

KfK 3237
November 1981

TRACEGS
Ein Programmsystem
für die Zeichnungserstellung
auf der Basis des
graphischen Systems (GS)

M. Politzky
Institut für Reaktorbauelemente
Projekt Schneller Brüter

Kernforschungszentrum Karlsruhe

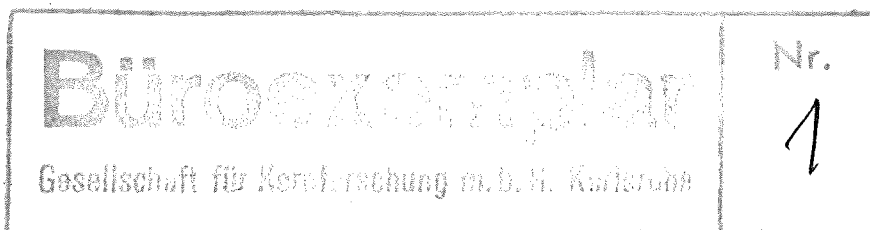
KERNFORSCHUNGSZENTRUM KARLSRUHE
Institut für Reaktorbauelemente
Projekt Schneller Brüter

KfK 3237

TRACEGS

Ein Programmsystem für die Zeichnungserstellung
auf der Basis des graphischen Systems (GS)

M. Politzky



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

Kernforschungszentrum Karlsruhe GmbH
ISSN 0303-4003

T R A C E G S

Ein Programmsystem für die Zeichnungserstellung auf der Basis des Graphischen Systems (GS).

Zusammenfassung:

Es wird ein Programmsystem vorgestellt, welches die Erstellung von Zeichnungen, Diagrammen usw. gestattet mit der Möglichkeit der Bildausgabe auf den On- bzw. Off-Line-Zeichengeräten VERSATEC, STATOS und XYNETICS und den Bildschirmgeräten 'Karlsruher Sichtgerät', 'Graphischer Anschluß' und 'TEKTRONIX 4015'.

Es handelt sich vorwiegend um Funktionssoftware auf der Basis des an unserer zentralen Rechenanlage verfügbaren Graphischen Systems (GS).

Neben einer detaillierten Beschreibung der Programm-Moduln wird auch auf die Handhabung des Systems eingegangen.

A Program-System for Preparing Drawings on the Basis of the Graphical System - GS.

Summary:

A program system for preparing drawings, diagrams etc. is presented. The possibility is given to plot these on the on-line respectively off-line plotters VERSATEC, STATOS, and XYNETICS as well as to display these on the graphic terminals ('Karlsruher Sichtgerät', 'Graphischer Anschluß', 'TEKTRONIX 4015').

It deals predominantly with the function software on the basis of the Graphical System, GS, available at our central computer.

In addition to a detailed description of the program modules the operation of the system is covered.



Inhaltsverzeichnis

	Seite
1. Einleitung.....	1
2. PLOTI - Die Initialisierungsroutine.....	2
3. PLOTP - Bewegen des Schreibstiftes.....	4
4. SYMBP/ - Zeichnen von Zeichenketten (Großbuchstaben) und Symbolen in... SYMBS Normal- oder Schrägschrift	5
4.1 SYMBP1/- Zeichnen von Zeichenketten (Kleinbuchstaben) und Symbolen in.. SYMBS1 Normal- oder Schrägschrift	7
4.2 SYMBP2/- Zeichnen von Zeichenketten (griechisch, groß) und Symbolen in.. SYMBS2 Normal- oder Schrägschrift	7
4.3 SYMBP3/- Zeichnen von Zeichenketten (griechisch, klein) und Symbolen.. SYMBS3 in Normal- oder Schrägschrift	8
5. NUMBP/ - Zeichnen von Zahlen in Normal- oder Schrägschrift..... NUMBS	8
6. WHEREP - Ermittlung der aktuellen Koordinaten.....	10
7. FACTOR - Bild-Maßstabsfaktor.....	11
8. PEN - Auswahl der Zeichenfarbe.....	12
9. AXLIN1 - Zeichnen einer linearen X-Achse.....	12
10. AYLIN1 - Zeichnen einer linearen Y-Achse.....	15
11. AXLOG1 - Zeichnen einer logarithmischen X-Achse.....	19
12. AYLOG1 - Zeichnen einer logarithmischen Y-Achse.....	22
13. POL - Zeichnen eines Polarkoordinatensystems.....	25
14. CURVE - Zeichnen eines Kurvenverlaufes in ein kartesisches..... Koordinatensystem	27
15. CURVEP - Zeichnen eines Kurvenverlaufes in ein Polarkoordinatensystem.	29
16. CURVLV - Zeichnen eines durchgezogenen oder gestrichelten..... Streckenzeuges	30
17. CADRE - Zeichnen eines Rahmens.....	32
18. CADREF - Festlegen eines fiktiven Rahmens.....	32
19. KFK - Zeichnen des KfK-Symbols.....	33
20. TEXT/ - Zeichnerische Darstellung von Texten in Normal- oder..... TEXTS Schrägschrift	35
21. Symbole für den Bereich: Allgemeines Zeichnen.....	37
21.1 FLECHE - Zeichnen von Linien und Pfeilen.....	38
21.2 DISY - Zeichnen eines Speicher-Symbols (Zylinder).....	39
21.3 TASY - Zeichnen eines Magnetband-Symbols.....	40
21.4 ECRAN - Zeichnen eines Symbols für ein Bildschirmgerät.....	40

21.5 STRIP - Zeichnen von verschiedenartigen Linien.....	41
21.6 CERCLE - Zeichnen von Kreisen und Kreisteilen.....	42
21.7 BALKEN - Zeichnen von schraffierten und unschraffierten Balken.....	44
22. Routinen für spezielle Anwendungen.....	45
22.1 EX - Vergrößerung, Verkleinerung, Verzerrung bei der Bild- ausgabe	45
22.2 ROT90 - Drehen eines Bildes um 90 Grad bei der Bildschirmausgabe.....	46
23. TRACEGS im TSO-Foreground.....	47
24. TRACEGS im Stapelbetrieb.....	54
25. Kompatibilität mit der konventionellen Software.....	57
26. Programm-Konfiguration.....	58
27. Literatur.....	60
28. Anhang: Abbildungen - Programmbeispiele.....	61

1. Einleitung

TRACEGS ist ein in der Programmiersprache Fortran IV geschriebenes, modular aufgebautes Programmsystem, welches dem Benutzer erweiterte Möglichkeiten bietet, auf relativ unkomplizierte Weise Bildausgaben zu erzeugen.

Als Basis für die Programmentwicklung diente das an unserer zentralen Rechenanlage (IBM /370 - /3033) residente Graphische System (GS) /1/. Dieses System hat aufgrund seines moderneren Konzeptes gegenüber der konventionellen Zeichen-Software (CALCOMP) /2/ Anwendungsvorteile, die, soweit es sich realisieren ließ, auch in TRACEGS verfügbar sind.

Diese Vorteile betreffen insbesondere die Bildausgabe auf allen z. Zt. verfügbaren Bildausgabegeräten wie Bildschirme (Graphischer Anschluß, TEKTRONIX 4015, Karlsruher Sichtgerät) und On- bzw. Off-Line-Zeichengeräte (STATOS, XYNETICS, VERSATEC).

Darüberhinaus besteht die Möglichkeit zur Speicherung von Bildinformation in permanenten Bildfiles (Bücher) auf Speichermedien mit direktem Zugriff. Der Vorteil ist darin zu sehen, daß ein einmal erstelltes Bildfile auf beliebigen Ausgabegeräten wiederholt graphisch dargestellt werden kann.

Um Benutzern der konventionellen Zeichen-Software einen aufwendigen Umlernprozeß zu ersparen wurde versucht, die Moduln namentlich und von der Funktion her möglichst so aufzubauen, daß sich deren Anwendung in der althergebrachten Denkweise vollziehen kann.

Dies führte dazu, daß in einigen Moduln (z. B. PLOT, SYMBOL, NUMBER) bestimmte Zustände (z. B. aktuelle Koordinaten) simuliert werden mußten, da GS keine der konventionellen Zeichen-Software entsprechenden Basis-Programme enthält.

Um mit diesem Bericht eine brauchbare Arbeitsunterlage (Handbuch) anzubieten, werden im folgenden alle verfügbaren Moduln in bezug auf die Funktion und die erforderliche Eingabe detailliert beschrieben.

Hinweis für externe Leser

TRACEGS benötigt zur Lösung der Referenzen neben der Basis-Software des Graphischen Systems auch andere KfK-spezifische Bibliotheken.

Aufgrund des relativ komplexen Systemaufbaues werden Interessenten gebeten, sich mit dem Autor in Verbindung zu setzen.

2. PLOTI - Die Initialisierungsroutine

Bevor in einem Programm Instruktionen programmiert werden, die der Erstellung einer Zeichnung dienen, muß der Aufruf des Unterprogrammes PLOTI erfolgen.

In der Routine PLOTI werden COMMON-Variablen Ursprungswerte zugewiesen und das GS-Zeichensystem initialisiert.

Als Bezugspunkt für künftige Zeichenaktivitäten ist die linke untere Ecke der Zeichenfläche definiert.

Aufruf: CALL PLOTI(IPT,IWS)

Die Argumente haben folgende Bedeutung:

IPT	(-)	Auswahl des Bildgenerators (Bildausgabe).
		IPT = 1 oder 10 Foreground-Version_: Graphischer Anschluß
		Batch-Version_____: STATOS-Plotter (Off-Line)
		IPT = 2 oder 20 Foreground-Version_: TEKTRONIX-4015-Display
		Batch-Version_____: XYNETICS-Plotter (Off-Line)
		IPT = 3 oder 30 Foreground-Version_: Karlsruher Sichtgerät
		Batch-Version_____: VERSATEC-Plotter (On-Line)
		IPT = 0 Foreground/Batch___: nur Speicherung der Bild-
		information

Für IPT = 10, 20 oder 30 erfolgt zusätzlich die Speicherung der Bildinformation. In diesen Fällen und auch für IPT = 0 muß zur Programmausführung ein Bild-Data-Set verfügbar sein (siehe hierzu auch die Punkte 20. und 21.).

IWS (k Byte) Größe des Arbeitsspeichers für die Bilderstellung. Die Bereitstellung eines Arbeitsspeichers ist GS-spezifisch. Die notwendige Region ist abhängig vom Bildgenerator (KSG, GA usw.).

Da es für den Benutzer schwierig ist abzuschätzen, wieviel Kernspeicher ein darzustellendes Bild benötigt, sollte ein möglichst großer Wert der Variablen IWS zugewiesen werden. Kann der geforderte Arbeitsspeicher vom System nicht zur Verfügung gestellt werden (passiert zuweilen im TSO-Fore-

ground), reduziert das GS die Region auf die maximal verfügbare.

Die Erfahrung hat gezeigt, daß in den meisten Fällen ein Arbeitsspeicher von 500 ...1000 k Byte benötigt wird.

Von PLOTI aufgerufene Unterprogramme:

GS-Routinen_____ : INITFG, PI, PA, SCO

TRACEGS-Routinen_____ : keine

periphere Routinen_____ : IBCOM#

Anmerkungen zum 'Bild' und zur 'Bildspeicherung'.

Unter einem 'Bild' ist hier eine bestimmte Menge an Zeicheninformation zu verstehen. Eingeschlossen wird ein Bild immer durch den Initialisierungsaufruf mit PLOTI(IPT,IWS) und den Abschlußaufruf mit PLOTP(x-Arg.,y-Arg.,999). Dazwischen können beliebig viele andere Aufrufe von Basis- und Funktions-Zeichenroutinen erfolgen.

Unter 'Bildspeicherung' ist eine temporäre oder permanente Ablage von Zeichnungsinformation in einer Bild-Datei (Buch) zum Zweck der Archivierung oder Bildreproduktion zu verstehen.

Da ein Bild prinzipiell in einer vom Bildausgabegerät unabhängigen Form abgespeichert wird, besteht somit die Möglichkeit, dieses wiederholt auf einem beliebigen Zeichengerät mit Hilfe des entsprechenden Interface-Programmes darzustellen, ohne daß ein erneutes Erarbeiten der Bildinformation erforderlich ist.

3. PLOTP - Bewegen des Schreibstiftes

Das Unterprogramm PLOTP dient der Ausführung von Schreibstift- bzw. Lichtzeiger-Bewegungen. Die Arbeitsweise von PLOTP ist sinngemäß die gleiche wie die des Unterprogrammes PLOT bei der CALCOMP-Software. Für die Einbindung von PLOTP in das Graphische System war es erforderlich, die Funktion dieses Unterprogrammes mit den verfügbaren Basisroutinen des GS zu simulieren.

Aufruf: CALL PLOTP(XK,YK,IP)

Die Argumente haben folgende Bedeutung:

XK (cm) Koordinaten, die, bezogen auf den gültigen Bezugspunkt, den Zielpunkt des Schreibstiftes bzw. Lichtzeigers angeben.

YK (cm) Koordinaten, die, bezogen auf den gültigen Bezugspunkt, den Zielpunkt des Schreibstiftes bzw. Lichtzeigers angeben.

IP (-) Dieses Argument bietet die Möglichkeit, die Stellung des Schreibstiftes während einer Positionsänderung zu steuern.

Es bedeuten:

IP = 2 Schreibstift in abgesenkter Stellung. Die aktuelle Position und der Zielpunkt (XK,YK) werden mit einer sichtbaren Linie verbunden.

IP = 3 Schreibstift bewegt sich in abgehobener Stellung zu den Zielkoordinaten.

IP = -2 Gleiche Bedeutung wie IP=2 bzw. IP=3. Die und Zielkoordinaten bilden jedoch einen neuen Bezugspunkt für folgende Koordinatenangaben.

IP = -3 zugspunkt für folgende Koordinatenangaben.

IP =999 Abschluß eines Bildes. Der Aufruf von PLOTP mit IP=999 muß immer der letzte Aufruf einer Zeichenroutine sein. Wird dieser Wert nicht angegeben, erfolgt keine Bildausgabe.

Anmerkung: Für die zeichnerische Darstellung eines Streckenzuges, der aus mehreren Wertepaaren gebildet wird, ist die Verwendung von PLOTP wegen des relativ hohen Speicherbedarfs nicht zu empfehlen. Hierfür sei

die Verwendung des Unterprogrammes CURVLV empfohlen (s. Punkt 13.1).

Von PLOTP aufgerufene Unterprogramme:

GS-Routinen _____: ECO, ENDGS, GSZA, LIN, PA, SCA, SCO, SPIB,
STR, RG, RPI, ROUT

TRACEGS-Routinen _____: keine

periphere Routinen _____: IBCOM#

4. SYMBP/SYMBS - Zeichnen von Zeichenketten (Großbuchstaben) und Symbolen in Normal- oder Schrägschrift

SYMBP und SYMBS sind allgemeine Unterprogramme für die zeichnerische Darstellung von alphanumerischen Zeichen (Großbuchstaben, Ziffern, Sonderzeichen) und Symbolen. SYMBP ist die Normalschrift-, SYMBS die Schrägschrift-Version. Es können alle Zeichen des TSO-Zeichenvorrates dargestellt werden. Symbole werden sowohl in SYMBP als auch in SYMBS nur in Normalschreibweise wiedergegeben. Die äußere Form der Routinen (Eingabe, Ausgabe) ist identisch mit derjenigen, wie sie die konventionelle Software (CALCOMP-SYMBOL) anbietet.

Für TRACEGS wurde die Funktionsweise von SYMBP und SYMBS mit den GS-Basis-Routinen simuliert.

Neben dem Zeichnen von Zeichenketten oder Symbolen werden in SYMBP und SYMBS auch die neuen Schreibstiftkoordinaten berechnet. Es sind diejenigen, die erforderlich wären, wenn eine weitere Zeichenkette unmittelbar im Anschluß an die soeben gezeichnete folgen würde. Die neuen aktuellen Koordinaten werden an die Variablen X01 und Y01 des benannten 'COMMON /XYPOS/' übergeben und sind damit über WHEREP (siehe Punkt 6.) allgemein verfügbar.

Aufruf: CALL SYMBP(X,Y,HS,TEXT,WINK,IT)

oder

CALL SYMBS(X,Y,HS,TEXT,WINK,IT)

Die Argumente haben folgende Bedeutung:

X	(cm)	Koordinaten für den Beginn des Textes oder des zu
Y	(cm)	zeichnenden Symbols, bezogen auf den zuletzt definier-

ten Bezugspunkt.

Bei Textdarstellung weisen die Koordinaten auf die linke untere Ecke der zu zeichnenden Textzeile, bei Symboldarstellung auf das Zentrum der Symbole.

Soll der Text bzw. ein Symbol bei den Koordinaten beginnen, die vor Aufruf von SYMBP/SYMBS aktuell waren, so ist für X und/oder Y der Wert 999. einzugeben.

- HS (cm) Höhe der Textzeichen.
Bei der Darstellung von Symbolen ist zu beachten, daß diese immer mit einer Höhe von ca. $0.857 * HS$ gezeichnet werden.
- TEXT (-) Zu zeichnender Text oder Angabe der Ordnungszahl eines Symbols. Die maximale Länge beträgt für Texte 80 Zeichen. Der Text kann im Hollerithformat (nH...), als Literal (in Hochkommas eingeschlossen) oder als vordefinierte Textvariable angegeben werden.
Für die Symboldarstellung steht folgende Auswahl zur Verfügung:
Ordnungszahl 0 1 2 3 4 5 6 7 8 9 10 11 12 13
Symbol \square \odot \triangle + \times \diamond \uparrow \times Z Y \boxtimes * \boxtimes |
- WINK (Grad) Winkel unter dem der Text oder das gewünschte Symbol gezeichnet werden soll.
- IT (-) Ein positiver Wert für IT gibt die Anzahl der zu zeichnenden Textzeichen an.
Ist der Wert -1 oder ≤ -2 wird angezeigt, daß ein Symbol (s. auch Argument TEXT) zu zeichnen ist. Bei -1 wird die Symbolposition mit abgehobenem, bei ≤ -2 mit abgesenktem Schreibstift angefahren.

Von SYMBP/SYMBS aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, STRO (SYMBP), STR10 (SYMBS), (WPIO10)

TRACEGS-Routinen _____: PLOTP

periphere Routinen _____: COS, SIN

4.1 SYMBP1/SYMBS1 - Zeichnen von Zeichenketten (Kleinbuchstaben) und Symbolen in Normal- oder Schrägschrift

SYMBP1 und SYMBS1 sind Unterprogramme für die zeichnerische Darstellung von Zeichenketten mit kleinen Buchstaben und Symbolen.

Für SYMBP1 und SYMBS1 gilt sinngemäß dasgleiche wie für die unter Punkt 4. beschriebenen Unterprogramme SYMBP und SYMBS.

Aufruf: CALL SYMBP1(X,Y,HS,TEXT,WINK,IT)
oder
CALL SYMBS1(X,Y,HS,TEXT,WINK,IT)

Die Zeichenketten werden in Großbuchstaben vorgegeben. Für die Umlaute (klein), das Eszett und eckige Klammer auf bez. zu hat folgende Korrespondenz Gültigkeit:

Eingabe__: 1 2 3 4 ()
Abbildung: ä ö ü ß []

Von SYMBP1/SYMBS1 aufgerufene Unterprogramme:

GS-Routinen_____: GSZA, STR1 (SYMBP1), STR11 (SYMBS1), (WPIO)
TRACEGS-Routinen_____: PLOTP
periphere Routinen_____: COS, SIN

4.2 SYMBP2/SYMBS2 - Zeichnen von Zeichenketten (griechisch, groß) und Symbolen in Normal- oder Schrägschrift

SYMBP2 und SYMBS2 sind Unterprogramme für die zeichnerische Darstellung von Zeichenketten mit griechischen Großbuchstaben und Symbolen.

