt und dabei wegen der beschränkten Anzahl möglicher Vektoren auf das Wesentliche reduziert.

..., GERAET=KO, MODUS=DRUCK,...

Die Zeichnung wird auf dem Terminal als Druckinformation ausgegeben.

PROTOKOLL

Angaben zur Protokollierung

Spezifikationswert:

"undefiniert"	: Kein Protokoll, nur Fehlermeldungen.
-STD-	: Wichtige Daten über den Ausgabevorgang werden im Ablaufprotokoll festgehalten.
KO	: Protokollierung erfolgt zusätzlich auf Konsole.
A	: Im MODUS=DRUCK wird ein Adressbuch ausgegeben.
Z	: Im MODUS=DRUCK, GERAET=KO wird zusätzlich ins Ablaufprotokoll gezeichnet.
&Z	: Die Endmeldung des ZEICHNE-Kommandos mit Angabe der verbrauchten Rechenzeit unterbleibt. Dies ist nützlich, wenn das Kommando in einer Prozedur aufgerufen wird und "unsichtbar" bleiben soll.
KW	: Es werden keine Warnungen in die Zeichnung eingetragen.

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Die Angaben unter PROTOKOLL steuern die Protokollierung. Wird bei der Interpretation des geräteunabhängigen Bildinformation ein Fehler festgestellt, so wird gewöhnlich die Stelle, an der sich der Zeichenstift in diesem Augenblick befindet, in der Zeichnung durch

Fehler

markiert. Diese Markierung läßt sich durch Angabe von KW unterdrücken.

formal:

$$\begin{aligned} \langle \text{Wertzuweisung PROTOKOLL} \rangle & ::= [\text{PROTOKOLL} =] \left\{ \begin{array}{l} \text{-STD-} \\ \langle \text{Teilwert} \rangle [\langle \text{Teilwert} \rangle]^\infty \end{array} \right\} \\ \langle \text{Teilwert} \rangle & ::= \text{KO} | \text{A} | \text{Z} | \&\text{Z} | \text{KW} \end{aligned}$$

Beispiel:

..., PROT.=A'KO, ...

BEREICH

Angabe eines Dateibereichs

Spezifikationswert:

"undefiniert" : Die gesamte unter DATEI angegebene Information wird ausgewertet.

a : Der Satz mit der Nummer a wird ausgewertet.

a-b : Der Satzbereich a-b wird ausgewertet.

Mehrere Angaben sind durch Apostroph zu trennen.

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Eine Angabe zu BEREICH wird nicht ausgewertet, wenn unter DATEI ein Fremdstring oder KO angegeben wurde.

Wirkung:

Mit dieser Spezifikation lassen sich bei Bedarf Zeilenbereiche aus einer Datei, die eine Bildbeschreibung enthält, auswählen.

formal:

$$\langle \text{Vestruweisung BEREICH} \rangle ::= \left\{ \langle \text{Bereichsangabe} \rangle [\langle \text{Bereichsangabe} \rangle^*] \right\}$$
$$\langle \text{Bereichsangabe} \rangle ::= \left\{ \langle \text{natürliche Zahl} \rangle - \langle \text{natürliche Zahl} \rangle \right\}$$

Beispiel:

..., BEREICH=10'20'10'100-1000'999000-999999, ...

PARAMETER

Angaben zur Parametrisierung der Zeichnung

Spezifikationswert:

"undefiniert" : Es werden keine Parameter definiert.

Liste von durch Apostroph getrennten Zahlen $p_1'p_2'...$

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Enthält die graphische Information Bezüge auf p_i , so müssen alle Parameter $p_1'p_2'...'p_i$ angegeben werden. Eine Voreinstellung gilt nicht.

formal:

$$\langle \text{Wertzuweisung PARAMETER} \rangle ::= [\text{PARAMETER} =] \left\{ \begin{array}{l} - \\ \langle p \rangle [\langle p \rangle] \end{array} \right\}$$
$$\langle p \rangle ::= \left[\begin{array}{l} \{ + \} \\ \{ - \} \end{array} \right] [\langle \text{Ziffernfolge} \rangle] [\langle \text{Ziffernfolge} \rangle]$$

Beispiel:

..., PARAMETER=.5'13.7'-2'-15'-0.18,...

