

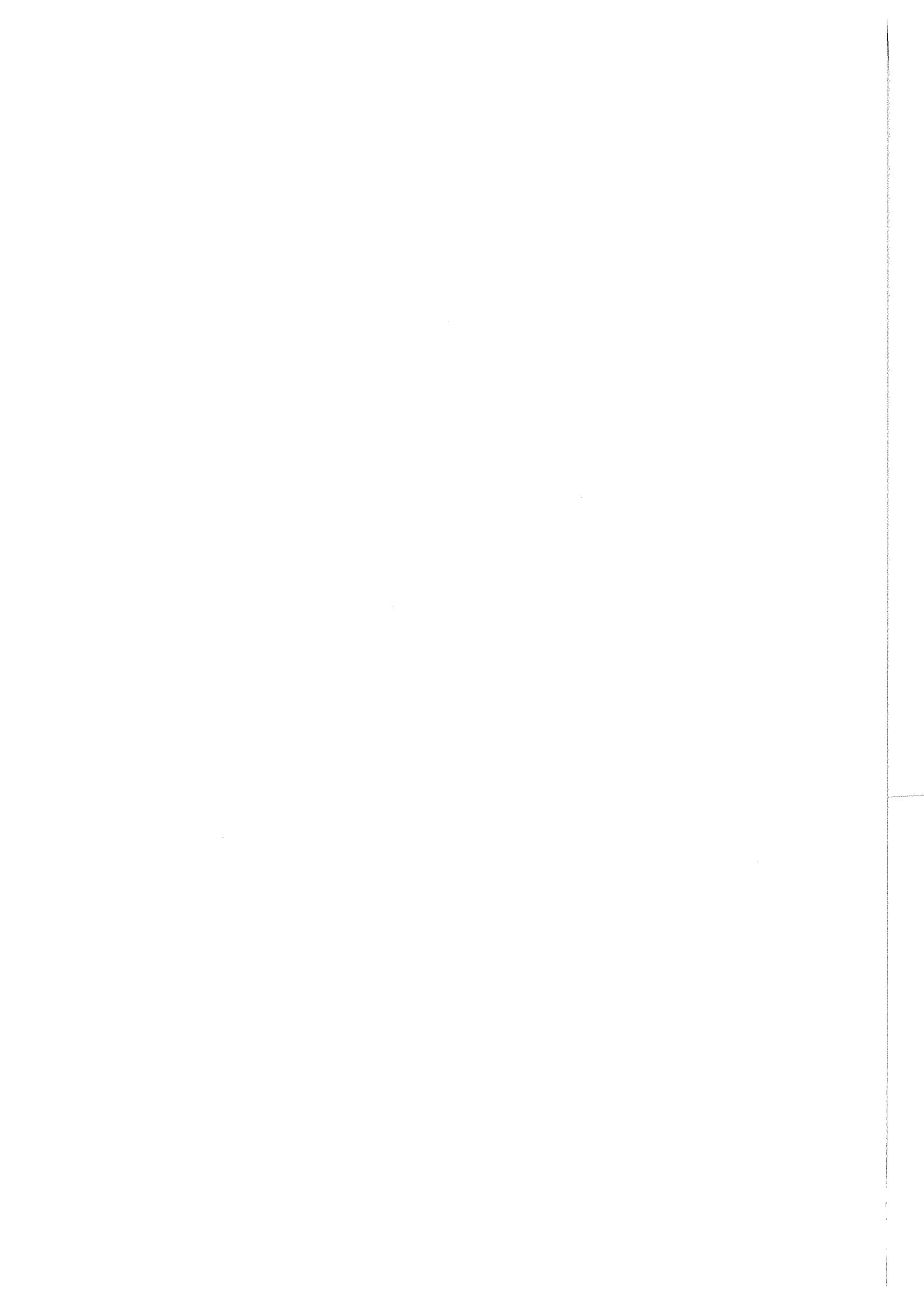
KfK 4496
Januar 1989

TRACE/PC

**Graphik-Software für IBM-PCs und
IBM-kompatible PCs**

M. Politzky
Institut für Reaktorbauelemente

Kernforschungszentrum Karlsruhe



KERNFORSCHUNGSZENTRUM KARLSRUHE
Institut für Reaktorbauelemente
KfK 4496

TRACE/PC
Graphik-Software
für IBM-PCs und IBM-kompatible PCs

M. Politzky

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

TRACE/PC

Graphik-Software für IBM-PCs und IBM-kompatible PCs

Zusammenfassung

TRACE/PC ist ein Programmsystem für die Entwicklung anwendungsorientierter Graphiksoftware auf IBM-PCs und IBM-kompatiblen PCs.

Der Inhalt des Systems besteht überwiegend aus funktionaler Software. Diese bedient sich dabei eines Basis-Systems (GSW-Grafik Software), das nicht Bestandteil von TRACE/PC ist.

Der modulare Aufbau von TRACE/PC bietet dem Anwender in leicht verständlicher Form Möglichkeiten, Graphiken irgendwelcher Art auf weitgehend individuelle Weise zu erstellen.

Neben einer detaillierten Modulbeschreibung wird auch auf die Installation und die Handhabung des Programmsystems eingegangen.

TRACE/PC

Graphic Software for IBM PCs and IBM Compatible PCs

Summary

TRACE/PC is a program system for the development of user oriented graphic software on IBM PCs and IBM compatible PCs.

The system consists mainly of functional software which makes use of a basic system (GSW Grafik Software) not part of TRACE/PC.

The modular structure of TRACE/PC in an easily understandable way enables the user to plot largely individually every kind of graph.

A detailed description of the moduls is given, and, in addition, installation and handling of the program system are reported.

Inhaltsverzeichnis

		Kopfzeile	Seite
		TRACE/PC V.2.0 - ...	
1.	Einleitung	Einleitung	1
2.	Installation von TRACE/PC	Installation	3
3.	Initialisierung einer Zeichnung	PLOTI	5
4.	Wahl der Zeichenfarbe	PEN	11
5.	Zeichnen eines Rahmens	CADRE/CADRE2	13
6.	Änderung der Zeichenposition	PLOTTP	15
7.	Aktuelle Koordinaten und Darstellungs- faktor	WHEREP	17
8.	Zeichnen eines Rahmens (Fenster) in ein kartesisches Koordinatensystem	AXYFEN/AXYRES ..	19
9.	Zeichnen einer linearen X-Achse	AXLIN1	21
10.	Zeichnen einer logarithmischen X-Achse	AXLOG1	25
11.	Zeichnen einer linearen Y-Achse	AYLIN1	29
12.	Zeichnen einer logarithmischen Y-Achse	AYLOG1	33
13.	Zeichnen eines Kurvenverlaufs in ein karte- sisches Koordinatensystem ..	CURVE	37
14.	Zeichnen eines Kurvenverlaufs in ein kar- tesisches Koordinatensystem unter Kontrolle von Bereichsüberschreitungen ...	CURVE2	39
15.	Konstruktion eines Polarkoordinatensystems	POL	43
16.	Zeichnen eines Kurvenverlaufs in ein Polar- koordinatensystem ..	CURVEP	47
17.	Zeichnen eines Kurvenverlaufs in ein Polar- koordinatensystem unter Kontrolle von Bereichsüberschreitungen ..	CURVP2	49
18.	Zeichnen eines Streckenzuges	CURVLV	53
19.	Zeichnen von Textstrings und Symbolen	SYMBP1-4/SYMBBS1 .	55
20.	Zeichnen von Gleitkommazahlen	NUMBP/NUMBS	59
	(Ziffernstrings)		
21.	Verwendung von Sonderschriftarten	FNTX	61

TRACE/PC V.2.0 - Inhalt

	Kopfzeile	Seite
	TRACE/PC V.2.0 - ...	
22. Zeichnerische Wiedergabe von Texten in Normal- und Schrägschrift	TEXT/TEXTS	65
23. Zeichnen des KfK-Symbols	KFK	69
24. Zeichnen von Kreisen oder Kreisteilen	CERCLE	71
25. Zeichnen von Ellipsen oder Ellipsenteilen	ELLIPS	73
26. Zeichnen von Geraden mit unterscheidbarer Liniencharakteristik	STRIP	75
27. Zeichnen von Linien und Pfeilen	FLECHE	77
28. Zeichnen von schraffierten oder ungeschraffierten Flächen (Balken)	BALKEN	79
29. Zeichnen eines Rahmens mit integriertem Fenster	SFCAD	81
30. Zeichnen eines cm- bzw. Zoll-Rasters zu Orientierungszwecken	RASTER	83
31. Zeichnerische Darstellung von Symbolen aus dem DV- bzw. PC-Hardware-Bereich	DV-/PC-Symbole	85
31.1 Zeichnen eines Magnetplattensymbols (Fixed Disk)	DISY	86
31.2 Zeichnen eines Datenspeichersymbols (Tape, Cartridge)	TASY	87
31.3 Zeichnen eines Bildschirmsymbols (Screen)	ECRAN	88
31.4 Zeichnen eines Zeichengerätsymbols (Plotter)	GRAPH	89
31.5 Zeichnen eines Druckersymbols (Printer)	IMPRIM	90
31.6 Zeichnen eines Tastatursymbols (Konsole)	CONSOL	91
31.7 Zeichnen eines Symbols für eine Systemeinheit (PC-Unit)	SYSUNI	92
32. Bestimmung geeigneter Parameter zur Definition von linearen Achsen	ROUND	93
33. Perspektivische Darstellung einer Funktion in Abhängigkeit zweier Variablen	Extern	95
34. Visualisierung eines Bildfile's	Bildfile	97

	Kopfzeile TRACE/PC V.2.0 - ...	Seite
35. Reservierte COMMON's	COMMON's	99
36. Hilfsroutinen	Hilfsroutinen	103
37. Literatur	Literatur	105
38. Programmbeispiele	Beispiele	107
39. Alphabetische Übersicht der Graphik-Module	Modul-Übersicht	129

1. Einleitung

Mit TRACE/PC wird ein Softwareprodukt für graphische Applikationen auf IBM-PCs und IBM-kompatiblen PCs vorgestellt. Es handelt sich hierbei um ein modular aufgebautes System, dessen Inhalt im wesentlichen aus funktionalen Unterprogrammen besteht. Diese können aus einem Anwendungsprogramm durch Aufruf (CALL) angesprochen werden, womit die direkte Visualisierung gerechneter oder anderweitig erfaßter Daten ermöglicht wird. Als Programmiersprache für die Software wurde Fortran 77 in der RM/FORTRAN™-Version 2.10 verwendet.

Ein allgemeiner Bedarf an einer auf PCs verfügbaren Grafiksoftware, die dem Anwender weitgehende individuelle Freiheit bei der Gestaltung grafischer Aufgaben bietet, führte zur Entwicklung des Programmsystems. Als Muster und Orientierungshilfe diente dabei die im KfK an der IBM-Großrechenanlage installierte Zeichensoftware TRACEGS [1]. Es wurde dabei ein Höchstmaß an Kompatibilität zwischen den beiden Softwareprodukten angestrebt. Mit einer nahezu identischen Benutzeroberfläche kann der Anwender relativ problemlos TRACEGS-Programme in die TRACE/PC-Umgebung übertragen, da die verwendeten Graphik-Module namentlich und funktionell übereinstimmen. Eine Ausnahme bildet in diesem Zusammenhang die Initialisierungsroutine PLOTI. Sie muß in jedem Fall an die entsprechende Softwareumgebung angepaßt werden.

In der Anwendung wird von TRACE/PC lediglich der graphisch-funktionale Bereich abgedeckt. Die dazu erforderliche Basis liefert das Softwareprodukt GSW [2], das darüber hinaus auch die Schnittstelle zur Calcomp-Software bietet.

Das gesamte Softwarepaket (TRACE/PC und GSW) umfaßt 3 Bibliotheken, eine Interrupt Service Routine für die Bildschirmausgabe, einen Treiber zur Visualisierung von Bildfile's und 15 Direct-Access-Files (3 File's je Schriftart) für die Sonderschriftarten.

Bei der Systeminstallation ist hinsichtlich der verwendeten Compiler sorgfältig auf die Verträglichkeit der Bibliotheken sowohl untereinander als auch in Verbindung mit dem Anwenderprogramm zu achten.

Es kann im allgemeinen davon ausgegangen werden, daß für alle gängigen Fortran-Compiler (RM/FORTRAN™, MicroSoft®) eine geeignete Softwarekonfiguration installierbar ist.

Um diesem Bericht den Charakter eines Nachschlagewerks bzw. Handbuchs zu geben, werden unter den folgenden Punkten die Graphikmodule einzeln in detaillierter Weise beschrieben.

TRACE/PC V.2.0 - Einleitung

Die im Text erwähnten Warenzeichen oder Kurzbezeichnungen sind folgenden Firmen bzw. Herstellern zuzuordnen:

Calcomp	California Computer Products, Inc., U.S.A.
HP	Hewlett Packard Company, U.S.A.
IBM	International Business Machines Corporation, U.S.A.
MicroSoft	MicroSoft Corporation, U.S.A.
NEC	NEC Corporation, Japan
PEGA	Paradise Systems, Inc., U.S.A.
PLINK86	Phoenix Software Associated Ltd., U.S.A.
RM/FORTRAN	Ryan-McFarland Corporation, U.S.A.

2. Installation von TRACE/PC

Bevor eine graphische Applikation mit TRACE/PC-Call's ausgeführt werden kann, müssen alle notwendigen Bibliotheken auf dem PC resident sein. Eine Installation auf der Festplatte ist hierfür zu empfehlen.

Stellvertretend für eine Vielzahl von unterschiedlichen Hard- und Softwarekonfigurationen wird eine Möglichkeit zur Installation des graphischen Systems vorgeschlagen.

Folgende Hard- und Softwareausstattung liegt dabei zugrunde:

- I) Hardware:
- IBM PC/AT3, 512kB Memory, math. Co-Prozessor/80287 FPP, 30MB-Festplatte
 - US MF II - Tastatur
 - Color Monitor NEC Multisync JC-1401P3ED
 - PEGA-Adapter (640×350/480 Bildpunkte)
 - Farbstrahlprinter IBM 3852 2 (für "Print Screen")
 - 8-Farben-Stift-Plotter HP-ColorPro
- II) Software:
- DOS 3.2 oder höher.
 - RM/FORTRAN - Compiler einschließlich der Runtime-Libraries als Numerical Data Processor Version (NDP).
 - GSW [2] - Grafiksoftware-Bibliotheken (16-Bit-Integer!) "GSW0.LIB" und "GSW1.LIB" einschließlich der Interrupt Service Routine "EGAIN.T.COM" für die direkte Darstellung einer Graphik auf dem Bildschirm ("On-Line") sowie eines Bildschirmtreibers "PLTDRV.SYS" zur Visualisierung von Plotfiles.
 - TRACE/PC-Bibliothek (16-Bit-Integer!) "TPC.LIB".
 - TRACE/PC-Spezial-Schriftarten:
"FTIn.DAT", "FTXn.DAT", "FTYn.DAT", n = 1 - 5.

Es ist ratsam unter DOS für alle Graphik-File's ein eigenes Subdirectory anzulegen. In dieses werden dann die Bibliotheken von GSW und TRACE/PC, die Spezial-schriftarten, der Bildschirmtreiber und die Interrupt Service Routine kopiert. Die RM/FORTRAN-Bibliotheken sollten zur sachlichen Trennung der Programmprodukte in einem separaten Subdirectory residieren. Dies ist kein Umstand, da über eine geeignete Pfaddefinition der Zugriff auf alle RM/FORTRAN-File's grundsätzlich möglich ist. Um die Konsistenz mit den Bibliotheksprogrammen sicherzustellen, müssen bei zu übersetzenden Programmen die Compileroptionen "/iy"

TRACE/PC V.2.0 - Installation

angegeben werden. Auch sei an dieser Stelle explizit erwähnt, daß der Zugriff auf die RMFORT.LIB-Bibliothek, sollte sie in einem Subdirectory residieren, nur möglich ist, wenn dies durch die Anweisung "set obj=\path lib" bekannt gemacht wurde (z. B. beim Systemstart im AUTOEXEC.BAT-File).

Dem ungeübten Anwender wird nachstehend eine Prozedur angeboten, mit der ein Fortran-Programm übersetzt, gelinkt und ausgeführt wird (Code and Go-Procedure). Die Prozedur (.bat-File) befindet sich dabei in dem gleichen Subdirectory, in dem sich auch die Graphik-Bibliotheken befinden.

```
\path\rmfort %1.for /iy
\path\plink86 fi %1.obj search tpc, gsw0, gsw1 verbose
%1
```

Anmerkung: In vorstehender Prozedur ist "\path" durch den Namen des Subdirectory's zu ersetzen, in dem sich die RM/FORTRAN-Bibliotheken befinden.

Für die Ausgabe ("On-Line") einer Graphik auf dem Bildschirm muß die Interrupt Service Routine EGAIN.T.COM durch Aufruf von "egaint" einmal für die gesamte Einschaltdauer des PCs aktiviert werden. Um dem Vergessen vorzubeugen, wird empfohlen, die Initialisierung beim Systemstart im AUTOEXEC.BAT-File ("path\egaint") vorzunehmen.

Bei Verwendung eines On-Line-Zeichengerätes ist es erforderlich, vor der Anwendung einmal das Schnittstellenprotokoll zu setzen. Auch hier kann dies unmittelbar vor Gebrauch oder besser schon beim Systemstart (AUTOEXEC.BAT) erfolgen. Die erforderliche Anweisung für den hier verwendeten Plotter (HP-ColorPro) lautet: "mode com1:9600,n,8,1,p".

Sollen Graphikfile's auf dem Bildschirm dargestellt werden, muß zusätzlich zur Interrupt Service Routine der Bildschirmtreiber "PLTDRV.SYS" installiert sein. Die geschieht in geeigneter Weise im CONFIG.SYS-File des DOS-Betriebssystems durch Angabe von "device=path\pltdrv.sys". Weitere Angaben hierzu sind unter dem Punkt "Visualisierung eines gespeicherten Bildes" zu finden.

Hinweis: Darüber hinausgehende und auch detailliertere Informationen zur Softwareinstallation, insbesondere auch hinsichtlich anderer Systemkonfigurationen, sind der GSW-Benutzeranleitung zu entnehmen.

3. Initialisierung einer Zeichnung

Die Initialisierung einer Zeichnung erfolgt mit Hilfe des Unterprogramms **PLOTI**. Der Aufruf von **PLOTI** ist obligatorisch, da hierdurch Default-Werte und andere relevante Parameter für den folgenden Programmablauf gesetzt werden. Danach können dann andere Graphik-Module in beliebiger aber logischer Reihenfolge aufgerufen werden. Unterbleibt die Initialisierung einer Zeichnung durch **PLOTI**, führen Graphik-Call's in der Regel zu unkontrollierten Programmabläufen.

PLOTI ist im wesentlichen identisch mit der GSW-Basisroutine **PLOTS** [2]. Über zwei weitere Argumente in **PLOTI** werden dem Anwender Möglichkeiten geboten, den Eröffnungsaufwand für eine Zeichnung durch Standardisierung weitgehend zu reduzieren.

Nach der Initialisierung ist die linke untere Ecke der Zeichenebene als Startbezugspunkt definiert.

Die Argumentliste von **PLOTI** enthält 5 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL PLOTI (IDEV,IC,IUN,IFORM,INST)

Argumentbeschreibung:

IDEV	(-)	<p>Angabe des Gerätes, auf dem die Visualisierung der graphischen Information erfolgen soll.</p> <p>Der Variablen IDEV kann sowohl eine "0" (Integer-Null) als auch eine Zeichenkette (CHARACTER-Variable, Literal) zugewiesen werden.</p> <p>Folgende, für TRACE/PC relevante Optionen sind wählbar:</p> <p>IDEV = 0:</p> <p style="padding-left: 40px;">Ausgabe der Bildinformation auf dem Default-Gerät (im allgemeinen der Bildschirm).</p> <p>= 'AUX' oder 'COM1':</p> <p style="padding-left: 40px;">Die graphische Ausgabe erfolgt auf einem Stiftplotter über die serielle Schnittstelle (On-Line-Mode).</p>
-------------	-----	--

= 'dsname.ext':

Die Bildinformation soll in das spezifizierte Bildfile geschrieben werden (Off-Line-Mode). Die Disposition des File's kann dabei 'NEW' oder 'OLD' sein. Für die Extension "ext" sollte aus praktischen Gründen "PIC" verwendet werden.

Die Revisualisierung eines Bildfile's auf einem Plotter (serielle Schnittstelle) kann mit Hilfe der DOS-Kommando-Sequenz

```
copy dsname.PIC aux
```

erfolgen.

Für die Bildwiederholung auf einem Bildschirm ist ein Treiber (GSW) erforderlich. Siehe hierzu auch den Punkt "System-Installation".

- IC** (-) Für den Fall, daß **IDEV** eine Zeichenkette zugewiesen wurde, gibt **IC** die Anzahl der darin enthaltenen Zeichen an.
Bei **IDEV** = 0 ist auch **IC** = 0 zu setzen.
- IUN** (-) Die Funktion der Variablen **IUN** hängt von dem gewählten Bildausgabegerät ab (siehe **IDEV**).
Bei Bildschirmausgabe (**IDEV** = 0) enthält der **IUN** zugewiesene Wert den Code für die Bildschirm-Hintergrundfarbe. Es gilt die Farbzusordnungstabelle wie sie bei der Beschreibung des Unterprogramms **PEN** angegeben ist.
Bei der Bildfilegenerierung stellt die Variable **IUN** eine Schreibeinheit (Unit) dar, über die das Programm mit dem File-Namen korrespondieren soll. Als Unit kommen die Ziffern 1-9 in Betracht, sofern sie nicht schon anderweitig für Ein- bzw. Ausgabeoperationen verwendet werden.
Ist die Ausgabe der Bildinformation lediglich für einen On-Line-Plotter bestimmt, sollte **IUN**

ebenfalls eine der Ziffern 1-9 zugewiesen werden. Der Zahlenwert selbst ist in der TRACE/PC-Umgebung bedeutungslos.

IFORM (-)

Multifunktionsvariable, mit der das Zeichnungs-Layout beeinflusst werden kann (Standardisierung). Auch der Aufwand für die Zeichnungseröffnung läßt sich damit reduzieren.

IFORM ist wie folgt zu bestimmen:

$$\text{IFORM} = \text{IF} + 100 * \text{IC} + 10000 * \text{IRK}$$

Erläuterung der Komponenten:

IF Zeichnungsformat. Unter folgenden Optionen kann gewählt werden:

30 oder 31 (DIN-A3 quer/hoch)

40 oder 41 (DIN-A4 quer/hoch)

IC Farbe, in der die Zeichnungsumrandung (Rahmen) und/oder das KfK-Symbol (s. Komponente IRK) gezeichnet werden soll. Es gilt die Farbzuordnungstabelle wie sie bei der Beschreibung zur "Wahl der Zeichenfarbe" angegeben ist. Die hier gewählte Farbe bleibt solange gültig, bis sie durch einen Aufruf von PEN geändert wird.

IRK Mit dieser Komponenten wird festgelegt, ob die Zeichnung mit einer Umrandung und/oder dem KfK-Symbol versehen werden soll. Unter nachstehenden Möglichkeiten kann gewählt werden:

0: Weder Rahmen noch KfK-Symbol

1: nur KfK-Symbol

2: nur Rahmen

3: Rahmen und KfK-Symbol

Ist die vorgestellte Form der Standardisierung nicht erwünscht, wird IFORM = 0 gesetzt. In diesem Fall gelten die GSW-[2]-Konventionen.

Siehe auch die untenstehenden Anmerkungen zu diesem Argument.

INST (-) Textvariable oder Literal (CHARACTER*4).
Sofern die Darstellung des KfK-Symbols mit **IFORM** festgelegt wurde, bietet INST die Möglichkeit, dieses darüber hinaus mit einer Bezeichnung (z. B. Institutsname) zu versehen.

Anmerkungen zum Argument IFORM.

Eine von GSW [2] vorgegebene Randbedingung (Einschränkung) besteht darin, daß auf den Zeichengeräten (Stiftplotter), in Bezug auf die genormten Formate (A3/A4), jeweils nur ein reduzierter Zeichenbereich zur Verfügung steht. Die Software gestattet es nicht, den Zeichenstift bis an die Randzonen der Zeichenfläche zu führen. Der Grund hierfür besteht wohl darin, ein Verhaken bzw. Verklemmen des Zeichenstiftes mit dem Zeichnungsträger (Papier/Transparent) zu vermeiden, was in der Nähe der Ränder leicht möglich ist. Eine 1:1-Abbildung des gesamten Inhalts einer A4- bzw. A3-Fläche auf dem entsprechenden Zeichengerät ist somit nicht realisierbar.

Da sich in der Regel der Aufbau einer Graphik auf einer genormten Zeichenfläche vollzieht, wurde versucht, dieser Gegebenheit in TRACE/PC in geeigneter Weise Rechnung zu tragen:

Durch entsprechende Definition des Arguments **IFORM** (>0) wird der gesamte Inhalt eines genormten Format's auf der reduzierten Zeichenfläche durch Verkleinerung der Bildgegenstände konform abgebildet.

Der Verkleinerungsfaktor ist abhängig von dem verwendeten Zeichengerät. Er kann vom Anwender mit nachstehender Anweisungsfolge erfragt werden:

```
C ... ROUTINE ZUR BESTIMMUNG DES VERKLEINERUNGSFAKTORS
      CHARACTER*4 CC(3), A3
      REAL GP(3)
      DATA A3/'A3  '/
      CALL PLOTS('AUX',3,1)
      CALL PAPIER(GP)
      CALL PAPIER(CC)
C ... A4/PR(ColorPro, US A)-ZEICHENGERÄT
      XL=29.7
      YL=21.0
C ... A3-ZEICHENGERÄT
      IF(CC(1).EQ.A3) THEN
                XL=42.0
                YL=29.7
```



```

END IF
FACTX=GP(2)/XL
FACTY=GP(3)/YL
FACT=FACTX
IF(FACTY.LT.FACTX) FACT=FACTY
CALL PLOT(0.,0.,999)
WRITE(*,(' VERWENDBARER FAKTOR FÜR DEN ',A4,' PLOTTER --->',
+      F6.3)') CC(1), FACT
STOP
END

```

Für die Ausgabe der graphischen Information auf einem Bildschirm gilt ähnliches.

Soll eine Graphik im Maßstab 1:1 abgebildet werden, ist die Variable IFORM = 0 zu setzen. Dem Anwender muß dabei lediglich bewußt sein, daß nicht die gesamte Zeichenfläche nutzbar ist. Die Größe der zur Verfügung stehenden Zeichenfläche ist aus den entsprechenden Verkleinerungsfaktoren zu bestimmen.

Verwendete Unterprogramme und Common's:

GSW_____: PAPIER, PLOT, PLOTS, FACTOR, SETFMT, WHERE
TRACE/PC__: CADRE, FORM, KFK, PEN, PLOTX
COMMON__: DDD, FAKTOR, FNSTR, CENTER

4. Wahl der Zeichenfarbe

Mit Hilfe des Unterprogramms **PEN** kann der Anwender auf farbgraphikfähigen Anlagen die Zeichenfarbe bestimmen.

Die Argumentliste von **PEN** enthält ein Element, das beim Aufruf initialisiert sein muß.

Aufruf:

CALL PEN (IPEN)

Das Argument **IPEN** stellt dabei den Zuordnungscode zu der umseitig angegebenen Farbtafel dar.

Nach dem Aufruf von **PEN** werden alle folgenden graphischen Darstellungen in der gewählten Farbe gezeichnet und zwar so lange, bis ein weiterer Aufruf von **PEN** mit einer anderen Zeichenfarbe erfolgt.

Hierzu einige Hinweise:

Zu den Geräten mit farbgraphischen Eigenschaften zählen insbesondere die Mehrfarbenzeichengeräte (Mehrfarbenstift-Plotter) und die Farbbildschirmgeräte mit ihren entsprechenden Adapterkarten (VEGA, PEGA, Quad EGA usw.). Darüberhinaus gibt es noch eine Reihe von Farbgraphikdruckern, mit denen per "Print Screen" die Bildschirmgraphik auf Papier oder Transparent kopiert werden kann.

Bei Verwendung von Mehrfarbenstift-Plottern wird darauf hingewiesen, daß das Argument **IPEN** auch zur Auswahl von Stiften verwendet werden kann, die sich nicht nur in der Farbe sondern auch in der Strichstärke unterscheiden. Die z. Zt. erhältlichen Strichstärken betragen 0.3 und 0.7 mm. Um eine Vieldeutigkeit des Zuordnungscode zu vermeiden, muß die Stiftzuordnung für diesen Fall individuell vom Anwender festgelegt werden.

Bei den Farbbildschirmgeräten sind unter den verschiedenen Fabrikaten häufig Unterschiede in der verfügbaren Farbskala, der Farbqualität, und der Farbintensität zu finden. Daher ist es nicht möglich, an dieser Stelle verbindliche Angaben zur Farbzuordnung zu machen.

Um dem Anwender jedoch Anhaltswerte für die Farbzuordnung zu geben, wird umseitig eine Tabelle angeboten, die die Korrespondenz von Farbcode zu dem verwendeten Zeichengerät herstellt. Es wurden hierzu die z. Zt. gebräuchlichsten Bildausgabegeräte ausgewählt.

Farbzuordnungstabelle zum Unterprogramm PEN

Farbcode IPEN	Plotter (HP-Color Pro)	Farbbildschirm EGA-Adapter (NEC-Multisync)	Farbstrahlprinter (IBM 3852 2)
1	schwarz	schwarz	schwarz
2	rot	rot	rot
3	blau	blau (mittel)	blau (mittel)
4	gold	gelb	gelb
5	grün	grün	grün
6	violett	violett	violett (magenta)
7	türkis	türkis	blau (hell)
8	orange	braun	orange (rötlich)
9	schwarz	grau	weiß
10	schwarz	braun	orange (rötlich)
11	schwarz	violett	violett (magenta)
12	schwarz	braun	orange (rötlich)
13	schwarz	türkis	blau (hell)
14	schwarz	grün	grün
15	schwarz	blau (dunkel)	blau (dunkel)
16	schwarz	schwarz	schwarz
17	schwarz	weiß	weiß

Verwendete Unterprogramme und COMMON's:

GSW_____: NEWPEN

TRACE/PC__: keine

COMMON__: DDD, ORIGIN

5. Zeichnen eines Rahmens

Mit den Unterprogrammen CADRE und CADRE2 lassen sich auf der Zeichenebene Rahmen unterschiedlicher Größe darstellen.

Der Unterschied zwischen CADRE und CADRE2 besteht darin, daß CADRE einen eckigen Rahmen und CADRE2 einen Rahmen mit abgerundeten Ecken zeichnet.

Die Ausdehnung des Rahmens in horizontaler und in vertikaler Richtung wird in der Argumentliste angegeben.

Die linke untere Rahmenecke ist mit dem zuletzt definierten Bezugspunkt (CALL PLOTP(...,-2/-3)) identisch. Bei CADRE2 ist es der Schnittpunkt der Verlängerungen von linker und unterer Rahmenseite.