Für SYMBP2 und SYMBS2 gilt sinngemäß dasgleiche wie für die unter Punkt 4. beschriebenen Unterprogramme SYMBP und SYMBS.

Aufruf: CALL SYMBP2(X,Y,HS,TEXT,WINK,IT)
oder
CALL SYMBS2(X,Y,HS,TEXT,WINK,IT)

Die Zeichenketten werden in lateinischen Großbuchstaben bzw. Ziffern vorgegeben. Folgende Korrespondenz hat Gültigkeit:

Eingabe__:	C	D	F	G	L	P	R	S	2	3	4
Abbildung:	χ	Δ	Φ	Γ	Λ	Π	Ρ	Σ	Θ	Ψ	Ω

Von SYMBP2/SYMBS2 aufgerufene Unterprogramme:

GS-Routinen_____ : GSZA, STR2 (SYMBP2), STR12 (SYMBS2), (WPOIO)
TRACEGS-Routinen_____ : PLOTP
periphere Routinen_____ : COS, SIN

4.3 SYMBP3/SYMBS3 - Zeichnen von Zeichenketten (griechisch, klein) und Symbolen in Normal- oder Schrägschrift

SYMBP3 und SYMBS3 sind Unterprogramme für die zeichnerische Darstellung von Zeichenketten mit griechischen Kleinbuchstaben und Symbolen.

Für SYMBP3 und SYMBS3 gilt sinngemäß dasgleiche wie für die unter Punkt 4. beschriebenen Unterprogramme SYMBP und SYMBS.

Aufruf: CALL SYMBP3(X,Y,HS,TEXT,WINK,IT)
oder
CALL SYMBS3(X,Y,HS,TEXT,WINK,IT)

Die Zeichenketten werden in lateinischen Großbuchstaben bzw. Ziffern vorgegeben. Folgende Korrespondenz hat Gültigkeit:

A	B	C	D	E	F	G	I	K	L	M	N	O	P	R	S	T	X	Y	Z	1	2	3	4
α	β	χ	δ	ε	φ	γ	ι	κ	λ	μ	ν	ο	π	ρ	σ	τ	ξ	υ	ζ	η	θ	ψ	ω

Von SYMBP3/SYMBS3 aufgerufene Unterprogramme:

GS-Routinen_____ : GSZA, STR3 (SYMBP3), STR13 (SYMBS3), (WPOIO)
TRACEGS-Routinen_____ : PLOTP
periphere Routinen_____ : COS, SIN

5. NUMBP/NUMBS - Zeichnen von Zahlen in Normal- oder Schrägschrift

Die Unterprogramme NUMBP und NUMBS ermöglichen die zeichnerische Darstellung von Gleitkommazahlen entsprechend dem F-Ausgabeformat

(Fortran). NUMBP ist die Normalschrift-, NUMBS die Schrägschrift-Version.

Die Form der Ein- und Ausgabe von NUMBP und NUMBS ist mit der konventionellen Software (CALCOMP-NUMBER) identisch.

Die Funktionsweise von NUMBP/NUMBS wurde in TRACEGS mit den Basis-Routinen des Graphischen Systems simuliert.

Neben der zeichnerischen Darstellung von Gleitkommazahlen werden während der Ausführung von NUMBP/NUMBS auch die neuen Schreibstiftkoordinaten bestimmt. Es sind diejenigen, die erforderlich sind, wenn unmittelbar im Anschluß an die gezeichnete Zahl weitere Bilddarstellungen (Zahl, Text, Linienzug usw.) folgen sollen. Bezüglich der neu berechneten Koordinaten gilt für NUMBP und NUMBS sinngemäß dasselbe wie für die unter Punkt 4. beschriebenen Unterprogramme SYMBP und SYMBS.

Aufruf: CALL NUMBP(X,Y,HS,ZAHL,WINK,NK)
oder
CALL NUMBS(X,Y,HS,ZAHL,WINK,NK)

Die Argumente haben folgende Bedeutung:

X (cm) Koordinaten für den Beginn der Zahlendarstellung (linke
Y (cm) untere Ecke des Ziffernstrings) bezogen auf den zuletzt definierten Bezugspunkt.

Soll eine Zahl bei den Koordinaten beginnen, die vor Aufruf von NUMBP/NUMBS aktuell waren, so ist für X und/oder Y der Wert 999. einzugeben.

HS (cm) Höhe der Zahlzeichen.

ZAHL (-) Darzustellende Gleitkommazahl.

WINK (Grad) Winkel, unter dem die darzustellende Zahl gezeichnet werden soll.

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

Beispiel: ZAHL = 27.123

Darstellung bei NK = 3: 27.123

NK = 2: 27.12

NK = 0: 27.

NK = -1: 27

Von NUMBP/NUMBS aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, STRO (NUMBP), STR10 (NUMBS)

TRACEGS-Routinen _____: keine

periphere Routinen _____: ALOG10, COS, SIN, ZST

6. WHEREP - Ermittlung der aktuellen Koordinaten

Mit Aufruf des Unterprogrammes WHEREP lassen sich während der Programmausführung die augenblicklichen Schreibstift- (Lichtzeiger-) Koordinaten und der Maßstabsfaktor ermitteln. Dabei ist die Kenntnis über den aktuellen Maßstabsfaktor ohne Bedeutung, da dieser nur einmal während der Bildgenerierung angegeben werden kann (siehe hierzu auch Punkt 7.). Um jedoch die Kompatibilität mit der CALCOMP-Software zu gewährleisten, wurde die originale äußere Form von WHEREP beibehalten.

Da das graphische System kein Basis-Programm enthält, das die aktuellen Koordinaten bereitstellt, mußte dieser Prozeß in TRACEGS simuliert werden. Die Simulation wird dabei auf folgende Weise durchgeführt: Das Unterprogramm WHEREP kommuniziert über den benannten 'COMMON /XYPOS/ X00, Y00, X01, Y01, IFB' mit den Unterprogrammen PLOTI, PLOTP, NUMBP, NUMBS, SYMBP, SYMBS, SYMBP1, SYMBS1, SYMBP2, SYMBS2, SYMBP3, SYMBS3, CURVE, CURVLV und STRIP. Bei Aufruf und Ausführung der genannten Programme werden die aktuellen Schreibstiftkoordinaten an die COMMON-Variablen X01 und Y01 übergeben, die damit auch in WHEREP bekannt sind und erfragt werden können.

Die Verwendung von WHEREP kann insbesondere bei Zahlen- und Textfortsetzungen, Zeichnen von Indizes usw. sehr hilfreich sein, da eine umständliche Berechnung von Anschlußkoordinaten entfällt.

Nach Ausführung der Unterprogramme CURVE, CURVLV und STRIP (s. auch die Punkte 13., 13.1 und 18.5) hat der Schreibstift die Koordinaten des zuletzt dargestellten Wertepaares.

Den Maßstabsfaktor bezieht WHEREP von dem benannten 'COMMON /SCALA/ SXY'.

Aufruf: CALL WHEREP(XK0,YK0,FACT)

Die Argumente haben folgende Bedeutung:

XKO (cm) Aktuelle Schreibstift- (Lichtzeiger-) Koordinaten.

YKO (cm)

FACT (-) Bild-Maßstabsfaktor.

Wichtig: Als Argumente dürfen keine Konstanten verwendet werden!

Von WHEREP werden keine anderen Unterprogramme aufgerufen.

7. FACTOR - Bild-Maßstabsfaktor

Das Unterprogramm FACTOR gestattet die beliebige Vergrößerung bzw. Verkleinerung eines Bildes.

Gegenüber der konventionellen Software besteht in TRACEGS die Einschränkung, daß FACTOR nicht auf einzelne Bildteile angewendet werden kann, sondern nur auf das gesamte Bild. Das bedeutet, daß während der Generierung eines Bildes nur ein Aufruf von FACTOR berücksichtigt werden kann. Wird in einem Programm mehrmals das Unterprogramm FACTOR aufgerufen, hat nur der letzte Aufruf Gültigkeit.

Aufruf: CALL FACTOR(FACT)

Das Argument FACT stellt einen Multiplikator für den Bildmaßstab dar. Ein Wert größer als 1. bewirkt eine Bildvergrößerung, ein Wert kleiner als 1. eine Bildverkleinerung. Der Defaultwert für FACT ist 1..

Anmerkung: Die Bild-Vergrößerung bzw. -Verkleinerung ist an Bildschirmgeräten unwirksam, da hier grundsätzlich auf der maximalen Bildfläche abgebildet wird.

Im Unterprogramm FACTOR werden weder GS-Basis-Routinen noch andere Unterprogramme aufgerufen.

Der Wert von FACT ist im benannten 'COMMON /SCALA/ SXY' resident. Er wird erst bei der Ausgabe des fertigen Bildes mit PLOTP(X,Y,999) benötigt.

Hinweis: Eine Bildvergrößerung oder -verkleinerung, die auch an den

Bildschirmgeräten wirksam ist, kann mit Hilfe des Unterprogrammes EX (siehe Punkt 19.1) erzielt werden.

8. PEN - Auswahl der Zeichenfarbe

Das Unterprogramm PEN ermöglicht die farbige Darstellung von Achsen, Linienzügen, Symbolen, Texten usw..

Da die Wahl der Farbe im Graphischen System in anderer Form ausgeführt wird (String-Argument) als es bei der CALCOMP-Software üblich ist, wurde die Routine PEN in TRACEGS in der Weise entwickelt, daß sie äußerlich dem Gebrauch der konventionellen Software entspricht.

PEN kommuniziert über den benannten 'COMMON /XYPOS/ X00, Y00, X01, Y01, IFB' mit allen Zeichenunterprogrammen, die aktiv der Bilderstellung dienen.

Aufruf: CALL PEN(IFB)

Über das Argument IFB wird die Farbe gewählt. Es gelten folgende Zuordnungen:

IFB = 1 schwarze Farbe
 = 2 rote "
 = 3 blaue "
 = 4 grüne "

Die Wahl einer bestimmten Farbe ist nur bei Verwendung des XYNETICS-Plotters relevant. Bei der Darstellung von Bildern auf dem STATOS-Plotter oder den Bildschirmen (GA, T4015, Graphischer Anschluß) werden Farbangaben nicht berücksichtigt.

Von PEN werden keine anderen Unterprogramme aufgerufen.

9. AXLIN1 - Zeichnen einer linearen X-Achse

Das Unterprogramm AXLIN1 gestattet die Konstruktion einer linearen X-Achse. Die Richtung der Achse auf dem Zeichnungsträger liegt fest. Am STATOS- und VERSATEC-Plotter fällt die Richtung der X-Achse mit der Papierflußrichtung zusammen. Bei dem XYNETICS-Plotter erstreckt sich die X-Achse bei den z. Zt. verwendeten Papierbögen immer in Richtung der längeren Bogenseite. An den Bildschirmgeräten (T4015, KSG, Graphischer Anschluß) wird die X-Achse parallel zum unteren/oberen Bildrand abgebildet.

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

Aufruf: CALL AXLIN1(XL,XMIN,XMAX,DX,XW,YL,IKM,HKM,HGM,ZSH,IGEN,IT,TEXT)

Die Argumente haben folgende Bedeutung:

XL (cm) Länge der X-Achse.

XMIN (phys. Einh.) Wert, mit dem die X-Achse beginnen soll (linkes Ende). Er muß nicht zwangsläufig als Graduation erscheinen, da dies von den angegebenen Werten für DX und XW abhängt.

XMAX (phys. Einh.) Wert am rechten Ende der X-Achse. Auch hier hängt das Zeichnen der entsprechenden Graduation von der Angabe der Werte für DX und XW ab.

DX (phys. Einh.) Der Wert von DX gibt an, nach wieviel physikalischen Einheiten jeweils eine große Markierung und die entsprechende Graduation gezeichnet werden soll.

XW (phys. Einh.) Wert für den gilt: $XMIN \leq XW \leq XMAX$. Es wird empfohlen, für XW einen Wert zu wählen, der mit einer großen Markierung und der entsprechenden Graduation an der Achse erscheinen soll.

YL (cm) Dieses Argument bietet die Möglichkeit, eine parallel zur X-Achse gespiegelte Achse ohne Graduationen, aber mit Markierungen im Abstand YL zeichnen zu lassen.
Sinnvollerweise sollte für YL die Länge der Y-Achse angegeben werden (Rahmen).
Wird für $YL = 0$. angegeben, entfällt das Zeichnen einer gespiegelten zweiten Achse.
YL muß auch dann ein entsprechender Wert zugewiesen werden, wenn das Zeichnen von Rasterlinien gewünscht wird. Ist hierbei YL positiv, wird das

Raster mit durchgezogenen Linien dargestellt, bei negativem YL gestrichelt. Für diesen Fall muß den Variablen HKM und/oder HGM der Wert 0. zugewiesen werden.

IKM (-) Anzahl der kleinen Markierungen an der Achse zwischen zwei großen Markierungen bzw. zwei Graduationen.

HKM (cm) Der Wert von HKM gibt die Höhe der kleinen Markierungen an der Achse an. Sollen Rasterlinien gezeichnet werden (siehe auch Argument YL), muß HKM = 0. gesetzt werden.

HGM (cm) Höhe der großen Markierungen an der Achse. Bezüglich Rasterlinien gilt das gleiche wie zuvor bei HKM beschrieben.

ZSH (cm) Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text, Graduationen). Standardmäßig werden die Markierungen auf und die Graduationen mit dem Achsentext unter die Achse gezeichnet. Die Angabe eines negativen Wertes für ZSH bedeutet hier, daß Graduationen und Achsentext über und die Markierungen unter die Achse gezeichnet werden.

Anmerkung: Unabhängig von dem Wert für YL wird bei negativem Wert für ZSH keine gespiegelte zweite Achse (Rahmen) gezeichnet.

IGEN (-) Mit dieser Variablen wird die Anzahl der Nachkommastellen der zu zeichnenden Graduationen festgelegt. Bei der Darstellung der Graduationen ohne Dezimal-exponenten bedeutet:

IGEN = 1 Graduationen mit einer Nachkommastelle.

= 0 Graduationen nur mit Dezimalpunkt.

= -1 Graduationen ohne Dezimalpunkt.

usw.

Für die Darstellung der Graduationen mit dem Dezimal-exponenten 'E' gilt:

(-11) >= IGEN >= (-14)

Es bedeutet z. B.:

IGEN = -11 Eine signifikante Nachkommastelle:
0.3E+03.

IGEN = -13 Drei signifikante Nachkommastellen:
0.123E-05.

IT (-) Die Achse kann mit einer Beschriftung versehen werden. Diese wird rechtsbündig an die Achse und parallel zu dieser gezeichnet. IT gibt die Anzahl der zu zeichnenden Textzeichen an.

TEXT (-) Text für die X-Achse. Die Zeichenkette darf maximal 80 Zeichen beinhalten. Besteht der Text aus bis zu vier Zeichen, kann er in der Argumentliste als Literal, in Hochkommas eingeschlossen, angegeben werden. Bei mehr als vier Zeichen Text muß eine Textvariable entsprechend dimensioniert und eventuell mit einer DATA-Anweisung definiert werden.

Das Unterprogramm AXLIN1 übergibt an den benannten 'COMMON /FAKTOR/' folgende für das Unterprogramm CURVE (Zeichnen eines Kurvenverlaufs, s. auch Punkt 13.) relevante Größen:

1. Kleinster Wert an der X-Achse $BXMIN = XMIN$
2. Maßstabsfaktor $XL/(XMAX-XMIN)$
3. Information (Steuergröße), daß es sich bei dieser Achse um eine lineare X-Achse handelt.

Von AXLIN1 aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN, NUM1, STRO

TRAC EGS-Routinen _____: PLOTP

per iphere Routinen _____: EXPD (CONVY /3/), NETLIN, XPOS

10. AYLINI - Zeichnen einer linearen Y-Achse

Das Unterprogramm AYLINI gestattet die Konstruktion einer linearen Y-Achse. Die Richtung der Achse auf dem Zeichnungsträger liegt fest. STATOS- und VERSATEC-Plotter stellen die Y-Achse senkrecht zur Papierflußrichtung dar. Bei dem XYNETICS-Plotter erstreckt sich die Y-Achse bei den z. Zt. verwendeten Papierbögen immer in Richtung der kürzeren Bogenseite. An den Bildschirmgeräten (T4015, KSG, Graphischer

Anschluß) verläuft die Y-Achse parallel zum rechten/linken Bildrand. Die Argumentliste von AYLINI enthält 13 Parameter, die beim Aufruf initialisiert sein müssen.

Aufruf: CALL AYLINI(YL,YMIN,YMAX,DY,YW,XL,IKM,HKM,HGM,ZSH,IGEN,IT,TEXT)

Die Argumente haben folgende Bedeutung:

- YL (cm) Länge der Y-Achse.
- YMIN (phys. Einh.) Wert, mit dem die Y-Achse beginnen soll (unteres Ende). Er muß nicht zwangsläufig als Graduation erscheinen, da dies von den angegebenen Werten für DY und YW abhängt.
- YMAX (phys. Einh.) Wert am oberen Ende der Y-Achse. Auch hier hängt das Zeichnen der entsprechenden Graduation von der Angabe der Werte für DY und YW ab.
- DY (phys. Einh.) Der Wert von DY gibt an, nach wieviel physikalischen Einheiten jeweils eine große Markierung und die entsprechende Graduation gezeichnet werden soll.
- YW (phys. Einh.) Wert für den gilt: $YMIN \leq YW \leq YMAX$. Es wird empfohlen, für YW einen Wert zu wählen, der mit einer großen Markierung und der entsprechenden Graduation an der Achse erscheinen soll.
- XL (cm) Dieses Argument bietet die Möglichkeit, eine parallel zur Y-Achse gespiegelte Achse ohne Graduation, aber mit Markierungen im Abstand XL zeichnen zu lassen.
Sinnvollerweise sollte für XL die Länge der X-Achse angegeben werden (Rahmen).
Wird für $XL = 0$. angegeben, entfällt das Zeichnen einer gespiegelten zweiten Achse.
XL muß auch dann ein entsprechender Wert zugewiesen werden, wenn das Zeichnen von Rasterlinien gewünscht wird. Ist hierbei XL positiv, wird das

Raster mit durchgezogenen Linien dargestellt, bei negativem XL gestrichelt. Für diesen Fall muß den Variablen HKM und/oder HGM der Wert 0. zugewiesen werden.

☐HKM (-) Anzahl der kleinen Markierungen an der Achse zwischen zwei großen Markierungen bzw. zwei Graduationen.

■HKM (cm) Der Wert von HKM gibt die Höhe der kleinen Markierungen an der Achse an. Sollen Rasterlinien gezeichnet werden (siehe auch Argument XL), muß HKM = 0. gesetzt werden.

■HGM (cm) Der Wert von HGM gibt die Höhe der großen Markierungen an der Achse an.
Bezüglich Rasterlinien gilt das gleich wie zuvor bei HKM beschrieben.

ZSH (cm) Der Betrag von ZSH gibt die Höhe der zu zeichnenden Zeichen und Ziffern an (Text, Graduationen). Standardmäßig werden die Markierungen und der Achsentext rechts und die Graduationen links von der Achse gezeichnet. Die Angabe eines negativen Wertes für ZSH bedeutet hier, daß Markierungen und Achsentext links und die Graduationen rechts von der Achse gezeichnet werden sollen.

Anmerkung: Unabhängig von dem Wert für XL wird bei negativem ZSH keine gespiegelte zweite Achse (Rahmen) gezeichnet.

IGEN (-) Mit dieser Variablen wird die Anzahl der Nachkommastellen der zu zeichnenden Graduationen festgelegt.

Bei der Darstellung der Graduationen ohne Dezimal-exponenten bedeutet:

IGEN = 1 Graduationen mit einer Nachkommastelle.