ZEICHNE

⑧ D I N A F O R M A T

D I N A F O R M A T

Angabe der gewünschten Größe der Zeichnung

Spezifikationswert:

"undefiniert"	Die gesamte Zeichenfläche kann verwendet werden.
n	Die Zeichnung wird im Format DIN A n quer ausgegeben.
-n	Die Zeichnung wird im Format DIN A n hoch ausgegeben.

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung: Wird unter dieser Spezifikation nichts angegeben, so steht die gesamte Fläche des Zeichengerätes zur Verfügung. Ihre Größe ist geräteabhängig (vgl. Kap. 1 des Graphik-Handbuches); sie reicht von einigen Quadratzentimetern bis in den Quadratmeterbereich (Papierplotter).

Zeichnungen genormter Größe lassen sich herstellen durch die Angabe

DINAFORMAT = n DIN A n quer oder

DINAFORMAT = -n DIN A n hoch

(z.B. DINA.=4 für DIN A 4 quer) - natürlich nur, wenn die Zeichenfläche des Gerätes hierfür groß genug ist.

ACHTUNG: Ob die Zeichnung vergrößert oder verkleinert oder aber am Rande des Formates einfach abgeschnitten wird, hängt von den unter AUSSCHNITT gemachten Angaben ab!

Eine Zeichnung kann aus mehreren Teilbildern bestehen. Die Fortschaltung zum nächsten Teilbild erfolgt mit NEXT picture oder CALL NEXT (1) (vgl. Kap. IVe des Graphik-Handbuches). Je nach Größe der gesamten Zeichenfläche werden dann die einzelnen Teilbilder so angeordnet, daß möglichst wenig verschwendet wird. Für n sind auch Zwischenwerte in Form von Dezimalbrüchen angebbar; ist DINAFORMAT=p, so berechnet sich das Format, wie folgt:

$$p > 0: y = 2 ** (-1/4 - p/2) * 100$$

$$x = y * \sqrt{2}$$

p < 0 : x und y vertauscht.

Der Inhalt einer Teilzeichnung wird durch AUSSCHNITT bestimmt.

formal:

$$\langle \text{Wertzuweisung DINAFORMAT} \rangle ::= [\text{DINAFORMAT=}] \left\{ \begin{array}{c} - \\ \langle p \rangle \end{array} \right\}$$
$$\langle p \rangle ::= [-] \langle \text{Ziffernfolge} \rangle [\langle \text{Ziffernfolge} \rangle]$$

Beispiel:

DINA.=4

Es wird im Format DIN A4 quer gezeichnet.

A U S S C H N I T T

Definition eines rechteckigen Ausschnitts

Spezifikationswert:

"undefiniert" Der linke untere Eckpunkt der Zeichenfläche entspricht dem Koordinatenursprung.

Liste von bis zu 4 Dezimalzahlen:

Die angegebenen Teilwerte werden als Koordinaten interpretiert und bestimmen den zu zeichnenden Ausschnitt des Bildes.