Die Argumentlisten von CADRE und CADRE2 enthalten jeweils 2 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL CADRE (XDIR, YDIR)

oder

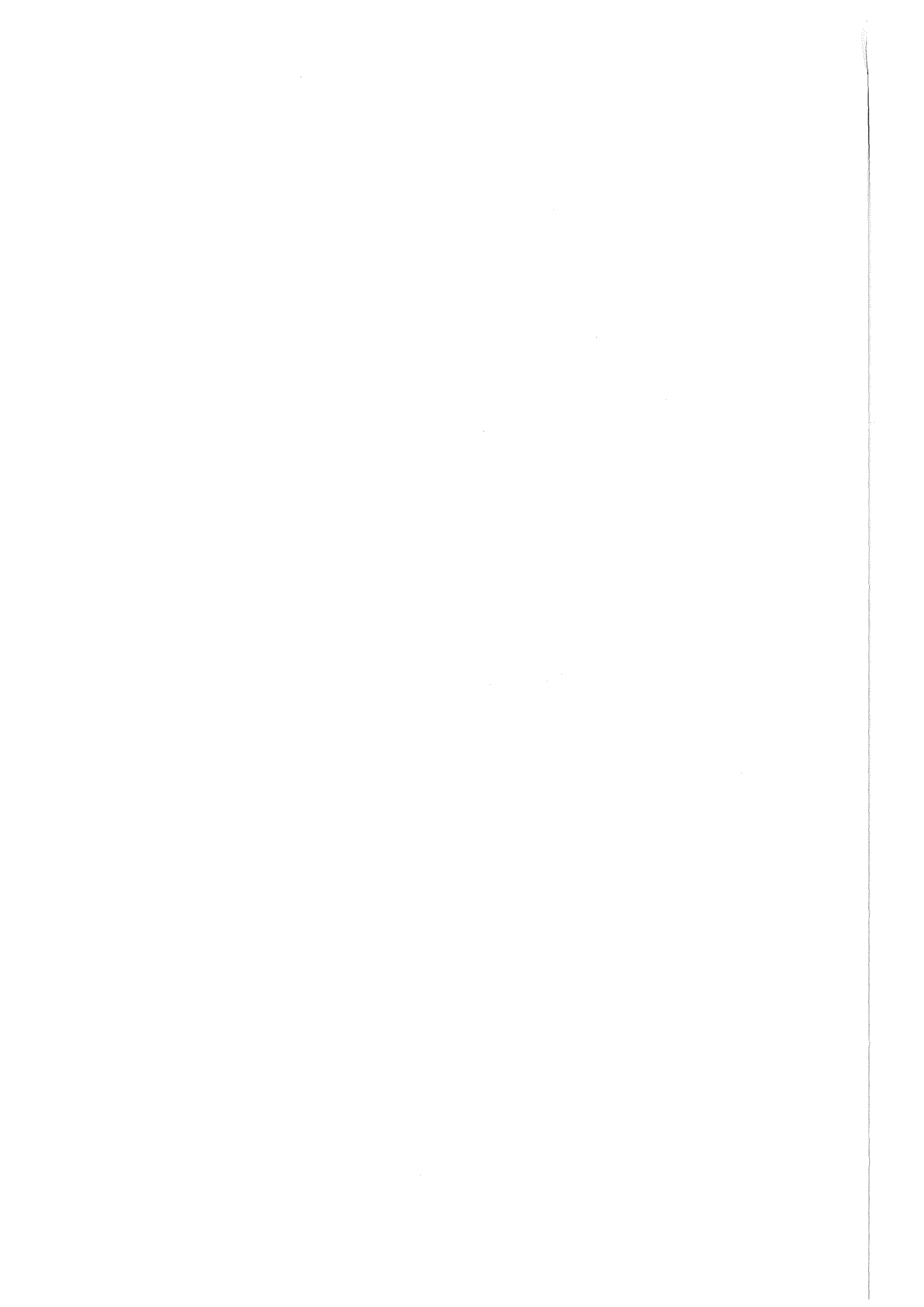
CALL CADRE2 (XDIR, YDIR)

Argumentbeschreibung:

XDIR	(cm)	Ausdehnung des Rahmens in horizontaler Richtung.
YDIR	(cm)	Ausdehnung des Rahmens in vertikaler Richtung.

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT
 TRACE/PC__: keine
 COMMON__: keine



6. Änderung der Zeichenposition

Die Anwendung des Unterprogramms PLOTP ist im allgemeinen eine Anweisung an das Zeichenmedium, z. B. der Zeichenstift eines Plotters, eine vorgegebene Zeichenposition zu erreichen. Dabei wird das Ziel mit angehobenem (inaktiv) oder abgesenktem (aktiv) Zeichenstift geradlinig angefahren. Darüber hinaus können die Zielkoordinaten über eine entsprechende Option zum aktuellen Bezugspunkt deklariert werden.

Mit der Anweisung "CALL PLOTP (.....,999)" wird eine Zeichnung abgeschlossen. Dieser Aufruf ist obligatorisch. Er bewirkt das ordnungsgemäße Zurücksetzen relevanter Zeichenparameter.

PLOTP in TRACE/PC ist lediglich ein Synonym der GSW-Basisroutine PLOT [2]. Damit soll die Kompatibilität mit der namens- und funktionsgleichen TRACEGS-Routine [1] gewährleistet werden.

Aus Gründen der Vollständigkeit in der Modulbeschreibung werden die Argumente von PLOTP auch an dieser Stelle detailliert beschrieben.

Die Argumentliste von PLOTP enthält 3 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL PLOTP (XK,YK,IP)

Argumentbeschreibung:

XK	(cm)	Zielkoordinaten in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
IP	(-)	Steuervariable. Mit ihr wird die Stellung des Zeichenstiftes (aktiv oder inaktiv) bei der Positionsänderung bestimmt. Das Negieren des IP zugewiesenen Wertes bewirkt, daß der Zielpunkt zum aktuellen Bezugspunkt wird.

In einzelnen gilt:

IP = 2 Der Schreibstift ist aktiv. Die aktuelle Position und der Zielpunkt (XK, YK) werden mit einer sichtbaren Linie verbunden.

- = 3 Der Schreibstift ist inaktiv. Er bewegt sich in abgehobener Stellung zum Zielpunkt.
- = -2 Gleiche Bedeutung wie bei IP=2 bzw.
- = -3 IP=3. Die Zielkoordinaten werden jedoch zum neuen Bezugspunkt für noch folgende Zeichenaktivitäten.
- = 999 Abschluß einer Zeichnung (obligatorisch).

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT
TRACE/PC__: keine
COMMON__: keine

7. Aktuelle Koordinaten und Darstellungsfaktor

Das Unterprogramm **WHEREP** liefert nach dessen Aufruf Informationen über die aktuelle Position des Zeichenstiftes und den Wert des gerade gültigen Darstellungsfaktors.

Die Anwendung von **WHEREP** ist insbesondere beim Fortsetzen von Ziffern- und Textstrings hilfreich, da hierbei die umständliche Bestimmung der Anschlußkoordinaten entfällt. Aber auch überall dort, wo graphische Objekte in Relation zu irgendwelchen nicht vorherbestimmbaren Koordinaten auf der Zeichenebene dargestellt werden sollen, ist die Verwendung von **WHEREP** zu empfehlen.

WHEREP ist in **TRACE/PC** ein Synonym der GSW-Basisroutine **WHERE** [2]. Damit ist die Kompatibilität mit der namens- und funktionsgleichen **TRACEGS**-Routine [1] sichergestellt.

Aus Gründen der Vollständigkeit in der Modulbeschreibung werden die Argumente von **WHEREP** auch hier beschrieben.

Die Argumentliste von **WHEREP** enthält 3 Elemente, die ausnahmslos Ausgabeparameter darstellen. Sie müssen daher in Form von Variablen in der Argumentliste erscheinen.

Aufruf:

CALL WHEREP (XK,YK,IFAC)

Argumentbeschreibung:

XK	(cm)	Rückgabeparameter.
YK	(cm)	Zeichnungskoordinaten in Relation zum aktuellen Bezugspunkt.
IFAC	(-)	Rückgabeparameter, der den zum Zeitpunkt der Abfrage gültigen Darstellungsfaktor repräsentiert.

TRACE/PC V.2.0 - WHEREP

Verwendete Unterprogramme und COMMON's:

GSW _____: WHERE

TRACE/PC __: keine

COMMON __: keine

8. Zeichnen eines Rahmens (Fenster) in ein kartesisches Koordinatensystem

Mit Hilfe des Unterprogramms AXYFEN kann im allgemeinen Sinn ein Rahmen mit wählbarer Strichcharakteristik an beliebiger Stelle auf der Zeichenebene dargestellt werden.

Im besonderen wurde dieser Modul jedoch zur Initialisierung, Definition und Realisierung eines Fensters innerhalb eines kartesischen Koordinatensystems entwickelt. Die Initialisierung und Definition wird, sofern später aufgerufen, den Achsenprogrammen AXLIN1 bzw. AXLOG1 und AYLIN1 bzw. AYLOG1 über den Common /FNSTR/ mitgeteilt. Damit wird sichergestellt, daß im Fall eines gewünschten Rasters im Koordinatensystem das Fenster von den Rasterlinien befreit bleibt (siehe hierzu auch die referierten Achsenprogramme).

Aber auch ohne Raster kann ein Rahmen mit AXYFEN, welches lediglich zum Hervorheben eines Textes dienen soll, an beliebiger Stelle im Koordinatensystemfenster plaziert werden. Bei diesem Anwendungsfall sollte jedoch überlegt werden, ob es nicht sinnvoller wäre, die Unterprogramme CADRE oder CADRE2 zur Realisierung heranzuziehen, mit der Einschränkung, daß der Rahmen nur als Vollinie dargestellt werden kann.

Um einen unkontrollierten Programmablauf zu vermeiden ist das Zurücksetzen der Fensteroption mit dem Aufruf des Moduls AXYRES zu empfehlen.

Unter den Beispielen im Anhang findet sich hierfür ein Anwendungsfall.

Die Argumentliste von AXYFEN enthält 5 Elemente, die beim Aufruf initialisiert sein müssen. AXYRES hat keine Argumentliste.

Aufruf:

Initialisierung, Definition und Realisierung:

```
CALL AXYFEN (X1,Y1,X2,Y2,IST)
```

Zurücksetzen auf Normalfunktion:

```
CALL AXYRES
```

Argumentbeschreibung:

X1,Y1	(cm)	Koordinaten der linken unteren Fensterecke, bezogen auf den aktuellen Bezugspunkt (in der Regel der Koordinatenursprung).
-------	------	---

TRACE/PC V.2.0 - AXYFEN/AXYRES

X2,Y2 (cm) Koordinaten der rechten oberen Fensterecke, bezogen auf den aktuellen Bezugspunkt (in der Regel der Koordinatenursprung).

IST (-) Steuervariable. Mit ihr entscheidet der Anwender ob der Rahmen des Fensters gestrichelt oder als Vollinie dargestellt werden soll. Es gilt folgendes:

IST = 0 Rahmen als Vollinie
≠ 0 gestrichelter Rahmen

Verwendete Unterprogramme und COMMON's:

GSW_____: keine
TRACE/PC__: nur AXYFEN: PLOTP, STRIP
COMMON__: FNSTR

9. Zeichnen einer linearen X-Achse

Das Unterprogramm AXLIN1 ermöglicht die Konstruktion einer linearen X-Achse (Abszisse). Die Lage der Achse auf dem Zeichnungsträger ist abhängig von dem verwendeten Gerätetyp (Bildschirm, Plotter) und dem Ausgabeformat (hoch od. quer). Mit den Eingangsargumenten hat der Anwender die Möglichkeit auf individuelle Weise die Achse zu gestalten. Dabei wird den Anforderungen an die Repräsentationsgraphik weitgehend Rechnung getragen.

Die Argumentliste von AXLIN1 enthält 13 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

```
CALL AXLIN1 (XL,XMIN,XMAX,DX,XA,YL,IKM,HKM,HGM,ZSH,IGEN,IT,TEXT)
```

Argumentbeschreibung:

XL	(cm)	Länge der X-Achse.
XMIN	(phys. Einh.)	Kleinster Wert an der X-Achse (linkes Ende). Er muß nicht zwangsläufig als Graduation erscheinen, da dies auch von den für DX und XA zugewiesenen Werten abhängt.
XMAX	(phys. Einh.)	Größter Wert an der X-Achse (rechtes Ende). Auch hier hängt das Zeichnen der entsprechenden Graduation von den für DX und XA zugewiesenen Werten ab.
DX	(phys. Einh.)	Der Wert von DX gibt an, nach wieviel physikalischen Einheiten jeweils ein großer Teilstrich mit der entsprechenden Graduation gezeichnet werden soll (Differenz zwischen zwei benachbarten Graduationen).
XA	(phys. Einh.)	XA repräsentiert eine auf der X-Achse darzustellende Graduation. Die weiteren Graduationen wer-

den programmintern gemäß den **DX** entsprechenden Abständen ermittelt.

Dabei gilt: $XMIN \leq XA \leq XMAX$.

Es wird empfohlen für **XA** einen Wert zu wählen, der mit einem großen Teilstrich an der Achse übereinstimmt.

YL	(cm)	<p>Multifunktionsvariable. Folgende Wertzuweisungen sind sinnvoll:</p> <p>YL = Länge der Y-Achse (Ordinate). Es wird eine parallel zur X-Achse gespiegelte Achse im Abstand YL mit Teilstrichen aber ohne Graduationen gezeichnet (Rahmeneffekt). Sind zusätzlich Rasterlinien erwünscht, werden bei positivem YL diese durchgezogen, bei negativem YL gestrichelt dargestellt. Für diesen Fall muß HKM und/oder HGM der Wert 0. zugewiesen werden.</p> <p>YL = 0. Das Zeichnen einer gespiegelten Achse entfällt.</p>
IKM	(-)	Anzahl der kleinen Teilstriche zwischen zwei Graduationen.
HKM	(cm)	Länge der kleinen Teilstriche. Falls Rasterlinien gezeichnet werden sollen, ist HKM = 0. zu setzen!
HGM	(cm)	Länge der großen Teilstriche. Falls Rasterlinien gezeichnet werden sollen, ist HGM = 0. zu setzen!
ZSH	(cm)	Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text, Graduationen).

Darüberhinaus entscheidet das Vorzeichen von ZSH über die Lage von Graduationen, Achsenbeschriftung und Teilstriche zur Achse. Ist ZSH positiv, werden Graduationen und Achsentext unter und die Teilstriche über die Achse gezeichnet. Bei negativem ZSH ist es umgekehrt.

Anmerkung: Unabhängig von dem für YL eingegebenen Wert wird bei negativem ZSH keine gespiegelte Achse gezeichnet.

IGEN (-)

Anzahl der Nachkommastellen für die zu zeichnenden Graduationen. Bei der Darstellung der Graduationen ohne Dezimalexponenten bedeutet

IGEN = -1: Graduationen ohne Dezimalpunkt.
 = 0: Graduationen nur mit Dezimalpunkt.
 = 1: Graduationen mit einer Nachkommastelle.

usw.

Für die Darstellung der Graduationen mit Dezimalexponenten 'E' gilt:

$$[-11] \geq \text{IGEN} \geq [-14]$$

Es bedeutet z. B.:

IGEN = -11: Eine signifikante Nachkommastelle: 0.3E+03
 = -13: Drei signifikante Nachkommastellen: 0.123E-05

IT (-)

Anzahl der Textzeichen für die Achsenbeschriftung. $IT_{\max} = 80$.

Bei $IT = 0$ entfällt die Beschriftung der Achse.

TEXT (-)

Text für die Achsenbeschriftung. Die Zeichenkette kann maximal 80 Zeichen beinhalten. Sie wird als vordefinierte CHARACTER-Variable (DATA,

READ) oder direkt als Literal (eingeschlossen in Hochkommas) angegeben.

Die Anordnung des Textes erfolgt rechtsbündig und parallel zur Achse.

AXLIN1 kommuniziert über verschiedene benannte COMMON's mit den Modulen CURVE, CURVE2, AXYFEN und AXYRES.

Folgende relevante Größen werden in AXLIN1 entweder definiert und übergeben oder empfangen:

a.) Weitergabe von Informationen an CURVE und CURVE2 über den COMMON /FAKTOR/.

1. Kleinster Wert an der X-Achse [XMIN]
2. Maßstabsfaktor [$XI/(XMAX-XMIN)$]
3. Mitteilung, daß es sich hier um eine lineare X-Achse handelt.

b.) Weitergabe von Informationen an CURVE2 über den COMMON /UGOG/.

- Feldbegrenzungen XMIN, XMAX

c.) Kommunikation mit AXYFEN über den COMMON/FNSTR/.

- AXLIN1 erhält über den COMMON/FNSTR/ die Information, ob durch einen vor AXLIN1 ausgeführten Aufruf von AXYFEN ein Fenster (Rahmen) im Koordinatenraum etabliert wurde. Dies ist hier insofern von Bedeutung, da beim eventuellen Zeichnen von Rasterlinien das Fenster von diesen ausgespart bleiben soll.

Siehe hierzu auch die Beschreibung der Unterprogramme AXYFEN und AXYRES.

Verwendete Unterprogramme und COMMON's:

GSW_____ : NUMBER, PLOT, SYMBOL
TRACE/PC__ : EXPD, NETLIN, NETLN2
COMMON___ : FAKTOR, FNSTR, UGOG

10. Zeichnen einer logarithmischen X-Achse

Das Unterprogramm AXLOG1 ermöglicht die Konstruktion einer logarithmischen X-Achse (Abszisse).

Hinsichtlich der Lage der Achse auf dem Zeichnungsträger und deren Gestaltungsmöglichkeiten im Bereich der Repräsentationsgraphik gilt das gleiche wie für das Unterprogramm AXLIN1 beschrieben.

Die Argumentliste von AXLOG1 enthält 11 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL AXLOG1 (XL,XMIN,XMAX,IM,YL,IKM,HKM,HGM,ZSH,IT,TEXT)

Argumentbeschreibung:

XL	(cm)	Länge der X-Achse.
XMIN	(phys. Einh.)	Kleinsten Wert an der X-Achse (linkes Ende). Entspricht XMIN <u>nicht</u> einer Zehnerpotenz oder einer ganzzahligen Fraktion einer Zehnerpotenz (2,3, ...,9*10 ⁿ), wird XMIN nicht als Graduation dargestellt.
XMAX	(phys. Einh.)	Größter Wert an der X-Achse (rechtes Ende). Hinsichtlich der Darstellung von XMAX als Graduation gilt das gleiche wie zuvor bei XMIN beschrieben.
IM	(-)	Zwischen den Dekaden können 8 Teilstriche gezeichnet werden, von 2*10 ⁿ bis 9*10 ⁿ . Siehe hierzu auch Argument IKM . Es gibt 4 Möglichkeiten, um an die Teilstriche auch die entsprechenden Graduationen zu zeichnen: IM = 0 Es werden keine Zwischengraduationen gezeichnet.

- = 1 An die Teilstriche 2, 4 und $7 \cdot 10^n$ werden die entsprechenden Graduationen gezeichnet.
- = 2 An die Teilstriche 2, 3, 5 und $8 \cdot 10^n$ werden die entsprechenden Graduationen gezeichnet.
- = 3 Es werden an alle Teilstriche Graduationen gezeichnet.

YL	(cm)	<p>Multifunktionsvariable. Folgende Wertzuweisungen sind sinnvoll:</p> <p>YL = Länge der Y-Achse (Ordinate). Es wird eine parallel zur X-Achse gespiegelte Achse im Abstand YL ohne Graduationen jedoch mit Teilstrichen gezeichnet (Rahmeneffekt). Sind zusätzlich Rasterlinien erwünscht, werden bei positivem YL diese durchgezogen, bei negativem YL gestrichelt dargestellt. Für diesen Fall muß HKM und/oder HGM der Wert 0. zugewiesen werden.</p> <p>YL = 0. Das Zeichnen einer gespiegelten Achse entfällt.</p>
IKM	(-)	<p>Steuervariable.</p> <p>IKM \leq 0 Zwischen den Dekaden werden weder Teilstriche noch Graduationen gezeichnet.</p> <p>IKM $>$ 0 Zwischen den Dekaden werden 8 Teilstriche ($2 \cdot 10^n$ bis $9 \cdot 10^n$) und in Abhängigkeit des Arguments IM die entsprechenden Graduationen gezeichnet.</p>
HKM	(cm)	<p>Länge der kleinen Teilstriche. Falls Rasterlinien gezeichnet werden sollen, ist HKM = 0. zu setzen!</p>

HGM	(cm)	Länge der großen Teilstriche. Falls Rasterlinien gezeichnet werden sollen, ist HGM = 0. zu setzen!
ZSH	(cm)	Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text, Graduationen). Darüberhinaus entscheidet das Vorzeichen von ZSH über die Lage von Graduationen, Achsenbeschriftung und Teilstriche zur Achse. Ist ZSH positiv, werden Graduationen und Achsentext unter und die Teilstriche über die Achse gezeichnet. Bei negativem ZSH ist es umgekehrt. <u>Anmerkung:</u> Unabhängig von dem für YL eingegebenen Wert wird bei negativem ZSH <u>keine</u> gespiegelte Achse gezeichnet.
IT	(-)	Anzahl der Textzeichen für die Achsenbeschriftung. $IT_{\max} = 80$. Bei IT = 0 entfällt die Beschriftung der Achse.
TEXT	(-)	Text für die Achsenbeschriftung. Die Zeichenkette kann maximal 80 Zeichen beinhalten. Sie wird als vordefinierte CHARACTER-Variable (DATA, READ) oder direkt als Literal (eingeschlossen in Hochkommata) angegeben. Die Anordnung des Textes erfolgt rechtsbündig und parallel zur Achse.

AXLOG1 kommuniziert über verschiedene benannte COMMON's mit den Modulen CURVE, CURVE2, AXYFEN und AXYRES.

Folgende relevante Größen werden in AXLOG1 entweder definiert und übergeben oder empfangen:

- a.) Weitergabe von Informationen an CURVE und CURVE2 über den COMMON /FAKTOR/.
 - 1. Kleinster Wert an der X-Achse [XMIN]
 - 2. Maßstabsfaktor [$XL/\lg(XMAX/XMIN)$]
 - 3. Mitteilung, daß es sich hier um eine logarithmische X-Achse handelt.

- b.) Weitergabe von Informationen an CURVE2 über den COMMON /UGOG/.
 - Feldbegrenzungen XMIN, XMAX

- c.) Kommunikation mit AXYFEN über den COMMON/FNSTR/.
 - AXLOG1 erhält über den COMMON/FNSTR/ die Information, ob durch einen vor AXLOG1 ausgeführten Aufruf von AXYFEN ein Fenster (Rahmen) im Koordinatenraum etabliert wurde. Dies ist hier insofern von Bedeutung, da beim eventuellen Zeichnen von Rasterlinien das Fenster von diesen ausgespart bleiben soll.
Siehe hierzu auch die Beschreibung der Unterprogramme AXYFEN und AXYRES.

Verwendete Unterprogramme und COMMON's:

GSW _____ : NUMBER, PLOT, SYMBOL
TRACE/PC __ : NETLIN, NETLN2
COMMON ___ : FAKTOR, FNSTR, UGOG

11 . Zeichnen einer linearen Y-Achse

Das Unterprogramm AYLIN1 ermöglicht die Konstruktion einer linearen Y-Achse (Ordinate).

Hinsichtlich der Lage der Achse auf dem Zeichnungsträger und deren Gestaltungsmöglichkeiten im Bereich der Repräsentationsgraphik gilt das gleiche wie für das Unterprogramm AXLIN1 beschrieben.

Die Argumentliste von AYLIN1 enthält 13 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL AYLIN1 (YL,YMIN,YMAX,DY,YA,XL,IKM,HKM,HGM,ZSH,IGEN,IT,TEXT)

Argumentbeschreibung:

YL	(cm)	Länge der Y-Achse.
YMIN	(phys. Einh.)	Kleinster Wert an der Y-Achse (unteres Ende). Er muß nicht zwangsläufig als Graduation erscheinen, da dies auch von den für DY und YA zugewiesenen Werten abhängt.
YMAX	(phys. Einh.)	Größter Wert an der Y-Achse (oberes Ende). Auch hier hängt das Zeichnen der entsprechenden Graduation von den für DY und YA zugewiesenen Werten ab.
DY	(phys. Einh.)	Der Wert von DY gibt an, nach wieviel physikalischen Einheiten jeweils ein großer Teilstrich mit der entsprechenden Graduation gezeichnet werden soll (Differenz zwischen zwei benachbarten Graduationen).
YA	(phys. Einh.)	YA repräsentiert eine auf der Y-Achse darzustellende Graduation. Die weiteren Graduationen werden programmintern gemäß den DY entsprechenden Abständen ermittelt.

Dabei gilt: $YMIN \leq YA \leq YMAX$.

Es wird empfohlen für YA einen Wert zu wählen, der mit einem großen Teilstrich an der Achse übereinstimmt.

XL	(cm)	<p>Multifunktionsvariable. Folgende Wertzuweisungen sind sinnvoll:</p> <p>XL = Länge der X-Achse (Abszisse).</p> <p>Es wird eine parallel zur Y-Achse gespiegelte Achse im Abstand XL mit Teilstrichen aber ohne Graduationen gezeichnet (Rahmeneffekt).</p> <p>Sind zusätzlich Rasterlinien erwünscht, werden bei positivem XL diese durchgezogen, bei negativem XL gestrichelt dargestellt. Für diesen Fall muß HKM und/oder HGM der Wert 0. zugewiesen werden.</p> <p>XL = 0.</p> <p>Das Zeichnen einer gespiegelten Achse entfällt.</p>
IKM	(-)	Anzahl der kleinen Teilstriche zwischen zwei Graduationen.
HKM	(cm)	<p>Länge der kleinen Teilstriche.</p> <p>Falls Rasterlinien gezeichnet werden sollen, ist HKM = 0. zu setzen!</p>
HGM	(cm)	<p>Länge der großen Teilstriche.</p> <p>Falls Rasterlinien gezeichnet werden sollen, ist HGM = 0. zu setzen!</p>
ZSH	(cm)	<p>Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text, Graduationen).</p> <p>Darüberhinaus entscheidet das Vorzeichen von ZSH über die Lage von Graduationen, Achsen-</p>

beschriftung und Teilstriche zur Achse. Ist ZSH positiv, werden Teilstriche und Achsentext rechts und die Graduationen links von der Achse gezeichnet. Bei negativem ZSH ist es umgekehrt.

Anmerkung: Unabhängig von dem für XL eingegebenen Wert wird bei negativem ZSH keine gespiegelte Achse gezeichnet.

IGEN (-) Anzahl der Nachkommastellen für die zu zeichnenden Graduationen. Bei der Darstellung der Graduationen ohne Dezimalexponenten bedeutet

IGEN = -1: Graduationen ohne Dezimalpunkt.
 = 0: Graduationen nur mit Dezimalpunkt.
 = 1: Graduationen mit einer Nachkommastelle.

usw.

Für die Darstellung der Graduationen mit Dezimalexponenten 'E' gilt:

$$[-11] > = \text{IGEN} > = [-14]$$

Es bedeutet z. B.:

IGEN = -11: Eine signifikante Nachkommastelle: 0,3E+03
 = -13: Drei signifikante Nachkommastellen: 0,123E-05

IT (-) Der Betrag von IT gibt die Anzahl der Textzeichen für die Achsenbeschriftung an. $IT_{\max} = 80$.
 Ist IT positiv erscheint die Beschriftung im rechten Winkel, bei negativem IT parallel zur Y-Achse.
 Bei IT = 0 entfällt die Beschriftung.

TEXT (-) Text für die Achsenbeschriftung. Die Zeichenkette kann maximal 80 Zeichen beinhalten. Sie wird als vordefinierte CHARACTER-Variable (DATA,

READ) oder direkt als Literal (eingeschlossen in Hochkommas) angegeben.

AYLIN1 kommuniziert über verschiedene benannte COMMON's mit den Modulen CURVE, CURVE2, AXYFEN und AXYRES.

Folgende relevante Größen werden in AYLIN1 entweder definiert und übergeben oder empfangen:

- a.) Weitergabe von Informationen an CURVE und CURVE2 über den COMMON /FAKTOR/.
 - 1. Kleinster Wert an der Y-Achse [YMIN]
 - 2. Maßstabsfaktor [YL/(YMAX-YMIN)]
 - 3. Mitteilung, daß es sich hier um eine lineare Y-Achse handelt.

- b.) Weitergabe von Informationen an CURVE2 über den COMMON /UGOG/.
 - Feldbegrenzungen YMIN, YMAX

- c.) Kommunikation mit AXYFEN über den COMMON/FNSTR/.
 - AYLIN1 erhält über den COMMON/FNSTR/ die Information, ob durch einen vor AYLIN1 ausgeführten Aufruf von AXYFEN ein Fenster (Rahmen) im Koordinatenraum etabliert wurde. Dies ist hier insofern von Bedeutung, da beim eventuellen Zeichnen von Rasterlinien das Fenster von diesen ausgespart bleiben soll.
Siehe hierzu auch die Beschreibung der Unterprogramme AXYFEN und AXYRES.

Verwendete Unterprogramme und COMMON's:

GSW _____: NUMBER, PLOT, SYMBOL
TRACE/PC __: EXPD, NETLIN, NETLN2
COMMON __: FAKTOR, FNSTR, UGOG

12. Zeichnen einer logarithmischen Y-Achse

Das Unterprogramm AYLOG1 ermöglicht die Konstruktion einer logarithmischen Y-Achse (Ordinate).

Hinsichtlich der Lage der Achse auf dem Zeichnungsträger und deren Gestaltungsmöglichkeiten im Bereich der Repräsentationsgraphik gilt das gleiche wie für das Unterprogramm AXLIN1 beschrieben.

Die Argumentliste von AYLOG1 enthält 11 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL AYLOG1 (YL, YMIN, YMAX, IM, XL, IKM, HKM, HGM, ZSH, IT, TEXT)

Argumentbeschreibung:

YL	(cm)	Länge der Y-Achse.
YMIN	(phys. Einh.)	Kleinsten Wert an der Y-Achse (linkes Ende). Entspricht YMIN <u>nicht</u> einer Zehnerpotenz oder einer ganzzahligen Fraktion einer Zehnerpotenz (2,3, ..., $9 \cdot 10^n$), wird YMIN nicht als Graduation dargestellt.
YMAX	(phys. Einh.)	Größter Wert an der Y-Achse (rechtes Ende). Hinsichtlich der Darstellung von YMAX als Graduation gilt das gleiche wie zuvor bei YMIN beschrieben.
IM	(-)	Zwischen den Dekaden können 8 Teilstriche gezeichnet werden, von $2 \cdot 10^n$ bis $9 \cdot 10^n$. Siehe hierzu auch Argument IKM. Es gibt 4 Möglichkeiten, um an die Teilstriche auch die entsprechenden Graduationen zu zeichnen: IM = 0 Es werden keine Zwischengraduationen gezeichnet.