= 0 Graduationen nur mit Dezimalpunkt.

= -1 Graduationen ohne Dezimalpunkt.

usw.

Für die Darstellung der Graduationen mit dem Dezimal-exponenten 'E' gilt:

$$(-11) \geq \text{IGEN} \geq (-14)$$

Es bedeutet z. B.:

IGEN = -11 Eine signifikante Nachkommastelle:
0.3E+03.

IGEN = -13 Drei signifikante Nachkommastellen:
0.123E-05.

IT (-) Die Achse kann mit einer Beschriftung versehen werden. Der Betrag von IT gibt die Anzahl der darzustellenden Textzeichen an.

Entsprechend dem Vorzeichen von ZSH wird der Text rechts oder links oben an die Achse gezeichnet. Ist dabei IT positiv, erscheint die Beschriftung im rechten Winkel, bei negativem IT parallel zur Y-Achse.

TEXT (-) Text für die Y-Achse. Die Zeichenkette darf maximal 80 Zeichen beinhalten. Besteht der Text aus bis zu vier Zeichen, kann er in der Argumentliste als Literal, in Hochkommas eingeschlossen, angegeben werden. Bei mehr als vier Zeichen Text muß eine Textvariable entsprechend dimensioniert und eventuell mit einer DATA-Anweisung definiert werden.

Das Unterprogramm AYLIN1 übergibt an den benannten 'COMMON /FAKTOR/' folgende für das Unterprogramm CURVE (Zeichnen eines Kurvenverlaufs, s. auch Punkt 13.) relevante Größen:

1. Kleinster Wert an der Y-Achse BYMIN = YMIN
2. Maßstabsfaktor YL/(YMAX-YMIN)
3. Information (Steuergröße), daß es sich bei dieser Achse um eine lineare Y-Achse handelt.

Von AYLIN1 aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN, NUM1, STRO

TRACEGS-Routinen _____: PLOTP

periphere Routinen _____: AMAX1, EXPD (CONVY /3/), NETLIN, YPOS

1. AXLOG1 - Zeichnen einer logarithmischen X-Achse

Das Unterprogramm AXLOG1 gestattet die Konstruktion einer X-Achse mit logarithmischer Teilung.

Für die Richtung der Achse auf dem Zeichnungsträger gilt das gleiche, wie es bei der Erläuterung des Unterprogrammes AXLIN1 (s. Punkt 9.) beschrieben wurde.

Beim Aufruf von AXLOG1 müssen 11 Argumente initialisiert sein.

Aufruf: CALL AXLOG1(XL,XMIN,XMAX,IX,YL,IKM,HKM,HGM,ZSH,IT,TEXT)

Die Argumente haben folgende Bedeutung:

XL (cm) Länge der X-Achse.

XMIN (phys. Einh.) Wert mit dem die X-Achse beginnen soll (linkes Ende). Entspricht der Wert nicht einer ganzen oder dem ganzzahligen Vielfachen $(2, 3, \dots, 9 \cdot 10^{**n})$ einer Dekade, wird keine Graduation gezeichnet.

XMAX (phys. Einh.) Wert am rechten Ende der X-Achse. Für das Zeichnen des Wertes als Graduation gilt sinngemäß das gleiche wie zuvor bei XMIN angegeben.

IX (-) Steuergröße. Zwischen den Dekaden können 8 Markierungen gezeichnet werden von $2 \cdot 10^{**n}$ bis $9 \cdot 10^{**n}$. Es gibt 4 Möglichkeiten, um an die Markierungen die entsprechenden Graduationen zu zeichnen:

IX = 0 Es werden keine Zwischengraduationen gezeichnet.

IX = 1 An die Markierungen 2, 4 und $7 \cdot 10^{**n}$ werden die entsprechenden Graduationen gezeichnet.

IX = 2 An die Markierungen 2, 3, 5 und $8 \cdot 10^{**n}$ werden die entsprechenden Graduationen gezeichnet.

IX = 3 Es werden alle Zwischengraduationen gezeichnet von $2 \cdot 10^{**n}$ bis $9 \cdot 10^{**n}$.

- YL (cm) Dieses Argument bietet die Möglichkeit, eine parallel zur X-Achse gespiegelte Achse ohne Graduationen, aber mit Markierungen im Abstand YL zeichnen zu lassen.
Sinnvollerweise sollte für YL die Länge der Y-Achse angegeben werden (Rahmen).
Wird für $YL = 0$. angegeben, entfällt das Zeichnen einer gespiegelten zweiten Achse.
YL muß auch dann ein entsprechender Wert zugewiesen werden, wenn das Zeichnen von Rasterlinien gewünscht wird. Ist hierbei YL positiv, wird das Raster mit durchgezogenen Linien dargestellt, bei negativem YL gestrichelt. Für diesen Fall muß den Variablen HKM und/oder HGM der Wert 0. zugewiesen werden.
- IKM (-) Steuergröße:
IKM ≤ 0 Zwischen den Dekaden werden weder Markierungen noch Graduationen gezeichnet.
IKM > 0 Zwischen den Dekaden werden 8 Markierungen ($2 \cdot 10^{*n}$ bis $9 \cdot 10^{*n}$) und in Abhängigkeit des Argumentes IX die entsprechenden Graduationen gezeichnet.
- HKM (cm) Der Wert von HKM gibt die Höhe der kleinen Markierungen an der Achse an. Sollen Rasterlinien gezeichnet werden (siehe auch Argument YL), muß HKM = 0. gesetzt werden.
- HGM (cm) Höhe der großen Markierungen an der Achse.
Bezüglich Rasterlinien gilt das gleiche wie zuvor bei HKM beschrieben.
- ZSH (cm) Der Betrag von ZSH gibt die Höhe der zu zeichnenden Ziffern (Graduationen) und Zeichen (Text) an.

Standardmäßig werden Markierungen auf und Graduationen mit Achsentext unter die Achse gezeichnet. Die Angabe eines negativen Wertes für ZSH bedeutet hier, daß die Markierungen unter und die Graduationen mit dem Achsentext über die Achse gezeichnet werden sollen.

Anmerkung: Unabhängig von dem Wert für YL wird bei negativem ZSH kein Rahmen (gespiegelte Achse) gezeichnet.

IT (-) Die Achse kann mit einer Beschriftung versehen werden. Diese wird rechtsbündig an die Achse und parallel zu dieser gezeichnet. IT gibt die Anzahl der zu zeichnenden Textzeichen an.

TEXT (-) Text für die X-Achse. Die Zeichenkette darf maximal 80 Zeichen beinhalten. Besteht der Text aus bis zu vier Zeichen, kann er in der Argumentliste als Literal, in Hochkommas eingeschlossen, angegeben werden. Bei mehr als vier Zeichen Text muß eine Textvariable entsprechend dimensioniert und eventuell mit einer DATA-Anweisung definiert werden.

Das Unterprogramm AXLOG1 übergibt an den benannten 'COMMON /FAKTOR/' folgende für das Unterprogramm CURVE (Zeichnen eines Kurvenverlaufs, s. auch Punkt 13.) relevante Größen:

1. Kleinster Wert an der X-Achse $BXMIN=XMIN$
2. Maßstabsfaktor $XL/1g(XMAX/XMIN)$
3. Information (Steuergröße), daß es sich bei dieser Achse um eine X-Achse mit logarithmischer Teilung handelt.

Von AXLOG1 aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN, NUM1, STRO

TRACEGS-Routinen _____: PLOTP

periphere Routinen _____: ALOG10, FRXPR#, NETLIN

12. AYLOG1 - Zeichnen einer logarithmischen Y-Achse

Das Unterprogramm AYLOG1 gestattet die Konstruktion einer Y-Achse mit logarithmischer Teilung.

Für die Richtung der Achse auf dem Zeichnungsträger gilt das gleiche wie es bei der Erläuterung des Unterprogrammes AYLIN1 (s. Punkt 10.) beschrieben wurde.

Beim Aufruf von AYLOG1 müssen 11 Argumente initialisiert sein.

Aufruf: CALL AYLOG1(YL,YMIN,YMAX,IY,XL,IKM,HKM,HGM,ZSH,IT,TEXT)

Die Argumente haben folgende Bedeutung:

- YL (cm) Länge der Y-Achse.
- YMIN (phys. Einh.) Wert mit dem die Y-Achse beginnen soll (unteres Ende). Entspricht der Wert nicht einer ganzen oder dem ganzzahligen Vielfachen $(2, 3, \dots, 9 \cdot 10^{**n})$ einer Dekade, wird keine Graduation gezeichnet.
- YMAX (phys. Einh.) Wert am oberen Ende der Y-Achse. Für das Zeichnen des Wertes als Graduation gilt sinngemäß das gleiche, wie zuvor bei YMIN angegeben.
- IY (-) Steuergröße. Zwischen den Dekaden können 8 Markierungen gezeichnet werden von $2 \cdot 10^{**n}$ bis $9 \cdot 10^{**n}$. Es gibt 4 Möglichkeiten, um an die Markierungen die entsprechenden Graduationen zu zeichnen:
- IY = 0 Es werden keine Zwischengraduationen gezeichnet.
- IY = 1 An die Markierungen 2, 4 und $7 \cdot 10^{**n}$ werden die entsprechenden Graduationen gezeichnet.
- IY = 2 An die Markierungen 2, 3, 5 und $8 \cdot 10^{**n}$ werden die entsprechenden Graduationen gezeichnet.
- IY = 3 Es werden alle Zwischengraduationen gezeichnet von $2 \cdot 10^{**n}$ bis $9 \cdot 10^{**n}$.

- XL (cm) Dieses Argument bietet die Möglichkeit, eine parallel zur Y-Achse gespiegelte Achse ohne Graduation, aber mit Markierungen im Abstand XL zeichnen zu lassen.
Sinnvollerweise sollte für XL die Länge der X-Achse angegeben werden (Rahmen).
Wird für $XL = 0$. angegeben, entfällt das Zeichnen einer gespiegelten zweiten Achse.
XL muß auch dann ein entsprechender Wert zugewiesen werden, wenn das Zeichnen von Rasterlinien gewünscht wird. Ist hierbei XL positiv, wird das Raster mit durchgezogenen Linien dargestellt, bei negativem XL gestrichelt. Für diesen Fall muß den Variablen HKM und/oder HGM der Wert 0. zugewiesen werden.
- HKM (cm) Der Wert von HKM gibt die Höhe der kleinen Markierungen an der Achse an. Sollen Rasterlinien gezeichnet werden (siehe auch Argument XL), muß $HKM = 0$. gesetzt werden.
- HGM (cm) Der Wert von HGM gibt die Höhe der großen Markierungen an der Achse an.
Bezüglich Rasterlinien gilt das gleiche wie zuvor bei HKM beschrieben.
- IKM (-) Steuergröße:
 $IKM \leq 0$ Zwischen den Dekaden werden weder Markierungen noch Graduationen gezeichnet.
 $IKM > 0$ Zwischen den Dekaden werden 8 Markierungen ($2 \cdot 10^{*n}$ bis $9 \cdot 10^{*n}$) und in Abhängigkeit des Argumentes IY die entsprechenden Graduationen gezeichnet.
- ZSH (cm) Der Betrag von ZSH gibt die Höhe der zu zeichnenden Ziffern (Graduationen) und Zeichen (Text) an. Standardmäßig werden Markierungen und Achsentext rechts, Graduationen links von der Achse gezeich-

net. Die Angabe eines negativen Wertes für ZSH bedeutet hier, daß Markierungen und Achsentext links, Graduationen rechts von der Achse gezeichnet werden sollen.

Anmerkung: Unabhängig von dem Wert für XL wird bei negativem ZSH kein Rahmen (gespiegelte Achse) gezeichnet.

IT (-) Die Achse kann mit einer Beschriftung versehen werden. Der Betrag von IT gibt die Anzahl der darzustellenden Textzeichen an.
Entsprechend dem Vorzeichen von ZSH wird der Text rechts oder links oben an die Achse gezeichnet. Ist dabei IT positiv, erscheint die Beschriftung im rechten Winkel, bei negativem IT parallel zur Y-Achse.

TEXT (-) Text für die Y-Achse. Die Zeichenkette darf maximal 80 Zeichen beinhalten. Besteht der Text aus bis zu vier Zeichen, kann er in der Argumentliste als Literal, in Hochkommas eingeschlossen, angegeben werden. Bei mehr als vier Zeichen Text muß eine Textvariable entsprechend dimensioniert und eventuell mit einer DATA-Anweisung definiert werden.

Das Unterprogramm AYLOG1 übergibt an den benannten 'COMMON /FAKTOR/' folgende für das Unterprogramm CURVE (Zeichnen eines Kurvenverlaufs, s. auch Punkt 13.) relevante Größen:

1. Kleinster Wert an der Y-Achse $BXMIN=XMIN$
2. Maßstabsfaktor $YL/1g(YMAX/YMIN)$
3. Information (Steuergröße), daß es sich bei dieser Achse um eine Y-Achse mit logarithmischer Teilung handelt.

Von AYLOG1 aufgerufene Unterprogramme:

GS-Routinen_____ : GSZA, LIN, NUM1, STRO

TRACEGS-Routinen_____ : PLOTP

periphere Routinen_____ : ALOG10, FRXPR#, NETLIN

13. POL - Zeichnen eines Polarkoordinatensystems.

Das Unterprogramm POL zeichnet ein Polarkoordinatensystem. In diesem können Funktionen (dargestellt durch Wertetabellen) abgebildet werden, deren Wertepaare durch radiale und azimutale Koordinaten bestimmt sind (siehe hierzu auch Punkt 15.).

Die Richtungen der beiden Hauptachsen ($\phi = 0^\circ$ und $\phi = 90^\circ$) auf dem Zeichnungsträger liegen fest und sind nicht veränderbar. Die Lage der Achse mit dem Winkel $\phi = 0^\circ$ auf den verschiedenen Zeichengeräten entspricht der Anordnung der X-Achse bei Verwendung des Unterprogrammes AXLINI (siehe Punkt 9.).

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

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

Die Argumente haben folgende Bedeutung.

- RL (cm) Maximaler Halbmesser des Polarkoordinatensystems.
- RMAX (phys. Einh.) Maximale radiale Koordinate. Sie muß nicht zwangsläufig als Graduation erscheinen, da dies von dem angegebenen Wert für DR 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, 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 angegeben.

IGEN (-)

Mit dieser Variablen wird die Anzahl der Nachkommastellen der zu zeichnenden Graduationen festgelegt. Bei der Darstellung der Graduationen ohne Dezimal-exponenten bedeutet:

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

usw.

Für die Darstellung der Graduationen mit dem Dezimal-exponenten 'E' gilt:

$$(-11) \geq \text{IGEN} \geq (-14)$$

Es bedeutet z. B.:

IGEN = -11 Eine signifikante Nachkommastelle:

0.3E+03.

IGEN = -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: $\phi = 0^\circ$ und $\phi = 90^\circ$
- = 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. Besteht der Text aus bis zu vier Zeichen, kann er in der Argumentliste als Literal, in Hochkommas eingeschlossen, angegeben werden. Bei mehr als vier Zeichen Text muß eine Textvariable entsprechend dimensioniert und eventuell mit einer DATA-Anweisung definiert werden.

Das Unterprogramm POL übergibt dem benannten 'COMMON /FAKTOR/' den Maßstabsfaktor (RL/RMAX), der für die Ausführung des Unterprogrammes CURVEP erforderlich ist.

Von POL aufgerufene Unterprogramme:

GS-Routinen _____: LIN, NUM1, STRO

TRACEGS-Routinen ____: CERCLE, NUMBP, PLOTP, SYMBP, SYMBP3, WHEREP

periphere Routinen __: COS, EXPD, SIN

14. CURVE - Zeichnen eines Kurvenverlaufes in ein kartesisches Koordinatensystem

Das Unterprogramm CURVE gestattet das Zeichnen einer Wertetabelle. Es können maximal 2000 Koordinatenpaare je Aufruf von CURVE gezeichnet werden. Die Werte bleiben dabei unverändert. 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 CURVE aufgerufen wird, müssen eine X- und eine Y-Achse durch Aufruf der Unterprogramme AXLIN1 oder AXLOG1 und AYLIN1 oder AYLOG1 definiert sein.

Das Unterprogramm CURVE hat 6 Argumente, die beim Aufruf initialisiert sein müssen.

Aufruf: CALL CURVE(XARRAY,YARRAY,IP,ISYMB,HZS,INCS)

Die Argumente haben folgende Bedeutung:

XARRAY(I), (phys. Einh.) Feldvariable. Sie enthält die X-Koordinaten I=1, IP der zu zeichnenden Wertetabelle.

XARRAY(max) = XARRAY(2000)

YARRAY(I), (phys. Einh.) Feldvariable. Sie enthält die Y-Koordinaten I=1, IP der zu zeichnenden Wertetabelle.

YARRAY(max) = YARRAY(2000)

IP (-) Anzahl der zu zeichnenden Koordinatenpaare.
IPmax = 2000.

ISYMB	(-)	Wenn die Kurve punktweise (mit Symbolen) oder als Linienzug mit Symbolen dargestellt werden soll, wird der Variablen ISYMB die Ordnungsnummer des gewünschten Symbolen zugewiesen. Folgende Symbole stehen zur Auswahl: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 □ ○ △ + × ◇ † ‡ ⊥ ∟ * ∞ Soll die zu zeichnende Kurve nur als Linienzug wiedergegeben werden (INCS = 0), ist eine Wertzuweisung für ISYMB bedeutungslos.
HZS	(cm)	Höhe der zu zeichnenden Symbole. <u>Hinweis:</u> Bei der Wiedergabe auf einer Zeichnung beträgt die wahre Höhe der Symbole ca. $0.857 * HZS$.
INCS	(-)	Steuervariable. Über die Variable INCS wird bestimmt, ob die Kurve nur durch Symbole (INCS <= -1), nur als Linienzug (INCS = 0) oder als Linienzug mit Symbolen (INCS >= 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 CURVE weitere für die Ausführung notwendige Informationen:

Das Unterprogramm zum Zeichnen einer X-Achse (AXLIN1/AXLOG1) liefert den kleinsten Wert an der X-Achse (XMIN), den Maßstabsfaktor für die X-Koordinaten (XL/(XMAX-XMIN) oder XL/lg(XMAX-XMIN)) und die Information (Steuergröße), ob es sich um eine lineare oder logarithmische Achse handelt.

Das Achsenprogramm für die Y-Achse liefert die korrespondierenden Werte.

Von CURVE aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN, POIO

TRACEGS-Routinen _____: keine

periphere Routinen _____: ALOG10

15. CURVEP - Zeichnen eines Kurvenverlaufes in ein Polarkoordinaten- system

Das Unterprogramm CURVEP ermöglicht die graphische Darstellung von Funktionen (Radius als Funktion des Umfangswinkels) in einem Polarkoordinatensystem. Es können maximal 2000 Koordinatenpaare je Aufruf von CURVEP gezeichnet werden.

Bezüglich der Darstellung des Funktionsverlaufes als Linienzug, punktweise usw. gilt das gleiche, wie es bei der Erläuterung des Unterprogrammes CURVE beschrieben wurde (siehe hierzu auch Punkt 14.).

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

Das Unterprogramm CURVEP hat 6 Argumente, die beim Aufruf initialisiert sein müssen.

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

Die Argumente haben folgende Bedeutung:

PHI(I), I=1, IP	(Grad)	Feldvariable. Sie enthält die radialen Koordinaten der darzustellenden Wertetabelle. PHI(max) = PHI(2000)
RAD(I), I=1, IP	(phys. Einh.)	Feldvariable. Sie enthält die azimutalen Koordinaten der darzustellenden Wertetabelle. RAD(max) = RAD(2000)
IP	(-)	Anzahl der zu zeichnenden Koordinatenpaare. IPmax = 2000.
ISYMB	(-)	Wenn die Kurve punktweise (mit Symbolen) oder als Linienzug mit Symbolen dargestellt werden soll, wird der Variablen ISYMB die Ordnungsnummer des gewünschten Symbolen zugewiesen. Folgende Symbole stehen zur Auswahl: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 □ ⊙ △ + × ◇ ♣ × Z Y ✕ * ×

Soll die zu zeichnende Kurve nur als Linienzug wiedergegeben werden (INCS = 0), ist eine Wertzuweisung für ISYMB bedeutungslos.