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Hier kann man angeben, welcher Teil der in der Bildbeschreibung definierten Zeichnung überhaupt ausgegeben werden soll: mit verschiedenen ZEICHNE-Aufrufen lassen sich nacheinander verschiedene Teile der Zeichnung erstellen. Im wesentlichen sind drei verschiedene Arten von Angaben möglich:

"undefiniert" : Der linke untere Eckpunkt der Zeichenfläche - bzw. des unter DINAFORMAT angegebenen Formates- entspricht den Koordinaten (0,0). Die Zeichnung wird in unveränderter Größe ausgegeben; was nicht in die Zeichenfläche oder ins gewünschte Format paßt, wird abgeschnitten.

x'y : Der linke untere Eckpunkt der Zeichenfläche oder des Formates entspricht den Koordinaten (x,y); die Zeichnung wird im Maßstab 1:1 ausgegeben und notfalls am Rande abgeschnitten.

x'y'x̄'ȳ : Der Maßstab wird so (in allen Richtungen gleich) verändert, daß der Ausschnitt der Zeichnung mit linkem unteren Eckpunkt (x,y) und rechtem oberen Eckpunkt (x̄,ȳ) gerade in das gewünschte Format paßt (in diesem Fall muß unter DINAFORMAT eine Angabe gemacht werden). Falls x,y angegeben sind, muß gelten x<x̄ und y<ȳ.

formal:

$$\langle \text{Wertzuweisung AUSSCHNITT} \rangle ::= [\text{AUSSCHNITT} =] \left\{ \begin{array}{l} \langle p \rangle \\ \langle p \rangle \langle p \rangle [\langle p \rangle \langle p \rangle] \end{array} \right\}$$
$$\langle p \rangle ::= \left[\begin{array}{c} + \\ - \end{array} \right] \langle \text{Ziffernfolge} \rangle [. \text{Ziffernfolge}]$$

Beispiel:

#ZEICHNE,...,DINA.=7,AUSSCHNITT=3'15

Der Teil der Zeichnung mit linkem unteren Eckpunkt (3,15) und rechtem oberen Eckpunkt (3 + Breite des Formates DIN A7, 15 + Höhe des Formates DIN A7) wird gezeichnet.

ZEICHNE

⑩ S T I F T E

S T I F T E

Angabe der gewünschten Zeichenstifte

Spezifikationswert:

"undefiniert" , Nur der Standard-Stift des betreffenden Gerätes ist verfügbar.

$p_1'p_2'p_3'p_4\dots$ Erfolgt in der graphischen Information ein Stiftwechsel zu Stift n , so wird der durch p_n definierte Stift benutzt. Ist $p_n=0$, so wird bis zum nächsten Stiftwechsel überhaupt nicht gezeichnet.

optionale Spezifikation zum Kommando ZEICHNE

anlagenspezifische
Voreinstellung: "undefiniert"

Einschränkung:

Wirkung:

Da jedes Gerät über eine andere Auswahl von Zeichenstiften verfügt, sind in der geräteunabhängigen Bildbeschreibung nur formale ("symbolische") Stiftangaben vorhanden.

durch PEN n

oder CALL PEN(n) (vgl. Kap IVd des Graphikhandbuches wird der symbolische Stift Nr. n ausgewählt. Jedem solchen n muß im ZEICHNE-Kommando ein "reeller", aktueller Stift zugeordnet werden:

#ZEICHNE,...,STIFTE=<aktueller Wert für PEN₁>'<aktueller Wert für PEN₂>'...

Dabei bedeuten bei Gerät = PL:

- Ø: Zeichnung wird unterdrückt,
- 1: Stift 1, schwarz
- 2: Stift 2, rot
- 3: Stift 3, grün

Am Tektronix-Sichtgerät so wie bei Modus=Druck gilt:

- Ø: Zeichnung wird unterdrückt
- 1: normale Zeichnung

(andere Angaben sind hier nicht sinnvoll)

Wichtig: Wird einem formalen Stift der aktuelle Stift Ø zugeordnet, so wird der betreffende Teil der Zeichnung gar nicht gezeichnet. Durch Zusammenspiel von NEWPEN in der Bildbeschreibung und STIFTE im ZEICHNE-Kommando lassen sich beliebige Teile der Zeichnung (z.B. Rahmen und sonstige Verzierungen) bei Bedarf unterdrücken.