- = 1 An die Teilstriche 2, 4 und $7 \cdot 10^n$ werden die entsprechenden Graduationen gezeichnet.
- = 2 An die Teilstriche 2, 3, 5 und $8 \cdot 10^n$ werden die entsprechenden Graduationen gezeichnet.
- = 3 Es werden an alle Teilstriche Graduationen gezeichnet.

XL (cm) Multifunktionsvariable. Folgende Wertzuweisungen sind sinnvoll:

XL = Länge der X-Achse (Ordinate).
 Es wird eine parallel zur Y-Achse gespiegelte Achse im Abstand **XL** ohne Graduationen jedoch mit Teilstrichen gezeichnet (Rahmeneffekt).
 Sind zusätzlich Rasterlinien erwünscht, werden bei positivem **XL** diese durchgezogen, bei negativem **XL** gestrichelt dargestellt. Für diesen Fall muß **HKM** und/oder **HGM** der Wert 0. zugewiesen werden.

XL = 0.
 Das Zeichnen einer gespiegelten Achse entfällt.

IKM (-) Steuervariable.

IKM \leq 0 Zwischen den Dekaden werden weder Teilstriche noch Graduationen gezeichnet.

IKM $>$ 0 Zwischen den Dekaden werden 8 Teilstriche ($2 \cdot 10^n$ bis $9 \cdot 10^n$) und in Abhängigkeit des Arguments **IM** die entsprechenden Graduationen gezeichnet.

HKM (cm) Länge der kleinen Teilstriche.
 Falls Rasterlinien gezeichnet werden sollen, ist **HKM** = 0. zu setzen!

HGM	(cm)	<p>Länge der großen Teilstriche.</p> <p>Falls Rasterlinien gezeichnet werden sollen, ist HGM = 0. zu setzen!</p>
ZSH	(cm)	<p>Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text, Graduatio- nen).</p> <p>Darüberhinaus entscheidet das Vorzeichen von ZSH über die Lage von Graduationen, Achsenbeschriftung und Teilstriche zur Achse. Ist ZSH positiv, werden Teilstriche und Achsentext rechts und die Graduationen links von der Achse gezeichnet. Bei negativem ZSH ist es umgekehrt.</p> <p><u>Anmerkung:</u> Unabhängig von dem für XL eingegebenen Wert wird bei negativem ZSH <u>keine</u> gespiegelte Achse gezeichnet.</p>
IT	(-)	<p>Der Betrag von IT gibt die Anzahl der Textzeichen für die Achsenbeschriftung an. $IT_{max} = 80$.</p> <p>Ist IT positiv erscheint die Beschriftung im rechten Winkel, bei negativem IT parallel zur Y-Achse.</p> <p>Bei IT = 0 entfällt die Beschriftung.</p>
TEXT	(-)	<p>Text für die Achsenbeschriftung. Die Zeichenkette kann maximal 80 Zeichen beinhalten. Sie wird als vordefinierte CHARACTER-Variable (DATA, READ) oder direkt als Literal (eingeschlossen in Hochkommata) angegeben.</p>

AYLOG1 kommuniziert über verschiedene benannte COMMON's mit den Modulen CURVE, CURVE2, AXYFEN und AXYRES.

Folgende relevante Größen werden in AYLOG1 entweder definiert und übergeben oder empfangen:

- a.) Weitergabe von Informationen an CURVE und CURVE2 über den COMMON /FAKTOR/.
 - 1. Kleinster Wert an der Y-Achse [YMIN]
 - 2. Maßstabsfaktor [YL/lg(YMAX/YMIN)]
 - 3. Mitteilung, daß es sich hier um eine logarithmische Y-Achse handelt.

- b.) Weitergabe von Informationen an CURVE2 über den COMMON /UGOG/.
 - Feldbegrenzungen YMIN, YMAX

- c.) Kommunikation mit AXYFEN über den COMMON/FNSTR/.
 - AYLOG1 erhält über den COMMON/FNSTR/ die Information, ob durch einen vor AYLOG1 ausgeführten Aufruf von AXYFEN ein Fenster (Rahmen) im Koordinatenraum etabliert wurde. Dies ist hier insofern von Bedeutung, da beim eventuellen Zeichnen von Rasterlinien das Fenster von diesen ausgespart bleiben soll.
Siehe hierzu auch die Beschreibung der Unterprogramme AXYFEN und AXYRES.

Verwendete Unterprogramme und COMMON's:

GSW_____: NUMBER, PLOT, SYMBOL

TRACE/PC__: NETLIN, NETLN2

COMMON___: FAKTOR, FNSTR, UGOG

13. Zeichnen eines Kurvenverlaufs in ein kartesisches Koordinatensystem

Das Unterprogramm **CURVE** zeichnet den Verlauf einer Funktion, deren Komponenten in zwei eindimensionalen Feldern (X,Y) vorliegen.

Der Kurvenverlauf kann punktweise (in Form von Symbolen), als Linienzug oder als Linienzug mit Symbolen gezeichnet werden.

Bei der Darstellung als Linienzug werden die Punkte linear miteinander verbunden. In halb- oder doppeltlogarithmischen Koordinatensystemen sollte daher die Vertretbarkeit der linearen Verbindungsweise abgewogen werden.

Dem Aufruf von **CURVE** müssen die beiden Aufrufe zur Definition des Koordinatensystems (**AXLIN1** oder **AXLOG1** und **AYLIN1** oder **AYLOG1**) vorausgegangen sein.

Die Argumentliste von **CURVE** enthält 6 Elemente, die beim Aufruf definiert sein müssen.

Aufruf:

CALL CURVE (X,Y,IP,ISYMB,HS,INCR)

Argumentbeschreibung:

X(I), I=1,IP	(phys. Einh.)	Feldvariable. Sie enthält die X-Koordinaten der graphisch darzustellenden Wertetabelle.
Y(I), I=1,IP	(phys. Einh.)	Feldvariable. Sie enthält die Y-Koordinaten der graphisch darzustellenden Wertetabelle.
IP	(-)	Anzahl der darzustellenden Koordinatenpaare.
ISYMB	(-)	Wahl des Symboltyps. Sie ist für den Fall INCR=0 (Darstellung der Kurve <u>nur als Linienzug</u>) bedeutungslos.

TRACE/PC V.2.0 - CURVE

Folgende Symboltypen stehen zur Auswahl:

0 1 2 3 4 5 6 7 8 9 10 11 12
□ ⊙ △ + × ◇ † ∞ ∞ ∞ ∞ ∞ ∞

HS (cm)

Höhe der zu zeichnenden Symbole.

INCR (-)

Steuervariable.

Hinsichtlich der Darstellungscharakteristik hat das Argument folgende Bedeutungen:

INCR ≤ -1 Darstellung nur mit Symbolen.

INCR = 0 Darstellung nur als Linienzug.

INCR ≥ 1 Darstellung als Linienzug mit Symbolen.

Ist der Betrag von INCR > 1 wird nur jedes INCR_{tes} Wertepaar durch ein Symbol dargestellt.

Über den benannten COMMON /FAKTOR/ erhält CURVE die zur maßstabsgetreuen Wiedergabe erforderlichen Faktoren.

Siehe hierzu auch die Unterprogramme AXLIN1, AYLIN1, AXLOG1, AYLOG1.

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT, SYMBOL

TRACE/PC__: keine

COMMON__: FAKTOR

14. Zeichnen eines Kurvenverlaufs in ein kartesisches Koordinatensystem unter Kontrolle von Bereichsüberschreitungen

Das Unterprogramm **CURVE2** zeichnet den Verlauf einer Funktion, deren Komponenten in zwei eindimensionalen Feldern (X,Y) vorliegen.

CURVE2 stellt eine Ergänzung zum Unterprogramm **CURVE** dar. Es sind im wesentlichen zwei Eigenschaften, die **CURVE2** von **CURVE** unterscheiden. **CURVE2** zeichnet nicht über die Achsengrenzen (XMIN↔XMAX und YMIN↔YMAX) hinaus und bietet die Wahl einer individuellen Strichcharakteristik für jede zu zeichnende Kurve.

Dem Aufruf von **CURVE2** müssen die beiden Aufrufe zur Definition des Koordinatensystems (AXLIN1 oder AXLOG1 und AYLIN1 oder AYLOG1) vorausgegangen sein.

Die Argumentliste von **CURVE2** enthält 8 Elemente, die bis auf das Feldelement **WORK** beim Aufruf definiert sein müssen.

Aufruf:

CALL CURVE2(X,Y,IP,IC,SL1,SL2,BL,WORK)

Argumentbeschreibung:

X(I), I=1,IP	(phys. Einh.)	Feldvariable. Sie enthält die X-Koordinaten der graphisch darzustellenden Wertetabelle.
Y(I), I=1,IP	(phys. Einh.)	Feldvariable. Sie enthält die Y-Koordinaten der graphisch darzustellenden Wertetabelle.
IP	(-)	Anzahl der darzustellenden Koordinatenpaare.
IC	(-)	Steuervariable. Mit ihr und den folgenden drei Variablen (SL1 , SL2 , BL) wird die Liniencharakteristik der graphisch darzustellenden Kurve bestimmt.

IC <= 0 Die Linie wird als durchgezogene Vollinie dargestellt. Die Variablen **SL1**, **SL2** und **BL** müssen definiert werden, der ihnen zugeordnete Wert ist jedoch bedeutungslos.

> 0 Die Linie wird entsprechend den Angaben für **SL1**, **SL2** und **BL** gestaltet.

SL1 (cm)
SL2 (cm)
BL (cm)

Mit den Variablen **SL1**, **SL2** und **BL** wird die Strichcharakteristik festgelegt.

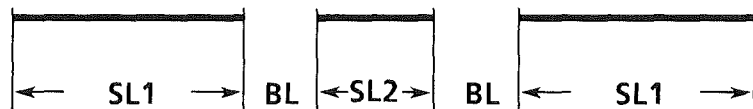
Eine sinnvolle Definition ist nur für den Fall IC > 0 erforderlich.

Der Kurvenverlauf wird mit der sich periodisch wiederholenden Strichsequenz

SL1→BL→SL2→BL→SL1→BL→SL2→BL

usw. dargestellt.

Folgende Prinzipskizze soll dies veranschaulichen:



WORK(I), (-)
 I=1, (2*IP)

Feldvariable, die im aufrufenden Programm dimensioniert werden muß.

Es handelt sich hierbei um ein Hilfsfeld der Größe 2*IP welches der Rettung der Eingangsfelder X und Y dient.

Über die benannten COMMON's /FAKTOR/ und /UGOG/ erhält CURVE2 die zur Darstellung des Kurvenverlaufs erforderlichen Maßstabsfaktoren und Feldbegrenzungsinformationen.

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT
TRACE/PC__: C2XP → CDL → CURVE1,CRV
COMMON__: FAKTOR, UGOG, S\$C2

15. Konstruktion eines Polarkoordinatensystems

Das Unterprogramm **POL** zeichnet ein Polarkoordinatensystem. Nach dessen Erstellung können darin Funktionen in Form von Wertepaaren, die jeweils Radius und Winkel repräsentieren, abgebildet werden (siehe hierzu auch die Unterprogramme **CURVEP** und **CURVP2**).

Die Richtungen der beiden Hauptachsen auf dem Zeichnungsträger liegen fest und sind nicht veränderbar.

Die Argumentliste von **POL** enthält 8 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL POL(RL,RMAX,DR,ZSH,IGEN,NAX,IT,TEXT)

Argumentbeschreibung:

RL	(cm)	Maximaler Halbmesser des Polarkoordinatensystems.
RMAX	(phys. Einh.)	Gößte radiale Koordinate. Sie muß nicht zwangsläufig als Graduation erscheinen, da dies von dem für DR gewählten Wert abhängt.
DR	(phys. Einh.)	Der Wert von DR gibt an, nach wieviel physikalischen Einheiten jeweils ein Koordinatenkreis gezeichnet werden soll. Sofern der Wert der Variablen ZSH positiv ist, erscheint die entsprechende Graduation zusätzlich an der 90°-Achse des Polarkoordinatensystems.
ZSH	(cm)	Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text und Graduationen). Bei positivem ZSH werden die Graduationen links vom Schnittpunkt der Koordinatenkreise mit der 90°-Achse gezeichnet.

Bei negativem ZSH entfällt das Zeichnen der Graduationen. Stattdessen wird rechts oben neben dem Polarkoordinatensystem die maximale radiale Koordinate gezeichnet.

IGEN (-)

Mit dieser Variablen wird die Anzahl der Nachkommastellen der zu zeichnenden Graduationen festgelegt.

Bei der Darstellung der Graduationen ohne Dezimalexponenten bedeutet:

IGEN = -1 Graduation ohne Dezimalpunkt
 = 0 Graduation mit Dezimalpunkt
 = 1 Graduation mit einer Nachkommastelle

usw.

Für die Darstellung der Graduationen mit Dezimalexponenten gilt:

$$[-11] \geq \text{IGEN} \geq [-14]$$

Es bedeutet z. B.:

IGEN = -11 Eine signifikante Nachkommastelle: 0.3E+03
 = -13 Drei signifikante Nachkommastellen: 0.123E-05

NAX (-)

Steuervariable, mit der die Anzahl der radialen Achsen im Polarkoordinatensystem festgelegt wird.

Es werden Achsen gezeichnet bei:

NAX = 0 bei 0° und 90°
 = 1 alle 45°
 = 2 " 30°
 = 3 " 15°
 = 4 " 10°

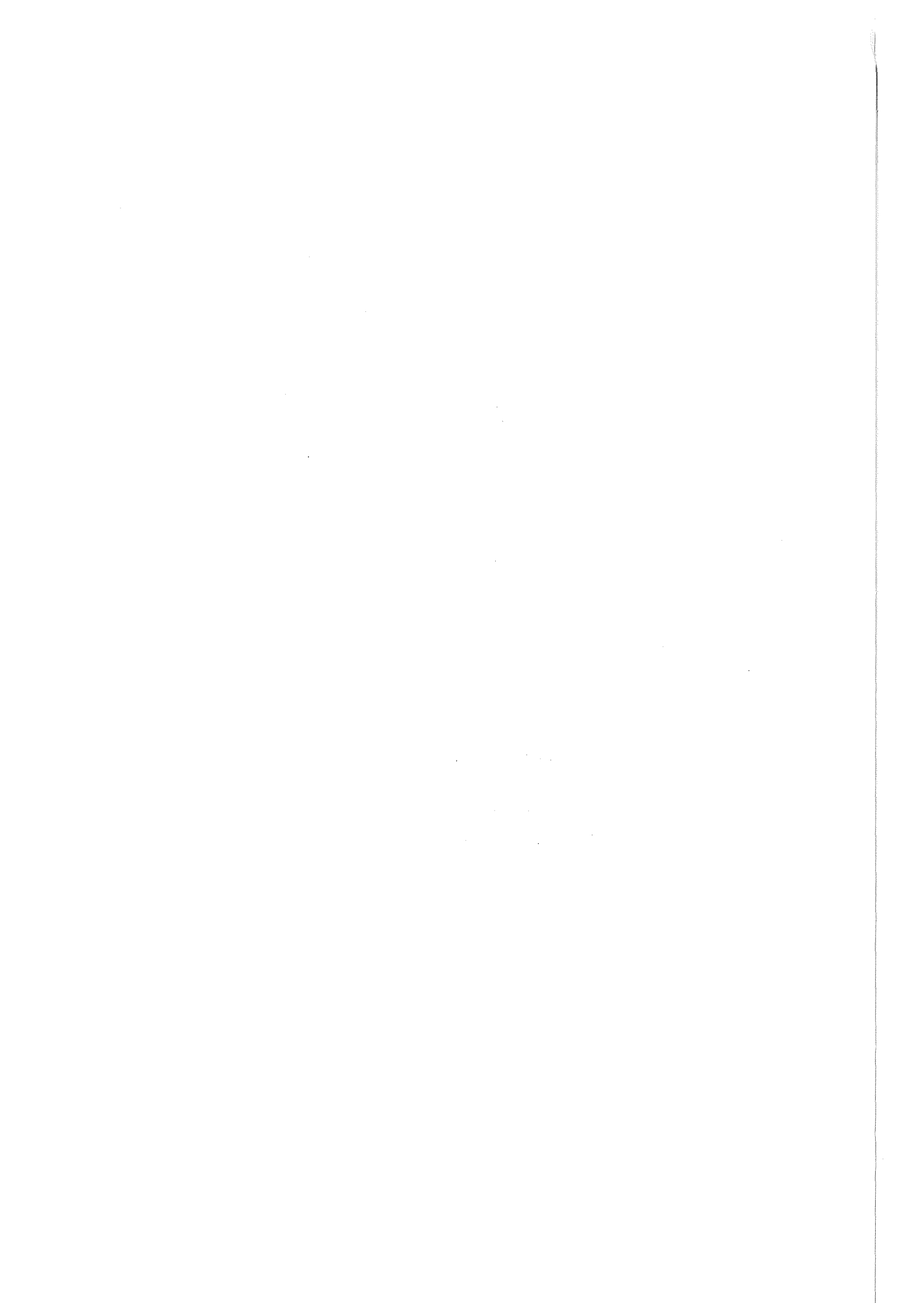
IT	(-)	Das Polarkoordinatensystem kann mit einer Beschriftung versehen werden. IT gibt dabei die Anzahl der darzustellenden Textzeichen an. Der Text wird, bezogen auf die 90°-Achse, zentriert unter dem Polarkoordinatensystem abgebildet.
TEXT	(-)	Text für das Polarkoordinatensystem. Die Zeichenkette darf maximal 80 Zeichen beinhalten. Sie kann in der Argumentliste als Literal (in Hochkommas eingeschlossen) oder als vordefinierte Textvariable (DATA, READ) erscheinen.

Das Unterprogramm **POL** übergibt dem benannten **COMMON /FAKTOR/** den Maßstabsfaktor (RL/RMAX), der für die Ausführung der Unterprogramme **CURVEP** und **CURVP2** erforderlich ist.

Darüberhinaus kommuniziert **POL** über den **COMMON /UGOG/** mit dem Unterprogramm **CURVP2** und übermittelt diesem die Feldbegrenzung ($-R_{\max} \leftrightarrow +R_{\max}$).

Verwendete Unterprogramme und Common's:

GSW _____: NUMBER, SYMBL3, SYMBOL, WHERE
 TRACE/PC __: CERCLE, EXPD, PLOTP, YPOS
 COMMON __: FAKTOR



16. Zeichnen eines Kurvenverlaufs in ein Polarkoordinatensystem

Das Unterprogramm **CURVEP** ermöglicht die graphische Darstellung von Funktionen (Radius als Funktion des Umfangswinkels) in einem Polarkoordinatensystem.

Über eine Steuervariable wird entschieden ob die Kurve punktweise (Symbole), als Linienzug oder als Linienzug mit Symbolen dargestellt werden soll. Bei der Darstellung als Linienzug werden die Punkte linear miteinander verbunden.

Bevor mit **CURVEP** eine Funktion dargestellt werden kann, ist das Unterprogramm **POL** aufzurufen, wobei das Polarkoordinatensystem gezeichnet und der Maßstabsfaktor bestimmt wird.

Die Argumentliste von **CURVEP** enthält 6 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL CURVEP (RAD,PHI,IP,ISYMB,HZS,INCS)

Argumentbeschreibung:

RAD(I), I=1, IP	(phys. Einh.)	Feldvariable. Sie enthält die radialen Koordinaten der darzustellenden Wertetabelle.
PHI(I), I=1, IP	(°)	Feldvariable. Sie enthält die azimuthalen Koordinaten (Winkel) der darzustellenden Wertetabelle.
IP	(-)	Anzahl der zu zeichnenden Koordinatenpaare.
ISYMB	(-)	Wenn die Kurve punktweise mit Symbolen oder als Linienzug mit Symbolen dargestellt werden soll, wird der Variablen ISYMB die Ordnungsnummer des gewünschten Symbols zugewiesen. Folgende Symboltypen sind verfügbar:

0	1	2	3	4	5	6	7	8	9	10	11	12
□	○	△	+	×	◇	♣	⌘	⌚	Υ	⌘	*	⌘

Soll die zu zeichnende Kurve nur als Linienzug wiedergegeben werden ($INCS=0$), ist der für **ISYMB** angegebene Wert ohne Bedeutung.

HZS	(cm)	Höhe der zu zeichnenden Symbole.
INCS	(-)	Mit der Variablen INCS wird festgelegt, ob die Kurve nur durch Symbole ($INCS \leq -1$), nur als Linienzug ($INCS=0$) oder als Linienzug mit Symbolen ($INCS \geq 1$) dargestellt werden soll. Ist der Betrag von INCS > 1 bedeutet dies, daß nur jedes INCS tes Wertepaar durch ein Symbol angedeutet wird.

Über den benannten **COMMON /FAKTOR/** erhält **CURVEP** vom Unterprogramm **POL** den für die maßstabgetreue Ausführung erforderlichen Faktor (**RL/RMAX**).

Verwendete Unterprogramme und COMMON's:

GSW_____ : PLOT, SYMBOL
TRACE/PC__ : keine
COMMON__ : FAKTOR, WORK

17. Zeichnen eines Kurvenverlaufs in ein Polarkoordinatensystem unter Kontrolle von Bereichsüberschreitungen

Das Unterprogramm **CURVP2** zeichnet den Verlauf einer Funktion, deren Komponenten aus Radius und Umfangswinkel bestehen, in ein Polarkoordinatensystem.

CURVP2 stellt eine Ergänzung zum Unterprogramm **CURVEP** dar. Es sind im wesentlichen zwei Eigenschaften, die **CURVP2** von **CURVEP** unterscheiden. **CURVP2** zeichnet nicht über die Systemgrenzen ($-RMAX \leftrightarrow +RMAX$) hinaus und bietet die Wahl einer individuellen Strichcharakteristik für jede zu zeichnende Kurve.

Dem Aufruf von **CURVP2** muß der Aufruf zur Definition des Polarkoordinatensystems (**POL**) vorausgegangen sein.

Die Argumentliste von **CURVP2** enthält 8 Elemente, die bis auf das Feldelement **WORK** beim Aufruf definiert sein müssen.

Aufruf:

CALL CURVP2 (RAD,PHI,IP,IC,SL1,SL2,BL,WORK)

Argumentbeschreibung:

RAD(I), I=1,IP	(phys. Einh.)	Feldvariable. Sie enthält die radialen Koordinaten der graphisch darzustellenden Wertetabelle.
PHI(I), I=1,IP	(°)	Feldvariable. Sie enthält die azimutalen Koordinaten (Winkel) der graphisch darzustellenden Wertetabelle.
IP	(-)	Anzahl der zu zeichnenden Koordinatenpaare.
IC	(-)	Steuervariable. Mit ihr und den folgenden drei Variablen (SL1, SL2, BL) wird die Liniencharakteristik der graphisch darzustellenden Kurve bestimmt.

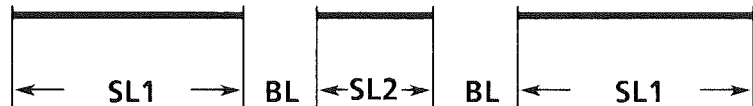
- IC \leq 0 Die Linie wird als durchgezogene Volllinie dargestellt. Die Variablen SL1, SL2 und BL müssen definiert werden, der ihnen zugeordnete Wert ist jedoch bedeutungslos.
- > 0 Die Linie wird entsprechend den Angaben für SL1, SL2 und BL gestaltet.

SL1	(cm)	Mit den Variablen SL1, SL2 und BL wird die Strichcharakteristik festgelegt. Eine <u>sinnvolle</u> Definition ist nur für den Fall IC > 0 erforderlich. Der Kurvenverlauf wird mit der sich periodisch wiederholenden Strichsequenz
SL2	(cm)	
BL	(cm)	

SL1→BL→SL2→BL→ SL1→BL→SL2→BL

usw. dargestellt.

Folgende Prinzipskizze soll dies veranschaulichen:



WORK(I),	(-)	Feldvariable, die im aufrufenden Programm dimensioniert werden muß. Es handelt sich hierbei um ein Hilfsfeld der Größe 2*IP welches der Rettung der Eingangsfelder RAD und PHI dient.
I=1, (2*IP)		

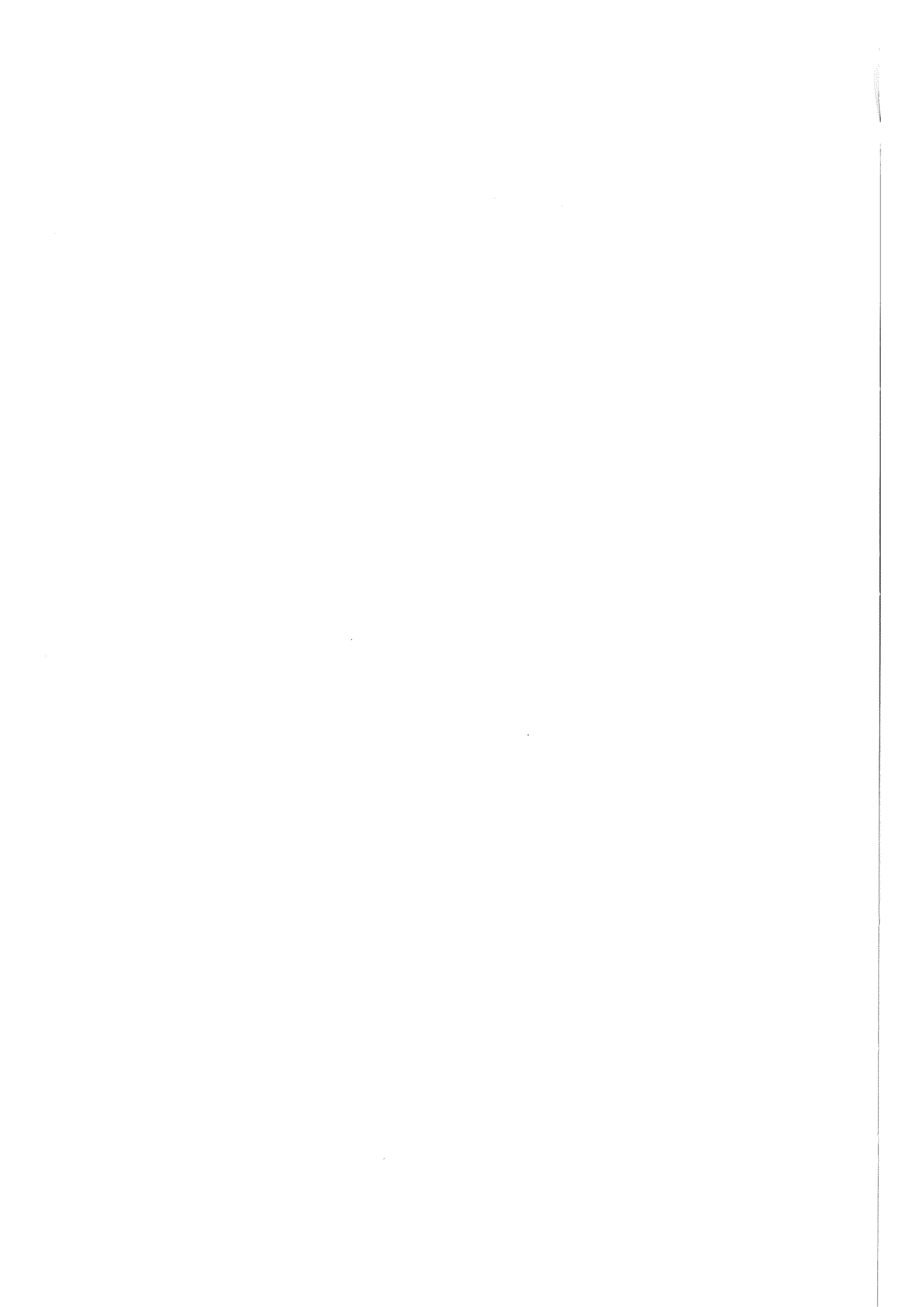
Über die benannten COMMON's /FAKTOR/ und /UGOG/ erhält CURVP2 die zur Darstellung des Kurvenverlaufs erforderlichen Maßstabsfaktoren und Feldbegrenzungsinformationen.

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT

TRACE/PC__: C2XPP → CDLP → CURVE1, CURVEP

COMMON__: FAKTOR, UGOG



18. Zeichnen eines Streckenzuges

Das Unterprogramm CURVLV ermöglicht die zeichnerische Darstellung eines Streckenzuges, dessen Elemente (Wertepaare) in eindimensionalen Feldern vorliegen. Die Darstellungseinheit ist "cm" und entspricht damit der Systembasis-Einheit. Eine Aufbereitung der Daten ist somit nicht erforderlich.

Der Streckenzug kann sowohl als durchgezogene Vollinie als auch in gestrichelter oder strichpunktierter Form dargestellt werden.

Die Argumentliste von CURVLV enthält 4 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL CURVLV (X,Y,IP,IST)

Argumentbeschreibung:

X(I), I=1, IP	(cm)	Feldvariable. Sie enthält die X-Koordinaten des zu zeichnenden Streckenzuges.
Y(I), I=1, IP	(cm)	Feldvariable. Sie enthält die Y-Koordinaten des zu zeichnenden Streckenzuges.
IP	(-)	Anzahl der darzustellenden Koordinatenpaare.
IST	(-)	Steuervariable, mit der die Strichcharakteristik des Streckenzuges festgelegt wird. Die Wertepaare werden bei IST = 0 mit einer durchgezogenen Linie verbunden. = 1 mit einer gestrichelten Linie verbunden. Strichlänge/Space: .4/.2 [cm] = 2 mit einer strichpunktierter Linie verbunden. Strichlänge/Punkt/Space: .4/.1/.2 [cm]

Verwendete Unterprogramme und COMMON's:

GSW_____ : PLOT
TRACE/PC__ : CURVE1
COMMON___ : keine

19. Zeichnen von Textstrings und Symbolen

Die Unterprogramme SYMBP1, SYMBP2, SYMBP3, SYMBP4 und SYMBS1 ermöglichen die zeichnerische Darstellung von alphanumerischen Zeichenketten (Textstrings) und Symbolen.

Jedem der angegebenen Module ist infolge der Zeichenvielfalt ein spezifischer Zeichensatz zugeordnet. So können mit SYMBP1 Zeichenketten, die sich aus lateinischen Klein- und Großbuchstaben (Umlaute und Eszett inbegriffen), numerischen Zeichen und Sonderzeichen zusammensetzen, dargestellt werden. Die Wiedergabe je eines zentrierten Symbols pro Aufruf ist ebenfalls möglich. Griechische Großbuchstaben werden mit SYMBP2, griechische Kleinbuchstaben mit SYMBP3 gezeichnet. SYMBP4 ermöglicht die Darstellung einer kleinen Auswahl an gefüllten zentrierten Symbolen.

SYMBP1 unterscheidet sich von SYMBS1 lediglich in der Schriftform. Dabei bedeutet "P" Normalschrift- und "S" Schrägschriftversion (kursiv, 75°-Normalschrift).

Die Schriftart aller hier erwähnten Zeichensätze ist aufgrund der Erscheinungsform in die Kategorie der Druckschriften einzuordnen.

Aus Gründen der besseren Übersicht wird im folgenden die soeben beschriebene Zuordnung der Module zu den Zeichensätzen in tabellarischer Form wiedergegeben.