HZS	(cm)	Höhe der zu zeichnenden Symbole. <u>Hinweis:</u> Bei der Wiedergabe auf einer Zeichnung beträgt die wahre Höhe der Symbole ca. $0.857 * HZS$.
INCS	(-)	Steuervariable. Über die Variable INCS wird bestimmt, ob die Kurve nur durch Symbole (INCS ≤ -1), nur als Linienzug (INCS = 0) oder als Linienzug mit Symbolen (INCS ≥ 1) dargestellt werden soll. Ist der Betrag von INCS > 1 , bedeutet dies, daß nur jedes INCS-Wertepaar durch ein Symbol angedeutet wird.

Über den benannten 'COMMON /FAKTOR/' erhält CURVEP den für die Ausführung erforderlichen Maßstabsfaktor RL/RMAX (siehe hierzu auch Punkt 13., Unterprogramm POL).

Von CURVEP aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN, POIO

TRACEGS-Routinen _____: keine

periphere Routinen _____: COS, SIN

16. CURVLV - Zeichnen eines durchgezogenen oder gestrichelten Streckenzuges

CURVLV ist ein allgemeines Unterprogramm zum Zeichnen von Streckenzügen mit der Möglichkeit, diese durchgezogen oder gestrichelt darzustellen.

Die zu zeichnenden Wertepaare werden in Feldern der Routine übergeben. Damit wird der GS-Workspace (Region) wesentlich optimaler ausgenutzt als beim Zeichnen von einzelnen Linien mit dem Unterprogramm PLOTP. Für das Zeichnen eines Linienzuges, der aus mehreren Wertepaaren besteht,

ist somit grundsätzlich die Verwendung des Unterprogrammes CURVLV zu empfehlen.

Die zu zeichnenden Wertepaare werden linear miteinander verbunden.

Aufruf: CALL CURVLV(X,Y,IP,IST)

Die Argumente haben folgende Bedeutung:

X(I), (cm) Feldvariable. Sie enthält die X-Koordinaten des zu zeichnenden Streckenzuges.
I=1, IP
X(max) = X(1000)

Y(I), (cm) Feldvariable. Sie enthält die Y-Koordinaten des zu zeichnenden Streckenzuges.
I=1, IP
Y(max) = Y(1000)

IP (-) Anzahl der darzustellenden Wertepaare. IPmax = 1000.

IST (-) Steuervariable.
IST = 0 Alle Wertepaare werden mit einer durchgezogenen Linie verbunden.
IST > 0 Alle Wertepaare werden mit einer gestrichelten Linie verbunden und zwar in der Weise, daß die erste Hälfte des Weges zwischen zwei Bildpunkten (in Richtung aufsteigender Wertepaarindizes) gezeichnet wird, die zweite nicht.

Über den benannten 'COMMON /FAKTOR/' erhält CURVLV die Information, mit welcher Farbe (PEN) der Streckenzug dargestellt werden soll.

Von CURVLV aufgerufene Unterprogramme:

GS-Routinen_____ : GSZA, LIN, VEC

TRACEGS-Routinen_____ : keine

periphere Routinen_____ : keine

17. CADRE - Zeichnen eines Rahmens

Das Unterprogramm CADRE zeichnet einen Rahmen mit der Ausdehnung X/Y, wobei von dem zuletzt definierten Bezugspunkt ausgegangen wird.

An den Bildschirmgeräten (GA/T4015/KSG) kann CADRE auch zur Festlegung eines physischen Rahmens verwendet werden.

Aufruf: CALL CADRE(RX,RY)

Die Argumente haben folgende Bedeutung:

RX (cm) Ausdehnung des Rahmens in X-Richtung.

RY (cm) Ausdehnung des Rahmens in Y-Richtung.

Von CADRE aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN

TRACEGS-Routinen _____: keine

periphere Routinen _____: keine

18. CADREF - Festlegen eines fiktiven Rahmens

Bei Aufruf des Unterprogrammes CADREF wird mit den vorgegebenen Werten in der Argumentliste lediglich ein physischer Rahmen festgelegt. Die Zeichnung des Rahmens selbst entfällt. Die Verwendung von CADREF ist insbesondere dann zu empfehlen, wenn an den Bildschirmgeräten konforme Abbildungen erwartet werden.

Aufruf: CALL CADREF(RX,RY)

Die Bedeutungen der Argumente RX und RY sind sinngemäß dieselben wie dies bei der Beschreibung von CADRE (s. Punkt 17.) angegeben wurde.

Von CADREF aufgerufene Unterprogramme:

GS-Routinen _____: STR

TRACEGS-Routinen _____: keine

periphere Routinen _____: keine

19. KFK - Zeichnen des KfK-Symbols

Die Verwendung des Unterprogrammes KFK ist dann zu empfehlen, wenn dokumentationsfähige Zeichnungen erstellt werden, die, entsprechend den KfK-Konventionen, auch das KfK-Symbol mit den ausgezogenen Linien beinhalten sollen.

Bei Aufruf von KFK wird, ausgehend von der linken unteren Ecke (Bezugspunkt!) der Hoch- bzw. Querformate DIN A4, DIN A3 und DIN A2, das KfK-Symbol einschließlich der ausgezogenen Linien formatgerecht gezeichnet. Falls gewünscht kann das KfK-Symbol mit dem Institutsnamen (als Zusatz) versehen werden.

Bei der Organisation der Zeichnung ist darauf zu achten, daß dieses Feld nicht überzeichnet wird.

Es ist zu empfehlen, das Unterprogramm KFK zusammen mit CADRE (s. Punkt 17.) zu verwenden. Zu beachten ist hierbei, daß zwischen beiden Aufrufen keine Änderung des Bezugspunktes vorgenommen wird.

Beim Aufruf des Unterprogrammes KFK müssen zwei Argumente initialisiert sein.

Aufruf: CALL KFK(IFORM,INST)

Das Argument IFORM besteht aus einer zweistelligen Integerzahl mit Doppelbedeutung. Die erste Ziffer gibt das DIN-Papierformat an, die zweite entscheidet über Hoch- oder Querformat. Gültige Werte sind der nachfolgenden Tabelle zu entnehmen.

Das Argument INST bietet die Möglichkeit, an das KfK-Symbol den Namen des Institutes zu zeichnen. Dieser wird in Form einer Zeichenkette der Länge 4 Byte's (= 4 Zeichen) als Literal oder im Hollerithformat linksbündig initialisiert.

Beispiel 1: Es soll das KfK-Symbol entsprechend dem DIN A3-Querformat gezeichnet werden. Der Name des Institutes, hier IRB, soll ebenfalls erscheinen.

```
CALL KFK(30,'IRB ')
```

Beispiel 2: Zeichnen eines DIN A4-Rahmens im Hochformat mit dem KfK-Symbol. Die Angabe des Institutes soll unterdrückt werden.

```
CALL CADRE(21.,29.7)
CALL KFK(41,'')
```

Im folgenden ist eine Tabelle angegeben, aus der die erforderlichen Parameter für die Unterprogramme CADRE und KFK entnommen werden können.

ARGUMENT	CALL KFK(IFORM,'INST')	CALL CADRE(X,Y)	Abst. der ausg.
FORMAT	IFORM	X/Y (cm)	KfK-Linie vom unt. Rand (cm)
DIN A4 Hochformat	41	21./29.7	4.25
DIN A4 Querformat	40	29.7/21.	3.00
DIN A3 Hochformat	31	29.7/42.	5.90
DIN A3 Querformat	30	42./29.7	4.25
DIN A2 Hochformat	21	42./59.4	8.40
DIN A2 Querformat	20	59.4/42.	6.00

Von KFK aufgerufene Unterprogramme:

GS-Routinen _____: keine

TRACEGS-Routinen _____: CURVLV, PLOTP, SYMBP

periphere Routinen _____: keine

20. TEXT/TEXTS - Zeichnerische Darstellung von Texten in Normal- oder Schrägschrift

Die Unterprogramme TEXT und TEXTS gestatten das Zeichnen von Texten, die mit Groß- und Kleinbuchstaben geschrieben vorliegen. TEXT ist die Normalschrift-, TEXTS die Schrägschrift-Version. Die Texte sind kein integraler Bestandteil eines Programmes. Sie befinden sich in separaten, unnummerierten Dateien (vorzugsweise TSO-Text-Dateien) mit einer festen Satzlänge von 80 Zeichen. In diesen Dateien kann der Text so strukturiert werden, wie er später auf dem Zeichnungsträger erscheinen soll. Damit entfallen umständliche Berechnungen der Anfangskoordinaten.

Die Dateien enthalten neben der eigentlichen Text-Information Steuerdaten, die angeben, in welcher Weise der Text gezeichnet werden soll. Diese Steuerdaten werden unformatiert in einer Zeile dem Text vorangestellt. Man unterscheidet fünf verschiedene Größen:

- I. Schrifthöhe _____ (cm) REAL*4
- II. Zeichen-Zwischenraum (Spalt) _____ (cm) REAL*4
- III. Zeilen-Zwischenraum (Spalt) _____ (cm) REAL*4
- IV. Strichstärke (0=dünn, 1=dick) _____ (-) INTEGER
- V. Farbe (1=schwarz, 2=rot, 3=blau, 4=grün) _____ (-) INTEGER

Der Text kann alle Zeichen des TSO-Zeichensatzes enthalten. Ausnahmen bilden die Zeichen '\$', '@' und '#'. Werden sie in dem zu zeichnenden Text angegeben, hat dies folgende Bedeutungen:

a) Die Eingabe eines \$-Zeichens einschließlich folgendem Leerzeichen bedeutet, daß die Zeile an dieser Stelle abgebrochen wird. Da ohne diese Maßnahme grundsätzlich immer 80 Spalten verarbeitet werden, kann man damit Rechenzeit einsparen.

b) Ein \$-Zeichen gefolgt von einer Zahl zwischen 0 und 9 oder den Großbuchstaben W - Z hat die Bedeutung eines Symbolerkennungszeichens. Folgende Symbole stehen zur Verfügung:

\$0 \$1 \$2 \$3 \$4 \$5 \$6 \$7 \$8 \$9 \$W \$X \$Y \$Z

☐ ○ △ + × ◇ ↑ ↘ Z Y ✕ ✕ |

- c) Das \$-Zeichen gefolgt von A, a, O, o, U oder u hat die Bedeutung eines Umlauterkennungszeichens. \$A wird zu Ä, \$a wird zu ä usw.. Für die Darstellung des 'ß' hat die Kombination \$B Gültigkeit.
- d) Je ein \$-Zeichen in der 1. und 2. Spalte einer Zeile bedeuten, daß in der nächsten Zeile neue Zeichnungsparameter (Steuerdaten) eingelesen werden sollen. Damit lassen sich innerhalb eines Textes beispielsweise die Zeichenhöhe und/oder die Strichstärke usw. ändern. Nach der Zeile mit der Steuereingabe wird mit dem Text fortgefahren. Der Parameterwechsel, hervorgerufen durch die beschriebene \$\$-Eingabe, kann beliebig oft wiederholt werden.
- e) Die Zeichen '¢' und '#' sind für die Darstellung von griechischen Klein- und Großbuchstaben reserviert. Treten sie in einem Text auf, gefolgt von alphanumerischen Zeichen, hat nachstehende Korrespondenz Gültigkeit:
- | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G | I | K | L | M | N | O | P | R | S | T | X | Y | Z | 1 | 2 | 3 | 4 |
| ¢_: | α | β | χ | δ | ε | φ | γ | 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 | |
- f) Zwei aufeinanderfolgende \$-Zeichen innerhalb der Spalten 2 - 80 bewirken das Zeichnen des \$-Zeichens selbst. Das gleiche gilt für die Darstellung der Zeichen '¢' und '#' innerhalb der Spalten 1 - 80.

Beim Aufruf des Unterprogrammes Text müssen drei Argumente initialisiert sein.

Aufruf: CALL TEXT(XKO,YKO,IUNIT)

Die Argumente haben folgende Bedeutung:

XKO (cm) Koordinaten für den Textbeginn ausgehend vom zuletzt festgelegten Bezugspunkt. Es ist die linke obere Ecke des Textfeldes.

IUNIT (-) Nummer der Eingabeeinheit für die zeichnerisch darzustellende Textdatei. Die Verbindung der Eingabeeinheit

mit der Textdatei geschieht im Stapelbetrieb über eine Datei-Definitions-Anweisung (DD-Rekord), im TSO-Foreground mit Hilfe des ALLOCATE-Kommandos.

Nachdem der Text gezeichnet ist, wird der Lese-Pointer wieder an den Anfang der Datei gesetzt (REWIND IUNIT). Damit ist es möglich, die gleiche Datei durch erneuten Aufruf des Unterprogrammes TEXT an anderer Stelle wiederholt zeichnen zu lassen.

Im Anhang ist in ABB. 3 der Inhalt einer Textdatei wiedergegeben und in Abb. 4 das auf einem Zeichengerät dargestellte Ebenbild.

Die Aufruffolge der Unterprogramme für den gezeichneten Text bestand in:

```
.  
.  
CALL PLOTI(2,500)  
CALL CADRE(29.7,21.)  
CALL KFK(40,'IRB '  
CALL TEXT(4.,19.,1)  
CALL PLOTP(0.,0.,999)  
.  
.
```

Von TEXT aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, STR1(TEXT), STR11(TEXTS)

TRACEGS-Routinen _____: PEN, PLOTP, SYMBP (TEXT), SYMBS (TEXTS),
SYMBP2 (TEXT), SYMBS2 (TEXTS),
SYMBP3 (TEXT), SYMBS3 (TEXTS)

periphere Hilfsprogramme_: IBCOM#, LDFIO#, UMLAUT(TEXT), UMLTS(TEXTS)

21. Symbole für den Bereich: Allgemeines Zeichnen

Es werden hier einige Routinen vorgestellt, die dem Programmierer bei der Erstellung von Datenflußplänen und ähnlichem als Hilfsmittel dienen sollen.

Bei den im folgenden näher beschriebenen Gerätesymbolen, wie peripherer Speicher (DISY), Magnetband (TASY) und Bildschirm (ECRAN), handelt es

sich um abstrahierte, nicht genormte Darstellungen.

Mit Hilfe des Unterprogrammes FLECHE lassen sich Pfeile auf normgerechte Weise darstellen.

STRIP ist ein Unterprogramm zum Zeichnen verschiedenartiger Linien.

Die Routine CERCLE gestattet die Darstellung von Kreisen und Kreisteilen.

Die Routine BALKEN ermöglicht das Zeichnen von orthogonalen Flächen mit und ohne Schraffur.

Anwendungsbeispiele sind in Abb. 1 und Abb. 2 im Anhang wiedergegeben.

21.1 FLECHE - Zeichnen von Linien und Pfeilen

Das Unterprogramm FLECHE verbindet zwei durch ihre Koordinaten vorgegebene Punkte mit einer Linie, deren Enden wahlweise mit Pfeilspitzen versehen werden können.

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

Die Argumente haben folgende Bedeutung:

XS (cm) Koordinaten des Startpunktes bezogen auf den zuletzt
YS (cm) definierten Bezugspunkt.

XZ (cm) Koordinaten des Zielpunktes bezogen auf den zuletzt
YZ (cm) definierten Bezugspunkt.

IPF (-) Steuergröße. Folgende Möglichkeiten sind vorgesehen:
IPF = 0 Start- und Zielpunkt werden mit einer geraden Linie verbunden.
IPF = 1 Wie IPF=0. Auf den Startpunkt weist zusätzlich eine Pfeilspitze.
IPF = 2 Wie IPF=0. Auf den Zielpunkt weist zusätzlich eine Pfeilspitze.
IPF = 3 Wie IPF=0. Auf Start- und Zielpunkt weisen zusätzlich Pfeilspitzen.

Von FLECHE aufgerufene Unterprogramme:

GS-Routinen _____ : GSZA, LIN
TRACEGS-Routinen _____ : PLOTP
periphere Routinen _____ : ATAN, COS, SIN

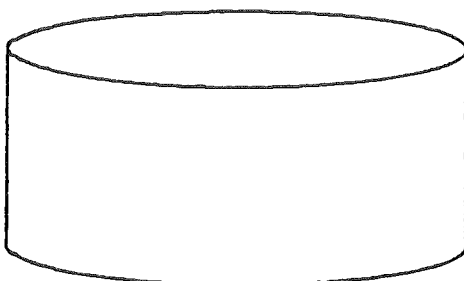
21.2 DISY - Zeichnen eines Speicher-Symbols (Zylinder)

Bei der zeichnerischen Darstellung von Datenflüssen wird zuweilen ein Symbol für einen peripheren Datenspeicher benötigt. In der Literatur werden solche Speicher häufig in Zylinderform abgebildet. Das Unterprogramm DISY zeichnet einen derartigen Zylinder mit der Möglichkeit, dessen Dimensionen nach Belieben gestalten zu können.

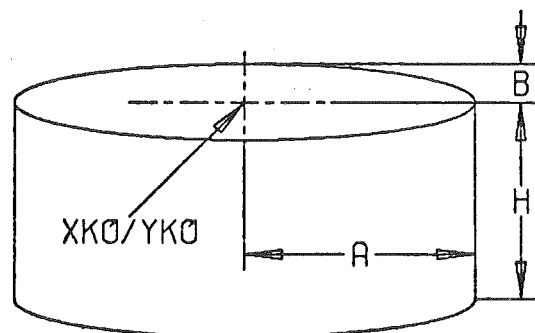
Aufruf: CALL DISY(XK0,YK0,A,B,H)

Die Argumente haben folgende Bedeutung:

- XK0 (cm) Koordinaten für die Positionierung des Speichersymbols (s. Skizze unten). Maßgebend ist der zuletzt definierte Bezugspunkt.
- YK0 (cm) Skizze unten).
- A (cm) Große (A) und kleine (B) Halbachse der elliptischen Zylinderoberfläche.
- B (cm) Zylinderoberfläche.
- H (cm) Höhe des Zylinders.



ORIGINAL



Von DISY aufgerufene Unterprogramme:

GS-Routinen _____ : GSZA, LIN
TRACEGS-Routinen _____ : PLOTP
periphere Routinen _____ : COS, SIN

21.3 TASY Zeichnen eines Magnetband-Symbols

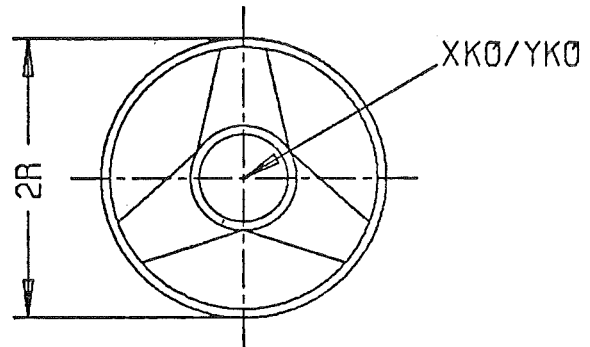
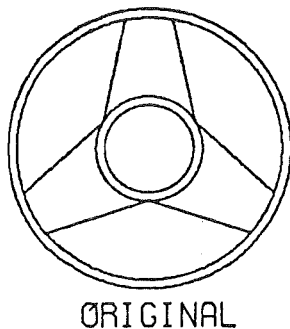
Das Unterprogramm TASY zeichnet ein Symbol, welches der äußeren Form eines Magnetbandes angenähert entspricht.