Z E I C H N E / S T I F T E

formal:

$$\begin{array}{l} \langle \text{Wertzuweisung STIFTE} \rangle ::= . \quad [\text{STIFTE} =] \left\{ \begin{array}{l} - \\ \langle p \rangle ' [\langle p \rangle]^\infty \end{array} \right. \\ \langle p \rangle ::= \langle \text{natürliche Zahl} \rangle \end{array}$$

Beispiel:

STIFTE = 2'0'1'3

Bei Aufruf CALL NEWPEN(1) zeichnet Stift 2, bei Aufruf CALL NEWPEN(2) zeichnet Stift 0 (d.h. Linienunterdrückung), bei Aufruf CALL NEWPEN(3) zeichnet Stift1, bei Aufruf CALL NEWPEN(4) zeichnet Stift 3.

Die bisherigen Beispiele bezogen sich immer auf die Programmiersprache FORTRAN IV. Fuer die Benutzung anderer Programmiersprachen gelten natuerlich in Folge der anderen Sprachstruktur noch andere Regeln. Alles was bisher ueber die Grafik - UP's gesagt wurde, gilt auch fuer den Gebrauch von der Programmiersprache ALG60 aus. Zusaetzlich muessen in ALG60 alle benutzten UP's auch deklariert werden. Da alle Grafik - UP's entweder in FORTRAN oder nach FTN - Konventionen in TAS geschrieben sind, muessen sie als:

"PROCEDURE" <grafik - UP - Name> (<Parameterliste>); "FORTRAN";
deklariert werden. Dabei kann die Parameterliste entweder eine vollstaendige Parameterliste sein, sie kann aber auch unvollstaendig sein; muss aber mindestens einen Parameter enthalten, zum Beispiel:

```
"PROCEDURE" PLOT (X, Y, I); "FORTRAN";
```

oder:

```
"PROCEDURE" SYMBOL(X); "FORTRAN";
```

(der PROCEDURE - Rumpf entfaellt dann natuerlich).

Auf alle Faelle ist jedoch beim Aufruf darauf zu achten, dass die Typen der aktuellen Parameter mit den Typen uebereinstimmen, die von den Grafik - UP's formal erwartet werden. Dies gilt ganz besonders fuer Konstanten, die als aktuelle Parameter

in Grafik - UP Aufrufen vorkommen koennen. Ein Aufruf in der Art:

```
CALL PLOT (2.0, 1, 2)
```

fuehrt in aller Regel zu einem Typenkennungsalarm (und damit zu einem Programmabbruch) weil der zweite Parameter nach ALG60 - Konventionen INTEGER ist, vom UP PLOT jedoch an dieser Stelle ein Wert vom Typ REAL erwartet wird.

Achtung! Strings werden intern in ALG60 anders als in FTN dargestellt. Alle Strings, die an Grafik - UP's uebergeben werden sollen, muessen vorher gewandelt werden. Dazu dient das UP STRING. (STRING gehoert nicht zur eigentlichen Grafik - Software und wird deshalb auch nicht im Grafik Handbuch aufgefuehrt. Bei weitergehenden Fragen wende man sich an die Benutzerberatung des RZ.) Die Deklaration von STRING sieht folgendermassen aus:

```
"PROCEDURE" STRING(I, ALTEXT, FOTEXT); "VALUE" I;
```

```
"INTEGER" I; "ARRAY" ALTEXT, FOTEXT; "CODE";
```

Der Aufruf von STRING geschieht als Aufruf einer eidentlichen Procedure, z.B.:

```
STRING(1, "(OTTO)", FOTEXT);
```

Der aktuelle ALGOL-String "OTTO" wird nach FORTRAN - Konventionen in einem REAL ARRAY FOTEXT abgelegt, oder:

```
STRING(1, ALTEXT, FOTEXT);
```

Der in ARRAY ALTEXT nach ALGOL - Konventionen enthaltene ALGOL - String wird im ARRAY FOTEXT nach FORTRAN - Konventionen abgelegt. Oder aber:

```
STRING(2, ALTEXT, FOTEXT);
```