Modul Zeichensatz	SYMBP1	SYMBP2	SYMBP3	SYMBP4	SYMBS1 (kursiv)
latein. Großbuchst.	●	—	—	—	●
latein. Kleinbuchst.	●	—	—	—	●
numerische Zeichen	●	—	—	—	●
Sonderzeichen	●	—	—	—	●
Symbole	●	—	—	—	—
griech. Großbuchst.	—	●	—	—	—
griech. Kleinbuchst.	—	—	●	—	—
gefüllte Symbole	—	—	—	●	—

Da sowohl die griechischen Zeichen als auch die Symbole nicht im Zeichenvorrat der Tastatur enthalten sind, wird über verfügbare Zeichen eine Korrespondenz hergestellt. In Verbindung mit dem entsprechenden Modul läßt sich dann die gewünschte Darstellung realisieren.

TRACE/PC V.2.0 - SYMBP1-4/SYMB51

Aus den folgenden Tabellen kann Zuordnung und Umfang der in Betracht kommenden Zeichen entnommen werden.

Großbuchstaben des griechischen Alphabets (SYMBP2):

Eingabe→	C	D	F	G	L	P	R	S	1	2	3
Zeichen	X	Δ	Φ	Γ	Λ	Π	Ρ	Σ	Θ	Ψ	Ω

Kleinbuchstaben des griechischen Alphabets (SYMBP3):

Eingabe→	A	B	C	D	E	F	G	I	K	L	M	N	O	P	R	S	T
Zeichen	α	β	χ	δ	ε	φ	γ	ι	κ	λ	μ	ν	ο	π	ρ	σ	τ

Fortsetzung:

Eingabe→	X	Y	Z	0	1	2	3
Zeichen	ξ	υ	ζ	η	θ	ψ	ω

zentrierte Symbole (SYMBP1):

Eingabe→	0	1	2	3	4	5	6	7	8	9	10	11	12
Symbol	□	○	△	+	×	◇	⊕	⊗	⊘	Υ	⊗	*	⊗

gefüllte zentrierte Symbole (SYMBP4):

Eingabe→	0	1	2	5	12
Symbol	■	●	▲	◆	⊗

Die Argumentlisten von SYMBP1-4 und SYMB51 enthalten jeweils 6 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL SYMBP_x (XK,YK,H,CHAR,W,IC) → x: 1 - 4

oder

CALL SYMB51 (XK,YK,H,CHAR,W,IC)

Argumentbeschreibung:

XK	(cm)	<p>Koordinaten, die die Position des darzustellenden Textes oder eines Symbols in Relation zu dem zuletzt definierten Bezugspunkt festlegen.</p> <p>Für zu zeichnende Texte gilt: Abhängig von dem Vorzeichen von H (Zeichenhöhe) stellen die Koordinaten den Anfang (+ H) bzw. das Zentrum (-H) der Textzeile dar.</p> <p>Bei der Symboldarstellung ist die Position [XK, YK] identisch mit dem Symbolzentrum. Das Vorzeichen von H ist hierbei bedeutungslos.</p> <p>Soll auf die Stringendekoordinaten des zuletzt ausgeführten SYMBP_x/SYMBS1- oder NUMBP/NUMBS-Aufrufs Bezug genommen werden, ist XK und/oder YK lediglich der Wert <u>999</u>, zuzuweisen. Damit lassen sich, ohne umständliche Bestimmung der Anschlußkoordinaten, in der gleichen Zeile Fortsetzungen von Zeichen-/Ziffernstrings realisieren.</p>
YK	(cm)	
H	(cm)	<p>Der Betrag von H gibt die Höhe der darzustellenden Textzeichen bzw. Symbole an.</p> <p>Darüber hinaus beeinflußt das Vorzeichen von H die Textposition (siehe dazu XK und YK).</p>
CHAR	(-)	<p>Zu zeichnender Text bzw. Angabe der Ordnungszahl eines zu zeichnenden Symbols.</p> <p>Der Textstring kann direkt in Form eines Literals (in Hochkommas eingeschlossen) oder als vordefinierte CHARACTER-Variable (READ, DATA) angegeben werden. Die Zeichenkette darf maximal 80 Zeichen beinhalten.</p>
W	(°)	<p>Winkel, unter dem der darzustellende Text oder das gewünschte Symbol gezeichnet werden soll.</p> <p>Bei W = 0, wird der Text horizontal dargestellt.</p>
IC	(-)	<p>Bei der Darstellung eines Textstrings gibt IC die</p>

Anzahl der Textzeichen an. $IC_{max} = 80$.

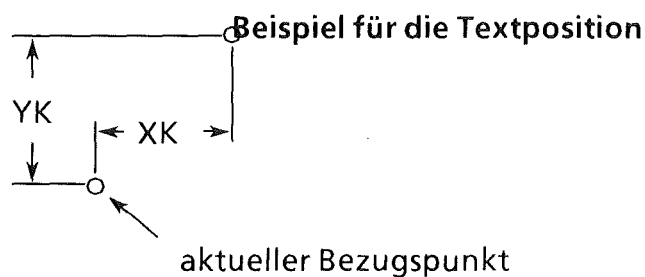
Wird IC der Wert -1 oder -2 zugewiesen, soll angezeigt werden, daß ein Symbol zu zeichnen ist.

Bei -1 wird die Symbolposition mit abgehobenem, bei -2 mit abgesenktem Zeichenstift angefahren.

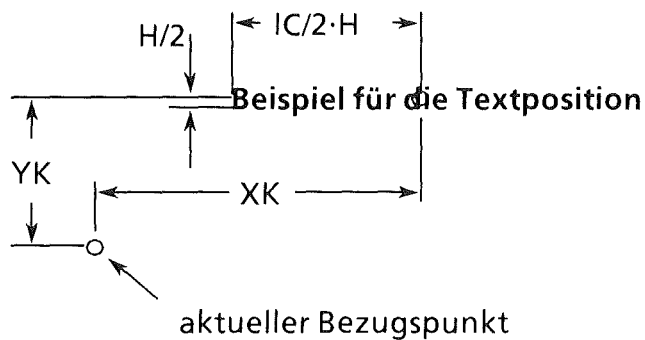
Beispiel zur Textpositionierung:

Eingabetext CHAR: 'Beispiel für die Textposition'; IC = 29

1. +H:



2. -H:



Verwendete Unterprogramme und Common's:

GSW _____: SYMBOL (SYMBP1), SYMBL2 (SYMBP2),
 SYMBL3 (SYMBP3), SYMBL4 (SYMBP4),
 PLTEXT (SYMSB1)

TRACE/PC __: POSXY

COMMON __: CENTER

20. Zeichnen von Gleitkommazahlen (Ziffernstrings)

Die Unterprogramme NUMBP und NUMBS ermöglichen die zeichnerische Darstellung von Gleitkommazahlen. Die Ausgabe erfolgt als Zeichenkette (Ziffernstring), die im Erscheinungsbild dem Fortran-F-Format entspricht.

NUMBP ist die Normal-, NUMBS die Schrägschriftversion (kursiv, 75°-Normschrift).

Die Argumentlisten von NUMBP und NUMBS enthalten jeweils 6 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL NUMBP (XK,YK,H,ZAHL,W,NK)

oder

CALL NUMBS (XK,YK,H,ZAHL,W,NK)

Argumentbeschreibung:

XK	(cm)	Koordinaten, die die Position des darzustellenden Ziffernstrings in Relation zu dem zuletzt definierten Bezugspunkt festlegen. Abhängig von dem Vorzeichen von H (Ziffernhöhe) stellen die Koordinaten den Anfang (+H) bzw. das Zentrum des Ziffernstrings (-H) dar. Soll auf die Stringendekoordinaten des zuletzt ausgeführten NUMBP/NUMBS- oder SYMBP _x /SYMBS1-Aufrufs Bezug genommen werden, ist XK und/oder YK lediglich der Wert <u>999</u> zuzuweisen. Damit lassen sich, ohne umständliche Bestimmung der Anschlußkoordinaten, in der gleichen Zeile Fortsetzungen von Zeichen-/Ziffernstrings realisieren.
YK	(cm)	
H	(cm)	Der Betrag von H gibt die Höhe der darzustellenden Ziffernzeichen an. Darüber hinaus beeinflußt das Vorzeichen von H die Position des Ziffernstrings (siehe dazu XK und YK).
ZAHL	(-)	Darzustellende <u>Gleitkommazahl</u> .

W (°) Winkel, unter dem der Ziffernstring gezeichnet werden soll.
Bei $W = 0$. wird die Zahl horizontal dargestellt.

NK (-) Der NK zugewiesene Wert gibt an, mit wieviel Nachkommastellen die betrachtete Zahl darzustellen ist.

Beispiel: ZAHL ← 123.456

- NK = 3 → 123.456
- = 2 → 123.45
- = 1 → 123.4
- = 0 → 123.
- = -1 → 123

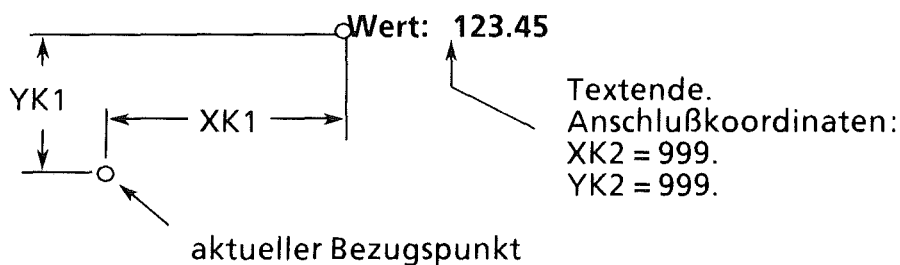
Beispiel zur Positionierung und Fortsetzung von Zeichen-/Ziffernstrings.

Anweisungsfolge:

```

.
.
CALL SYMBP1(XK1,YK1,H,'Wert: ',0.,6)
CALL NUMBP(XK2,YK2,H,123.456,0.,2)
.
.

```



Verwendete Unterprogramme und Common's:

- GSW_____ : SYMBOL (NUMBP), PLTEXT (NUMBS)
- TRACE/PC__ : POSXY, ZST
- COMMON___ : CENTER

21. Verwendung von Sonderschriftarten

Mit dem Unterprogramm FNTX hat der Anwender Zugriff auf spezielle kalligraphische Schriftzeichen (Font's). Die Kollektion umfaßt 5 verschiedene Zeichensätze zu je 86 Zeichen. Das Erscheinungsbild der Zeichensätze entspricht den Hershey Font's, die vom amerikanischen National Bureau of Standards [3] vertrieben werden. Folgende Schriftarten sind verfügbar:

Font	Schriftart
1	Triplex Roman
2	Complex Script
3	Gothic English
4	Triplex Italic
5	Gothic German

Jeder Zeichensatz besteht aus alphabetischen, numerischen und Sonderzeichen.

Die alphabetischen Zeichen umfassen alle Groß- und Kleinbuchstaben einschließlich der Umlaute und das 'ß'.

Neben den numerischen Zeichen 0 - 9 sind noch 17 Sonderzeichen verfügbar. Da diese Zeichen oft nicht direkt über die reguläre Tastatur erreichbar sind, sondern nur über deren Hexadezimal- bzw. ASCII-Code, wird in nachstehender Tabelle die Korrespondenz veranschaulicht.

Nummer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Zeichen	+	-	*	:	=	()	.	,	;	?	!	°	/	&	"	'
HEX	2 B	2 D	2 A	3 A	3 D	2 8	2 9	2 E	2 C	3 B	3 F	2 1	A 1	2 F	2 6	2 2	22 77
ASCII	0 4 3	0 4 5	0 4 2	0 5 8	0 6 1	0 4 0	0 4 1	0 4 6	0 4 4	0 5 9	0 6 3	0 3 3	1 6 1	0 4 7	0 3 8	0 3 4	00 33 99

Die Zeichenketten (Textstrings) werden nach dem Prinzip der Proportionalschrift dargestellt. Das Verhältnis von Höhe zu Breite ist bei jedem Zeichen individuell verschieden und als interner Parameter bekannt. Durch das Aneinanderreihen der Zeichen mit unterschiedlicher Schrittweite entsteht ein optischer Eindruck von ästhetischer Ausgewogenheit.

Die Argumentliste von FNTX enthält 10 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL FNTX(XK,YK,H,B,BS,XN,W,CHAR,IC,IFONT)

Argumentbeschreibung:

XK	(cm)	Koordinaten des Textanfangs oder des Textzentrums. Siehe hierzu auch das Argument H!
YK	(cm)	Soll der aktuelle Text am Ende eines unmittelbar zuvor mit FNTX gezeichneten Textes fortgesetzt werden, ist den Variablen XK und YK jeweils der Wert <u>999</u> , zuzuweisen.
H	(cm)	Der Betrag von H gibt die Höhe der Textzeichen an. Darüberhinaus hat das Argument noch folgende Bedeutung: Ist H positiv befindet sich der Textstringanfang bei der Position XK,YK (linke untere Ecke des 1. Zeichens). Bei negativem H wird der gesamte Textstring auf die Position XK,YK zentriert.
B	(cm)	"Breite" der Textzeichen. Unter "Breite" ist hier infolge des individuellen Platzbedarfs der einzelnen Zeichen eine mittlere Breite zu verstehen. Sie dient lediglich zur Orientierung.
BS	(cm)	Leerraum (Space). Er wird zwischen alle Zeichen des Textstrings eingefügt.
XN	(°)	Neigungswinkel der Textzeichen gegen die Vertikale in mathematisch entgegengesetztem Drehsinn (Uhrzeigersinn).
W	(°)	Winkel, unter dem der Textstring CHAR gezeichnet werden soll. Bei W = 0. wird der Text horizontal dargestellt.

CHAR	(-)	Textstring. Innerhalb eines Textstrings sind alle gültigen zuvor besprochenen Zeichen erlaubt. Die Zeichenkette darf maximal 80 Zeichen beinhalten.
IC	(-)	Anzahl der Zeichen, aus denen sich der Textstring zusammensetzt, Leerzeichen (Blanc's) inklusive. $IC_{\max} = 80.$
IFONT	(-)	Wahl der Schriftart. Entsprechend der zuvor angegebenen Tabelle wird für IFONT die gewünschte Font-Nummer angegeben.

Verwendete Unterprogramme und Common's:

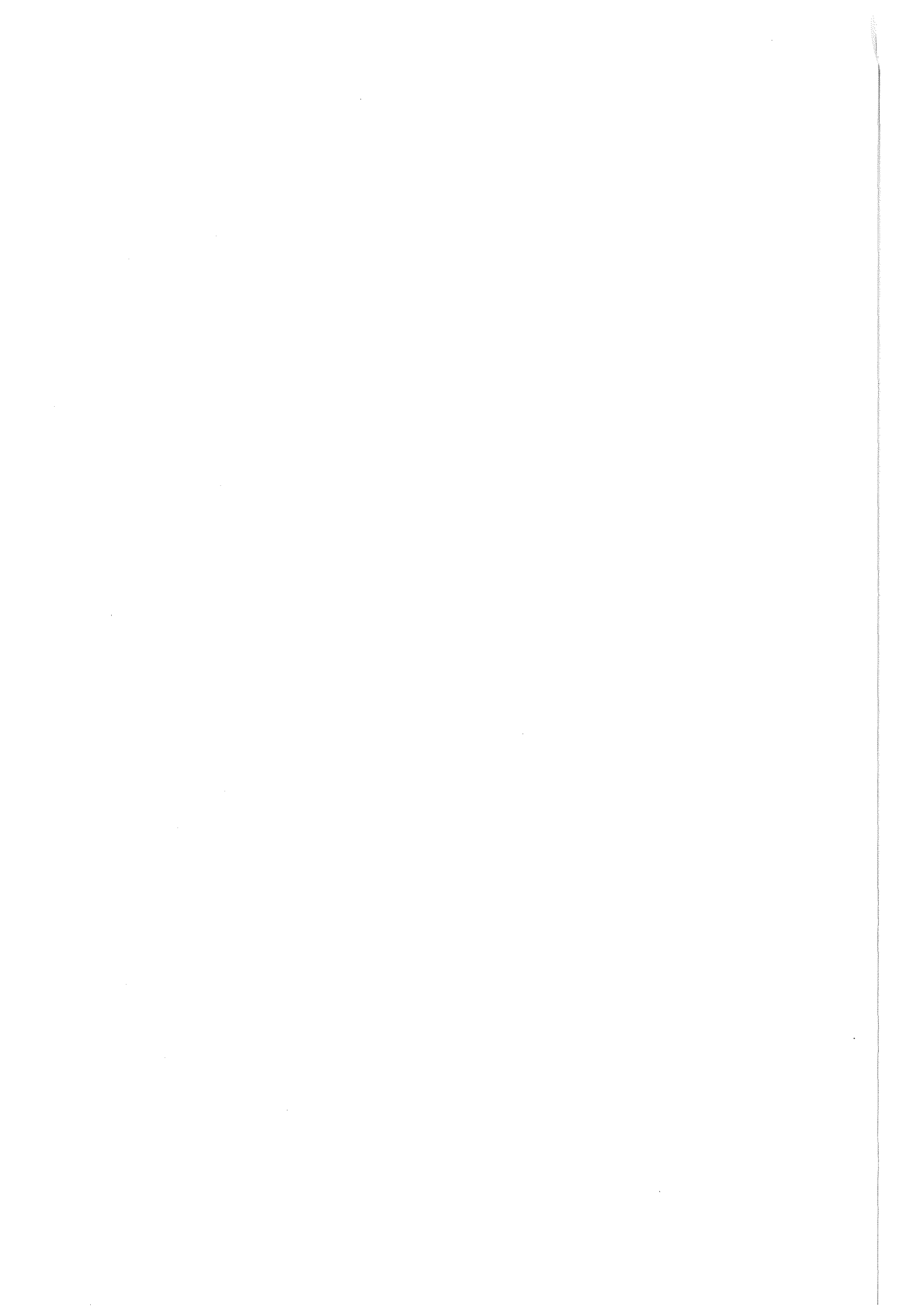
GSW_____ : PLOT
 TRACE/PC__ : POSFT
 COMMON___ : CENTER

Hinweis: Bei Anwendung von FNTX müssen die Schriftarten-Files resident sein und zwar in demselben Directory in dem auch das ausführbare Programm (EXE-Modul) gestartet wird.

Es ist darüber hinaus zu beachten, daß die File-Unit's 46, 47 und 48 für die Dauer des Programmablaufs exklusiv dem Sonderschriftarten-Modul zugeordnet sind.

Es handelt sich um folgende Files:

FTI1.DAT	FTX1.DAT	FTY1.DAT
FTI2.DAT	FTX2.DAT	FTY2.DAT
FTI3.DAT	FTX3.DAT	FTY3.DAT
FTI4.DAT	FTX4.DAT	FTY4.DAT
FTI5.DAT	FTX5.DAT	FTY5.DAT



22. Zeichnerische Wiedergabe von Texten in Normal- und Schrägschrift

Mit Hilfe der Unterprogramme TEXT bzw. TEXTS können Texte, die in separaten Datenfile's vorliegen und nicht integraler Bestandteil eines Programms sind, gezeichnet werden. Die Textdarstellung kann sowohl in Normal- (TEXT) als auch in Schrägschrift (TEXTS, kursiv) erfolgen.

Neben den lateinischen Buchstaben in Groß- und Kleinschrift, einschließlich der Umlaute und des Eszett's, können auch die Buchstaben des griechischen Alphabets, Sonderzeichen und Symbole zeichnerisch wiedergegeben werden.

Da über die Tastatur das griechische Alphabet und die Symbole nicht erreichbar sind, wurden Zeichenkombinationen festgelegt, die die Erkennung ermöglichen. Dabei dient die geschweifte Klammer auf ({}), in Verbindung mit einem lateinischen Großbuchstaben oder einer der Ziffern 0 - 3, zur Markierung eines kleinen griechischen Buchstabens. Mit dem numerischen Zeichen (#), in Kombination mit einem lateinischen Großbuchstaben oder den Ziffern 1 - 3, werden die griechischen Großbuchstaben markiert. Dies jedoch nur insoweit, wie sich die griechischen Zeichen von den lateinischen unterscheiden. Die Symbole werden durch ein \$-Zeichen markiert, gefolgt von einer der Ziffern 0 - 9 oder einem der Buchstaben W - Y.

Die Selbstdarstellung der Markierungszeichen erfolgt durch die zweimalige Aufeinanderfolge desselben Zeichens:

"{{" → "{"; "##" → "#"; "\$\$" → "\$".

Einschränkung: "\$\$" kann für die Selbstdarstellung des \$-Zeichens nicht in den Spalten 1 und 2 einer Zeile stehen, da bei dieser Position die Zeichenkombination eine andere Bedeutung bekommt (Wechsel der Textdarstellungsparameter; siehe später im Text).

Aus Gründen der besseren Übersicht wird im folgenden die Korrespondenz der Zeichen detailliert in tabellarischer Form wiedergegeben

Kleinbuchstaben des griechischen Alphabets:

{ + →	A	B	C	D	E	F	G	I	K	L	M	N	O	P	R	S	T
Zeichen	α	β	χ	δ	ϵ	ϕ	γ	ι	κ	λ	μ	ν	\omicron	π	ρ	σ	τ

Fortsetzung:

{ + →	X	Y	Z	0	1	2	3
Zeichen	ξ	υ	ζ	η	θ	ψ	ω

Großbuchstaben des griechischen Alphabets:

# + →	C	D	F	G	L	P	R	S	1	2	3
Zeichen	X	Δ	Φ	Γ	Λ	Π	Ρ	Σ	Θ	Ψ	Ω

Symbole:

\$ + →	0	1	2	3	4	5	6	7	8	9	W	X	Y
Symbol	□	⊙	△	+	×	◇	♠	⋈	⋈	Υ	⋈	*	⋈

Hinweis: Ein \$-Zeichen gefolgt von einem Leerzeichen (Blanc) bewirkt den Textabbruch in der betrachteten Zeile.

Da grundsätzlich 80 Spalten jeder Zeile verarbeitet werden, also auch die (sinnlosen) Leerstellen am Textende einer Zeile, kann man durch ein abschließendes \$-Zeichen (Rechen-) Zeit einsparen.

Neben der eigentlichen Textinformation enthalten die Datenfile's auch Steuerdaten, die die Textstruktur und die Gestaltung des Layout's festlegen. Es handelt sich hierbei um 6 Eingabeparameter, die dem eigentlichen Text in einer Zeile (free formatted) vorangestellt werden müssen.

Im einzelnen sind dies:

- I. Schrifthöhe [SH, cm] REAL
- II. Schriftbreite [SB, cm] REAL
- III. Spaltenbreite zwischen 2 Buchstaben ... [ZB, cm] REAL
- IV. Zeilenzwischenraum [ZZ, cm] REAL
- V. Strichstärke (0/1 --- dünn/dick) [IS, ----] INTEGER
- VI. Zeichenfarbe (s. PEN-Farbtafel) [IC, ----] INTEGER

Innerhalb eines Textes können nach jeder Zeile die Steuerdaten geändert werden. Dazu muß zunächst eine Zeile mit 2 aufeinanderfolgenden \$-Zeichen in der 1. und 2. Spalte in den Text eingefügt werden. Darauf folgt die Zeile mit den geänderten Steuerdaten. Danach kann mit dem Text in der gewohnten Weise fortgefahren werden.

Unter den Programmbeispielen befinden sich auch Textdateien, die dem Anwender die Verfahrensweise anschaulich darstellen.

Die Argumentlisten von **TEXT** und **TEXTS** enthalten jeweils 3 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL TEXT (XK,YK,IUNIT)

oder

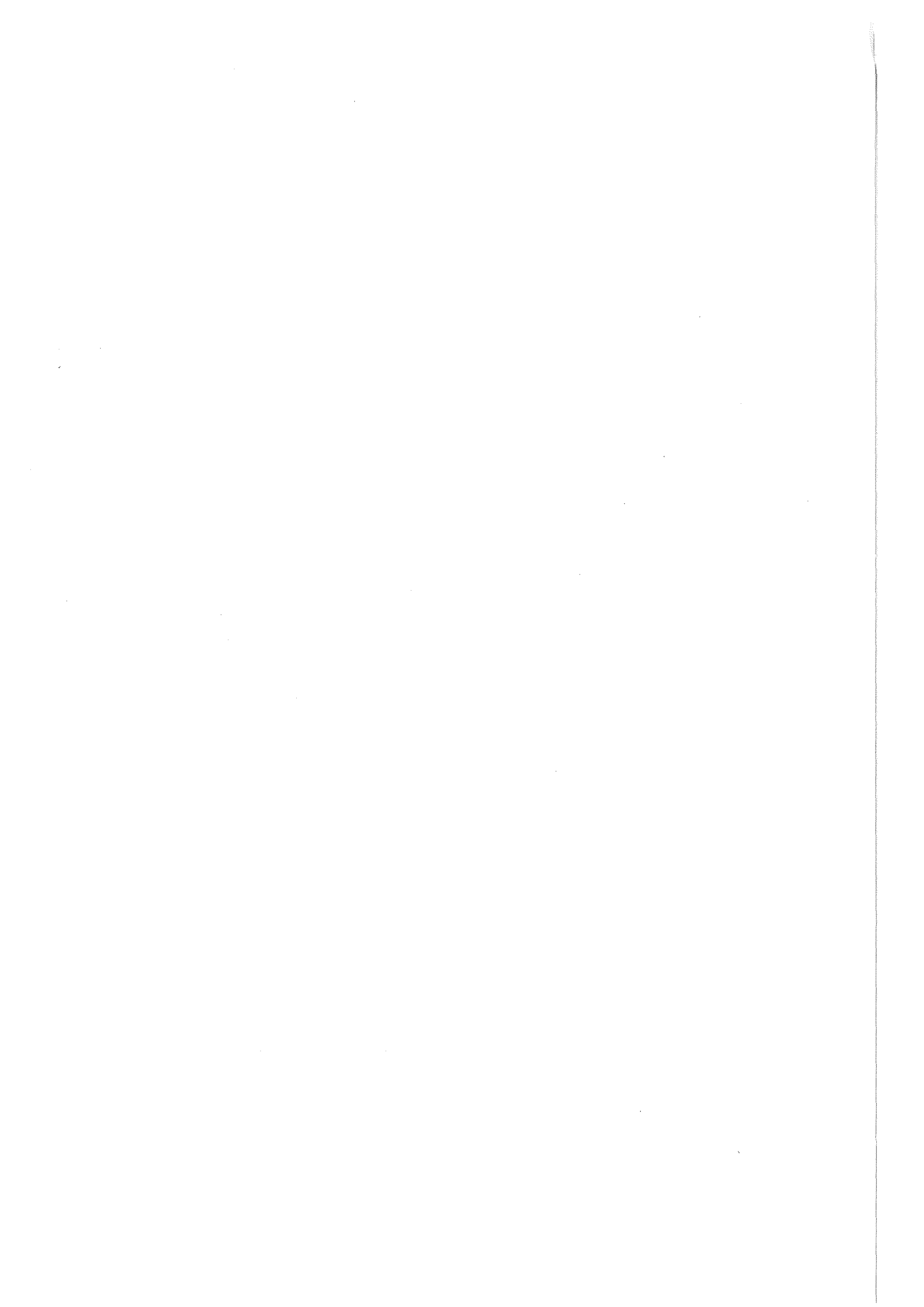
CALL TEXTS (XK,YK,IUNIT)

Argumentbeschreibung:

XK	(cm)	Koordinaten der linken oberen Ecke des Textkomplexes.
YK	(cm)	
IUNIT	(-)	Eingabeeinheit, über die ein zu zeichnender Text vom Programm gelesen wird [READ(IUNIT,...)]. Die Verbindung der Unit mit einer Textdatei muß im Anwenderprogramm vor dem Aufruf von TEXT bzw. TEXTS hergestellt werden (OPEN-Anweisung). Hinweis: Nachdem der Text vom Programm gelesen wurde, erfolgt intern unmittelbar das Zurücksetzen des Dateipointers an den Fileanfang. Damit ist es möglich die gleiche Textdatei wiederholt zu bearbeiten.

Verwendete Unterprogramme und Common's:

GSW_____: OLDPEN, PLOT, PLTEXT, SYMBL2, SYMBL3
TRACE/PC__: PEN, CONV
COMMON__: keine



23. Zeichnen des KfK-Symbols

Für Repräsentationszwecke empfiehlt sich die Integration des KfK-Symbols (einschließlich der ausgezogenen horizontalen Linien) in die graphische Darstellung.

Allgemeinen Richtlinien zufolge ist der Ort des KfK-Symbols auf dem Zeichnungsträger dem Format (DIN-Norm) entsprechend festgelegt. Prinzipiell ist es aber auch möglich das KfK-Symbol an beliebiger Stelle auf der Zeichenfläche zu positionieren. Es ist lediglich darauf zu achten, daß im Symbolbereich keine Überzeichnungen stattfinden.

Das KfK-Symbol wurde für das DIN-A4-Format in Hoch- und Querdarstellung programmiert. Mit der GSW-Basis-Routine FACTOR [2] läßt sich jedoch jedes andere Format einstellen.

Die Darstellung des KfK-Symbols orientiert sich am zuletzt festgelegten Bezugspunkt [CALL PLOTP(.....,-3)]. Im Standardfall ist dies die linke untere Ecke der Zeichenfläche. Der horizontale und vertikale Abstand des KfK-Symbols (linker Rand der ausgezogenen horizontalen Linie) vom Bezugspunkt ist abhängig vom gewählten Format für die Zeichenfläche. Zur Orientierung sind umseitig in einer Tabelle die relevanten Werte für das Standardformat DIN-A4 angegeben.

Die Argumentliste des Unterprogramms KfK enthält 2 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL KfK (IFORM,INST)

Argumentbeschreibung:

IFORM Steuervariable, der lediglich die beiden Werte 40 oder 41 zugewiesen werden können. Der Wert 40 gibt an, daß auf ein DIN-A4-Querformat, der Wert 41 auf ein DIN-A4-Hochformat Bezug genommen wird.

INST CHARACTER*4-Variable, mit der, in der Darstellungsform eines Literals, die Möglichkeit geboten wird, an das KfK-Symbol eine Bezeichnung (meist der Institutsname) zu schreiben.

Beispiel: Auf einer DIN A4-Zeichenfläche im Querformat soll das KfK-Symbol in der Standardanordnung dargestellt werden. Der Name des Instituts, hier IRB, soll ebenfalls erscheinen.

CALL KFK (40,' IRB')

Beispiel: Eine DIN A6-Zeichenfläche im Hochformat soll umrahmt und mit dem KfK-Symbol versehen werden. Infolge unzureichenden Platzbedarfs für einen unterzubringenden Zeichnungstext ist eine zusätzliche vertikale Verschiebung des KfK-Symbols um YD cm erforderlich. Die Zusatzbezeichnung soll entfallen.

...	Kommentar:
CALL CADRE(10.5,10.5*SQRT(2.))	(1) DIN-A6-Rahmen (hoch)
CALL PLOTP(0.,YD,-3)	(2) vert. Verschiebung um YD cm
CALL FACTOR(10.5/21.0)	(3) künftiger Plotfaktor (A6/A4)
CALL KFK (41,' ')	(4) KfK-Symbol
CALL FACTOR(1.)	(5) Reset Plot-Faktor
CALL PLOTP(0.,-YD,-3)	(6) Reset Bezugspunkt
...	