Aufruf: CALL TASY(XKO,YKO,R)

Die Argumente haben folgende Bedeutung:

XKO (cm) Zentrumskoordinaten des Magnetband-Symbols (s. Skizze
YKO (cm) unten) bezogen auf den zuletzt definierten Bezugspunkt.

R (cm) Außenradius des Magnetband-Symbols.



Von TASY aufgerufene Unterprogramme:

GS-Routinen _____ : GSZA, LIN
TRACEGS-Routinen _____ : CERCLE, PLOTP
periphere Routinen _____ : COS, SIN

21.4 ECRAN - Zeichnen eines Symbols für ein Bildschirmgerät

Das Unterprogramm ECRAN zeichnet ein stilisiertes allgemeines Symbol für einen Bildschirm.

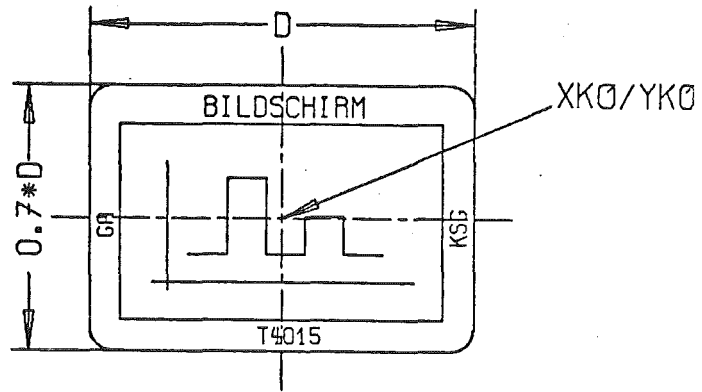
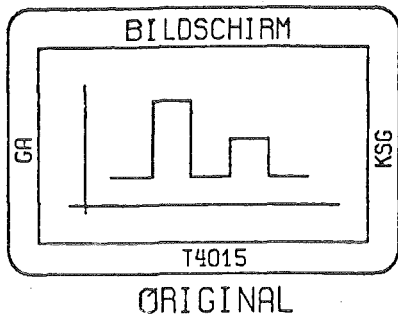
Die Größe des Symbols wird durch seine Breite festgelegt. Das Verhältnis Breite/Höhe beträgt 1.0/0.7.

Aufruf: CALL ECRAN(XKO,YKO,D)

Die Argumente haben folgende Bedeutung:

XK0 (cm) Zentrumskoordinaten des Bildschirm-Symbols bezogen auf den
YK0 (cm) zuletzt definierten Bezugspunkt (s. Skizze unten).

D (cm) Breite des Bildschirm-Symbols.



Von ECRAN aufgerufene Unterprogramme:

GS-Routinen _____ : GSZA, LIN
TRACEGS-Routinen _____ : CADRE, PLOTP, SYMBP
periphere Routinen _____ : COS, SIN

21.5 STRIP - Zeichnen von verschiedenartigen Linien

Das Unterprogramm STRIP gestattet das Zeichnen von gestrichelten, strichpunktieren und durchgezogenen Linien ausgehend vom zuletzt definierten Bezugspunkt.

Aufruf: CALL STRIP(XS,YS,XZ,YZ,SL,SP,ISTEU)

Die Argumente haben folgende Bedeutung:

XS (cm) Startkoordinaten der zu zeichnenden Linie.
YS (cm)

XZ (cm) Zielkoordinaten.
YZ (cm)

SL (cm) Strichlänge. Sie ist nur relevant bei gestrichelter oder strichpunktierter Darstellung einer Linie. Siehe auch Argument ISTEU.

SP (cm) Leerraum zwischen zwei Strichen bei gestrichelter oder strichpunktierter Darstellung. Siehe auch Argument ISTEU.

ISTEU (-) Steuergröße. Es gibt folgende Möglichkeiten:
ISTEU = 0 Liniendarstellung in gestrichelter Form
= 1 Liniendarstellung in strichpunktierter Form
= 2 Darstellung als durchgezogene Linie
Anmerkung: Bei ISTEU = 2 stellen SL und SP Dummy-Variable dar.

Hinweis: Da im Regelfall bei gestrichelter bzw. strichpunktierter Liniendarstellung die Summe aller Teilstrecken SL und SP mit der Strecke zwischen Start- und Zielpunkt nicht übereinstimmen wird, paßt STRIP die Strichlänge SL den gegebenen Verhältnissen automatisch an. Die Abweichung des angepaßten Wertes für SL vom vorgegebenen wird im allgemeinen vernachlässigbar klein sein.

Von STRIP aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN, VEC
TRACEGS-Routinen _____: keine
periphere Routinen _____: ATAN, COS, SIN, SQRT

21.6 CERCLE - Zeichnen von Kreisen und Kreisteilen

Mit Hilfe des Unterprogrammes CERCLE lassen sich Kreise und Kreisteile unter Vorgabe von Position, Winkelbereich und Größe zeichnen.

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

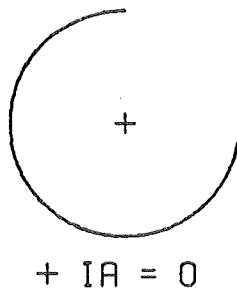
Die Argumente haben folgende Bedeutung:

XP (cm) Koordinaten, die die Kreisposition, bezogen auf den zuletzt definierten Bezugspunkt, angeben.
YP (cm) Siehe auch Argument IA.

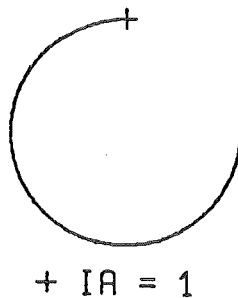
- WA (Grad) Anfangs- (WA) und Endwinkel (WE) des Kreises oder Kreis-
WE (Grad) teiles in mathematischem Sinne.
- R (cm) Radius des darzustellenden Kreises oder Kreisteiles.
- IA (-) Steuervariable. Mit ihr wird entschieden, ob die durch
die Koordinaten XP und YP angegebene Position das Zen-
trum des zu zeichnenden Kreises (IA = 0) oder dessen
Anfangspunkt bei WA (IA = 1) betrifft (s. Skizze unten).

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

Beispiel: CALL CERCLE(0.,0.,90.,360.,1.5,0)



Beispiel: CALL CERCLE(0.,0.,90.,360.,1.5,1)



Von CERCLE aufgerufene Unterprogramme:

- GS-Routinen _____ : keine
TRACEGS-Routinen _____ : CURVLV
periphere Routinen _____ : COS, SIN

21.7 BALKEN - Zeichnen von schraffierten und unschraffierten Balken.

Das Unterprogramm BALKEN zeichnet einen Rahmen (orthogonale Fläche) mit der Ausdehnung X/Y. Die linke untere Ecke des Rahmens liegt dabei an dem zuletzt definierten Bezugspunkt.

Es besteht wahlweise die Möglichkeit, die durch den Rahmen gebildete Fläche unter beliebigem Winkel und Feinheitsgrad zu schraffieren.

Die Verwendung von BALKEN ist insbesondere für die graphische Darstellung von Balkendiagrammen geeignet.

Das Unterprogramm BALKEN hat 4 Argumente, die beim Aufruf initialisiert sein müssen.

Aufruf: CALL BALKEN(RX,RY,W,DL)

Die Argumente haben folgende Bedeutung:

RX (cm) Ausdehnung des Balkens in X-Richtung.

RY (cm) Ausdehnung des Balkens in Y-Richtung.

W (Grad) Schraffurwinkel.

DL (cm) Abstand zwischen zwei Schraffurgeraden.

Im Fall $DL \leq 0$. wird der Balken ohne Schraffur gezeichnet.

Von CURVEP aufgerufene Unterprogramme:

GS-Routinen _____: GSZA, LIN

TRACEGS-Routinen _____: keine

periphere Routinen _____: SHADE (Calcomp)

22_ Routinen für spezielle Anwendungen

Es werden im folgenden zwei Unterprogramme beschrieben, deren Anwendung in manchen Fällen von Nutzen sein kann. Zum einen handelt es sich um die Routine EX zur Erzeugung verkleinerter, vergrößerter oder verzerrter Bildausgaben, zum anderen um die Routine ROT90, die an Bildschirmgeräten (nur TSO-Foreground!) die Drehung eines Bildes um 90 Grad bewirkt.

22 - 1 EX - Vergrößerung, Verkleinerung, Verzerrung bei der Bildausgabe

Zuweilen kann, entgegen der ursprünglichen Planung, aus optischen Gründen eine vergrößerte, verkleinerte oder verzerrte Bildausgabe gewünscht werden. Die Möglichkeit hierzu bietet die Routine EX. Mit ihr lassen sich alle Bilder in ihrer Ausdehnung (Breite und Höhe) auf einfache Weise beliebig verändern.

Wichtig: Der Aufruf von EX ersetzt den Aufruf des Unterprogrammes PLOTP(...,...,999). Eine Bildspeicherung ist bei dieser Ausgabeform nicht möglich. Diesbezügliche Angaben beim Aufruf der Initialisierungs-Routine PLOTI werden ignoriert.

Aufruf: CALL EX(XM,YM)

Die Argumente haben folgende Bedeutung:

XM (Einheit) Größe des Bildausgabefensters in X- und Y-Richtung
YM (Einheit) (Breite/Höhe).

Das darzustellende Bild dehnt sich bis an die äußersten Grenzen des definierten Fensters aus.

Ist das Verhältnis XM/YM identisch mit dem Seitenverhältnis des ursprünglich programmierten Bildes, erhält man eine vergrößerte bzw. verkleinerte konforme Abbildung. Ist dies nicht der Fall, wird das auszugebende Bild verzerrt dargestellt.

Die physikalische Einheit für XM und YM hängt von dem Gerät für die Zeichnungsausgabe ab. Für die On-/Off-Line-Zeichengeräte hat die Einheit 'cm' Gültigkeit, für die

Bildschirmgeräte sind es 'Bildrasterpunkte'.

Gültige Werte für die maximalen Gerätefenster sind dem Handbuch für das Graphische System zu entnehmen /1/.

Da jedes Zeichengerät voneinander verschiedene Bildausgabefenster besitzt, muß der Benutzer darauf achten, daß sein Bild in einem gültigen Bereich liegt.

Anmerkung: Bei der Bildschirmausgabe ist zu bemerken, daß konform zur gewünschten Ausgabe und in Relation zum gesamten Bildschirmfenster abgebildet wird.

Von EX aufgerufene Unterprogramme:

GS-Routinen _____: ENDGS, RG, ROUT

TRACEGS-Routinen _____: keine

periphere Routinen _____: keine

22.2 ROT90 Drehen eines Bildes um 90 Grad bei der Bildschirmausgabe

ROT90 ist ein Unterprogramm, das nur im TSO-Foreground bei der Darstellung von Zeichnungen im DIN-Hochformat auf den Bildschirmgeräten sinnvoll angewendet werden kann. Es bietet die Möglichkeit ein Bild vor der Ausgabe um 90 Grad in mathematischem Sinn zu drehen.

Bei der normalen Hochformat-Ausgabe wird die Bildgröße durch die maximale Schirmhöhe, die kleiner ist als die Schirmbreite, bestimmt. Aufgrund dieser Gegebenheit werden DIN-Hochformat-Bilder immer kleiner auf dem Bildschirm abgebildet als DIN-Querformat-Bilder.

Der Aufruf von ROT90 kann zwischen dem Startaufruf (PLOTI(...,...)) und dem Abschlufaufruf (PLOTI(...,...,999)) der TRACEGS-Zeichensoftware an beliebiger Stelle im Programm stehen.

Ein Bild kann lediglich nur einmal um 90 Grad gedreht werden. Ein mehrmaliger Aufruf von ROT90 ist wirkungslos.

Bei der Bilderstellung im Stapelbetrieb wird der Aufruf von ROT90 nur im Fall der ausschließlichen Bildspeicherung (CALL PLOTI(0,...)) akzeptiert.

Aufruf: CALL ROT90

Das Unterprogramm hat keine Argumentliste.

Von ROT90 aufgerufene Unterprogramme:

GS-Routinen_____ : ROT
TRACEGS-Routinen_____ : keine
periphere Routinen_____ : keine

23. TRACEGS im TSO-Foreground

Die Verwendung von TRACEGS im TSO-Foreground ist z. Zt. mit den Bildschirmgeräten 'TEKTRONIX 4015', 'Karlsruher Sichtgerät' und 'Graphischer Anschluß' möglich.

Der Benutzer muß beim Einschalten in die TSO-Sitzung mit 'LOGON' die Foreground-Region mit dem z. Zt. maximal möglichen Wert 'SIZE(1024)' überschreiben. Der Defaultwert ist 'SIZE(320)'.

Für den Foreground-Anwendungsfall wurde eine TSO-Kommandoprozedur (KP) erstellt, die den Benutzer weitgehend von spezifischen Kenntnissen, die die Bilderzeugung betreffen, entlastet. Der Benutzungsaufwand wird damit auf ein Minimum reduziert.

Im folgenden ist die Anweisungsfolge der KP zur Bilderstellung auf den verfügbaren Bildschirmgeräten wiedergegeben:

```
PROC 0
WRITE
WRITE ***** KOMMANDOPROZEDUR AGS *****
WRITE ***** T R A C E G S *****
WRITE
CONCAT ('IRB178.AGS.LOAD' 'SYS1.FORTLIB' 'SYS2.FORTLIB' 'SYS2.CALCOMP'-
        'SYS2.GSMAN')
SET &GA = &STR(GA)
SET &T611 = &STR(KSG)
```

```
SET &T415 = &STR(T4015)
SET &MS   = &STR(M)
WRITE EINGABE:
WRITE IST DAS ZEICHENPROGRAMM EIN SEQUENTIELLER
WRITENR DATA-SET (S) ODER MEMBER EINES PDS (M)?_:
READ &SM
IF &SM = &MS THEN GOTO M0
WRITENR PROGRAMM-NAME (OHNE QUALIFIER FORT)_____:
READ &PROG
GOTO M1
M0: WRITENR NAME DES PDS (OHNE QUALIFIER FORT)_____:
READ &PDSNAM
WRITENR MEMBER-NAME_____:
READ &MEMB
M1: WRITENR AUSGABEGERAET (GA/T4015/KSG)_____:
READ &DEVICE
IF &DEVICE. - = &T611 THEN GOTO M2
WRITENR ADRESSE (KARLSRUHER SICHTGERAET/KSG)____:
READ &ADDR
WRITE
FREE F(DISPL)
ALLOC UNIT(&ADDR.) F(DISPL) NEW
GOTO M4
M2: IF &DEVICE - = &T415 THEN GOTO M3
WRITE
GSA
GOTO M4
M3: IF &DEVICE - = &GA THEN GOTO M6
WRITE
GSA GT(GA)
M4: ALLOC DA(*) F(COMM)
ALLOC DA(*) F(FT06F001)
ALLOC DA(*) F(FT09F001)
IF &SM - = &MS THEN GOTO M5
SET &NAME = &STR(&PDSNAM. . FORT(&MEMB. .))
SET &PROG = &MEMB
COPY &NAME &PROG. . FORT
M5: FORT &PROG.
```



```
LINK &PROG..OBJ
CALL &PROG.(TEMPNAME)
CONTROL MAIN NOMSG
FREE DA(&PROG..LOAD)
ERASE (&PROG..OBJ &PROG..LOAD)
IF &SM = &MS THEN ERASE &PROG..FORT
CONTROL MSG
IF &DEVICE = &T611 THEN FREE FILE(DISPL)
FREE F(FT09F001)
GOTO M7
M6: WRITE EINGABEFehler: DEVICE WEDER GA, T4015 NOCH KSG
M7: CONCAT ('SYS1.FORTLIB' 'SYS2.HARWELL' 'SYS2.FORTLIB'-
           'SYS2.CALCOMP' 'SYS2.IMSL.DP' 'SYS2.GSMIN')
WRITE *** ENDE DER PROGRAMMAUSFUEHRUNG***
EXIT
```

Aufgerufen wird die KP mit:

```
exec @kp-name@
```

Anstelle von @kp-name@ wird der Name der KP (z. B.: AGS.CLIST) angegeben.

Die für die Ausführung der KP relevanten Daten werden von der KP durch selbsterklärenden Text erfragt und bedürfen hier keiner weiteren Erläuterung.

Eventuell problemorientierte Datei-Verknüpfungen (Allokationen) müssen vor der KP-Ausführung durchgeführt werden.

Es sei an dieser Stelle darauf hingewiesen, daß Datei-Verknüpfungen nicht über das File FT05F001 erfolgen können, da diese Unit bei TRACEGS für Terminaleingabe reserviert bleiben muß.

Will man das darzustellende Bild speichern, muß, ebenfalls vor Ausführung der KP, eine entsprechende Datei zur Verfügung gestellt werden. Die Bilddatei ist bei TRACEGS grundsätzlich nur über das File FT49F001 erreichbar.

Es ist folgende Allokation (mit Attributangabe) durchzuführen:

```
attrib dcb recfm(u) blksize(19069)
alloc da(@buchname@) f(ft49f001) new space(@quant1@ @quant2@) tracks
dir(1) using(dcb)
```

Anstelle von @quant1@ und @quant2@ (primary und secondary quantity)

wird die Anzahl der zu reservierenden Spuren angegeben. Ein mittlerer Wert ist 5, also z. B. space(5 5).

Für @buchname@ kann ein beliebiger Name gewählt werden. Ein Qualifier ist nicht erforderlich. Beispiel: GSBOOK.

Bilder, die in TSO-Dateien gespeichert vorliegen, lassen sich auf den Bildschirmgeräten auf einfache Weise unter Verwendung der im folgenden wiedergegebenen KP wiederholen.

```
PROC 0
WRITE
WRITE ***** KOMMANDOPROZEDUR REPI *****
WRITE ***** T R A C E G S *****
WRITE
SET &GA = &STR(GA)
SET &T611 = &STR(KSG)
SET &T415 = &STR(T4015)
SET &MS = &STR(M)
WRITE EINGABE (ERFORDERLICH FUER ALLOKATIONEN):
WRITE =====
WRITE NAME DES AUF DEM BILDSCHIRM ABZUBILDENDEN
WRITENR TSO-BILD-DATA-SET'S (OHNE TSO-USER-PREFIX)_:
READ &BDS
WRITENR AUSGABEGERAET (GA/T4015/KSG)_____:
READ &DEVICE
IF &DEVICE. -=& &T611 THEN GOTO M1
WRITENR ADRESSE (KARLSRUHER SICHTGERAET/KSG)_____:
READ &ADDR
WRITE
FREE F(DISPL)
ALLOC UNIT(&ADDR.) F(DISPL) NEW
GOTO M3
M1: IF &DEVICE -=& &T415 THEN GOTO M2
WRITE
GSA
GOTO M3
M2: IF &DEVICE -=& &GA THEN GOTO M4
WRITE
GSA GT(GA)
```

```
M3: ALLOC DA(*) F(COMM)
ALLOC DA(*) F(FT06F001)
ALLOC DA(*) F(FT09F001)
ALLOC DA(&BDS.) F($$GS$$)
CALL 'IRB178.AGS.LOAD(REPI)'  
IF &DEVICE = &T611 THEN FREE FILE(DISPL)
FREE F(FT09F001)
GOTO M5
M4: WRITE EINGABEFEHLER: DEVICE WEDER GA, T4015 NOCH KSG
M5: WRITE *** ENDE DER PROGRAMMAUSFUEHRUNG ***
EXIT
```

Gestartet wird die KP mit:

```
exec @kp-name@
```

Für @kp-name@ wird der Name der KP (z. B.: REPI.CLIST) angegeben. Die für die Ausführung der Prozedur relevanten Daten werden von der KP durch selbsterklärenden Text erfragt und sind hier keiner weiteren Erläuterung bedürftig.