Die im ARRAY FOTEXT nach FORTRAN - Konventionen enthaltene Hollerrithkonstante wird im ARRAY ALTEXT nach ALGOL - Konventionen abgelegt. Die Anzahl der Zeichen pro Wort ist ja in FORTRAN vier, in ALGOL dagegen sechs. Fuer die Laenge der Felder ALTEXT und FOTEXT muessen deshalb folgende Bedingungen gelten:

(n = Zeichenzahl des Strings bzw. der Hollerithkonstante):

Laenge von ALTEXT > $1 + [(n + 5) / 6]$

Laenge von FOTEXT > $[(n + 3) / 4]$

Als Beispiele seien die bereits in FORTRAN aufgefuehrte Beispielprogramme hier noch einmal nach ALG60 "uebersetzt":

```

'BEGIN'
  'INTEGER' I;
  'REAL' x, DELTA;
  'ARRAY' XWERT, YWERT[1:52], FOTEXT[1:6];
  'PROCEDURE' STRING(I, A, B); 'CODE';
  'PROCEDURE' SCALE(X); 'FORTRAN';
  'PROCEDURE' AXIS(X); 'FORTRAN';
  'PROCEDURE' LINE(X); 'FORTRAN';
  DELTA:=(8.0*ARCTAN(1.0))/49.0;
  X=-DELTA
  'FOR' I:=1 'STEP' 1 'UNTIL' 50 'DO'
    'BEGIN'
      X=X+DELTA;
      XWERT[I]=X;
      YWERT[I]=SIN(X)
    'END'
  'END'

```

```

"END";
SCALE(XWERT, 15.0, 50, 1);
SCALE(YWERT, 10.0, 50, 1);
STRING(1, "(X-ACHSE F(X)=SIN(X))", FOTEXT);
AXIS(0.0, 5.0, FOTEXT, 19, 15.0, 0.0, XWERT[51], XWERT[52]);
STRING(1, "(Y-ACHSE)", FOTEXT);
AXIS(0.0, 0.0, FOTEXT, -7, 10.0, 90.0, YWERT[51], YWERT[52]);
LINE(XWERT, YWERT, 50, 1, 0, 2)
"END";

```

Der so erzeugte OUTPUT ist derselbe wie der auf Seite 25 dieser Einfuehrung. Nun noch das Beispiel von Seite 35:

```

#4XRA, BEN=...
<PASSWORD>
#UEBERSETZE, SPRACHE=ALG60, QUELLE=/
"REAL" "PROCEDURE" ALGSIN(X);
  "VALUE" X;
  "REAL" X;
  "BEGIN"
    ALGSIN:=SIN(X)
  "END";
#UEBERSETZE, SPRACHE=FTN, QUELLE=/
  REAL FUNCTION FTNSIN(X)
  ALG60EXTERNAL ALGSIN
  FTNSIN=ALGSIN(X)
  RETURN
  END

```

```

-----
#UEBERSETZE, SPRACHE=ALG60, QUELLE=/
"BEGIN"
  "REAL" "PROCEDURE" FTNSIN(X); "FORTRAN";
  "PROCEDURE" PFUNC(A, B, C, D, E, F, G); "FORTRAN";
  "PROCEDURE" STRING(I, A, B); "CODE";
  "ARRAY" TITEL, TEXT1, TEXT2[1:4];
  STRING(1, "(F(X) = SIN(X))", TITEL);
  STRING(1, "(X - ACHSE)", TEXT1);
  STRING(1, "(Y - ACHSE)", TEXT2);
  PFUNC(FTNSIN, 0.0, 8.0*ARCTAN(1.0), 002, TITEL, TEXT1, TEXT2)
"END";
#TDEKLARIERE, PLDATEI
#ERSTELLE, SYSTEM , GRAFIK
#MONTIERE
#STARTE, DATEI=87-PLDATEI
#ZEICHNE, GERAET=PL, DATEI=PLDATEI, MODUS=DINA4F
4XEN#.