Relevante Daten für das Standardformat DIN-A4

DIN-A4 Format	IFORM (-)	Format horiz./vert. (cm)	Position des KfK-Symbols* zum Bezugspunkt (cm)
hoch	41	21.0/29.7	0.5/4.25
quer	40	29.7/21.0	0.5/3.00

Jedes andere Format ist mit Hilfe des Unterprogramms **FACTOR** einstellbar, indem auf das DIN-A4-Format Bezug genommen wird.

)* Linker Rand der ausgezogenen horizontalen Linie.

Verwendete Unterprogramme und COMMON's:

GSW_____ : PLOT, PLTEXT
 TRACE/PC__ : keine
 COMMON___ : keine

24. Zeichnen von Kreisen oder Kreisteilen

Mit dem Unterprogramm CERCLE lassen sich Kreise bzw. Kreisteile zeichnen. Die Argumentliste von CERCLE enthält 6 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL CERCLE (XP,YP,WA,WE,R,IA)

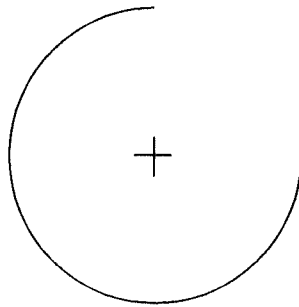
Argumentbeschreibung:

XP	(cm)	Koordinaten, die die Kreisposition in Relation auf den zuletzt definierten Bezugspunkt angeben. Die Koordinaten können dabei den Mittelpunkt des Kreises oder dessen Anfangspunkt bei WA darstellen. Siehe hierzu auch das Argument IA .
YP	(cm)	
WA	(°)	Winkel, bei denen die Kreisdarstellung begonnen (WA) und beendet (WE) wird. Es gilt der mathematische Drehsinn (gegen den Uhrzeigersinn). Bedingungen: $WE > WA; (WE - WA) \leq 362^\circ$
WE	(°)	
R	(cm)	Der Betrag von R gibt den Radius des zu zeichnenden Kreises bzw. Kreisteils an. Ist R negativ, wird der Kreis oder dessen Fragment entsprechend der aktuellen Farbe (PEN) gefüllt dargestellt.
IA	(-)	Steuervariable mit folgender Bedeutung: Die Koordinaten XP und YP stellen bei IA = 0 den Mittelpunkt des Kreises dar. ≠ 0 den Anfangspunkt des Kreises bei WA dar.

TRACE/PC V.2.0 - CERCLE

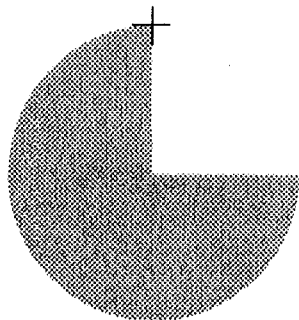
Hinweis: Die Genauigkeit der Kreisdarstellung ergibt sich aus der Anzahl der Stützpunkte, die immer für jedes Grad bestimmt werden.

Beispiel: CALL CERCLE (XP,YP,90.,360.,1.9,0)



+ : XP, YP; (IA = 0)

Beispiel: CALL CERCLE (XP,YP,90.,360.,1.9,1)



+ : XP, YP; (-R, IA ≠ 0)

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT, SHADE

TRACE/PC__: keine

COMMON__: keine

25. Zeichnen von Ellipsen bzw. Ellipsenteilen

Mit dem Unterprogramm ELLIPS lassen sich Ellipsen bzw. Ellipsenteile zeichnen. Die Argumentliste von ELLIPS enthält 7 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

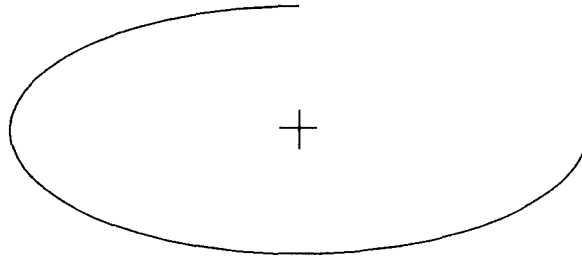
CALL ELLIPS (XP,YP,WA,WE,A,B,IA)

Argumentbeschreibung:

XP	(cm)	Koordinaten, die die Ellipsenposition in Relation auf den zuletzt definierten Bezugspunkt angeben. Die Koordinaten können dabei den Mittelpunkt der Ellipse oder deren Anfangspunkt bei WA darstellen. Siehe hierzu auch das Argument IA.
YP	(cm)	
WA	(°)	Winkel, bei denen die Ellipsendarstellung begonnen (WA) und beendet (WE) wird. Es gilt der mathematische Drehsinn (gegen den Uhrzeigersinn). Bedingungen: $WE > WA; (WE - WA) \leq 362^\circ$
WE	(°)	
A	(cm)	Der Betrag von A gibt die große Halbachse der zu zeichnenden Ellipse oder des Ellipsenteils an. Ist A negativ, wird die Ellipse oder deren Fragment entsprechend der aktuellen Farbe (PEN) gefüllt dargestellt.
B	(cm)	Kleine Halbachse der Ellipse.
IA	(-)	Steuervariable mit folgender Bedeutung: Die Koordinaten XP und YP stellen bei IA = 0 den Mittelpunkt der Ellipse dar. ≠ 0 den Anfangspunkt der Ellipse bei WA dar.

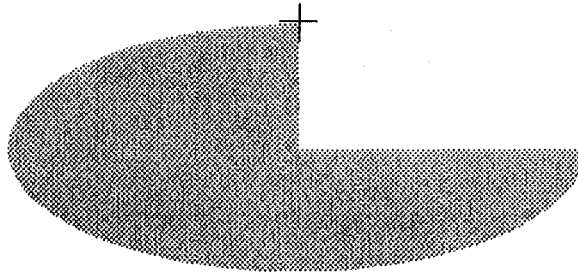
Hinweis: Die Genauigkeit der Ellipsendarstellung ergibt sich aus der Anzahl der Stützpunkte, die immer für jedes Grad bestimmt werden.

Beispiel: CALL ELLIPS (XP,YP,90.,360.,3.9,1.7,0)



+ : XP, YP; (IA = 0)

Beispiel: CALL ELLIPS (XP,YP,90.,360.,-3.9,1.7,1)



+ : XP, YP; (-A, IA ≠ 0)

Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT, SHADE

TRACE/PC__: keine

COMMON__: keine

26. Zeichnen von Geraden mit unterscheidbarer Liniencharakteristik

Das Unterprogramm **STRIP** gestattet die Darstellung von Linien (Geraden) mit unterscheidbarer Charakteristik.

Neben der durchgezogenen Vollinie ist auch die gestrichelte und die strichpunktierte Darstellung einer Geraden realisierbar.

Die Argumentliste von **STRIP** enthält 7 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL STRIP (XS,YS,XZ,YZ,SL,SP,IST)

Argumentbeschreibung:


XS	(cm)	Startkoordinaten der darzustellenden Geraden.
YS	(cm)	
XZ	(cm)	Zielkoordinaten der darzustellenden Geraden.
YZ	(cm)	
SL	(cm)	Länge der Teilstriche (Fragmente) bei gestrichelten bzw. strichpunktierten Linien (IST = 0 od. 1). Für den Fall, daß eine Vollinie (IST =2) gezeichnet werden soll, ist der SL zugewiesene Wert bedeutungslos. Hinweis: Der "Punkt" hat bei der strichpunktierten Darstellung immer eine Länge von 0.1cm.
SP	(cm)	Leerraum (Space) zwischen zwei Teilstrichen bei den gestrichelt gezeichneten Linien (IST =0). Bei der strichpunktierten Darstellung einer Geraden (IST =1) wird der Leerraum um den Platzbedarf für den "Punkt" (0.1cm) reduziert. Diesem Umstand muß bei der Wertzuweisung Rechnung getragen werden. SP sollte daher mindestens 0.3cm betragen.

Im Fall $IST=2$ (Vollinie) ist der SP zugewiesene Wert bedeutungslos.

IST (-) Steuervariable, mit der die Strichcharakteristik gewählt wird:
 $IST = 0$ gestrichelt
 $= 1$ strichpunktiert
 $= 2$ durchgezogen

Beispiele zur Strichcharakteristik:

$IST = 0; (SL, SP) \text{ ______} :$ 

$IST = 1; (SL, SP > = .3) :$ 

$IST = 2 \text{ ______} :$ 

Verwendete Unterprogramme und COMMON's:

$GSW \text{ ______} :$ PLOT
 $TRACE/PC \text{ ______} :$ keine
 $COMMON \text{ ______} :$ keine

27. Zeichnen von Linien und Pfeilen

Mit dem Unterprogramm **FLECHE** lassen sich zwei durch ihre Koordinaten vorgegebenen Punkte mit einer durchgezogenen Linie verbinden. Darüber hinaus besteht die Möglichkeit die Enden der Linie mit Pfeilspitzen zu versehen.

Die Pfeilgröße (Spitze) entspricht der DIN-Norm.

Die Argumentliste von **FLECHE** enthält 5 Elemente, die beim Aufruf initialisiert sein müssen.

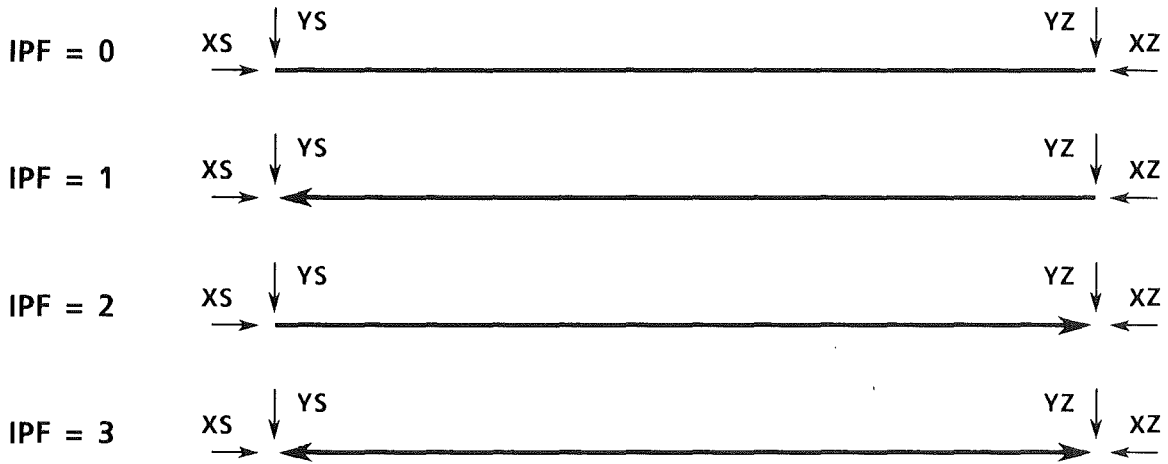
Aufruf:

CALL FLECHE (XS,YS,XZ,YZ,IPF)

Argumentbeschreibung:

XS	(cm)	Startkoordinaten der zu zeichnenden Linie (Pfeil).
YS	(cm)	
XZ	(cm)	Zielkoordinaten der zu zeichnenden Linie (Pfeil).
YZ	(cm)	
IPF	(-)	<p>Steuervariable, mit der die Wahl getroffen wird, ob eine Linie mit Pfeilspitzen zu versehen ist und wenn ja, an welchen Enden diese zu plazieren sind.</p> <p>Folgende Optionen stehen zur Auswahl:</p> <p>IPF = 0 Start- und Zielpunkt werden mit einer geraden Linie verbunden.</p> <p>= 1 Auf den Startpunkt weist eine Pfeilspitze, sonst wie bei IPF=0.</p> <p>= 2 Auf den Zielpunkt weist eine Pfeilspitze, sonst wie bei IPF=0.</p> <p>= 3 Sowohl auf den Start- als auch auf den Zielpunkt weisen Pfeilspitzen, sonst wie bei IPF=0.</p>

Beispiele:



Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT
TRACE/PC_: keine
COMMON__: keine

28. Zeichnen von schraffierten oder ungeschraffierten Flächen (Balken)

Mit Hilfe des Unterprogramms **BALKEN** können im allgemeinen Anwendungsfall orthogonale Flächen mit der Ausdehnung (X, Y) auf der Zeichenebene dargestellt werden. Die linke untere Rahmenecke ist dabei identisch mit dem zuletzt festgelegten Bezugspunkt [CALL PLOTP (...,-3)].

Die definierte Fläche kann wahlweise unter jedem Winkel und mit beliebigem Feinheitsgrad schraffiert werden.

Das Unterprogramm **BALKEN** ist insbesondere zur Konstruktion von Histogrammen geeignet.

Die Argumentliste von **BALKEN** enthält 4 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

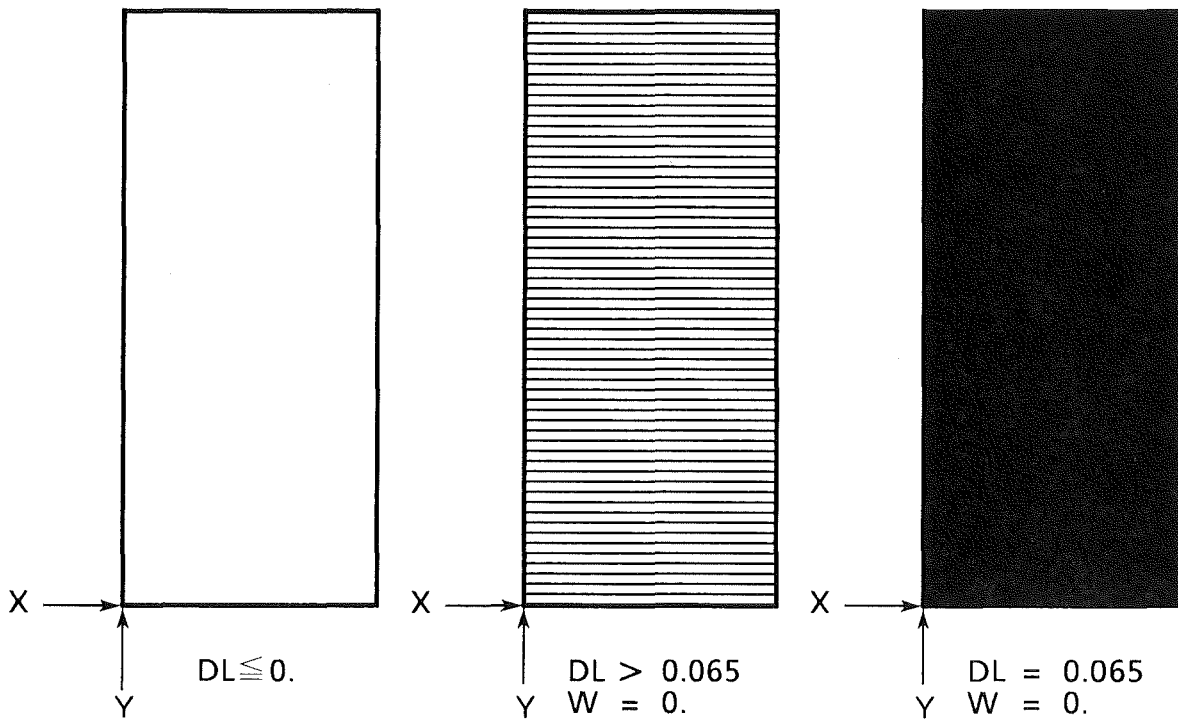
CALL BALKEN (X,Y,W,DL)

Argumentbeschreibung:

X	(cm)	Die beiden Argumente X und Y bestimmen die Ausdehnung der orthogonalen Fläche auf der Zeichenebene. Die linke untere Ecke der Fläche ist dabei mit dem aktuellen Bezugspunkt identisch.
Y	(cm)	
W	(°)	Winkel, unter dem die Schraffurgeraden gezeichnet werden sollen. Wird keine Schraffur gewünscht ($DL \leq 0.$), ist der hier angegebene Wert bedeutungslos.
DL	(cm)	Abstand der Schraffurgeraden. Bei $DL \leq 0.$ wird die Schraffur unterdrückt.

Hinweis: Mit einem hinreichend kleinen Wert für **DL** kann auch eine farblich gefüllte Fläche erzeugt werden. Der Wert von 0.065 cm sollte dabei jedoch nicht unterschritten werden. Zumindest bei Bildschirmgraphiken wäre dies wirkungslos (Auflösung gesättigt) und würde zudem nur unnützen Rechenaufwand verursachen.

Beispiele:



Verwendete Unterprogramme und COMMON's:

GSW_____: PLOT, SHADE

TRACE/PC__: keine

COMMON__: keine

29. Zeichnen eines Rahmens mit integriertem Fenster

Das Unterprogramm SFCAD zeichnet einen Außen- und einen Innenrahmen (leeres Fenster). Die Fläche zwischen den Rahmen kann dabei schraffiert, farbegefüllt (sehr dicht schraffiert) oder leer bleiben.

Der Anwendungsbereich von SFCAD ist sehr allgemein. So kann SFCAD beispielsweise für Histogramme, einfache Konstruktionen, Texthervorhebungen usf. verwendet werden.

Die Argumentliste von SFCAD enthält 13 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL SFCAD (XK,YK,XD,YD,DR,DO,DL,DU,IRC,ISF,IFC,DS,WS)

Argumentbeschreibung:

XK	(cm)	Koordinaten der linken unteren Rahmenecke (äußerer Rahmen) bezogen auf den zuletzt definierten Bezugspunkt.
YK	(cm)	
XD	(cm)	Ausdehnung des Rahmens (außen) in horizontaler (XD) und vertikaler (YD) Richtung.
YD	(cm)	
DR	(cm)	Differenz zwischen den Rahmenseiten von Außen- und Innenrahmen (Rahmendicke). Dabei bedeutet DR : Differenz rechts
DO	(cm)	
DL	(cm)	
DU	(cm)	
		DO : " oben
		DL : " links
		DU : " unten
IRC	(-)	Umrandungsfarbe der Rahmen. Siehe hierzu auch die Farbzusammenstellungstabelle bei der Beschreibung des Unterprogramms PEN.
ISF	(-)	Steuervariable, mit der die Füllstruktur der Fläche zwischen den Rahmen bestimmt wird.

Folgende Optionen sind anwendbar:

ISF = 0 Die Fläche bleibt leer.

< 0 Die Fläche wird schraffiert, wobei der Abstand der Schraffurgeraden **DS** cm beträgt.

> 0 Die Fläche wird farbgefüllt (sehr dicht schraffiert).

IFC (-)

Schraffur- bzw. Füllfarbe.

Nur relevant wenn **ISF** ein Wert $\neq 0$ zugewiesen wurde.

DS (cm)

Abstand der Schraffurgeraden.

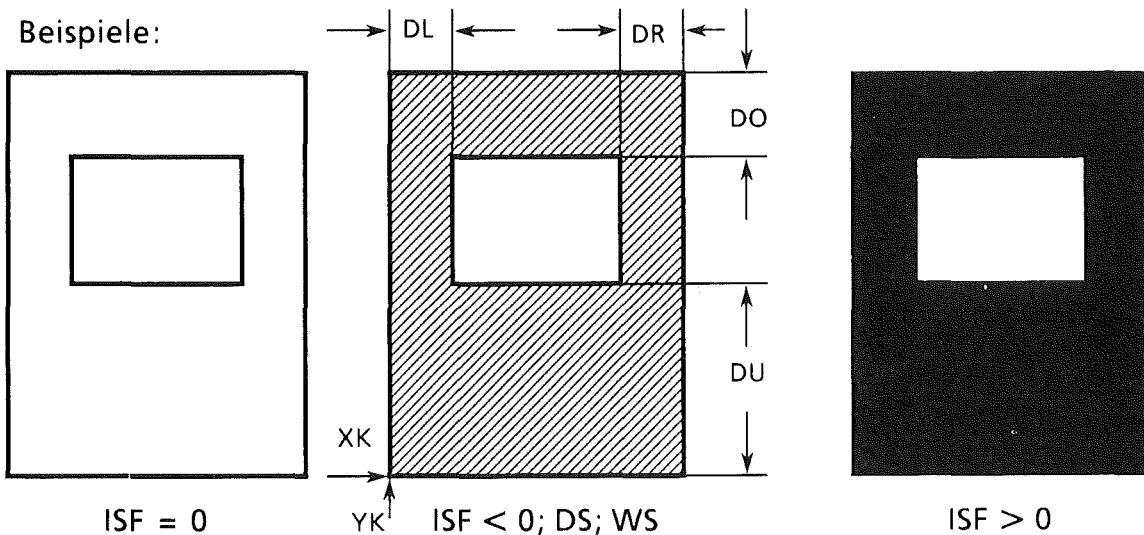
Eine realistische Wertzuweisung ist nur im Fall **ISF** < 0 erforderlich.

WS (cm)

Winkel, unter dem die Schraffurgeraden gezeichnet werden sollen.

Nur relevant im Fall **ISF** < 0.

Beispiele:



Verwendete Unterprogramme und COMMON's:

GSW_____: OLDPEN, SHADE

TRACE/PC__: CURVLV, PEN

COMMON__: keine

30. Zeichnen eines cm- bzw. Zoll-Rasters zu Orientierungszwecken

Das Unterprogramm **RASTER** zeichnet ein cm- bzw. Zoll-Raster mit entsprechenden Maßangaben. Es handelt sich hierbei lediglich um eine graphische Hilfsroutine, die bei der Programmentwicklung für das gezielte Plazieren von graphischen Objekten (Texte, Konstruktionen, Diagramme usw.) auf dem Zeichnungsträger eine wertvolle Orientierungshilfe bietet.

Mit der (temporären) Installation eines Rasters kann der Anwender auf einfache Weise diejenigen Orte (Koordinaten) lokalisieren, auf die dann später bei der Positionierung zusätzlich zu zeichnender Objekte Bezug genommen werden kann. Die linke untere Ecke des Rasterfeldes ist dabei mit dem aktuellen Bezugspunkt (`CALL PLOTP(...,-3)`) identisch.

Insbesondere geeignet erscheint die Anwendung von **RASTER** bei der Bildschirmgraphik. Das Lokalisieren einer Position ist hierbei durch die oft verzerrte Bildwiedergabe nur schätzungsweise möglich.

Die Argumentliste von **RASTER** enthält 3 Elemente, die beim Aufruf initialisiert sein müssen.

Aufruf:

CALL RASTER (XD,YD,INCM)

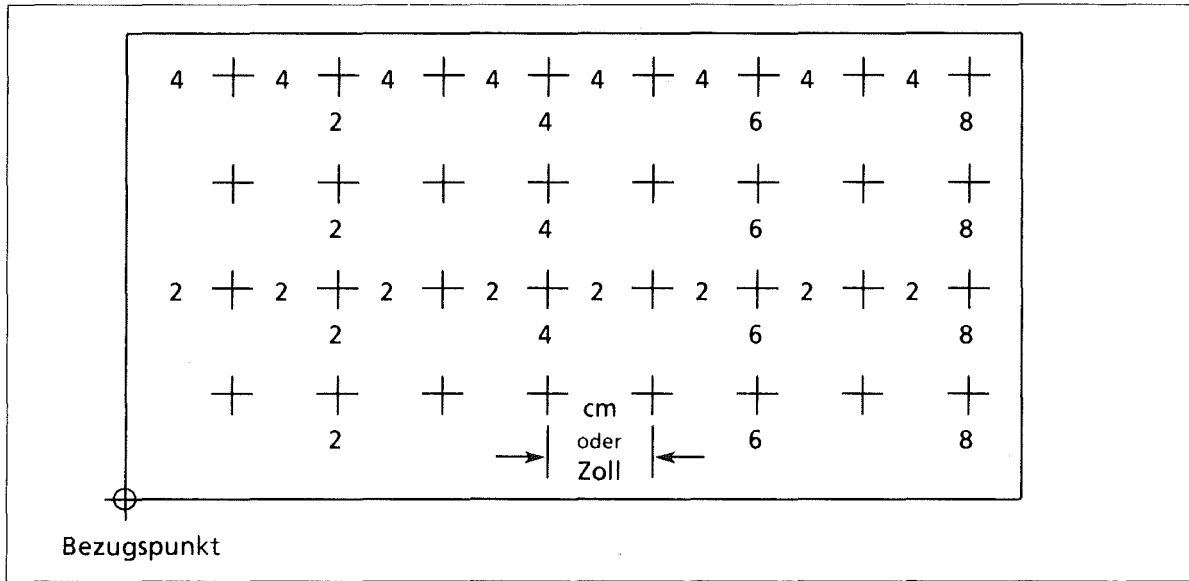
Argumentbeschreibung:

XD	(cm)	Ausdehnung des Rasters in horizontaler (XD)
YD	(cm)	und vertikaler (YD) Richtung.
INCM	(-)	Steuervariable, mit der der Rastertyp festgelegt wird.
		INCM = 0 cm-Raster
		≠ 0 Zoll-Raster

Die zeichnerische Darstellung des Rasters erfolgt in Form von Kreuzmarken. Diese werden entsprechend dem gewählten Rastertyp im cm- bzw. Zoll-Abstand in das definierte Feld gezeichnet. Dazu wird im 2 cm- bzw. 2 Zoll-Abstand links von einem betrachteten Kreuz die vertikale, unterhalb von diesem die horizontale Position als Zahlenwert dargestellt.

TRACE/PC V.2.0 - RASTER

Beispiel:



Verwendete Unterprogramme und COMMON's:

GSW_____: keine

TRACE/PC__: NUMBP, SYMBP1

COMMON__: keine

31. Zeichnerische Darstellung von Symbolen aus dem DV- bzw. PC-Hardware-Bereich

Es wird im folgenden eine Reihe von Unterprogrammen vorgestellt, die nicht der Grundsoftware angehören. Es handelt sich hierbei um nicht genormte, abstrahierte Darstellungen aus der DV- bzw. PC-Zubehör-Umgebung. Dem Anwender werden damit Möglichkeiten geboten, Konfigurationsvorstellungen und Datenflüsse zeichnerisch zu realisieren.

Die hier vorgestellten Symbole werden im allgemeinen in der zuvor mit PEN gewählten Farbe gezeichnet. Das farbliche Hinterlegen eines Symbols mit der aktuellen Farbe ist ebenfalls möglich. In diesem Fall werden alle Konturen und Bezeichnungen schwarz gezeichnet. Die Verfahrensweise wird bei der jeweiligen Modulbeschreibung erläutert.

Tabellarische Übersicht der verwendeten Unterprogramme

Modul → Library ↓	DISY	TASY	ECRAN	GRAPH	IMPRIM	CONSOL	SYSUNI
GSW:							
PLOT	•	•	•	•	•	•	•
PLTEXT					•	•	•
SHADE	•		•		•		
TRACE/PC:							
CADRE			•		•	•	•
CADRE2				•		•	•
CERCLE		•		•			•
CFILL				•		•	•
FLECHE				•			
ORGPEN	•	•	•	•	•	•	•
PEN	•	•	•	•	•	•	•
SYMBP1			•	•			
SYMBS1				•			

31.1 Zeichnen eines Magnetplattensymbols (Fixed Disk)

Aufruf:

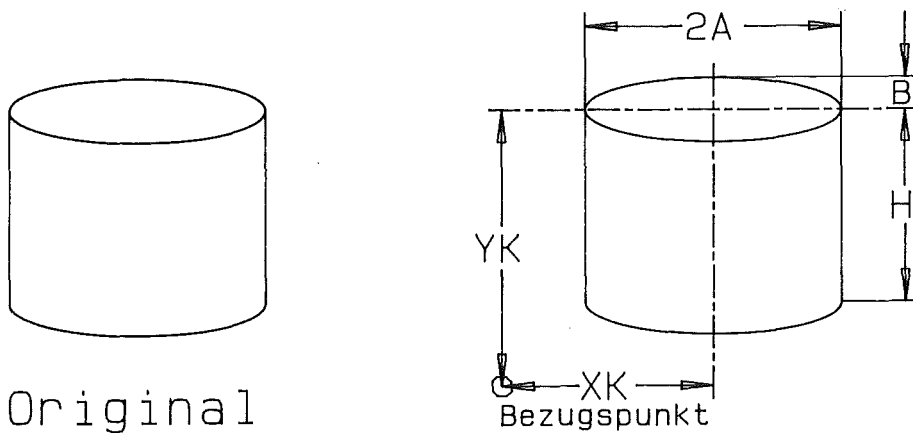
CALL DISY (XK,YK,A,B,H)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols in Relation zum aktuellen Bezugspunkt (siehe Skizze).
YK	(cm)	
A	(cm)	Große (A) und kleine (B) Halbachse der elliptischen Zylinderoberfläche.
B	(cm)	
H	(cm)	Der Betrag von H gibt die Höhe des Zylinders an. Ist H negativ, wird das Symbol farblich hinterlegt. Symbolhöhe = $H + 2 \cdot B$.

Darstellung:



3 1.2 Zeichnen eines Datenspeichersymbols (Tape, Cartridge)

Aufruf:

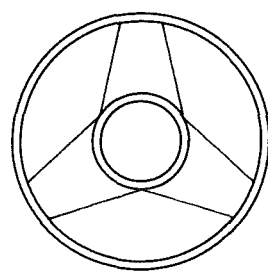
CALL TASY (XK,YK,R)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

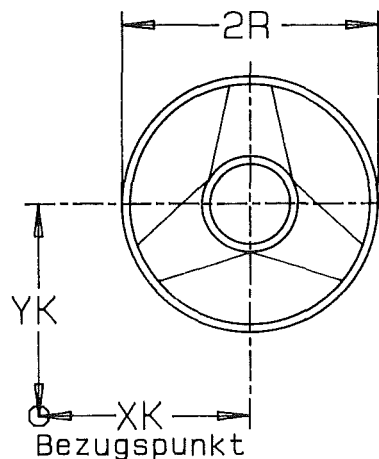
Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols (Zentrum) in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
R	(cm)	Der Betrag von R gibt den Außenradius des Symbols an. Ist R negativ, wird das Symbol farblich hinterlegt.

Darstellung:



Original



31.3 Zeichnen eines Bildschirmsymbols (Screen)

Aufruf:

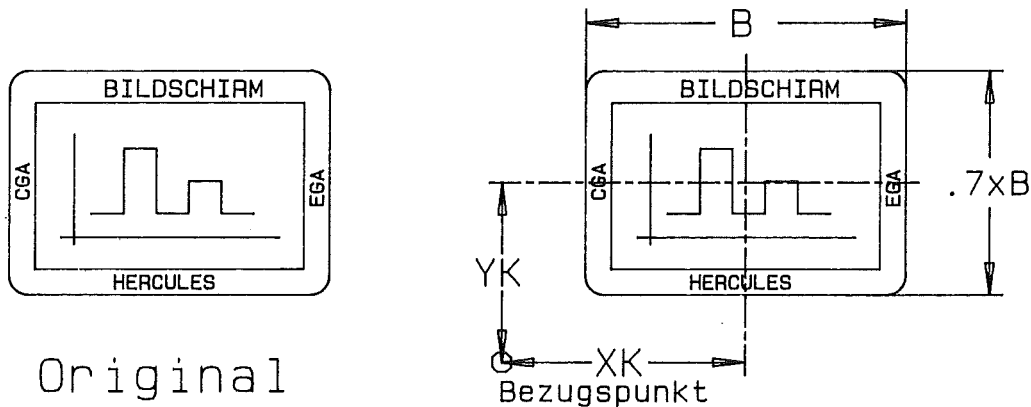
CALL ECRAN (XK, YK, B)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols (Zentrum) in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
B	(cm)	Der Betrag von B gibt die Breite des Symbols an. Ist B negativ, wird das Symbol farblich hinterlegt. Symbolhöhe = $0.7 \cdot B$.