Programmierhinweis: Für die Darstellung von Bildern auf den Bildschirmgeräten im TSO-Foreground kann eine vorteilhafte Verfahrensweise angewendet werden, die im Stapelbetrieb nur bei der Bildspeicherung (CALL PLOTI(0,...)) möglich ist. Es handelt sich hierbei um das mehrmalige Initialisieren und Abschließen von Bildern in einer Programmeinheit. Der Ablauf der Bilderstellung vollzieht sich dann in der Weise, daß Bild für Bild, gesteuert durch entsprechende Terminaleingabe, auf dem Bildschirm ausgegeben wird. Dieser Vorgang kann sich beliebig oft wiederholen.

Die Vorteile bei diesem Verfahren sind darin zu sehen, daß Einzelbilder immer auf den maximalen Bildschirmgrenzen (auf dem ohnehin kleinen Schirm) abgebildet werden und Kernspeicher für die Bildinformation gespart wird.

Bei der Bildspeicherung wird jedes Einzelbild als Member in dem spezifizierten Bild-Data-Set (partitioned organisiert!) gespeichert, entsprechend einer in TRACEGS fest vereinbarten Namensgenerierung.

Beispiel:			Namensgenerierung bei Bildspeicherung in ei- nem Data-Set (Member).
-----	Start	.	
P		.	
r	-----	CALL PLOTI(...,....)	
o	1. Bild	.	---> T00
g	-----	CALL PLOTP(...,....,999)	
r		.	
a		.	
m	-----	CALL PLOTI(...,....)	
m	2. Bild	.	---> T01
e	-----	CALL PLOTP(...,....,999)	
i		.	
n		.	
h	-----	CALL PLOTI(...,....)	
e	n. Bild	.	---> T0n
i	(n <= 100)	CALL PLOTP(...,....,999)	
t	-----	.	
-----	Stop		

Alle in einem Bild-Data-Set gespeicherten Bilder (Member) können während einer Job-Passage in der generierten Reihenfolge wiederholt dargestellt werden.

Anmerkung: An den Bildschirmterminals steht bei Verwendung von TRACEGS eine Bildfläche zur Verfügung, die im Seitenverhältnis der DIN-Papierformat-Norm entspricht (Bild-Höhe/-Breite = $1/\sqrt{2}$ und umgekehrt).

Da ein Bild, unabhängig von seinem Bildmaßstab, immer auf der gesamten definierten (DIN-Format) Bildschirmfläche abgebildet wird, kann unter Umständen eine erhebliche Bildverzerrung entstehen. Um dies zu vermeiden, sollte der Benutzer in seinem Programm einen DIN-gerechten physischen Rahmen definieren (z. B. mit CADRE oder CADREF), dessen Größe so bemessen werden muß, daß sämtliche Koordinatenangaben des darzustellenden Bildes im Rahmenfeld liegen.

Mit der TSO-Kommandoprozedur für die Bildwiederholung können an den Bildschirmgeräten nur Dateien bearbeitet werden, die den TSO-Konventionen (katalogisierter Name mit TSO-User-Prefix) entsprechen. Ist dies

nicht der Fall, muß entweder der Bild-Data-Set in das eigene TSO-Kontingent kopiert oder die Kommandoprozedur entsprechend abgeändert werden.

Eine Möglichkeit für das Umkopieren ist im folgenden wiedergegeben (Verwendung des IBM-Utility's 'IEHMOVE' /4/, Batch-Job).

```
// ... JOB - Karte ...
// EXEC   PGM=IEHMOVE
//SYSPRINT DD SYSOUT=*
//SYSUT1  DD UNIT=NOVIO,DISP=(,DELETE),SPACE=(TRK,50)
//DD1     DD UNIT=DISK,VOL=SER=@tp@,DISP=SHR
//DD2     DD UNIT=DISK,VOL=SER=@bp@,DISP=OLD           c72
//SYSIN   DD *                                         |
COPYDD COPY DSNAME=@dsold@,FROM=@unit1@=@bp@,TO=@unit2@=@tp@, *
          FROMDD=DD2,RENAME=@dsnew@,CATLG
          |
          c16
```

In vorstehender JCL muß ersetzt werden:

@tp@ durch den Volume-Serial-Namen der TSO-Magnetplatte, welche den zu kopierenden Bild-Data-Set aufnehmen soll.

@bp@ durch den Volume-Serial-Namen der Magnetplatte, auf welcher sich der zu kopierende Bild-Data-Set befindet.

@dsold@ durch den Namen des zu kopierenden und auf der Magnetplatte **@bp@** befindlichen Bild-Data-Set's.

@dsnew@ durch einen, den TSO-Konventionen entsprechenden Namen für die Kopie des Bild-Data-Set's.

Die RENAME-Option kann entfallen, wenn **@dsold@** schon den TSO-Namenskonventionen entspricht und kein neuer Name gewünscht wird.

@unit1@ durch den UNIT-Namen der Magnetplatteneinheit, auf welcher sich der zu kopierende Bild-Data-Set befindet.

@unit2@ durch den UNIT-Namen für die TSO-Magnetplatteneinheit, auf die der Bild-Data-Set kopiert werden soll.

In Abb. 1 (Anhang) ist die Verfahrensweise der Bilderstellung mit der Foreground-Version von TRACEGS vereinfacht in einem Programmflußplan dargestellt.

24. TRACEGS im Stapelbetrieb

Die Verwendung von TRACEGS im Stapelbetrieb ist dann zu empfehlen, wenn entweder infolge von Kernspeichermangel im TSO-Foreground keine Bilderstellung bzw. Bildspeicherung möglich ist oder auf direktem Weg Zeichnungen auf den On-/Off-Line-Zeichengeräten dargestellt werden sollen. Dies gilt sowohl für die Neuerarbeitung von Bildinformation, als auch für die Reproduktion von Zeichnungen aus Bild-Dateien.

Infolge der unterschiedlichen Verfahrensweise bei der Bilderstellung muß für die Off-Line-Zeichengeräte STATOS und XYNETICS in der JCL ein Plottape spezifiziert und für das On-Line-Zeichengerät VERSATEC ein zusätzlicher separater Ausgabe-Step an die JCL angehängt werden.

Wie bei der Foreground-Version ist auch hier die Speicherung von Bildern in Bild-Dateien möglich.

Die Load-Moduln für den Stapelbetrieb befinden sich in der Privat-Bibliothek 'IRB178.TRACEGS.LOAD' (katalogisiert) auf der Magnetplatte BATOOD.

Im folgenden ist eine vollständige JCL-Folge für die Benutzung von TRACEGS im Stapelbetrieb wiedergegeben. Es soll hierbei eine Zeichnung auf dem On-Line-Zeichengerät VERSATEC erstellt und eine Bild-Datei (DD-Rekord: FT49F001) angelegt werden.

Kommentarkarten (/*) deuten Alternativen an.

```
//      ... JOB - Karte mit Mindest-Region-Angabe von 1024K ...
//A EXEC   FHCLG,PLOT=GSMAN,USER='SYS2.CALCOMP'
//C.SYSRINT DD SYSOUT=*
//C.SYSIN   DD *
           (Fortran-IV-Programm mit TRACEGS-Anweisungen)
           (hier mit 'CALL PLOTI(3,...)' für On-Line-VERSATEC-Plotter)
/*
//L.SYSLIB   DD
//          DD DSN=IRB178.TRACEGS.LOAD,DISP=SHR
//L.SYSRINT DD SYSOUT=*
/*
//G.PLOTPARM DD *
           &PLOT XMAX=317., YMAX=23. &END      x:  S = STATOS
//G.COMM    DD SYSOUT=*                      | X/Y = XYNETICS (Kuli(X)/Tusche (Y))
/* G. PLOTTAPE DD UNIT=T0800,VOL=SER=x@jn@,LABEL=(,NL),DCB=DEN=2
//G.FT49F001 DD UNIT=DISK,VOL=SER=@platte@,SPACE=(TRK,(5,5,1)),
//          DISP=(NEW,CATLG),DSN=@bn@,DCB=RECFM=U,BLKSIZE=19069
```

```
//G.SYSIN DD *  
    (eventuell Programmeingabe)  
/*  
//B EXEC SVPLOT  
/*  
//
```

Es muß ersetzt werden:

@jn@ durch die Benutzernummer und die Job-Kennung. Wie auf der Job-Karte, jedoch ohne Institutsnamen. Z.B.: 178PL.

@platte@ durch den Namen einer Magnetplatte, auf welcher ein Bild-Data-Set angelegt und Bildinformation gespeichert werden soll. Z.B.: BAT00D.

@bn@ durch einen gültigen, den Konventionen entsprechenden Namen für die Bild-Datei. Z. B.: IRB178.GSBOOK.

Hinweise: Die angegebene User-Bibliothek auf der EXEC-Karte 'SYS2.CALCOMP' braucht man nur dann spezifizieren, wenn Referenzen aus der CALCOMP-Software gelöst werden sollen.

Besitzt die spezifizierte Magnetplatte, auf welcher eine Bild-Datei gespeichert werden soll, keinen Katalog, wird der Dispositions-Untersparameter 'CATLG' durch 'KEEP' ersetzt (DD-Record FT49F001).

Die DD-Karte '//G.PLOTPARM DD *' und die folgende Eingabezeile sind nur bei Verwendung des VERSATEC-Zeichengerätes erforderlich.

Die wiederholte zeichnerische Darstellung eines Bildes (auf den On-/Off-Line-Zeichengeräten), das in einer Bild-Datei gespeichert vorliegt, läßt sich mit Hilfe des Programmes REPIBP (IRB178.TRACEGS.LOAD) durchführen. Zur Ausführung ist nachstehende JCL erforderlich. Es soll hierbei ein Bild auf einem Off-Line-Zeichengerät reproduziert werden. Kommentarkarten (/* ...) deuten Alternativen an.

```
//      ... JOB - Karte mit Mindest-Region-Angabe von 1024K ...
//A EXEC      FHLG,PLOT=GSMAN
//L. LGS      DD DSN=IRB178.TRACEGS.LOAD,DISP=SHR
//L. SYSPRINT DD SYSOUT=*
//L. SYSIN    DD *
      INCLUDE  LGS(REPIBP)
/*
                        x:  S = STATOS
//G. COMM     DD SYSOUT=*      | X/Y = XYNETICS (Kuli (X)/Tusche (Y))
//G. PLOTTAPE DD UNIT=T0800,VOL=SER=x@jn@,LABEL=(,NL),DCB=DEN=2
//G. FT49F001 DD DSN=@bn@,DISP=SHR
//G. SYSIN    DD *
      Erforderliche Eingabe (unformatiert):
      1. Plotter-Typ: 1 = STATOS, 2 = XYNETICS, 3 = VERSATEC      (INTEGER)
      2. GS-Workspace: Z. Zt. max. 6 MB spezifizierbar          (INTEGER)
/*
//* B EXEC    SVPLOT
/*
//
```

Für die in @-Zeichen eingeschlossenen Bezeichnungen gilt sinngemäß das gleiche wie zuvor bei der JCL für die TRACEGS-Benutzung im Stapelbetrieb erläutert wurde.

Es ist zu beachten, daß der gespeicherte Bild-Data-Set nur über das File FT49F001 angesprochen werden kann.

Die Darstellung gespeicherter Bilder an den Bildschirmgeräten ist unter Punkt 23. beschrieben.

In Abb. 2 (Anhang) ist die Verfahrensweise der Bilderstellung mit der TRACEGS-Version für den Stapelbetrieb vereinfacht in einem Programmflußplan wiedergegeben.

25. Kompatibilität mit der konventionellen Software

Wie bereits in der Einleitung erwähnt, stehen in TRACEGS Moduln zur Verfügung, die hinsichtlich Syntax und Funktion den wesentlichsten Basis-Routinen der konventionellen Software (CALCOMP) /2/ entsprechen. Damit bietet sich die Möglichkeit, Programme, die CALCOMP-Aufrufe enthalten, ohne großen Umstellungsaufwand mit TRACEGS auszuführen.

Für TRACEGS wurden folgende Routinen der konventionellen Software simuliert:

UP's, die namentlich und funktionell der konventionellen Software entsprechen			entsprechende TRACEGS Unterprogramme
PLOT (Koordinaten in <u>inch</u>)		entspricht	PLOTP (Koordinaten in <u>cm</u>)
SYMBOL	"	"	SYMBP "
SYMBL1	"	"	SYMBP1 "
SYMBL2	"	"	SYMBP2 "
SYMBL3	"	"	SYMBP3 "
NUMBER	"	"	NUMBP "
WHERE	"	"	WHEREP "
FACTOR (*)		"	FACTOR
PLOTS (**)		"	PLOTI

(*) Das Unterprogramm FACTOR hat in TRACEGS gegenüber der konventionellen Software eine eingeschränkte Funktion. FACTOR kann in TRACEGS lediglich einmal und nur auf das gesamte Bild angewendet werden (siehe hierzu auch Punkt 7.).

(**) PLOTS ist die einzige Routine, die bezüglich Bedeutung und Anzahl der Argumente der TRACEGS-Routine PLOTI angepaßt werden muß.

Funktionelle Unterprogramme der konventionellen Software, die mit Hilfe der Basis-Routinen aufgebaut sind ('LINE', 'AXIS' usw.), können somit auch ausgeführt werden.

Eventuelle Benutzer, die konventionelle Software unter TRACEGS bearbeiten wollen, seien darauf hingewiesen, daß bei häufiger Verwendung des Unterprogrammes 'PLOT' der Kernspeicher unwirtschaftlich ausgenutzt wird. Die Folge davon ist, daß nur relativ 'kleine' Bilder mit wenig Bildinformation dargestellt werden können (siehe auch Punkt 3.).

26. Programm-Konfiguration

COMMON /XYPOS/	COMMON /FAKTOR/	COMMON /SCALA/	COMMON /BOOK/	COMMON /WORK/	Unterprogramme
*	*	*	*		PLOTI/PLOTS
*	*	*	*		PLOTP/PLOT
			*		EX
*	*				SYMBP/SYMBOL
*	*				SYMBS
*	*				SYMBP1/SYMBL1
*	*				SYMBS1
*	*				SYMBP2/SYMBL2
*	*				SYMBS2
*	*				SYMBP3/SYMBL3
*	*				SYMBS3
*	*				NUMBP/NUMBER
*	*				NUMBS
*	*	*			WHEREP/WHERE
		*			FACTOR
*					PEN
*	*				AXLINI
	*				XPOS (*)
*	*				AYLINI
	*				YPOS (*)
*	*				AXLOGI
*	*				AYLOGI
*					NETLIN (*)
*	*				POL
*	*			*	CURVE
*	*			*	CURVEP
*				*	CURVLV
*					CADRE
*					FLECHE
*					BALKEN
*					DISY
*					TASY
*					STRIP
*					ECRAN
*					TEXT/TEXTS
			*		ROT90
					CADREF
					CERCLE
					KFK
					EXPD (*)
					UMLAUT (*)
					UMLTS (*)
					ZST (*)

Die mit (*) gekennzeichneten Routinen sind Hilfs-Unterprogramme.

In den benannten COMMON's sind gespeichert:

/XYPOS/ X00, Y00, X01, Y01, IFB
Aktuelle Schreibstiftkoordinaten und Nummer für die Zeichenfarbe.

/FAKTOR/ FX, FY, BXMIN, BYMIN, LLX, LLY, FAKT
Maßstabsfaktoren und Steuergrößen für die Achsengestaltung.

/SCALA/ SXY
Zeichnungsmaßstab.

/BOOK/ IBOOK, KPT, BN, IROT
Steuergrößen für die Bildspeicherung und den Zeichengerätetyp.
Aktueller Bildname. Steuergröße, deren Wert darüber entscheidet, ob ein
Bild vor seiner Ausgabe um 90 Grad gedreht werden soll oder nicht.

/WORK/ X(2000), Y(2000)
Gemeinsamer Workspace der Unterprogramme CURVE, CURVEP und CURVLV.

27. Literatur

- /1/ K. Gogg u. a. : GS-Handbuch.
Unveröffentlichter Bericht des KfK, März 1980.
- /2/ P. Fette : Plot-Handbuch.
Unveröffentlichter Bericht des KfK, Juni 1979.
- /3/ H. Blesene : Fortran-Unterprogramm CONVY.
Unveröffentlichter Bericht des KfK, April 1971.
- /4/ IBM : Systems Reference Library. OS Utilities.
File Number S360-32.

28. Anhang

- Abbildungen	Seite
1. TRACEGS im TSO-Foreground.....	62
2. TRACEGS im Batch.....	63
3. Beispiel für eine TSO-Text-Datei.....	64
4. Gezeichnete Ausgabe der Subroutine Text.....	65
- Programmbeispiele	
1.a Programmbeispiel 1 (Achsendarstellungen).....	66
1.b Zeichnung zu Programmbeispiel 1.....	67
2.a Programmbeispiel 2 (Diagramm).....	68
2.b Zeichnung zu Programmbeispiel 2.....	69
3.a Programmbeispiel 3 (Sinus/Ellipsen).....	70
3.b Zeichnung zu Programmbeispiel 3.....	71
4.a Programmbeispiel 4 (Calcomp-Test).....	72
4.b Zeichnung zu Programmbeispiel 4.....	73
5.a Programmbeispiel 5 (Kreis-Zylinder-Schnitte)...	74
5.b Zeichnung zu Programmbeispiel 5.....	75
6.a Programmbeispiel 6 (Polarkoordinaten).....	76
6.b Zeichnung zu Programmbeispiel 6.....	77

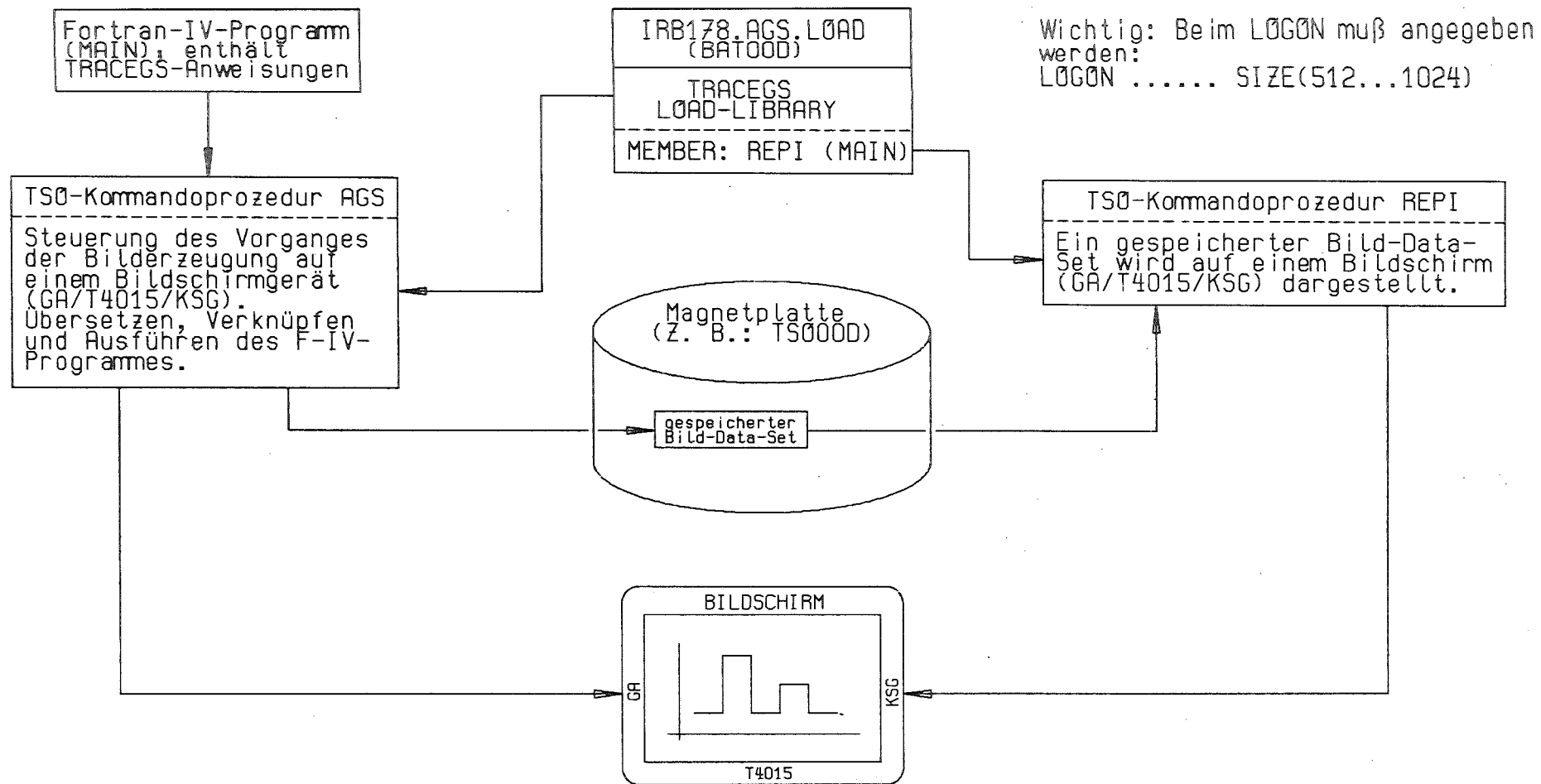


Abb. 1 TRACEGS im TSO-Foreground (vereinfachter Datenfluß)