```

```

-----
Einige Grafik - UPs koennen auch von der Programmiersprache
PL/1 aus aufgerufen werden. Hier treten aber besondere
Schwierigkeiten auf, weil die Sprachstrukturen von PL/1
und FORTRAN IV doch sehr unterschiedlich sind. Um diese
Sprachunterschiede zu ueberbruecken, wurde das UP S&PL1F
geschrieben, das es zumindest erlaubt, FORTRAN Unterprogramme

```

mit direkt kompatiblen Datentypen von PL/1 aus aufzurufen.

Funktionen koennen auf diese Art aber leider nicht uebergeben werden, so dass z.B. fast alle FUPLOT UPs nicht direkt von PL/1 aus aufgerufen werden koennen. (Abhilfe: Indirekter Aufruf ueber ein selbstgeschriebenes FORTRAN UP. Die von den FUPLOT UPs erwartete Funktion muss jedoch auch in diesem Fall ein "FORTRAN" EXTERNAL sein.) Hier also nun die Beschreibung von S&PL1F:

Anschluß von FORTRAN-Unterprogrammen an PL/1

1.) Anwendungszweck

Aufruf von FORTRAN-Subroutines und Functions von PL/1-Programmen aus; insbesondere die Benutzung der Programm-bibliotheken wie z.B.

PLOT (Plottersoftware FOPLOT/FUPLOT)
IMSL/NAG (numerische Bibliothek)

2.) Sprachschnittstelle

In die PL/1-Syntax ist eine FTN-Sprachschnittstelle integriert:

```
DCL FTNUP ENTRY(.....) OPTIONS(FTN);
```

```
CALL FTNUP(.....);
```

Zur Laufzeit wird das FORTRAN-Unterprogramm FTNUP nicht direkt angesprungen, sondern über eine Wandelroutine

```
S&PL1F
```

Dieses Unterprogramm wurde von CGK nicht geliefert. Die folgende Beschreibung bezieht sich auf eine im Rechenzentrum der Universität Osnabrück entwickelte Version des Unterprogramms S&PL1F.

3.) Leistungsumfang

Entsprechend dem Anwendungszweck wurde folgender Leistungsumfang realisiert:

a) Parameter nur auf Ganzwortgrenze

(PL/1: ALIGNED),

b) nur direkt kompatible Datentypen also als FUNCTION-Typ:

- INTEGER * 4 ~ REAL BIN FIXED(46)
- REAL * 4 ~ REAL BIN FLOAT(35)
- REAL * 8 ~ REAL BIN FLOAT(79)
- COMPLEX * 8 ~ COMPLEX BIN FLOAT(35)
- COMPLEX * 16 ~ COMPLEX BIN FLOAT(79)

als Parameter sind die oben angegebenen Typen zulässig, zusätzlich:

- FELDER der Datentypen mit
Dimension = 1 und untere Grenze = 1

- CHAR(*) VARYING ALIGNED

Achtung: Die Character werden auf ein REAL-Feld mit 4 Zeichen pro Wort (Typenkennung 0) umgespeichert.

Die Dimension errechnet sich aus

$\frac{\text{Anzahl Zeichen}}{4} + 1$. Das letzte Ganzwort wird mit Ignore aufgefüllt.

Es wird nicht zurückgespeichert!

- c) Die Verantwortung für die Einhaltung der Konventionen liegt beim Benutzer, die Schnittstelle überprüft lediglich a). Es wird also z.B. eine Variable vom Typ BIN FIXED(40,4) auf INTEGER * 4 abgebildet, was zu falschen Berechnungen führt!
- d) Der kompatible Aufbau von Feldern liegt ebenfalls in der Verantwortung des Benutzers. (evtl. Einsatz des Sprachmittels "i SUB" zum Vertauschen der Indices).