Darstellung:



31.4 Zeichnen eines Zeichengerätsymbols (Plotter)

Aufruf:

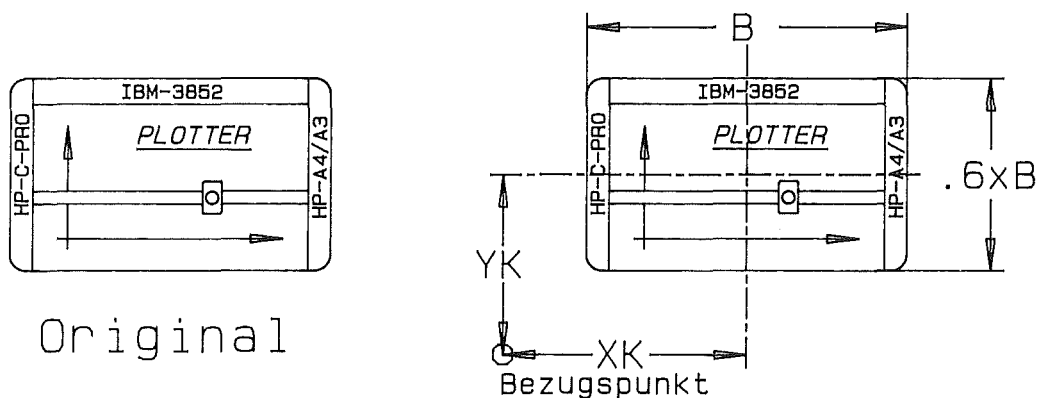
CALL GRAPH (XK,YK,B)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols (Zentrum) in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
B	(cm)	Der Betrag von B gibt die Breite des Symbols an. Ist B negativ, wird das Symbol farblich hinterlegt. Symbolhöhe = $0.6 \cdot B$.

Darstellung:



31.5 Zeichnen eines Druckersymbols (Printer)

Aufruf:

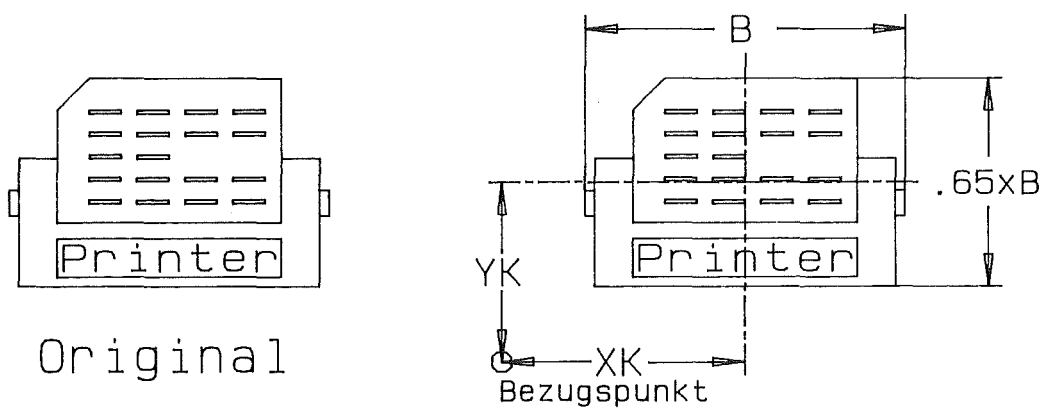
CALL IMPRIM (XK,YK,B)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols (Zentrum) in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
B	(cm)	Der Betrag von B gibt die Breite des Symbols an. Ist B negativ, wird das Symbol farblich hinterlegt. Symbolhöhe = $0.65 \cdot B$.

Darstellung:



31.6 Zeichnen eines Tastatursymbols (Konsole)

Aufruf:

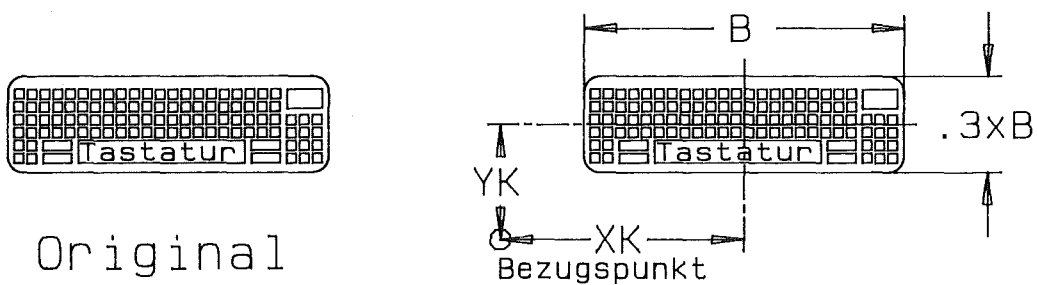
CALL CONSOL (XK,YK,B)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols (Zentrum) in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
B	(cm)	Der Betrag von B gibt die Breite des Symbols an. Ist B negativ, wird das Symbol farblich hinterlegt. Symbolhöhe = $0.3 \cdot B$.

Darstellung:



31.7 Zeichnen eines Symbols für eine Systemeinheit (PC-Unit)

Aufruf:

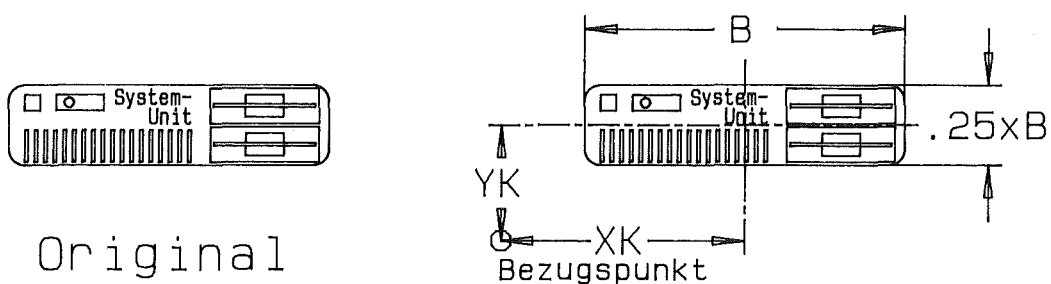
CALL SYSUNI (XK,YK,B)

Alle Elemente der Argumentliste stellen Eingangsparameter dar, die beim Aufruf initialisiert sein müssen.

Argumentbeschreibung:

XK	(cm)	Koordinaten zur Positionierung des Symbols (Zentrum) in Relation zum aktuellen Bezugspunkt.
YK	(cm)	
B	(cm)	Der Betrag von B gibt die Breite des Symbols an. Ist B negativ, wird das Symbol farblich hinterlegt. Symbolhöhe = $0.25 \cdot B$.

Darstellung:



32. Bestimmung geeigneter Parameter zur Definition von linearen Achsen

Das Unterprogramm **ROUND** dient im Bedarfsfall als Hilfsmittel zur Bestimmung geeigneter Parameter für die Konstruktion von linearen Achsen (**AXLIN1**, **AYLIN1**).

Die Anwendung von **ROUND** ist beispielsweise dann sinnvoll, wenn irgendeine Datenmenge in ein Koordinatensystem gezeichnet werden soll ohne daß Informationen über den darzustellenden Wertebereich vorliegen. Es sind hierzu lediglich die Extremwerte der Datenmenge zu bestimmen, die die Eingangsparameter für **ROUND** darstellen. Nach dem Aufruf von **ROUND** sind alle relevanten Parameter zur Definition einer linearen Achse verfügbar.

Diese Vorgehensweise kann sich sowohl auf die Definition der Abszisse als auch auf die der Ordinate oder auf beide beziehen.

Die Argumentliste von **ROUND** enthält 6 Elemente. Zwei Elemente stellen dabei sowohl Eingangs- als auch Ausgangsparameter dar, alle anderen nur Ausgangsparameter. Sie müssen daher in Form von Variablen in der Argumentliste erscheinen.

Aufruf:

CALL ROUND(WMIN,WMAX,WANF,DW,IM,IGEN)

Argumentbeschreibung:

WMIN	(phys. Einh.)	Kleinster (WMIN) und größter (WMAX) Wert einer Datenmenge. WMIN und WMAX stellen zunächst Eingabeparameter dar, die nach erfolgter Rundung überschrieben (!) zurückgegeben werden. Gültigkeitsbereich für die Eingabe: $ WMAX \leq 10^8$ und $[WMAX-WMIN] \geq 10^{-4}$.
WMAX	(phys. Einh.)	
WANF	(phys. Einh.)	Rückgabeparameter. Es wird ein Wert zurückgegeben, der mit einem großen Teilstrich und der entsprechenden Graduation an der betrachteten Achse identisch ist.

TRACE/PC V.2.0 - ROUND

Dabei gilt:

$$WMIN_{\text{gerundet}} \leq WANF \leq WMAX_{\text{gerundet}}$$

DW	(phys. Einh.)	Rückgabeparameter, der angibt, nach wieviel physikalischen Einheiten jeweils ein großer Teilstrich mit der entsprechenden Graduation gezeichnet werden soll (Differenz zwischen zwei benachbarten Graduationen).
IM	(-)	Rückgabeparameter, der die Anzahl der kleinen Teilstriche zwischen zwei Graduationen repräsentiert.
IGEN	(-)	Rückgabeparameter. Er stellt die Anzahl der Nachkommastellen für die zu zeichnenden Graduationen dar. Es gilt: IGEN = -1: Graduationen ohne Dezimalpunkt. = 0: Graduationen nur mit Dezimalpunkt. = 1: Graduationen mit einer Nachkommastelle. usf..

Verwendete Unterprogramme und COMMON's:

GSW_____ : keine
TRACE/PC__ : DT, HRMG
COMMON___ : keine

33. Perspektivische Darstellung einer Funktion in Abhängigkeit zweier Variablen.

Es wird hier ein Programm vorgestellt, das bereits seit einigen Jahren an der KfK-Großrechenanlage unter dem Modulnamen DT PLOT [4] verfügbar ist. Es handelt sich dabei um ein Fortran-IV-Unterprogramm zur graphischen Darstellung von Funktionen zweier Variablen in Zentral- und Parallelprojektion.

Infolge eines allgemeinen Bedarfs, den Modul auch auf PCs verfügbar zu haben, lag es nahe DT PLOT mit seinen Hilfsunterprogrammen DTPLO1, DTPLO2 und DTPLO3 in das TRACE/PC-System ("TPC.LIB"-Bibliothek) zu integrieren.

Mit relativ geringem Aufwand konnte die Kompatibilität zum Ryan-McFarland FORTRAN 77 hergestellt werden. Hinsichtlich der verwendeten Graphik-Call's gab es keine Anpassungsprobleme da gleichnamige und funktionsidentische Graphik-Unterprogramme in TRACE/PC und GSW [2] verfügbar waren.

Da sich hinsichtlich der Handhabung des Programms nichts änderte, wird auf die detaillierte Beschreibung des Autors von DT PLOT [4] verwiesen.

Verwendete Unterprogramme und COMMON's,

GSW _____: NUMBER, PLOT, SYMBOL
 TRACE/PC ____: PEN
 COMMON ____: DDD



3.4. Visualisierung eines Bildfile's

In Bildfile's gespeicherte graphische Informationen, die den GSW-TRACE/PC-Konventionen entsprechen, können mit der installierten Software sowohl auf Zeichengeräten (Plottern) als auch auf Bildschirmen visualisiert werden. Siehe hierzu auch den Punkt "Installation von TRACE/PC".

Die zeichnerische Wiedergabe eines Bildfile's auf einem Stiftplotter erfolgt mit Hilfe des DOS-Kommandos "copy". Darüber hinaus muß das Schnittstellenprotokoll gesetzt sein ("mode com1:9600,n,8,1,p").

Soll beispielsweise das Bildfile "bild.pic" über die serielle Schnittstelle ausgegeben werden, sieht die Anweisung hierfür wie folgt aus:

```
copy \path\bild.pic aux
```

Zur Visualisierung eines Bildfile's auf einem Bildschirm ist für die betrachtete Gerätekonfiguration die Interrupt Service Routine EGAIN.T.COM und der Bildschirmtreiber PLTDRV.SYS erforderlich. Auch hier läßt sich die gespeicherte Bildinformation mit dem DOS-Kommando "copy" auf dem Bildschirm visualisieren. Für das Bildfile "bild.pic" lautet die vollständige Anweisung :

```
copy/b \path\bild.pic plt
```

Bei der Bildschirmausgabe muß eventuell einem Umstand Rechnung getragen werden, der lediglich bei Anwendung der "Print Screen"-Option in Verbindung mit einem Drucker (z.B. Farbstrahlprinter IBM 3852 2) eine gewisse Relevanz hat. Es kann dabei vorkommen, daß das Bildschirmabbild verzerrt wiedergegeben wird. Die Ursache besteht darin, daß der Bildschirmtreiber den graphischen Inhalt auf der maximal möglichen Bildfläche abbildet, die Seitenverhältnisse von Bildfläche und Graphik aber nicht identisch sind.

Am Bildschirm selbst könnte man sich hardwareseitig durch Justieren der vertikalen Ausdehnung helfen um optisch einen unverzerrten Eindruck zu erhalten. Der Drucker hingegen läßt sich nicht auf diese Weise beeinflussen.

Abhilfe für dieses Problem bietet softwareseitig der Bildschirmtreiber selbst. Der Anwender erstellt sich hierzu auf manuelle Weise ein Bildfile, z.B. mit dem Namen "setform.pic". Der einzeilige Inhalt des Bildfile's hat dabei folgendes Aussehen:

```
IN;IFxparm,yparm;SP0;
```

Die beiden Parameter "xparm" und "yparm" stehen stellvertretend für die horizontale bzw. vertikale Ausdehnung des darzustellenden Bereichs. Die Parameter sind wie folgt definiert:

$$\begin{aligned} \text{xparm} &= 10000 \times \text{x-Faktor}, \quad 0 < \text{x-Faktor} \leq 1 \\ \text{yparm} &= 10000 \times \text{y-Faktor}, \quad 0 < \text{y-Faktor} \leq 1 \end{aligned}$$

Für TRACE/PC-Anwendungen in Verbindung mit der betrachteten Gerätekonfiguration hat sich für xparm ein Wert von 10000 und für yparm ein Wert von 9286 als geeignet erwiesen. Bei Verwendung anderer Geräte können sich davon leicht abweichende Werte einstellen.

Gefunden werden diese Werte indem eine Testzeichnung mit je einer horizontalen und vertikalen Linie gleicher Länge erstellt wird. Durch Ausmessen des Druckerbildes findet man den Faktor und damit den gesuchten Parameter.

Mit nachstehender Anweisung, die vor der ersten Bildausgabe erfolgt, läßt sich dann das Bildschirmfenster wunschgemäß einstellen:

```
copy/b \path\setform.pic plt
```

Darüber hinausgehende Informationen zur Einstellung des Bildschirmschreibers sind der GSW-Betriebsanleitung zu entnehmen.

35. Reservierte COMMON's

Um identische Namen in Anwendungsprogrammen zu vermeiden, werden nachstehend alle in TRACE/PC verwendeten COMMON's aufgelistet. Es handelt sich hierbei grundsätzlich um benannte Common's.

Eine Kurzbeschreibung der Variablen soll unter anderem dazu dienen, die Fehlersuche zu erleichtern.

Bei geeigneter Verwendung der COMMON's mit seinen Variablen kann der Benutzer die Gestaltung einer Graphik über die normalen Möglichkeiten hinaus beeinflussen. Dies soll lediglich als Hinweis dienen. Dem unerfahrenen Anwender ist die Manipulation einer Graphik über die Common's nicht zu empfehlen.

COMMON /CENTER/ XO ,YO

- XO** (cm) Textendekoordinaten eines gezeichneten Textstrings.
YO (cm) Sie werden zum Aneinanderreihen (Fortsetzen) von Texten verwendet.

Die Definition der Variablen erfolgt in den Unterprogrammen PLOTX (default: 0./0.), SYMBP1-4, SYMBS1, NUMBP, NUMBS und FNTX.

In den Hilfsunterprogrammen POSXY und POSFT werden XO und YO für die Positionierung eines Folgetextes benötigt.

COMMON /DDD/ IDV

- IDV** (-) Steuervariable, deren Wert das Ausgabegerät für die Graphik repräsentiert. Dabei bedeutet:
IDV = 0: Ausgabe auf einem Zeichengerät (Plotter) oder Ablage der graphischen Information in einem Bildfile.
 = 1: Bildschirmausgabe.

Der COMMON /DDD/ mit seiner Variablen IDV wurde aus Gründen der Farbkompatibilität eingeführt, da die Zuordnung der Farben bei jedem Ausgabegerät anders ist.

Die Variable IDV wird im Unterprogramm PLOTI in Abhängigkeit des für die Graphik gewählten Ausgabegerätes (Variable IDEV) gesetzt und in PEN abgefragt.

COMMON /FAKTOR/ FX ,FY ,BXMIN ,BYMIN ,LLX ,LLY

FX (cm/phys. Einh.) Maßstabsfaktor für die X-Achse oder ein polares System. Er ist wie folgt definiert:

od. (cm/log(p. E.))
 lin. X-Achse: $XL/(XMAX-XMIN)$ [cm/p. E.]
 log. X-Achse: $XL/ALOG10(XMAX/XMIN)$ [cm/log(p. E.)]
 pol. System: $RL/RMAX$ [cm/p. E.]

FY (cm/phys. Einh.) Maßstabsfaktor für die Y-Achse. Er ist wie folgt definiert:

od. (cm/log(p. E.))
 lin. Y-Achse: $YL/(YMAX-YMIN)$ [cm/p. E.]
 log. Y-Achse: $YL/ALOG10(YMAX/YMIN)$ [cm/log(p. E.)]

BXMIN (phys. Einh.) Kleinster Wert an der X-Achse (linear oder logarithmisch).

BYMIN (phys. Einh.) Kleinster Wert an der Y-Achse (linear oder logarithmisch).

LLX (-) Steuergröße, deren Wert den Typ der X-Achse repräsentiert. Dabei bedeutet:

LLX = 0: lineare X-Achse
 = 1: logarithmische X-Achse

LLY (-) Steuergröße, deren Wert den Typ der Y-Achse repräsentiert. Dabei bedeutet:

LLY = 0: lineare Y-Achse
 = 1: logarithmische Y-Achse

Die COMMON-Variablen werden in den Achsenprogrammen **AXLIN1**, **AYLIN1**, **AXLOG1**, **AYLOG1** und **POL** definiert. Verwendung finden sie in den Unterprogrammen **CURVE**, **XPOS**, **YPOS**, **CURVEP**, **CURVE2**, **CDL**, **CRV** und **CDLP**.

COMMON /FNSTR/ ISTV ,XXUL ,YYUL ,XXOR ,YYOR

ISTV	(-)	Steuervariable, deren Wert darüber entscheidet, ob ein Fenster (Rahmen) in ein kartesisches Koordinatensystem gezeichnet werden soll (ISTV=1) oder nicht (ISTV=0).
XXUL	(cm)	Koordinaten der linken unteren Fensterecke, bezogen
YYUL	(cm)	auf den zuvor definierten Bezugspunkt.
XXOR	(cm)	Koordinaten der rechten oberen Fensterecke, bezogen
YYOR	(cm)	auf den zuvor definierten Bezugspunkt.

Die Variablen des COMMON /FNSTR/ werden in PLOTI (default: ISTV=0) und AXYFEN (ISTV=1) definiert und in AXLIN1, AYLIN1, AXLOG1, AYLOG1, NETLN2 und AXYRES verwendet.

COMMON /ORIGIN/ IOP

IOP	(-)	Variable, deren Wert einen Farbcode (temporär gespeichert) repräsentiert.
------------	-----	---

Der COMMON /ORIGIN/ wird in einigen TRACE/PC-Unterprogrammen verwendet, in denen eine, vom Anwender nicht beeinflussbare, interne Farbgebung erfolgt. Dies ist beispielsweise der Fall beim Zeichnen eines farbgefüllten Kreises. Dabei ist die Farbe der Fläche vorgebbar (PEN-Arg. → IOP). Die Umrandung wird jedoch grundsätzlich mit schwarzer Farbe gezeichnet.

Um eine gezielte Farbzuordnung mit PEN zu gewährleisten, muß die intern gesetzte Farbe wieder mit der ursprünglich vorgegebenen Farbe getauscht werden: CALL PEN (IOP).

Für den Anwender ist der COMMON /ORIGIN/ ohne Bedeutung.

COMMON /S\$C2/ ABX ,ABY

ABX (log(phys. Einh.)) Logarithmus des Minimums an der betrachteten X-Achse.

ABY (log(phys. Einh.)) Logarithmus des Minimums an der betrachteten Y-Achse.

Die Variablen des COMMON /S\$C2/ werden im Unterprogramm CURVE2 definiert und in der Hilfsroutine CDL (zeichnen eines Kurvenverlaufs mit wählbarer Strichcharakteristik) verwendet.

Für den Anwender ist der COMMON /S\$C2/ ohne Bedeutung.

COMMON /UGOG/ YUG ,YOG ,XLG ,XRG

YUG (phys. Einh.) Unterer (YUG) und oberer (YOG) Grenzwert des Gültigkeitsbereichs für die Ordinaten eines Kurvenverlaufs.

XLG (phys. Einh.) Linker (XLG) und rechter (XRG) Grenzwert des Gültigkeitsbereichs für die Abszissen eines Kurvenverlaufs.

Bei polarer Darstellungsweise repräsentieren die Variablen einen Gültigkeitsbereich mit den Grenzen R_{\min} (XLG) und R_{\max} (XRG).

Die Definition der Variablen des COMMON /UGOG/ erfolgt in den Achsenprogrammen AXLIN1, AYLIN1, AXLOG1, AYLOG1 und POL. Verwendet werden sie in den Unterprogrammen CURVE2, C2XP und C2XPP.

36. Hilfsroutinen

Die TRACE/PC-Bibliothek "TPC.LIB" beinhaltet neben den Basisunterprogrammen auch noch eine Reihe von Hilfsroutinen zur Lösung systeminterner Aufgaben. Um identische Namen bei der Benennung von selbst programmierten Modulen in Anwenderprogrammen zu vermeiden, werden im folgenden alle in der Bibliothek "TPC.LIB" befindlichen Hilfsroutinen aufgelistet und ihre Funktion kurz beschrieben. Hinsichtlich ihrer Verwendungsmöglichkeiten sind sie jedoch für den Anwender ohne Bedeutung.

CDL	Koordinatenaufbereitung (kartesisch: $y = f(x)$) für einen Kurvenverlauf mit wählbarer Strichcharakteristik.
CDLP	Koordinatenaufbereitung (polar: $r = f(\phi)$) für einen Kurvenverlauf mit wählbarer Strichcharakteristik.
CFILL	Farbliches Unterlegen (Fill, dichte Schraffur) von bibliotheksinternen Objekten (GRAPH, CONSOL, SYSUNI).
CONV	Umlautkonversion (Unterprogramme TEXT und TEXTS).
CRV	Zeichnen eines durchgezogenen Kurvenverlaufs.
CURVE1	Bestimmung der Koordinaten für einen gestrichelten bzw. strichpunktierten Kurvenverlauf.
C2XP	Kontrolle von Überschreitungen für einen vorgegebenen Abszissenbereich.
C2XPP	Kontrolle von radialen Bereichsüberschreitungen.
DT	Bestimmung der Extremwerte an einer Achse.
EXPD	Umwandlung einer Zahl im E-Format in eine Zeichenkette.
FORM	Setzen von Zeichnungsparametern (Format, Rahmen, KfK-Symbol).

TRACE/PC V.2.0 - Hilfsroutinen

HRMG	Bestimmung der Anzahl von Marken und der Genauigkeit der Graduationen für eine Achse.
HYP0	Länge einer Hypotenuse.
NETLIN	Zeichnen von getrichelten oder durchgezogenen Rasterlinien.
NETLN2	Zeichnen von gestrichelten oder durchgezogenen Rasterlinien unter Berücksichtigung eines freibleibenden Fensters.
ORGPEN	Retten eines vorgegebenen Farbcodes.
PHI	Winkelbestimmung im rechtwinkligen Dreieck.
PLOTX	Zeichnungsinitialisierung und Argumentanalyse.
POSFT	Bestimmung der Startposition für eine zu zeichnende Textzeile (Sonderschriftarten, FNTX).
POSXY	Bestimmung der Startposition für einen Text- bzw. Ziffernstring (Standardschriften).
SLX	Setzen der Kennung für eine gewählte Strichcharakteristik.
XPOS	Bestimmung der horizontalen Ziffernposition für die Graduationen an einer Abszisse.
YPOS	Bestimmung der horizontalen Ziffernposition für die Graduationen an einer Ordinate.
VECPLT	Zeichnen eines gestrichelten bzw. strichpunktieren Kurvenverlaufs.
ZST	Umwandlung einer Zahl im F-Format in eine Zeichenkette.

37. Literatur

- [1] M. Politzky - TRACEGS - Ein Programmsystem für die Zeichnungserstellung auf der Basis des graphischen Systems (GS).
KfK 3237, November 1981.
- [2] W. Hame - GSW - Grafik Software, Version 2.05. Benutzeranleitung und Programmbeschreibungen.
Ingenieurbüro PTF, Karlsruhe, Januar 1988.
- [3] N. M. Walcott - A Contribution to Computer Typesetting Techniques (National Bureau of Standards, Washington, D.C. 20234, Special Publication 424, April 1976).
- [4] S. Kleinheins - DT PLOT - Programmbeschreibung.
Unveröffentlichter Bericht des KfK, Juli 1981.

38. Programmbeispiele

Zum besseren Verständnis wird an ausgewählten Beispielen der Aufbau einer Graphik sowie die Verfahrensweise bei der Programmierung demonstriert.

Der ungeübte Anwender kann dabei anhand von Programmlisten die Graphiken gedanklich nachvollziehen.

Auf den folgenden Seiten ist jeweils linksseitig das Programm und rechtsseitig die gezeichnete Darstellung desselben wiedergegeben.