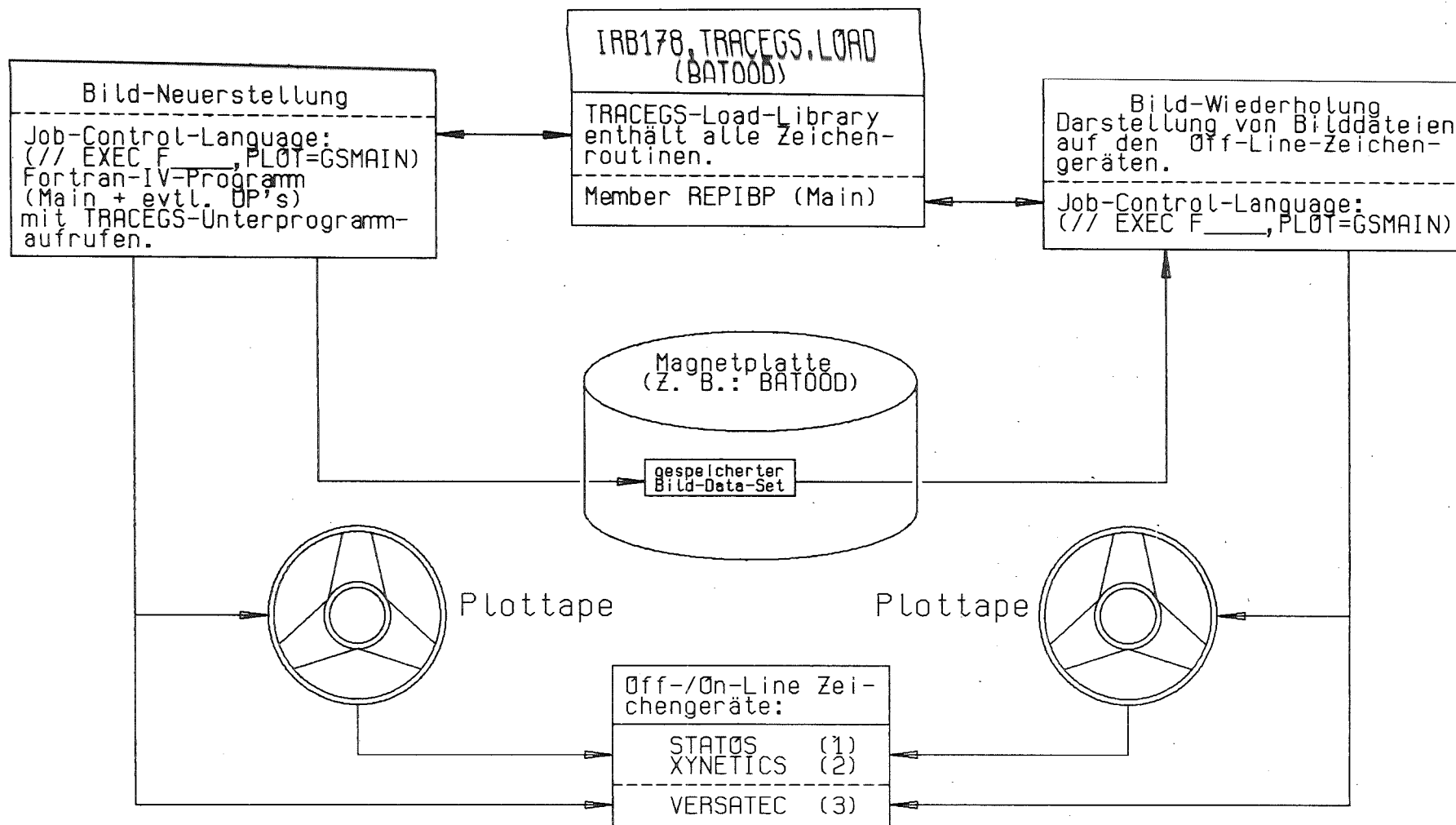


Abb. 2 TRACEGS im Batch (vereinfachter Datenfluß)

.25 .01 .2 0 1

Die Unterprogramme 'TEXT' und 'TEXTS' zeichnen Texte in Groß- und Kleinbuchstaben (lateinisch und griechisch). 'TEXT' ist die Normalschrift-, 'TEXTS' die Strichschrift-Version.

Es sind zwei verschiedene Strichstärken möglich.

Dem jeweils zu zeichnenden Text müssen die folgenden Informationen vorangestellt werden:

\$

Schriftgröße _____ (SH) cm REAL*4\$
Spaltenzwischenraum zwischen zwei Buchstaben (SZR) cm REAL*4\$
Zeilenzwischenraum zwischen zwei Zeilen _____ (ZZR) cm REAL*4\$
Schriftstärke, 0=dünn, 1=dick _____ (SD) -- INTEGER\$
Farbe des zu zeichnenden Textes _____ (IC) -- INTEGER\$

\$

Nach dieser Zeile kann beliebiger Text folgen.

Sollen die Parameter geändert werden, müssen die Spalten 1 und 2 einer neuen Zeile je ein Dollar-Zeichen enthalten.

Danach wird wie zuvor beschrieben verfahren.

Der Vorgang kann sich beliebig wiederholen.

Es ist sinnvoll, jede eingegebene Textzeile mit einem Dollarzeichen zu beenden, weil damit Rechenzeit gespart wird.

\$

Die Unterprogramme 'TEXT' und 'TEXTS' haben je drei Argumente:

\$

PXK ____ (cm): Textbeginn in X-Richtung

PYK ____ (cm): Textbeginn in Y-Richtung

IUNIT ____ (-): Nummer der Eingabeeinheit, über welche der zu zeichnende Text eingelesen werden soll.

\$

Aufruf: CALL TEXT(PXK,PYK,IUNIT)

oder

CALL TEXTS(PXK,PYK,IUNIT)

\$

Zeichensatz:

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz. : ; ! ? / + - _ () < > & = * % \$

Symbole ____ : \$1,\$2,\$3,\$4,\$5,\$6,\$7,\$8,\$9,\$W,\$X,\$Y,\$Z. \$\$\$\$\$\$-\$\$\$

Umlaute ____ : \$A,\$O,\$U,\$a,\$o,\$u

griech. klein: \$A,\$B,\$C,\$D,\$E,\$F,\$G,\$I,\$K,\$L,\$M,\$N,\$O,\$P,\$R,\$S,\$T,\$X,\$Y,\$Z,\$Z,\$1,\$2,\$3,\$4

griech. groß: \$#C,\$#D,\$#F,\$#G,\$#L,\$#P,\$#R,\$#S,\$#2,\$#3,\$#4

\$

\$\$

.5 .01 .3 1 1

\$A,\$O,\$U,\$a,\$o,\$u,A,O,U,a,o,u, \$\$\$

\$\$

.25 .01 .4 0 1

Die Unterprogramme 'TEXT' und 'TEXTS' sind Member der Load-Library 'IRB178.TRACEGS.LOAD'.

Abb.: 3. Beispiel für eine TSO-Text-Datei.

Die Unterprogramme 'TEXT' und 'TEXTS' zeichnen Texte in Groß- und Kleinbuchstaben (lateinisch und griechisch). 'TEXT' ist die Normalschrift-, 'TEXTS' die Schrägschrift-Version.

Es sind zwei verschiedene Strichstärken möglich.

Dem jeweils zu zeichnenden Text müssen die folgenden Informationen vorangestellt werden:

Schrifthöhe_____	(SH)	cm	REAL*4
Spaltenzwischenraum zwischen zwei Buchstaben_	(SZR)	cm	REAL*4
Zwischenraum zwischen zwei Zeilen_____	(ZZR)	cm	REAL*4
Schriftdicke, 0=dünn, 1=dick_____	(SD)	--	INTEGER
Farbe des zu zeichnenden Textes_____	(IC)	--	INTEGER

Nach dieser Zeile kann beliebiger Text folgen.

Sollen die Parameter geändert werden, müssen die Spalten 1 und 2 einer neuen Zeile je ein Dollar-Zeichen enthalten.

Danach wird wie zuvor beschrieben verfahren.

Der Vorgang kann sich beliebig wiederholen.

Es ist sinnvoll, jede eingegebene Textzeile mit einem Dollarzeichen zu beenden, weil damit Rechenzeit gespart wird.

Die Unterprogramme 'TEXT' und 'TEXTS' haben je drei Argumente:

PXK____(cm): Textbeginn in X-Richtung

PYK____(cm): Textbeginn in Y-Richtung

IUNIT____(-): Nummer der Eingabeeinheit, über welche der zu zeichnende Text eingelesen werden soll.

Aufruf: CALL TEXT(PXK,PYK,IUNIT)
oder
CALL TEXTS(PXK,PYK,IUNIT)

Zeichensatz:

ABCDEFGHIJKLMNÖPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz.,:;!/?/+_-_()<>&=*%

Symbole_____: @, Δ, +, x, φ, φ, x, z, y, π, *, z, l. \$\$\$ \$

Umlaute_____: Ä, Ö, Ü, ä, ö, ü

griech. klein: α, β, γ, δ, ε, φ, ρ, ι, κ, λ, μ, ν, ο, π, ρ, σ, τ, ξ, υ,
ξ, η, θ, ψ, ω

griech. groß_: X, Δ, Φ, Γ, Λ, Π, Ρ, Σ, Θ, Ψ, Ω

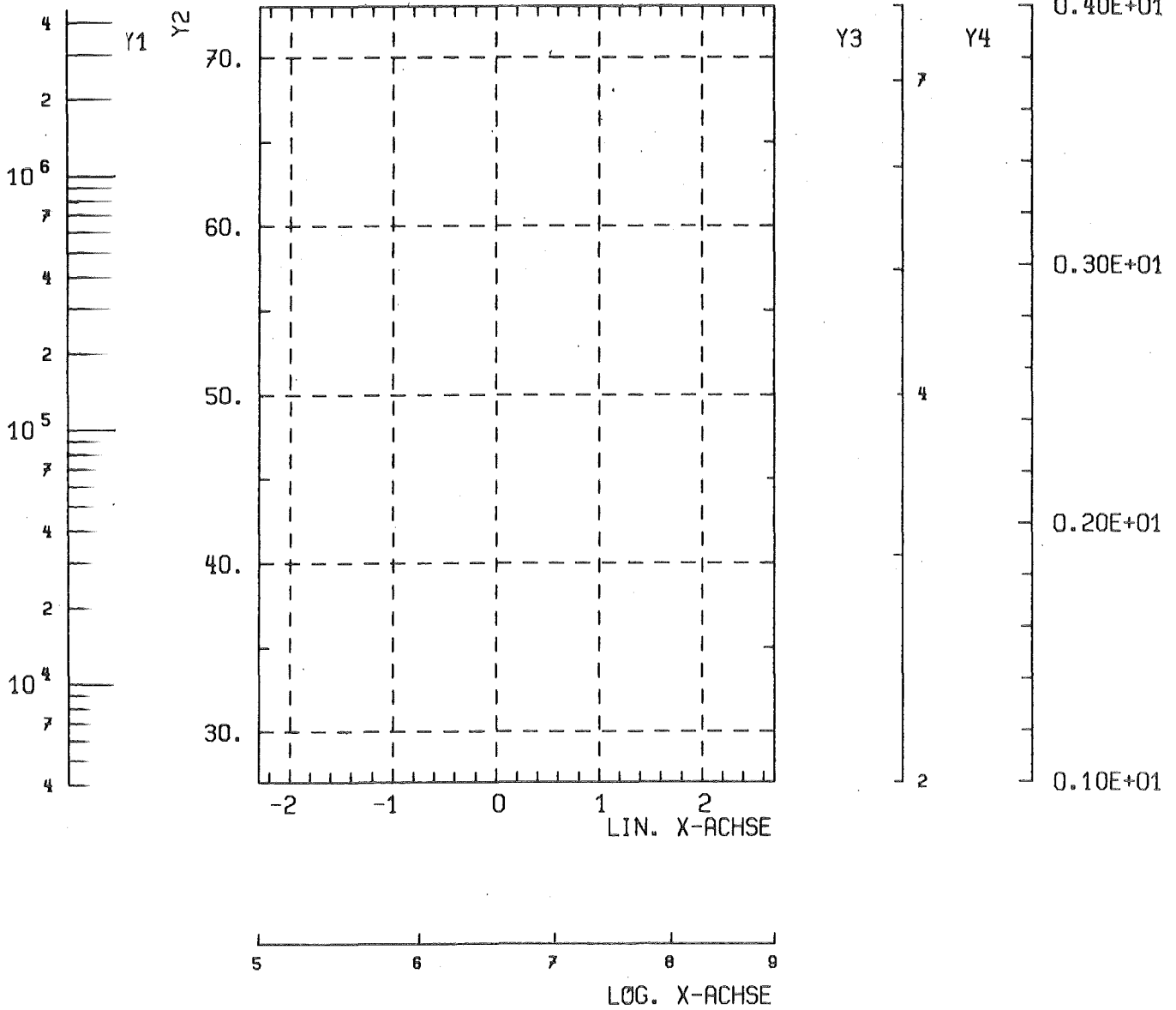
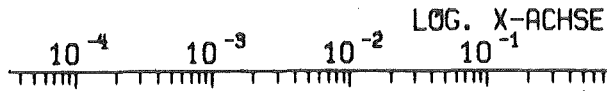
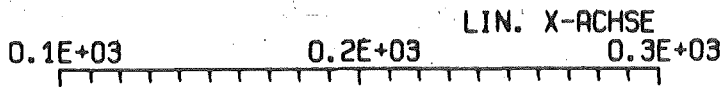
Ä, Ö, Ü, ä, ö, ü, A, O, U, a, o, u, \$

Die Unterprogramme 'TEXT' und 'TEXTS' sind Member der Load-Library
'IRB178.TRACEGS.LOAD'.



```
C      TESTPROGRAMM (ACHSENDARSTELLUNGEN)                                0000001
      REAL TEXT1(3)/'LOG. ',' X-A','CHSE'/                               0000002
      REAL TEXT2(3)/'LIN. ',' X-A','CHSE'/                               0000003
      CALL PLOTI(3,1024)                                                 0000004
      CALL CADRE(21.,29.7)                                               0000005
      CALL SYMBP (3.8,2.,.5,'Z',0.,1)                                     0000006
      CALL SYMBP1(999.,999.,.5,'EICHNUNG ZU ',0.,12)                     0000007
      CALL SYMBP (999.,999.,.5,'P',0.,1)                                 0000008
      CALL SYMBP1(999.,999.,.5,'ROGRAMMBEISPIEL ',0.,16)                0000009
      CALL SYMBP (999.,999.,.5,'1.',0.,2)                                0000010
      CALL PLOTP(5.5,5.5,-3)                                             0000011
      CALL AXLOG1(8.,5.0E+1,9.0E+1,3,0.,1,0.15,0.20,0.25,12,TEXT1)     0000012
      CALL PLOTP(0.,19.5,-3)                                             0000013
      CALL AXLINI(8.,100.,300.,100.,100.,0.,9,0.15,0.20,-0.25,-11,12,  0000014
*      TEXT2)                                                            0000015
      CALL PLOTP(0.,-2.5,-3)                                             0000016
      CALL AXLOG1(8.,3.3E-5,7.6E-1,0,0.,1,0.15,0.20,-0.25,12,TEXT1)    0000017
      CALL PLOTP(-2.5,-14.5,-3)                                         0000018
      CALL AYLOG1(12.,4.E+3,4.5E+6,1,0.,1,0.15,0.20,0.25,2,'Y1')       0000019
      CALL PLOTP(12.5,0.,-3)                                             0000020
      CALL AYLOG1(12.,2.0E+7,8.0E+7,1,0.,2,0.15,0.20,-0.25,4,' Y3')    0000021
      CALL PLOTP(2.0,0.,-3)                                             0000022
      CALL AYLINI(12.,1.,4.,1.,3.,0.,4,0.15,0.20,-0.25,-12,4,' Y4')    0000023
      CALL PLOTP(-12.0,0.,-3)                                           0000024
      CALL AXLINI(8.,-2.3,2.7,1.,-1.,-12.,4,0.15,0.,0.25,-1,12,TEXT2)  0000025
      CALL AYLINI(12.,27.,73.,10.,50.,-8.,1,0.15,0.,0.25,0,-2,'Y2')    0000026
      CALL PLOTP( 0., 0.,999)                                           0000027
      STOP                                                                0000028
      END                                                                0000029
```

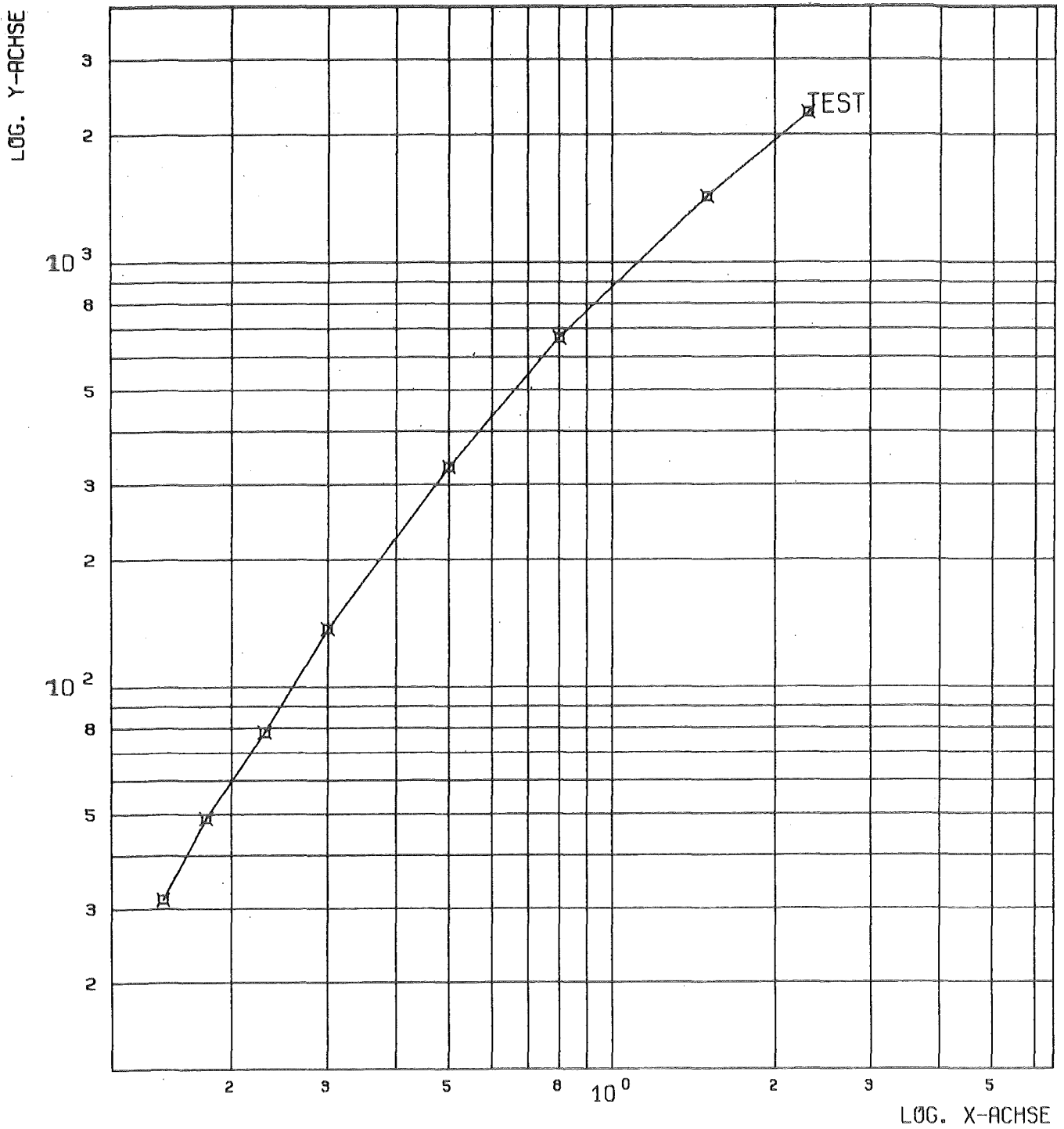
Programmbeispiel 1



Zeichnung zu Programmbeispiel 1.