4.) Technische Spezifikationen

Die Schnittstelle leistet:

- a) Umstellen der Indexbasis
- b) Erstellen eines FTN-Versorgungsblockes
- c) Umstellen der Alarmadresse
(da sonst entsprechend der ON-Bedingungen gehandelt wird)
- d) Ansprung des FTN-UP

- e) Bei Funktionsaufrufen Rückspeicherung des Funktionswertes
- f) Umstellen auf die alte Alarmadresse
- g) Rücksprung in das PL/1-Programm oder definiertes Ende mit Fehlermeldungen

5.) Fehlermeldungen

Bei Fehlern in der Versorgung usw. wird eine Fehlermeldung mit Fehlernummer ausgegeben und dann über S&CC+8 der Operatorlauf beendet.

Wird das FTN-UP über die Fehleradresse (1.HW im Versorgungsblock) verlassen, wird die Kontrollblockadresse in RH entsprechend an S&CC+8 weitergeleitet.

Beim Alarm im FTN-UP wird über S&CC+4 der Rückverfolger und die Dumproutinen gestartet und dann über S&CC+8 der Operatorlauf beendet.

6. Weitere Entwicklung

Die nächste Version von S&PL1F soll folgende Leistungserweiterung enthalten:

- Übergabe von Feldern mit Dimension >1
- Übergabe und Rückspeichern von BIT(1) als LOGICAL * 4
- ev. Abprüfen der Precision (siehe 3c)
- Fehlermeldungen im Klartext
- Ablage von Felddesreibungen und Hilfszellen im PL/1-Stack

FEHLERNUMMERN:

a) Fehler im S&PL1F

- 202 HILFSPUFFER ZU KLEIN [BEI ZU VIEL FELDERN/STRINGS]
- 201 ADRESSRAUM ZU GROSS

b) Fehler in der Umgebung von S&PL1F/Benutzerfehler

- 106 MEHR ALS 32 PARAMETER
- 105 S&PL1F REKURSIV AUFGERUFEN
- 104 FUNKTION NICHT ERLAUBT (FEHLERHAFTES RETURNS)

- 103 KEINE DATENBESCHREIBUNG ABER PARAMETER
BZW. FUNCTION
- 102 AUFRUFART UNDEFINIERT
- 101 AUFRUF NICHT VON PL/1

c) Fehler in den übergebenen Parametern

- 3 UNTERE GRENZE BEI FELD NICHT 1
- 2 DIMENSION GROESSER 1
- 1 FALSCHER PARAMETERTYP

Es erscheint dann jeweils:

++++ FEHLER(S&PL1F): FEHLERNUMMER xxx

bei Fehlernummer <100 zusätzlich:

BEI PARAMETER: ***

Stichwortregister

Ablaufprotokoll	38	Datenpunkte	26
Achsen	14	Deklaration	48
Achsenanfangspunkt	23	DELTA V	23
Achsenbeschriftung	23	DEVICE	13,31
Achsenkreuz	14,19,22,23,33	DINA4F	44
Anfangskoordinate		DINAFORMAT	36,52
von SYMBOL und NUMBER	19	DRP	38
Anfangspunkt	14	DRUCK	44
ANGLE	18	Drucker	4,38
ANZAHL	36,42	ERSTELLE	13,36
Ausgabegeraet	10,44	EXTERNAL FUNKTION	33
AUSSCHNITT	36,54	Fach fuer Plotterauftraege	7
AUTOSC	14	FACTOR	17,18
AXIS	23,38	Farben	6
Basissoftware	13	FINIT	13
Benutzerprogramm	9	FIRSTV	23
Benutzerstring	6	Format	6
BEREICH	36,48	Fremdstring	40
Bereich	48	Funktionen	14,33
Beschriftung	13,14,18,33	Funktionsfamilien	33
Bildbeschriftung	14	FUPLOT - Programmpaket	13,14,33
Bildrahmen	14	gehobener Stift	15
BO&ZEICHNE	9	GERAET	36,38
breites Papier	6	Geraetebezeichnung	38
cm	17	geraeteneutrale	
COMPUM	14	Information	36,40
Datei	9,11	geraetespezifische	
DATEI	36,40	Information	10
Datenfluss	7	gesenkter Stift	15