Folgende Anwendungsbeispiele werden gezeigt (Titel):

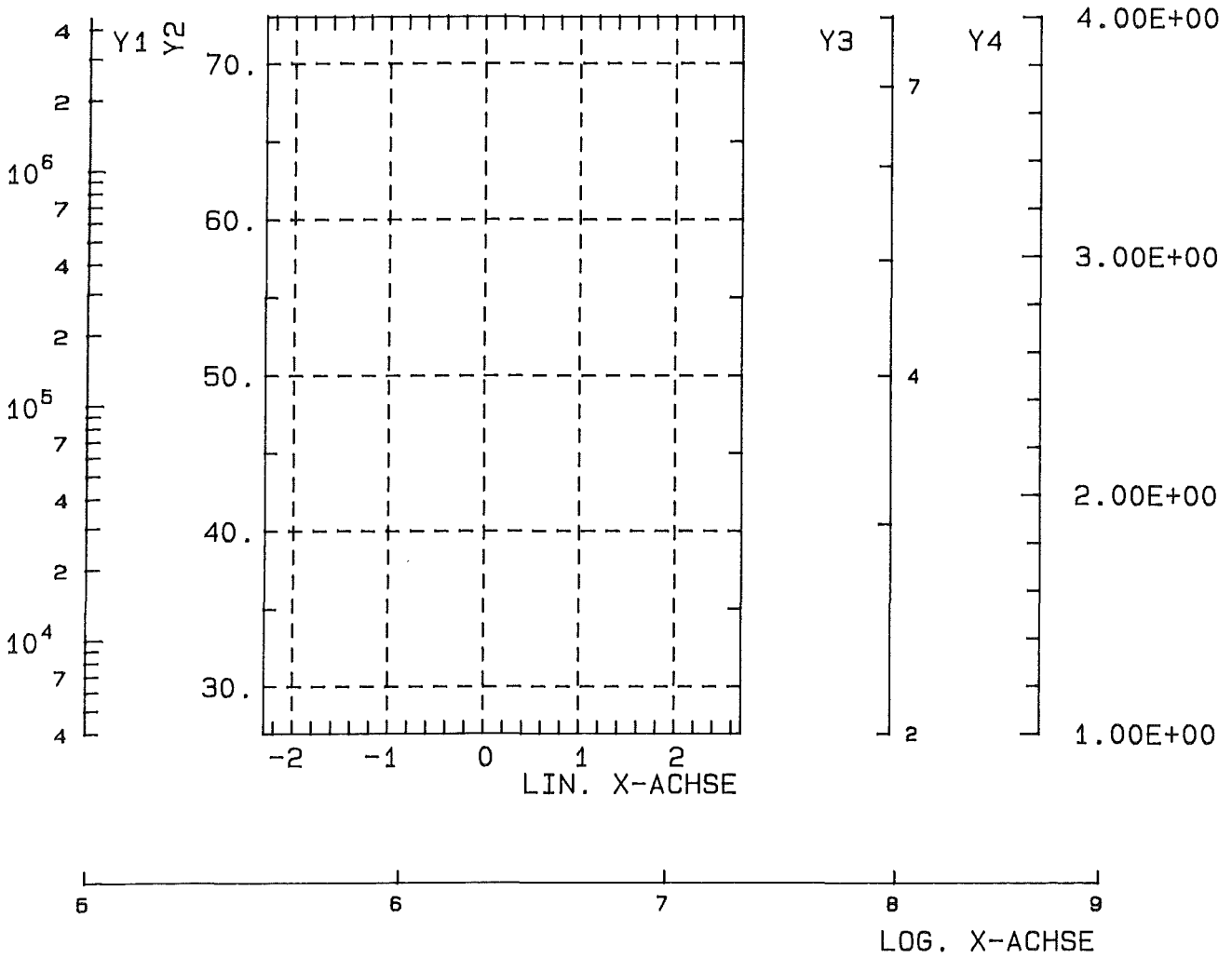
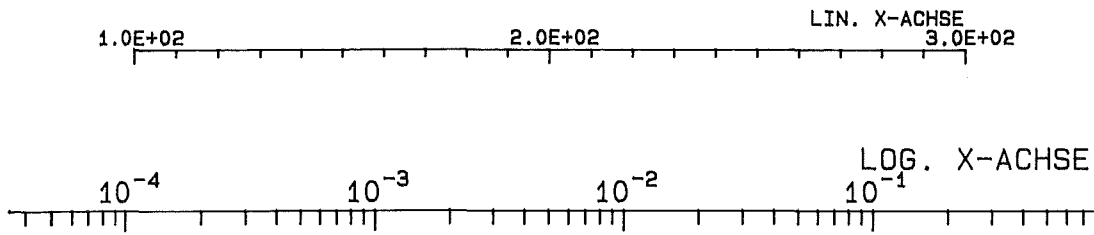
	Seite
1. PROGRAM AXIS - Anwendungen der Achsenprogramme	108
2. PROGRAM VIVIANI - Kugel/Zylinder-Schnitte - Ein Anwendungsbeispiel ...	110
3. PROGRAM POLAR - TRACEPC: POL und CURVEP	112
4. PROGRAM KURVE - Wirkungsweise von CURVE und CURVE2	114
5. PROGRAM WINDOW - Beispiel für die Verwendung von AXYFEN/AXYRES .	116
6. PROGRAM FONT - Hershey Font's	118
7. PROGRAM HISTO - Anwendungsbeispiel für ein Histogramm	120
8. PROGRAM SCRIPT - Zeichnen von Texten in Normal- und Schrägschrift ...	122
9. PROGRAM D3PLOT - Anwendung des Moduls "DTPLOT"	124
(nach S. Kleinheins, INR/KfK)	
10. PROGRAM CONFIG - Geeignete Hardware-Konfiguration für Farbgraphik- Applikationen	126

Programmbeispiel 1

```

PROGRAM AXIS
C ANWENDUNGEN DER ACHSENPROGRAMME
  CHARACTER*12 TEXT1, TEXT2
  DATA          TEXT1/'LOG. X-ACHSE'/
  DATA          TEXT2/'LIN. X-ACHSE'/
C HP-PLOTTER
  CALL PLOTI('COM1',4, 1,30141,' IRB')
C SCREEN
C CALL PLOTI( 0,0,17,30141,' IRB')
  CALL SYMBP1(10.5,2.6,-.59,'Anwendungsbeispiele für die',0.,27)
  CALL SYMBP1(10.5,1.7,-.59,'Achsenprogramme', 0.,15)
  CALL PLOTP(1.7,7.0,-3)
  CALL AXLOG1(17.,5.0E + 1,9.0E + 1,3,0.,1,0.2,0.3,0.3,12,TEXT1)
  CALL PLOTP(2.,19.5,-3)
  CALL AXLIN1(13.,100.,300.,100.,100.,0.,9,0.1,0.2,-0.2,-11,12,
+ TEXT2)
  CALL PLOTP(-2.,-2.5,-3)
  CALL AXLOG1(17.,3.3E-5,7.6E-1,0,0.,1,0.2,0.3,-0.3,12,TEXT1)
  CALL PLOTP(0.,-14.5,-3)
  CALL AYLOG1(12.,4.E + 3,4.5E + 6,1,0.,1,0.2,0.3,0.3,2,'Y1')
  CALL PLOTP(13.5,0.,-3)
  CALL AYLOG1(12.,2.0E + 7,8.0E + 7,1,0.,2,0.2,0.3,-0.3,2,'Y3')
  CALL PLOTP(2.5,0.,-3)
  CALL AYLIN1(12.,1.,4.,1.,3.,0.,4,0.2,0.3,-0.3,-12,2,'Y4')
  CALL PLOTP(-13.,0.,-3)
  CALL AXLIN1(8.,-2.3,2.7,1.,-1.,-12.,4,0.2,0.,0.3,-1,12,TEXT2)
  CALL AYLIN1(12.,27.,73.,10.,50.,-8.,1,0.2,0.,0.3,0,-2,'Y2')
  CALL PLOTP( 0., 0.,999)
  STOP
  END

```



Anwendungsbeispiele für die Achsenprogramme

Programmbeispiel 2

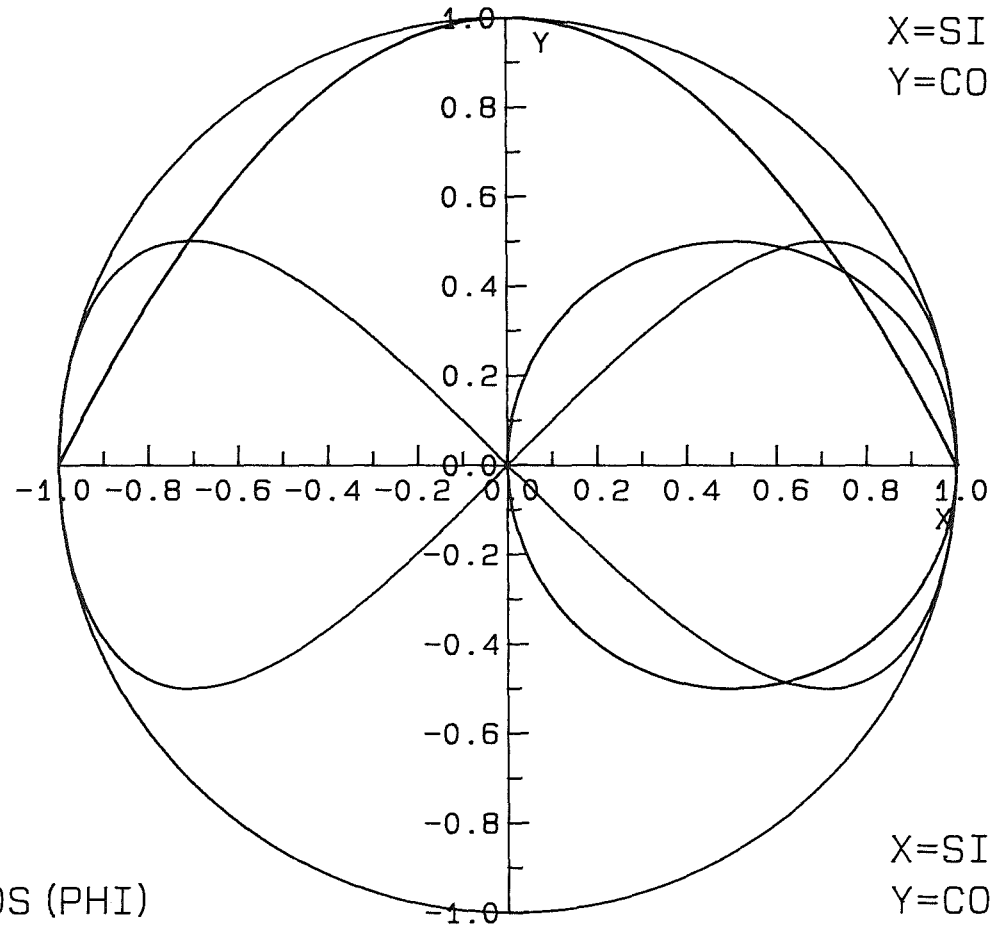
```

PROGRAM VIVIANI
C KUGEL/ZYLINDER-SCHNITTE - EIN ANWENDUNGSBEISPIEL
REAL W(722), X(722), Y(722), Z(722)
CHARACTER COMX1*13, COMY1*19, COMX2*19, COMY2*10
CHARACTER COMX3*10, COMY3*10, COMX4*13, COMY4*10, TEXT*48
DATA TEXT/'KUGEL/ZYLINDER-SCHNITTE - EIN ANWENDUNGSBEISPIEL'/
DATA COMX1/'X = SIN**2(PHI)'/
DATA COMY1/'Y = SIN(PHI)*COS(PHI)'/
DATA COMX2/'X = SIN(PHI)*COS(PHI)'/
DATA COMY2/'Y = COS(PHI)'/
DATA COMX3/'X = COS(PHI)'/
DATA COMY3/'Y = SIN(PHI)'/
DATA COMX4/'X = SIN**2(PHI)'/
DATA COMY4/'Y = COS(PHI)'/
CONST = 3.14159265/180.
DO 10 I = 1, 722
PHI = (I-1)*.5*CONST
W(I) = SIN(PHI)
X(I) = W(I)**2
Z(I) = COS(PHI)
10 Y(I) = W(I)*Z(I)
C HP-PLOTTER
CALL PLOTI('AUX',3, 1,30140,' IRB')
C SCREEN
C CALL PLOTI( 0,0,17,30140,' IRB')
CALL PLOTP(7.85,12.,-3)
CALL AXLIN1(14.,-1.,1.,2,0.,0.,1.,2.,3.,3,1,1,'X')
CALL PLOTP(7.,-7.,-3)
CALL AYLIN1(14.,-1.,1.,2,0.,0.,1.,2.,3.,3,1,1,'Y')
CALL PLOTP(0.,7.,-3)
CALL SYMBP1(0.,-10.5,-.5,TEXT,0.,48)
CALL PLOTP(-7.,-7.,-3)
CALL CURVE(Z,W,722,0.,15,0)
CALL SYMBP1(-6.,13.6,.4,COMX3,0.,10)
CALL SYMBP1(-6.,12.8,.4,COMY3,0.,10)
CALL CURVE(X,Y,361,0.,15,0)
CALL SYMBP1(-6.,.8,.4,COMX1,0.,13)
CALL SYMBP1(-6.,.0,.4,COMY1,0.,19)
CALL CURVE(Z,X,361,0.,15,0)
CALL SYMBP1(13.,13.6,.4,COMX4,0.,13)
CALL SYMBP1(13.,12.8,.4,COMY4,0.,10)
CALL CURVE(Z,Y,722,0.,15,0)
CALL SYMBP1(13.,.8,.4,COMX2,0.,19)
CALL SYMBP1(13.,.0,.4,COMY2,0.,10)
CALL PLOTP(0.,0.,999)
STOP
END

```

$$X = \cos(\text{PHI})$$
$$Y = \sin(\text{PHI})$$

$$X = \sin^2(\text{PHI})$$
$$Y = \cos(\text{PHI})$$



$$X = \sin^2(\text{PHI})$$
$$Y = \sin(\text{PHI}) * \cos(\text{PHI})$$

$$X = \sin(\text{PHI}) * \cos(\text{PHI})$$
$$Y = \cos(\text{PHI})$$



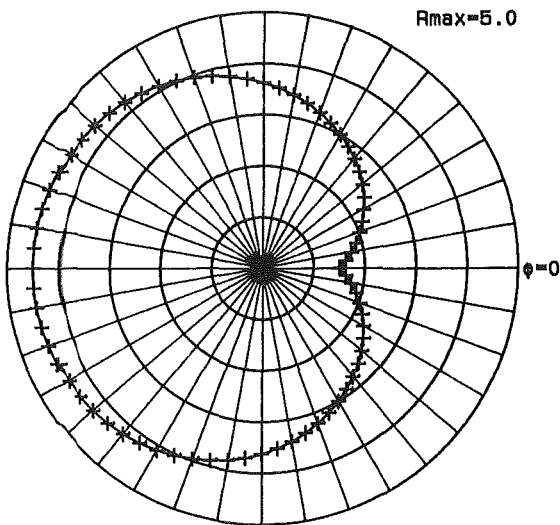
KUGEL/ZYLINDER-SCHNITTE - EIN ANWENDUNGSBEISPIEL

Programmbeispiel 3

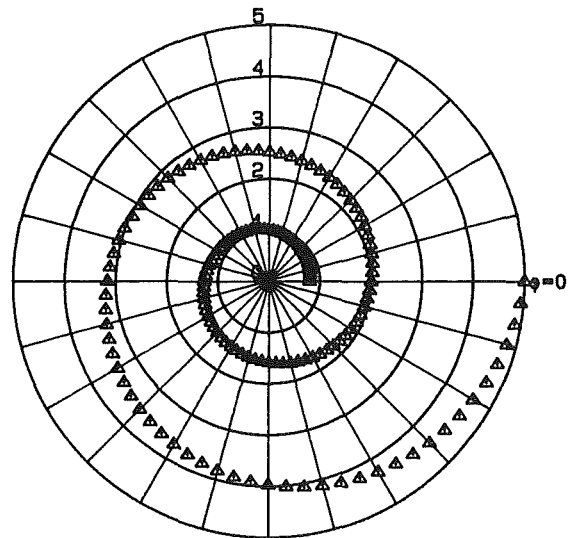
```

PROGRAM POLAR
C TRACEPC: POL UND CURVEP
CHARACTER*37 T1, T2*27, T3*47, T4*31
REAL PHI(721), RAD(721)
DATA T1/'ARCHIMEDISCHE SPIRALE: R = 0.138888*PHI'/
DATA T2/'FUNKTION: R = 3.*SIN(PHI) + 1.5'/
DATA T3/'LOGARITHMISCHE SPIRALE: R = 2.*E**(0.0025452*PHI)'/
DATA T4/'FUNKTION: R = 3.*SIN(PHI/2.) + 1.5)'/
DATA CONST/.0174532/
C HP-PLOTTER
CALL PLOTI('COM1',4, 1,30141,' IRB')
C SCREEN
C CALL PLOTI( 0,0,17,30141,' IRB')
CALL SYMBP1(10.5,1.8,-.5,'TRACEPC: POL UND CURVEP',0.,23)
CALL PLOTP(5.,11.5,-3)
CALL POL(4.,5.,1.,-.20,-12,0,37,T1)
DO 10 I = 1,361
PHI(I) = I-1
10 RAD(I) = .0138888*PHI(I)
CALL CURVEP(RAD,PHI,361,2.,2,5)
CALL PLOTP(10.5,0.,-3)
CALL POL(4.,5.,1.,-.20,-12,1,27,T2)
DO 20 I = 1,361
PHI(I) = I-1
20 RAD(I) = 3.*SIN(PHI(I)*CONST) + 1.5
CALL CURVEP(RAD,PHI,361,2.,2,0)
CALL PLOTP(0.,12.,-3)
CALL POL(4.,5.,1.,-.20,-1,3,47,T3)
DO 30 I = 1,721
PHI(I) = I-361
30 RAD(I) = 2.*EXP(.0025452*PHI(I))
CALL CURVEP(RAD,PHI,721,2.,2,-5)
CALL PLOTP(-11.,0.,-3)
CALL POL(4.,5.,1.,-.20,-1,4,31,T4)
DO 40 I = 1,361
PHI(I) = I-1
40 RAD(I) = 3.*SIN(PHI(I)/2.*CONST) + 1.5
CALL CURVEP(RAD,PHI,361,3.,2,5)
CALL PLOTP(0.,0.,999)
STOP
END

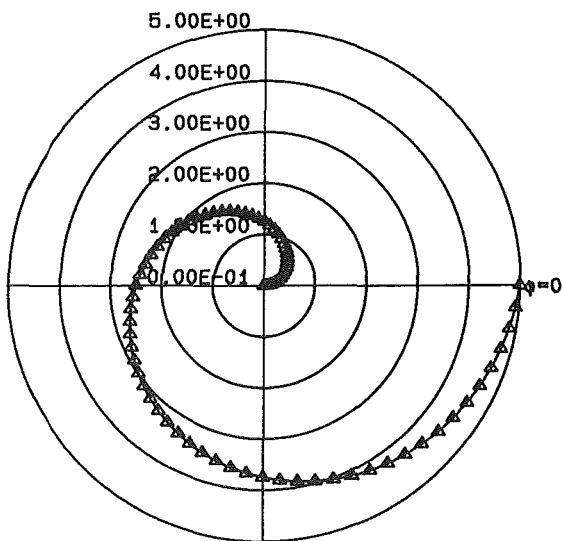
```

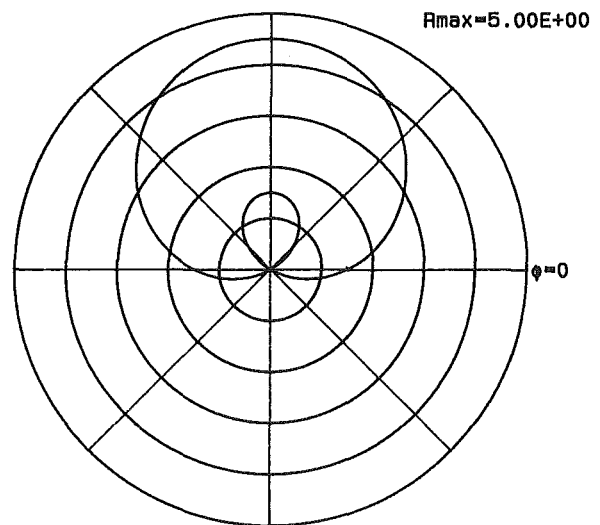
FUNKTION: $R=3 \cdot \sin(\Phi/2) + 1.5$



LOGARITHMISCHE SPIRALE: $R=2 \cdot e^{(0.0025452 \cdot \Phi)}$



ARCHIMEDISCHE SPIRALE: $R=0.138888 \cdot \Phi$



FUNKTION: $R=3 \cdot \sin(\Phi) + 1.5$



TRACEPC: POL UND CURVEP

Programmbeispiel 4

```

PROGRAM KURVE
C WIRKUNGSWEISE VON CURVE UND CURVE2
  CHARACTER TEXT*34, TIME*4, TEMP*10
  REAL X(2500), Y(2500), WORK(2*2500)
  DATA TEXT/'Wirkungsweise von CURVE und CURVE2'/
  DATA TIME/'Time'/, TEMP/'Temperatur'/
  OPEN (2,FILE = 'CV2.DAT',STATUS = 'OLD')
C READ: NUMBER OF POINTS AND TIMESTEP
  READ(2,*) NP, TS
C READ: TEMPERATURE POINTS
  READ(2,*) (Y(I), I = 1, NP)
C CREATE: TIME VALUE TABLE
  DO 10 I = 1, NP
10 X(I) = (I-1)*TS
C HP-PLOTTER
  CALL PLOTI('COM1',4, 1,30141,' IRB')
C SCREEN/DISPLAY
C CALL PLOTI( 0,0,17,30141,' IRB')
  CALL SYMBS1(10.5,1.7,-.55,TEXT,0.,34)
  CALL PLOTP(2.5,18.,-3)
  CALL SYMBS1(1.0,7.0,.4,'CURVE2',0.,6)
  CALL AXLIN1(17.,0.,4.,.5,0.,10.,4.,1.,2.,3, 1,4,TIME)
  CALL AYLIN1(10.,420.,450., 5.,420.,17.,4.,1.,2.,3,-1,-10,TEMP)
  CALL CURVE2(X,Y,NP,0.,0.,0,WORK)
  CALL PLOTP(0.,-11.5,-3)
  CALL SYMBS1(1.0,7.0,.4,'CURVE',0.,5)
  CALL AXLIN1(17.,0.,4.,.5,0.,10.,4.,1.,2.,3, 1,4,TIME)
  CALL AYLIN1(10.,420.,450., 5.,420.,17.,4.,1.,2.,3,-1,-10,TEMP)
  CALL CURVE (X,Y,NP,0.,0,0)
  CALL PLOTP(0.,0.,999)
  STOP
  END

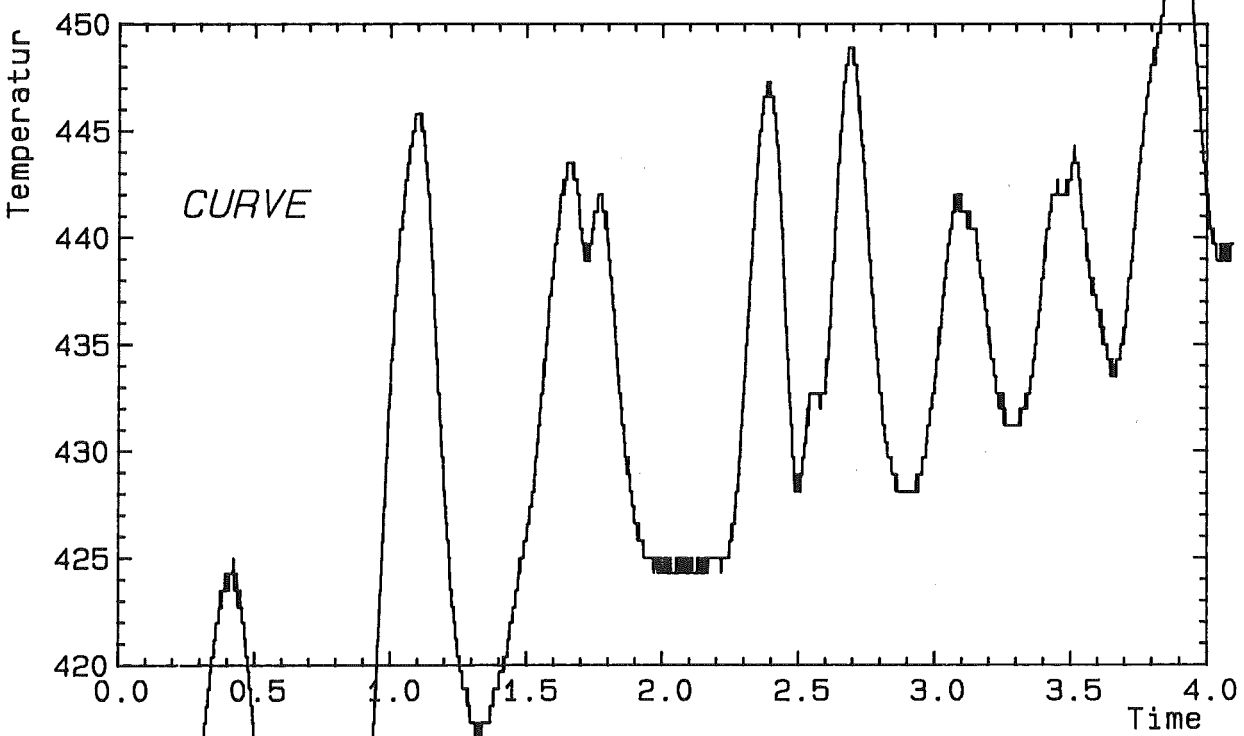
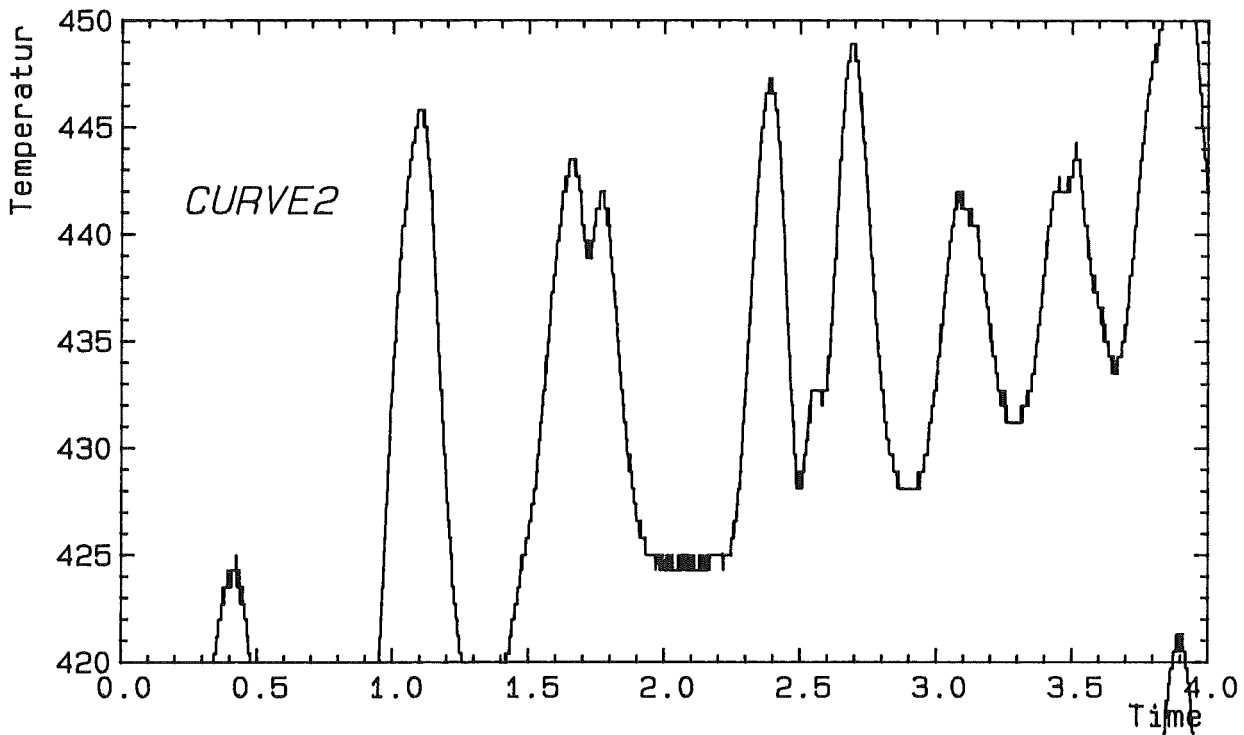
```

Inhalt (Auszug) des Datenfile's "CV2.DAT"

```

2048 .2E-2
0.4066E + 03 0.4066E + 03 0.4066E + 03 0.4066E + 03 0.4066E + 03 0.4066E + 03
0.4066E + 03 0.4066E + 03 0.4066E + 03 0.4066E + 03 0.4066E + 03 0.4066E + 03
....
....
0.4327E + 03 0.4327E + 03 0.4327E + 03 0.4327E + 03 0.4327E + 03 0.4327E + 03
0.4327E + 03 0.4327E + 03 0.4327E + 03 0.4327E + 03 0.4327E + 03 0.4327E + 03
....
....
0.4389E + 03 0.4389E + 03 0.4389E + 03 0.4397E + 03 0.4389E + 03 0.4389E + 03
0.4389E + 03 0.4397E + 03 0.4397E + 03 0.4389E + 03 0.4397E + 03 0.4397E + 03
0.4397E + 03 0.4397E + 03

```



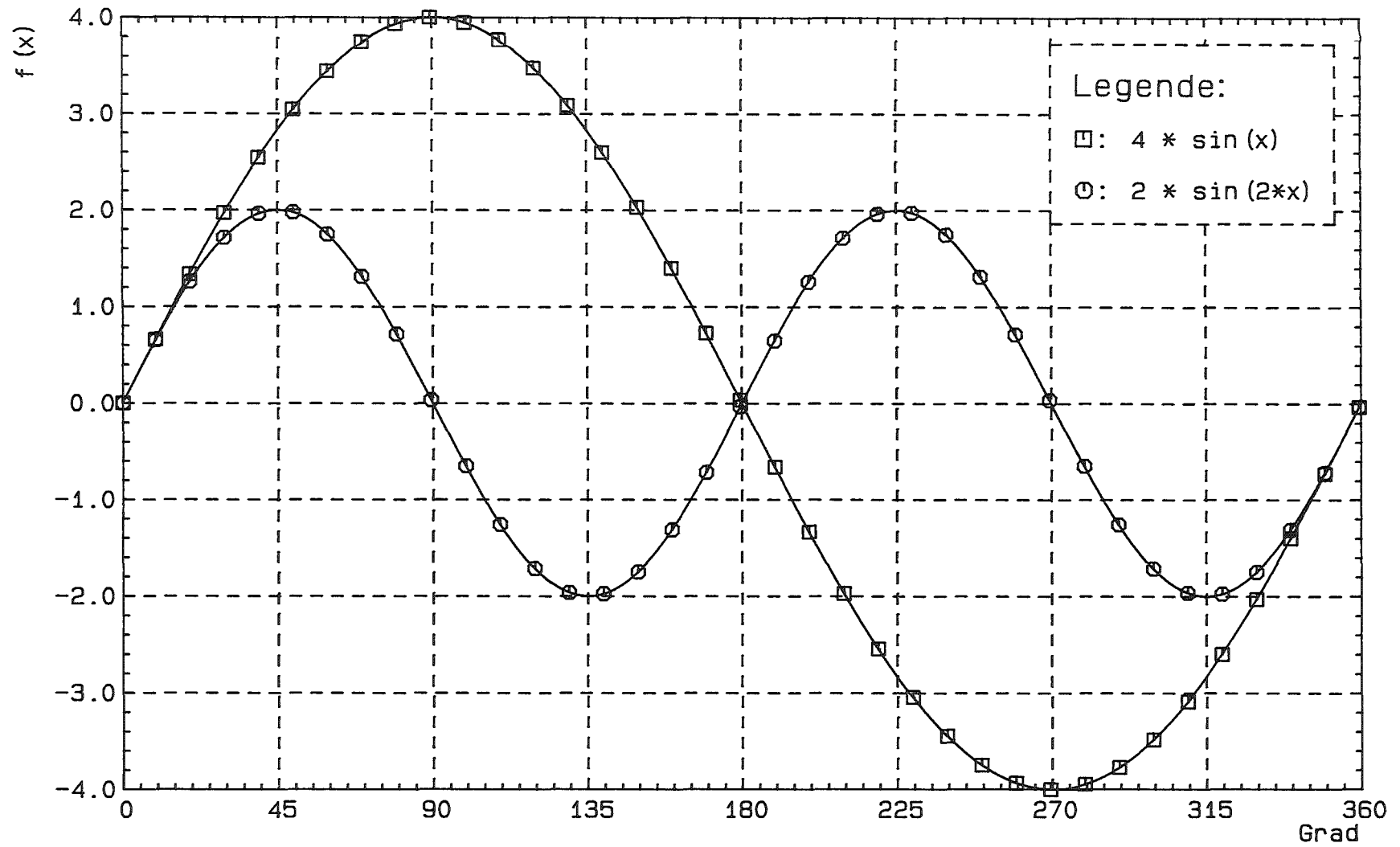
Wirkungsweise von CURVE und CURVE2

Programmbeispiel 5

```

PROGRAM WINDOW
C  BEISPIEL FÜR DIE VERWENDUNG VON AXYFEN/AXYRES
  CHARACTER TEXT*45
  REAL XARRAY(720), YARRAY(720)
  DATA TEXT/'Beispiel für die Verwendung von AXYFEN/AXYRES'/
  DATA CONST/.0174533/
C  HP-PLOTTER
  CALL PLOTI('AUX',3, 1,30140,' IRB')
C  SCREEN
C  CALL PLOTI( 0,0,17,30140,' IRB')
  CALL SYMBP1(14.85,1.5,-.6,TEXT,0.,45)
  CALL PLOTP(3.5,5.,-3)
  CALL AXYFEN(18.,11.,23.5,14.5,1)
  CALL AXLIN1(24.,0.,360.,45.,360.,-15.,8.,1,0.,.3,-1,4,'Grad')
  CALL AYLIN1(15.,-4.,4.,1.,-4.,-24.,4.,1,0.,.3,1,-4,'f(x)')
  CALL AXYRES
  CALL SYMBP1(18.5,13.5,.4,'Legende:',0., 8)
  CALL SYMBP1(18.5 + .15,12.5 + .15,.3,0,0.,-1)
  CALL SYMBP1(19.0,12.5,.3,': 4 * sin(x)',0.,12)
  CALL SYMBP1(18.5 + .15,11.5 + .15,.3, 1,0.,-1)
  CALL SYMBP1(19.0,11.5,.3,': 2 * sin(2*x)',0.,14)
  XARRAY(1) = 0.
  DO 10 I = 2,720
10 XARRAY(I) = 0.5*I
  DO 20 I = 1,720
  PHI = XARRAY(I)*CONST
20 YARRAY(I) = 4*SIN(PHI)
  CALL CURVE(XARRAY,YARRAY,720,0.,.25,20)
  DO 30 I = 1, 720
  PHI = XARRAY(I)*CONST
30 YARRAY(I) = 2*SIN(2*PHI)
  CALL CURVE(XARRAY,YARRAY,720,1,.,25,20)
  CALL PLOTP(0.,0.,999)
  STOP
  END