Stichwortregister

Grafikdatei	31	Output - Anweisungen	9
Groesse	52	Papierbreite	6
gruen	6,28,29,56	Papiervorschub	30
HEIGHT	18	PARAMETER	36,50
INIT	13,31	Parameter	50
Kanalnummer	31	PFRAME	14
Kanalnummer 87	9	PL1	52
KO	38,40,46	PLOT&DRUCK	9
Konsolprotokoll	38	PLOT&TEKTRON	9
Koordinate	14	PLOT&ZCH	9
Koordinatenkreuz	13	PLOT	13,14
Koordinatenursprungspunkt	14	Plotter	4,38
Verschiebung	15	Plotzeit	17
Kreise	14	Polygonzug	22
Kurve	13	mehrere in einem Bild	26
langfristige Datei	11	PROTOKOLL	36,46
LAYOUT	33	Protokoll	46
leerfahrten	17	PSB	12
LINE	13	Rahmen	14,16,33
Linien, gestrichelte	13	RASTER	14
Linienraster	14	rot	6,28,29,56
Massstabsfaktor	13,17	S&PL1F	52
MODUS	36,44	Satzbau	11,31
NEWPEN	13,27,28	Satzzahl	11
NEXT	52	SCALE	13,23
NUMBER	13,18,21	Scalierung	14
Optimierung		Schlussaufruf	32
Plotzeit	17	schmales Papier	6
Rechenzeit, Speicherplatz	13	Schnittlinie	6

Stichwortregister

<p>Schriftgrundlinie 19</p> <p>schwarz 6,28,29,56</p> <p>SELFSC 14</p> <p>sequentielle Datei 11</p> <p>Sichtgeraet 4</p> <p>SIG100 7</p> <p>SIG50 38</p> <p>Skalierungsparameter zu LINE 23</p> <p>Speicherplatz 31</p> <p>Spezialsymbol 19,20,26</p> <p>Standardstift 6,27,28</p> <p>Stift 6,13,27,56</p> <p>STIFTE 36,56</p> <p>Stiftposition 14,17</p> <p>Stiftzuordnung 29</p> <p>Strich 5,14</p> <p>Strich in X - Richtung 5</p> <p>Strich in Y - Richtung 5</p> <p>STRING (UP) 49</p> <p>Strings, wandeln von 49</p> <p>SYMBOL 13,18</p> <p>Teilauftrag 9</p> <p>Teile von Zeichnungen 54</p> <p>Teilzeichnungen 29</p> <p>TEKTRONIX 44</p> <p>Tektronix Sichtgeraet 7</p> <p>temporaere Datei 11</p> <p>Text 18</p> <p>TEXT1 33</p>	<p>TEXT2 33</p> <p>Texthaltungskommandos 11</p> <p>TITEL 33</p> <p>TSB 12</p> <p>Typ 48</p> <p>Unterdrueckung von Teilzeichnungen 56</p> <p>Ursprungsverschiebung 13</p> <p>Voreinstellungen in FUPLOT 14</p> <p>WHERE 17</p> <p>Winkel 18</p> <p>X - Achse 23</p> <p>Y - Achse 23</p> <p>Zahlen, zeichnen von 21</p> <p>ZCH123 44</p> <p>Zeichenflaeche, neue 13</p> <p>Zeichenflaeche 16</p> <p>Zeichenkette 21</p> <p>Zeichnungen, mehrere 30</p> <p>Zuwachs pro cm Achse 23</p>
--	---