```



Beispiel für die Verwendung von AXYFEN/AXYRES

Programmbeispiel 6

```

PROGRAM FONT
C  HERSHEY FONT'S
CHARACTER TEXT1*35, TEXT2*36, TEXT3*36, TEXT4*36, TEXT5*35,
+   TITEL*14
CHARACTER*26 AG,AK,AZ*7,SN*28
DATA TITEL/'Hershey Font's'/
DATA TEXT1/'Schriftprobe Font 1 (Triplex Roman)'/
DATA TEXT2/'Schriftprobe Font 2 (Complex Script)'/
DATA TEXT3/'Schriftprobe Font 3 (Gothic English)'/
DATA TEXT4/'Schriftprobe Font 4 (Triplex Italic)'/
DATA TEXT5/'Schriftprobe Font 5 (Gothic German)'/
C  Font Definition
C  Großes Alphabet
DATA AG/'ABCDEFGHIJKLMNPOQRSTUVWXYZ'/
C  Kleines Alphabet
DATA AK/'abcdefghijklmnopqrstuvwxyxz'/
C  zusätzliche alphabetische Zeichen
DATA AZ/'ÄÖÜäöüß'/
C  Sonder- und numerische Zeichen
DATA SN/' + -*: = ( ), ; ? ! / & "" ' 0123456789'/
C  HP-PLOTTER
CALL PLOTI('AUX',3, 1,30141,' IRB')
C  SCREEN
C  CALL PLOTI( 0,0,17,30141,' IRB')
CALL SYMBP1(10.5,2.1,-1.,TITEL,0.,14)
C  FONT 1 (TRIPLEX ROMAN)
CALL FNTX(10.5,28.2,-.6,.5,.1,0.,0.,TEXT1,35,1)
CALL FNTX(10.5,27.2,-.5,.5,.2,0.,0.,AG,26,1)
CALL FNTX(10.5,26.4,-.5,.5,.2,0.,0.,AK,26,1)
CALL FNTX(10.5,25.6,-.5,.5,.2,0.,0.,AZ, 7,1)
CALL FNTX(10.5,24.8,-.5,.5,.2,0.,0.,SN,28,1)
C  FONT 2 (COMPLEX SCRIPT)
CALL FNTX(10.5,23.4,-.6,.5,.0,0.,0.,TEXT2,36,2)
CALL FNTX(10.5,22.4,-.5,.5,.2,0.,0.,AG,26,2)
CALL FNTX(10.5,21.6,-.5,.5,.2,0.,0.,AK,26,2)
CALL FNTX(10.5,20.8,-.5,.5,.2,0.,0.,AZ, 7,2)
CALL FNTX(10.5,20.0,-.5,.5,.2,0.,0.,SN,28,2)
C  FONT 3 (GOTHIC ENGLISH)
CALL FNTX(10.5,18.6,-.6,.5,.1,0.,0.,TEXT3,36,3)
CALL FNTX(10.5,17.6,-.5,.5,.2,0.,0.,AG,26,3)
CALL FNTX(10.5,16.8,-.5,.5,.2,0.,0.,AK,26,3)
CALL FNTX(10.5,16.0,-.5,.5,.2,0.,0.,AZ, 7,3)
CALL FNTX(10.5,15.2,-.5,.5,.2,0.,0.,SN,28,3)
C  FONT 4 (TRIPLEX ITALIC)
CALL FNTX(10.5,13.8,-.6,.5,.1,0.,0.,TEXT4,36,4)
CALL FNTX(10.5,12.8,-.5,.5,.2,0.,0.,AG,26,4)
CALL FNTX(10.5,12.0,-.5,.5,.2,0.,0.,AK,26,4)
CALL FNTX(10.5,11.2,-.5,.5,.2,0.,0.,AZ, 7,4)
CALL FNTX(10.5,10.4,-.5,.5,.2,0.,0.,SN,28,4)
C  FONT 5 (GOTHIC GERMAN)
CALL FNTX(10.5, 9.0,-.6,.5,.1,0.,0.,TEXT5,35,5)
CALL FNTX(10.5, 8.0,-.5,.5,.2,0.,0.,AG,26,5)
CALL FNTX(10.5, 7.2,-.5,.5,.2,0.,0.,AK,26,5)
CALL FNTX(10.5, 6.4,-.5,.5,.2,0.,0.,AZ, 7,5)
CALL FNTX(10.5, 5.6,-.5,.5,.2,0.,0.,SN,28,5)
CALL PLOTP(0.,0.,999)
STOP
END

```

Schriftprobe Font 1 (Triplex Roman)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Ä Ö Ü ä ö ü ß

+ - * : = () . , ; ? ! ° / & ' " 0 1 2 3 4 5 6 7 8 9

*Schriftprobe Font 2 (Complex Script)**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**a b c d e f g h i j k l m n o p q r s t u v w x y z**Ä Ö Ü ä ö ü ß**+ - * : = () . , ; ? ! ° / & ' " 0 1 2 3 4 5 6 7 8 9*

Schriftprobe Font 3 (Gothic english)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Ä Ö Ü ä ö ü ß

+ - * : = () . , ; ? ! ° / & ' " 0 1 2 3 4 5 6 7 8 9

*Schriftprobe Font 4 (Triplex Italic)**A B C D E F G H I J K L M N O P Q R S T U V W X Y Z**a b c d e f g h i j k l m n o p q r s t u v w x y z**Ä Ö Ü ä ö ü ß**+ - * : = () . , ; ? ! ° / & ' " 0 1 2 3 4 5 6 7 8 9*

Schriftprobe Font 5 (Gothic German)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

a b c d e f g h i j k l m n o p q r s t u v w x y z

Ä Ö Ü ä ö ü ß

+ - * : = () . , ; ? ! ° / & ' " 0 1 2 3 4 5 6 7 8 9



Hershey Font 's

Programmbeispiel 7

```

PROGRAM HISTO
C ANWENDUNGSBEISPIEL FÜR EIN HISTOGRAMM
REAL PCA(8),PCN(8)
DATA PCA/0.,1.,2.,7.,9.,10.,12.,16./
DATA PCN/1.,1.,5.,2.,1.,2.,4.,2./
C HP-PLOTTER
CALL PLOTI('COM1',4,1,20140)
C SCREEN
C CALL PLOTI( 0,0,17,20140)
CALL PLOTP(3.,2.,-3)
CALL AXLIN1(24.,1980.5,1988.5,1.,1981.,0.,0,0.,0.,3,-1,0,0)
CALL AYLIN1(18.,0.,18.,5.,0.,0.,4.,1.,2.,3,-1,-6,'Anzahl')
CALL PLOTP(2.,15.,-3)
CALL CADRE2(8.,2.)
CALL SYMBS1(4.,1.,-.35,'PC - Installationen',0.,19)
CALL PLOTP(0.,-2.5,-3)
CALL BALKEN(1.,1.,0.,.1)
CALL SYMBP1(1.8,0.,.3,'Bestand aus dem Vorjahr',0.,23)
CALL PLOTP(0.,-2.5,-3)
CALL BALKEN(1.,1.,90.,.1)
CALL SYMBP1(1.8,0.,.3,'Jährliche Zuwachsrates',0.,21)
CALL PLOTP(-2.,-10.,-3)
BB = 2.8
X = -2.9
DO 10 I = 1, 8
X = X + 3.
CALL PLOTP(X,0.,-3)
CALL BALKEN(BB,PCA(I),0.,.1)
CALL PLOTP(0.,PCA(I),-3)
CALL BALKEN(BB,PCN(I),90.,.1)
CALL PLOTP(0.,-PCA(I),-3)
CALL PLOTP(-X,0.,-3)
10 CONTINUE
CALL PLOTP(0.,0.,999)
STOP
END

```


Anzahl

PC - Installationen



Bestand aus dem Vorjahr



Jährliche Zuwachsrage

15

10

5

0

1981

1982

1983

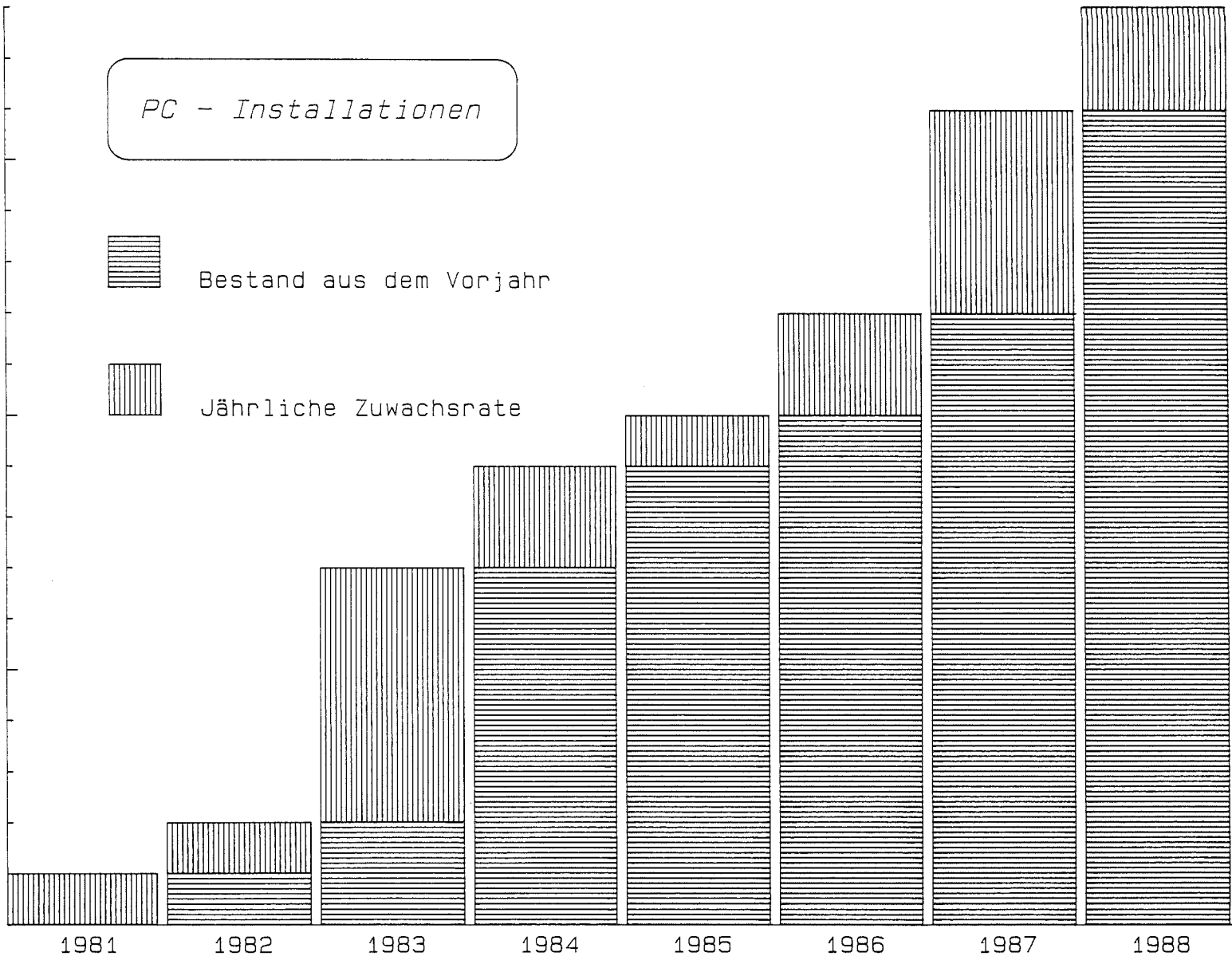
1984

1985

1986

1987

1988



Programmbeispiel 8

```

PROGRAM SCRIPT
C ZEICHNEN VON TEXTEN IN NORMAL- UND SCHRÄGSCHRIFT
  OPEN (1,FILE = 'TEXTT.DAT',STATUS = 'OLD')
  OPEN (2,FILE = 'TEXT.DAT',STATUS = 'OLD')
  OPEN (3,FILE = 'TEXTS.DAT',STATUS = 'OLD')
  CALL PLOTI('AUX',3,1,30141,' IRB')
  CALL TEXT (1.,2.20,1)
  CALL TEXT (.6,28.8,2)
  CALL TEXTS(.6,14.8,3)
  CALL PLOTP(0.,0.,999)
  STOP
  END

```

Inhalt des Textfile's "TEXT.DAT"

```

.4 .29 .0 .2 0 1
Die Unterprogramme TEXT und TEXTS gestatten das Zeichnen von Texten.$
Die alphabetischen und numerischen Zeichen werden bei der Anwendung$
von TEXT in Normalschrift, bei der Anwendung von TEXTS in Schräg-$
schrift (kursiv) dargestellt.$
Mit folgenden Eingabe-Parametern, die am Anfang von jedem Textkom-$
plex angegeben werden müssen, läßt sich das Schriftbild weitgehend$
den Erfordernissen anpassen:$
$
  I. Schrifthöhe..... [SH, cm] REAL$
  II. Schriftbreite..... [SB, cm] REAL$
  III. Spaltenbreite zwischen 2 Buchstaben... [ZB, cm] REAL$
  IV. Zeilenzwischenraum..... [ZZ, cm] REAL$
  V. Strichstärke (0/1---dünn/dick)..... [IS, --] INTEGER$
  VI. Zeichenfarbe (s. PEN-Farbtafel)..... [IC, --] INTEGER$
$
Wird innerhalb eines Textes (nur zeilenweise möglich!) die Änderung$
der Zeichen-Parameter gewünscht, muß zunächst eine Zeile mit 2 $$-$
Zeichen in der 1. und 2. Spalte eingefügt werden.$
Eine größere Zeichenhöhe und ein Wechsel der Strichstärke hätte fol-$
gende$
$$
.6 .40 .0 .2 1 1
  A u s w i r k u n g $
  -----$

```

Inhalt des Textfile's "TEXTS.DAT"

```

.4 .29 .0 .16 0 1
Für das Schreiben der Texte stehen die folgenden alphabetischen,$
numerischen, Sonder- und Symbol-Zeichen zur Verfügung:$
$
Lateinisches Alphabet:$
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - Ä Ö Ü$
a b c d e f g h i j k l m n o p q r s t u v w x y z - ä ö ü ß$
$
Griechisches Alphabet:$
{A {B {C {D {E {F {G {I {K {L {M {N {O {P {R {S {T {X {Y {Z {0 {1 {2 {3$
  #C #D #F #G #L #P #R #S #1 #2 #3$
$
Symbole:$
$0 , $1 , $2 , $3 , $4 , $5 , $6 , $7 , $8 , $9 , $W , $X , $Y
$
Numerische Zeichen: Sonderzeichen:$
0 1 2 3 4 5 6 7 8 9 [# { [ } $$ . ' " ; ; ! ? / + - _ ( )$
[ ] < > & = % ' " ; ; @ $

```

Inhalt des Textfile's "TEXTT.DAT"

```

.5 .306 .1 0. 1 1
Zeichnen von Texten in Normal- und Schrägschrift$
-----$

```

Die Unterprogramme TEXT und TEXTS gestatten das Zeichnen von Texten. Die alphabetischen und numerischen Zeichen werden bei der Anwendung von TEXT in Normalschrift, bei der Anwendung von TEXTS in Schrägschrift (kursiv) dargestellt.

Mit folgenden Eingabe-Parametern, die am Anfang von jedem Textkomplex angegeben werden müssen, läßt sich das Schriftbild weitgehend den Erfordernissen anpassen:

I. Schrifthöhe.....	[SH, cm]	REAL
II. Schriftbreite.....	[SB, cm]	REAL
III. Spaltenbreite zwischen 2 Buchstaben...	[ZB, cm]	REAL
IV. Zeilenzwischenraum.....	[ZZ, cm]	REAL
V. Strichstärke (0/1---dünn/dick).....	[IS, --]	INTEGER
VI. Zeichenfarbe (s. PEN-Farbtabelle).....	[IC, --]	INTEGER

Wird innerhalb eines Textes (nur zeilenweise möglich!) die Änderung der Zeichen-Parameter gewünscht, muß zunächst eine Zeile mit 2 \$-Zeichen in der 1. und 2. Spalte eingefügt werden. Eine größere Zeichenhöhe und ein Wechsel der Strichstärke hätte folgende

A u s w i r k u n g

Für das Schreiben der Texte stehen die folgenden alphabetischen, numerischen, Sonder- und Symbol-Zeichen zur Verfügung:

Lateinisches Alphabet:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - Ä Ö Ü
a b c d e f g h i j k l m n o p q r s t u v w x y z - ä ö ü ß

Griechisches Alphabet:

α β γ δ ε φ χ ι κ λ μ ν ο π ρ σ τ ξ υ ζ η θ ψ ω
Χ Δ Φ Γ Λ Π Ρ Σ Θ Ψ Ω

Symbole:

□, ⊙, △, +, ×, ◇, ♠, ⌘, ⌘, ⌘, ⌘, ⌘, ⌘, ⌘, ⌘, ⌘, ⌘

Numerische Zeichen:

0 1 2 3 4 5 6 7 8 9

Sonderzeichen:

{ } \$. , : ; ! ? / + - _ ()
[] < > & = % " ' @



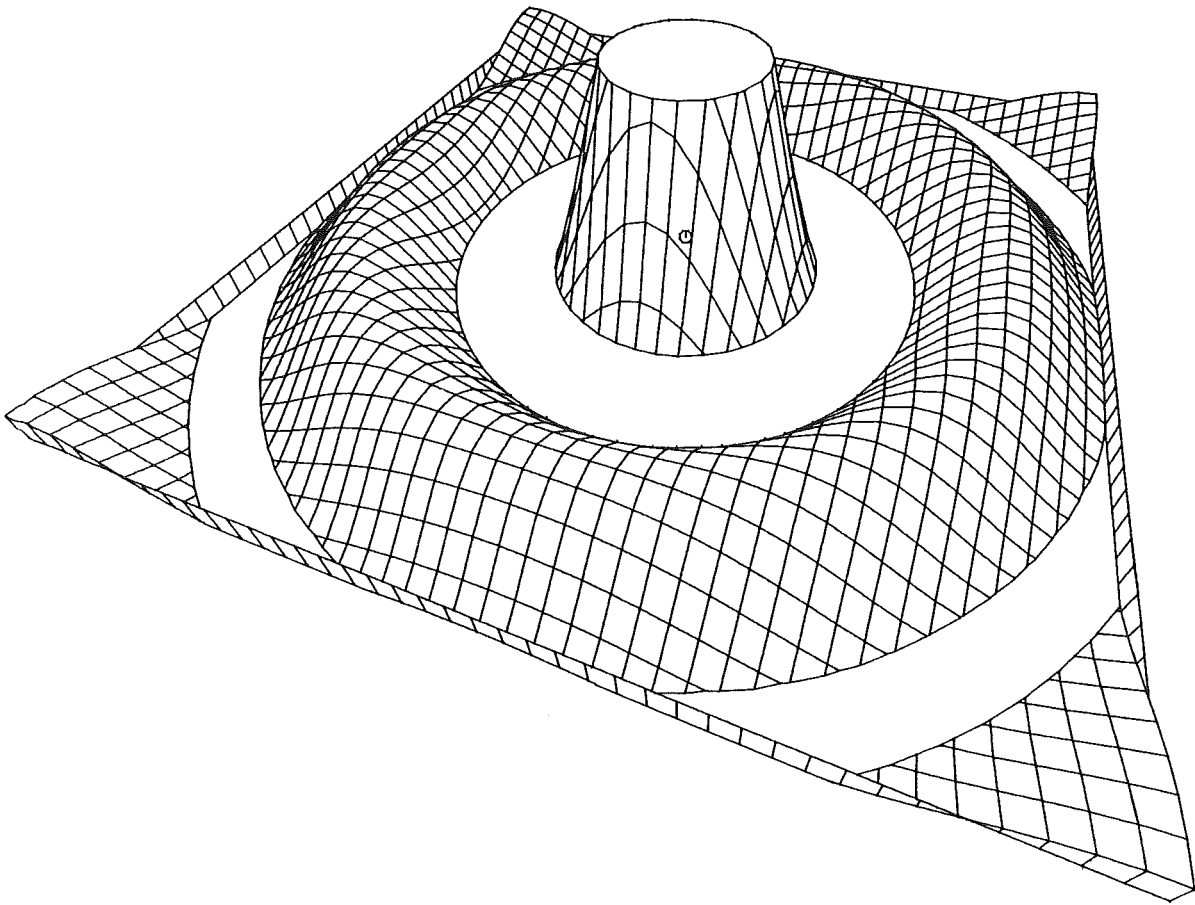
Zeichnen von Texten in Normal- und Schrägschrift

Programmbeispiel 9

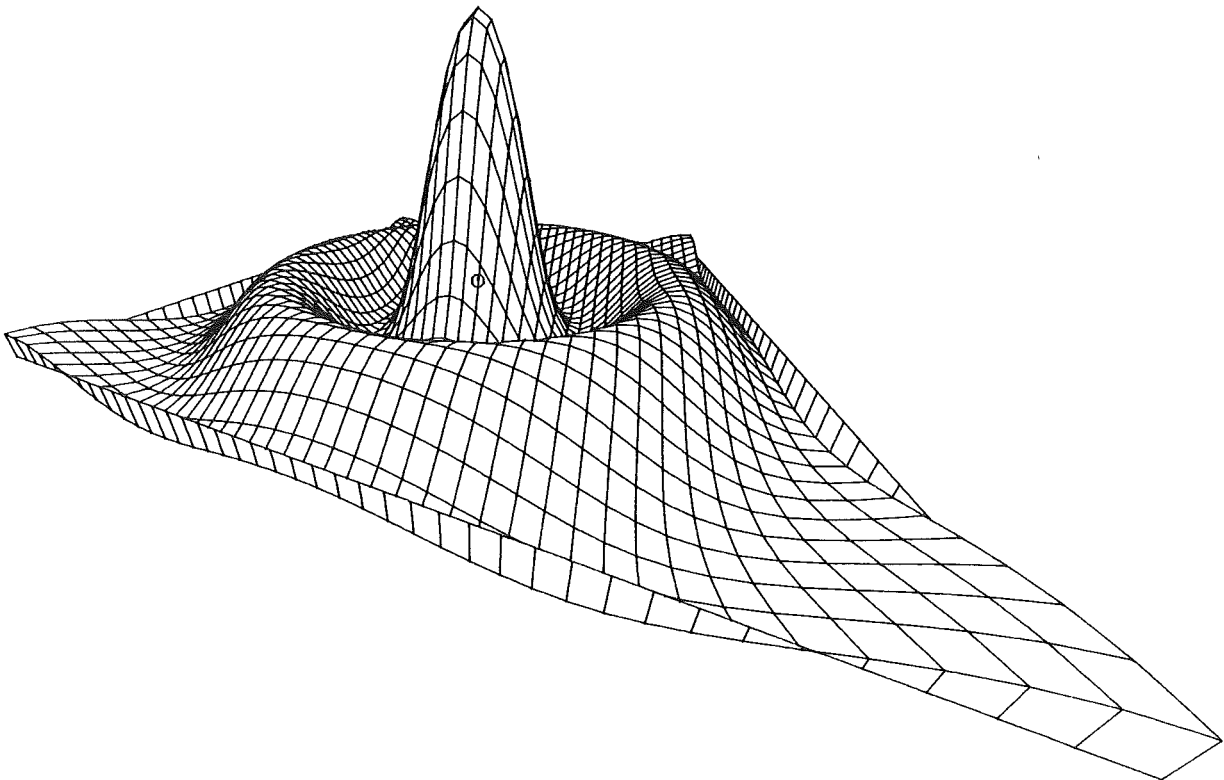
```

PROGRAM D3PLOT
C ANWENDUNG DES MODULS "DTPLOTN" (NACH S. KLEINHEINS, INR/KfK)
CHARACTER TEXT1*9, TEXT2*9
INTEGER*2 IC(4)
INTEGER*4 LZ(1681)
REAL Z(41,41), X(41), Y(41)
DATA TEXT1/'D3-PLOT/1'/, TEXT2/'D3-PLOT/2'/
DATA IC/ 1, 1, 1, 1/
C DEFINITION DER FUNKTIONSOBERFLÄCHE
DO 10 IY = 1,41
YIY = FLOAT(21-IY)*.5
DO 10 IX = 1,41
XIX = FLOAT(21-IX)*.5
D = YIY**2 + XIX**2
R = SQRT(D)
IF(R.EQ.0.) R = 1.E-6
10 Z(IY,IX) = SIN(R)/R
DO 20 I = 1,41
X(I) = FLOAT(-21 + I)
20 Y(I) = X(I)
C HP-PLOTTER
CALL PLOTI('COM1',4, 1,20141)
C SCREEN
C CALL PLOTI( 0,0,17,20141)
CALL DTPLOT(2,1, 0,1., 1.,19.,14.,-9,TEXT1,0.,0.,0.,30.,70.,20.,
+ 20.,41,41,X,Y,Z,3.,-2.,0,0)
CALL PLOTP(0.,14.85,-3)
CALL DTPLOT(4,3,-1,1.,-1.,19.,14.,-9,TEXT2,0.,0.,0.,38.,50.,20.,
+ 20.,41,41,X,Y,Z,.5,-.07,LZ,IC)
CALL PLOTP(0.,0.,999)
STOP
END

```



D3-PLOT/2



D3-PLOT/1

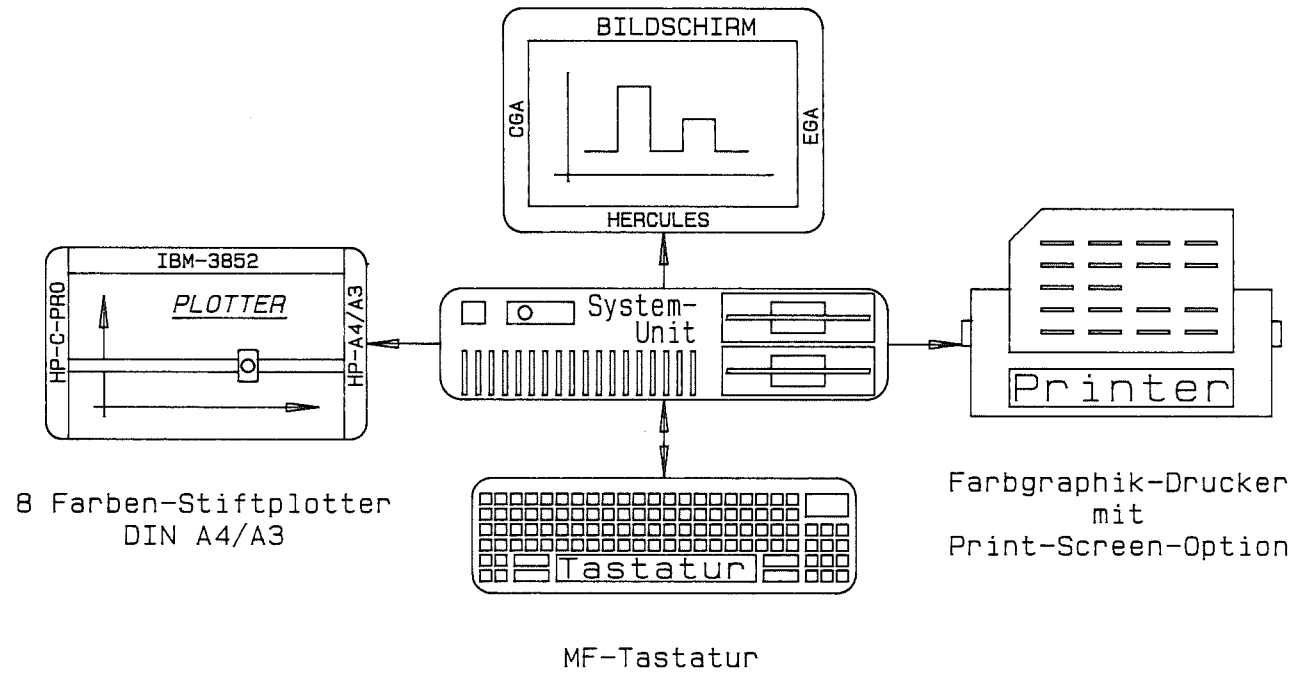
Programmbeispiel 10

```

PROGRAM CONFIG
C   HP-PLOTTER
    CALL PLOTI('AUX',3,1,30140,'IRB')
C   SCREEN
C   CALL PLOTI(0,0,17,30140,'IRB')
    CALL SYMBP1(14.85,1.5,-.45,
+ 'Geeignete Hardware-Konfiguration für Farbgraphik-Applikationen',
+ 0.,62)
    CALL SYSUNI(14.85,11.,7.)
    CALL IMPRIM(22.00,11.5,5.)
    CALL SYMBP1(22.,8.85,-.28,'Farbgraphik-Drucker',0.,19)
    CALL SYMBP1(22.,8.35,-.28,'mit',0.,3)
    CALL SYMBP1(22.,7.85,-.28,'Print-Screen-Option',0.,19)
    CALL GRAPH(7.70,11.,5.)
    CALL SYMBP1(7.7,8.5,-.28,'8 Farben-Stiftplotter',0.,21)
    CALL SYMBP1(7.7,8.0,-.28,'DIN A4/A3',0.,9)
    CALL CONSOL(14.85,8.,6.)
    CALL SYMBP1(14.85,6.1,-.28,'MF-Tastatur',0.,11)
    CALL ECRAN(14.85,14.5,5.)
    CALL SYMBP1(14.85,17.7,-.28,'farbgraphikfähiger',0.,18)
    CALL SYMBP1(14.85,17.2,-.28,'Bildschirm',0.,10)
    CALL FLECHE(14.85,8.9,14.85,10.125,3)
    CALL FLECHE(14.85,11.875,14.85,12.75,2)
    CALL FLECHE(11.35,11.,10.20,11.,2)
    CALL FLECHE(18.35,11.,19.50,11.,2)
    CALL PLOTP(0.,0.,999)
STOP
END

```

farbgraphikfähiger
Bildschirm



Geeignete Hardware-Konfiguration für Farbgraphik-Applikationen



39. Alphabetische Übersicht der Graphikmodule

	Modul	Seite
1.	AXLIN1	21
2.	AXLOG1	25
3.	AXYFEN/AXYRES	19
4.	AYLIN1	29
5.	AYLOG1	33
6.	BALKEN	79
7.	CADRE/CADRE2	13
8.	CERCLE	71
9.	CONSOL	91
10.	CURVE	37
11.	CURVE2	39
12.	CURVEP	47
13.	CURVLV	53
14.	CURVP2	49
15.	DISY	86
16.	DTPLOT	95
17.	ECRAN	88
18.	ELLIPS	73
19.	FLECHE	77
20.	FNTX	61
21.	GRAPH	89
22.	IMPRIM	90
23.	KFK	69
24.	NUMBP/NUMBS	59
25.	PEN	11
26.	PLOTI	5
27.	PLOTP	15
28.	POL	43
29.	RASTER	83
30.	ROUND	93
31.	SFCAD	81
32.	STRIP	75
33.	SYMBP1-4/SYMBBS1	55
34.	SYSUNI	92
35.	TASY	87
36.	TEXT/TEXTS	65
37.	WHEREP	17