C	TESTPROGRAMM (DIAGRAMM)	0000001
	REAL X(8), Y(8)	0000002
	REAL TEXTX(3)/'LOG.', ' X-A', 'CHSE'/'	0000003
	REAL TEXTY(3)/'LOG.', ' Y-A', 'CHSE'/'	0000004
	DATA X/ .15, .18, .23, .30, .50, .80, 1.5, 2.3/	0000005
	DATA Y/31.5,49.0,78.0,137.,330.,665.,1430.,2250./	0000006
	CALL PLOTI(3,1024)	0000007
	CALL CADRE(21.,29.7)	0000008
	CALL KFK(41,'IRB ')	0000009
	CALL PLOTP(3.5,7.,-3)	0000010
	CALL AXLOG1(16.,1.2E-1,6.5,2,18.,1,0.,0.,.25,12,TEXTX)	0000011
	CALL AYLOG1(18.,1.25E+1,4.E+3,2,16.,1,0.,0.,.25, -12,TEXTY)	0000012
	CALL CURVE(X,Y,8,10.,.25,1)	0000013
	CALL SYMBP(999.,999.,.3,'TEST',0.,4)	0000014
	CALL SYMBP (.35,-5.,.5,'Z',0.,1)	0000015
	CALL SYMBP1(999.,999.,.5,'EICHNUNG ZU ',0.,12)	0000016
	CALL SYMBP (999.,999.,.5,'P',0.,1)	0000017
	CALL SYMBP1(999.,999.,.5,'ROGRAMMBEISPIEL ',0.,16)	0000018
	CALL SYMBP (999.,999.,.5,'2.',0.,2)	0000019
	CALL PLOTP(0., 0.,999)	0000020
	STOP	0000021
	END	0000022

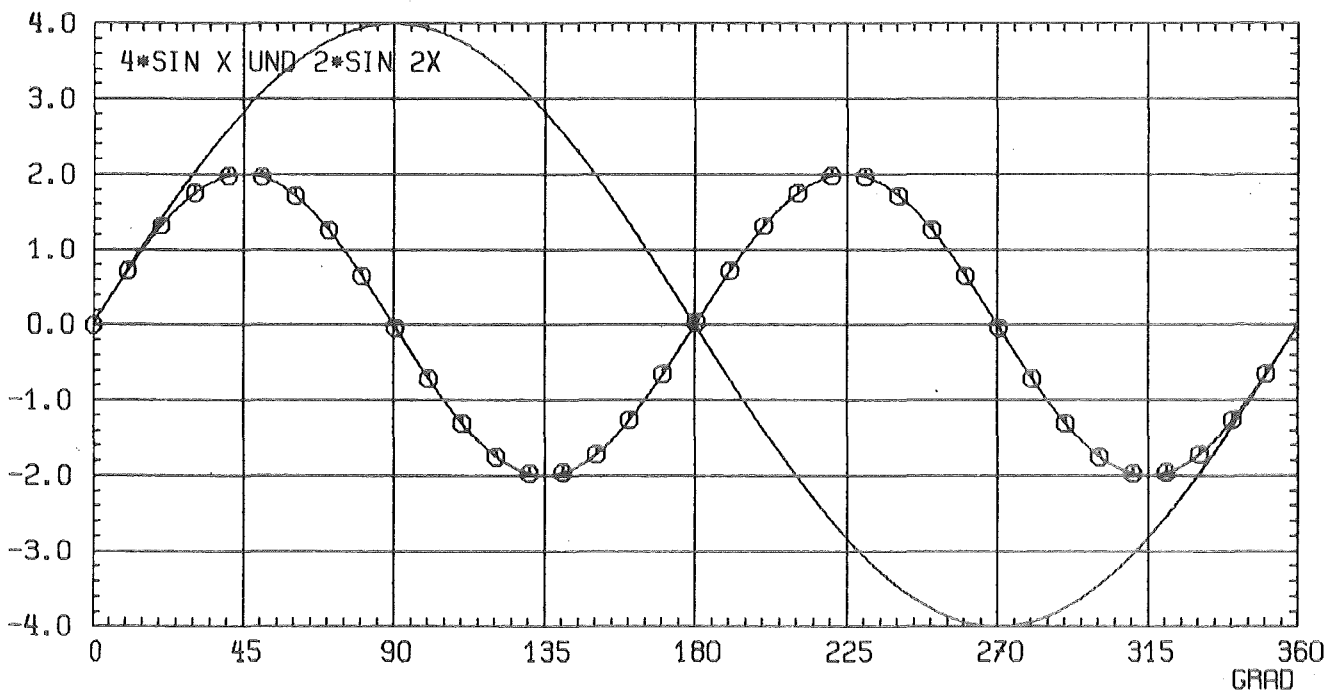
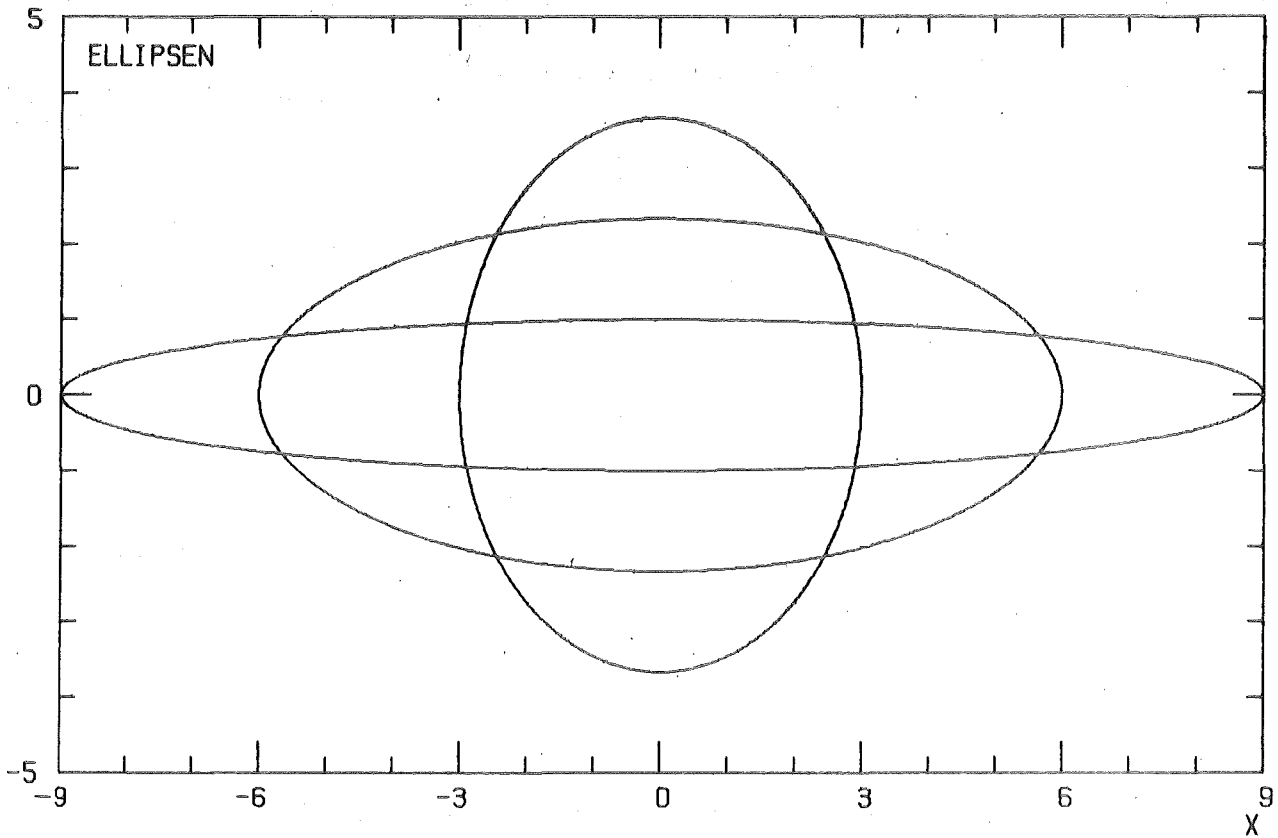
Programmbeispiel 2



Zeichnung zu Programmbeispiel 2.

```
C TESTPROGRAMM (SINUS/ELLIPSEN) 0000001
REAL XARRAY(750), YARRAY(750) 0000002
REAL TEXT1(1)/'GRAD'/ 0000003
REAL TEXT2(5)/'4*SI', 'N X ', 'UND ', '2*SI', 'N 2X'/ 0000004
REAL TEXT3(1)/'X ' 0000005
REAL TEXT4(2)/'ELLI', 'PSEN'/ 0000006
DATA CONST/0.0174533/ 0000007
CALL PLOTI(3,1024) 0000008
CALL CADRE(21.,29.7) 0000009
CALL SYMBP (3.85,1.6,..5,'Z',0.,1) 0000010
CALL SYMBP1(999.,999.,.5,'EICHNUNG ZU ',0.,12) 0000011
CALL SYMBP (999.,999.,.5,'P',0.,1) 0000012
CALL SYMBP1(999.,999.,.5,'ROGRAMMBEISPIEL ',0.,16) 0000013
CALL SYMBP (999.,999.,.5,'3.',0.,2) 0000014
C 1. BILD (SINUS) 0000015
CALL PLOTP(3.,5.,-3) 0000016
XMIN=0. 0000017
YMIN=-4. 0000018
CALL AXLIN1(16.,XMIN,360.,45.,360.,8.,8.,1,0.,.25,-1,4,TEXT1) 0000019
CALL AYLIN1(8.,YMIN,4.,1.,YMIN,16.,4.,1,0.,.25,1,20,TEXT2) 0000020
XARRAY(1)=0. 0000022
DO 1 I=2,720 0000023
1 XARRAY(I)=0.5*I 0000024
DO 2 I=1,720 0000025
PHI=XARRAY(I)*CONST 0000026
2 YARRAY(I)=4*SIN(PHI) 0000027
CALL CURVE(XARRAY,YARRAY,720,0,.25,0) 0000028
DO 3 I=1,720 0000029
PHI=XARRAY(I)*CONST 0000030
3 YARRAY(I)=2*SIN(2*PHI) 0000031
CALL CURVE(XARRAY,YARRAY,720,1,.25,20) 0000032
C 2. BILD (ELLIPSEN) 0000033
CALL PLOTP(0.,11.5,-3) 0000034
XMIN=-9. 0000035
YMIN=-5. 0000036
CALL AXLIN1(16.,XMIN,9.,3.,0.,10.,2,0.2,0.4,.25,-1,1,TEXT3) 0000037
CALL AYLIN1(10.,YMIN,5.,5.,YMIN,16.,4,0.2,0.4,.25,-1,8,TEXT4) 0000038
A=0. 0000040
B=5. 0000041
DO 5 K=1,3 0000042
A=A+3. 0000043
B=B-1.33333 0000044
DO 4 I=1,722 0000045
PHI=(I-1)*0.5*CONST 0000046
XARRAY(I)=A*COS(PHI) 0000047
YARRAY(I)=B*SIN(PHI) 0000048
4 CONTINUE 0000049
CALL CURVE(XARRAY,YARRAY,722,0,.25,0) 0000050
5 CONTINUE 0000051
CALL PLOTP(0.,0.,999) 0000052
STOP 0000053
END 0000054
```

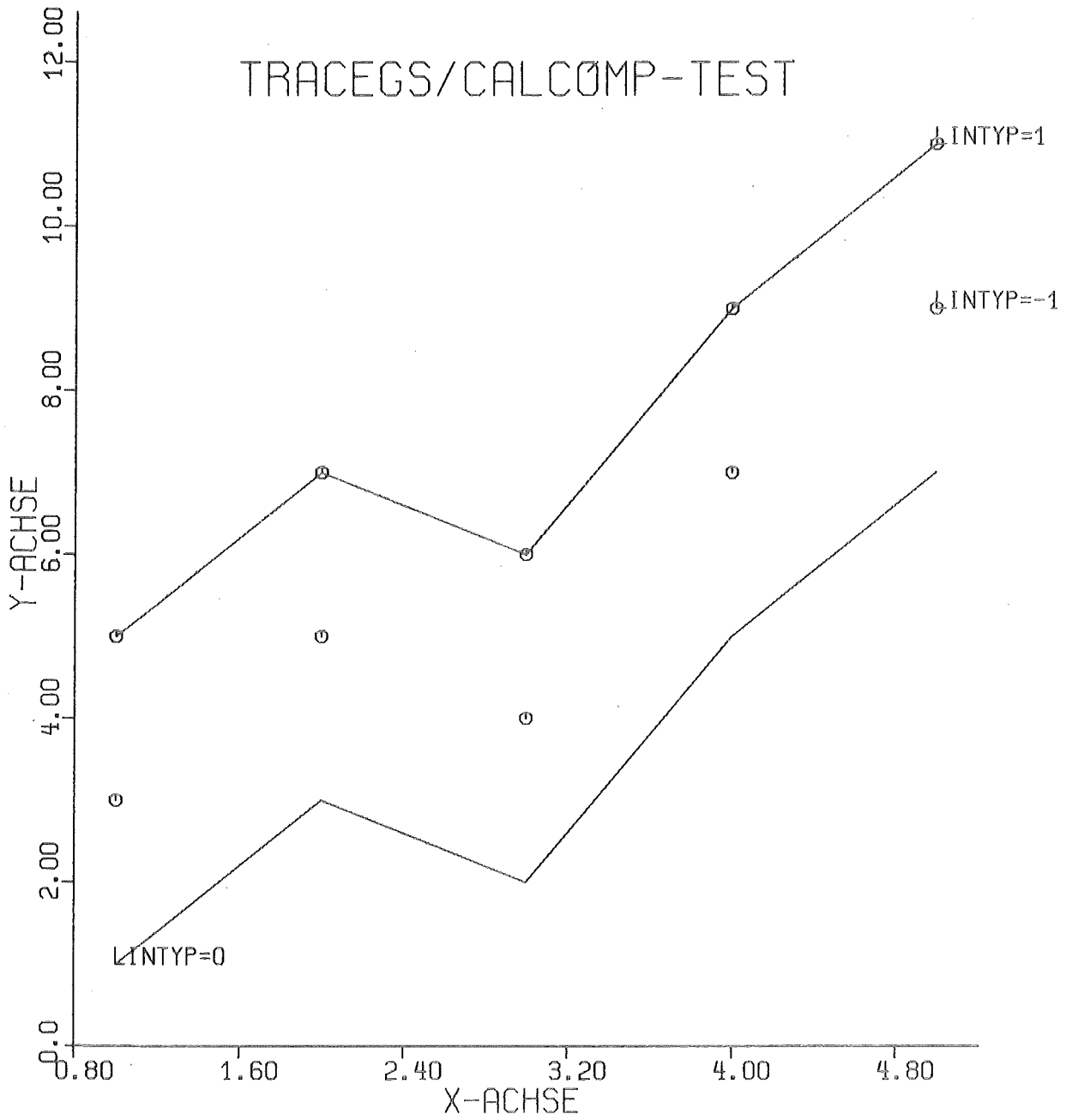
Programmbeispiel 3



Zeichnung zu Programmbeispiel 3.

```
C TESTPROGRAMM (CALCOMP) 0000001
REAL X(7)/1.,2.,3.,4.,5.,0.,0./ 0000002
REAL Y(7)/1.,3.,2.,5.,7.,0.,0./ 0000003
DATA ZOLL/2.54/ 0000004
CALL PLOTS(3,1024) 0000005
CALL PLOT(0.,29.7/ZOLL,2) 0000006
CALL PLOT(21./ZOLL,29.7/ZOLL,2) 0000007
CALL PLOT(21./ZOLL,0.,2) 0000008
CALL PLOT(0.,0.,2) 0000009
CALL SYMBOL(3.85/ZOLL,2.5/ZOLL,.5/ZOLL,'Z',0.,1) 0000010
CALL SYMBL1(999.,999.,.5/ZOLL,'EICHNUNG ZU ',0.,12) 0000011
CALL SYMBOL(999.,999.,.5/ZOLL,'P',0.,1) 0000012
CALL SYMBL1(999.,999.,.5/ZOLL,'ROGRAMMBEISPIEL ',0.,16) 0000013
CALL SYMBOL(999.,999.,.5/ZOLL,'4.',0.,2) 0000014
CALL PLOT(4./ZOLL,7./ZOLL,-3) 0000015
CALL SCALE(X,15./ZOLL,5,1) 0000016
CALL SCALE(Y,15./ZOLL,5,1) 0000017
CALL AXIS(0.,0.,'X-ACHSE',-7,14./ZOLL,0.,X(6),X(7)) 0000018
CALL AXIS(0.,0.,'Y-ACHSE',7,16./ZOLL,90.,Y(6),Y(7)) 0000019
CALL SYMBOL(1.,5.8,.2,'TRACEGS/CALCOMP-TEST',0.,20) 0000020
CALL LINE(X,Y,5,1,0,1) 0000021
CALL SYMBOL(999.,999.,.1,'LINTYP=',0.,7) 0000022
CALL NUMBER(999.,999.,.1,0.,0.,-1) 0000023
CALL PLOT(0.,1.,-3) 0000024
CALL LINE(X,Y,5,1,-1,1) 0000025
CALL SYMBOL(999.,999.,.1,'LINTYP=',0.,7) 0000026
CALL NUMBER(999.,999.,.1,-1.,0.,-1) 0000027
CALL PLOT(0.,1.,-3) 0000028
CALL LINE(X,Y,5,1,1,1) 0000029
CALL SYMBOL(999.,999.,.1,'LINTYP=',0.,7) 0000030
CALL NUMBER(999.,999.,.1,1.,0.,-1) 0000031
CALL PLOT(0.,0.,999) 0000032
STOP 0000033
END 0000034
```

Programmbeispiel 4



Zeichnung zu Programmbeispiel 4.

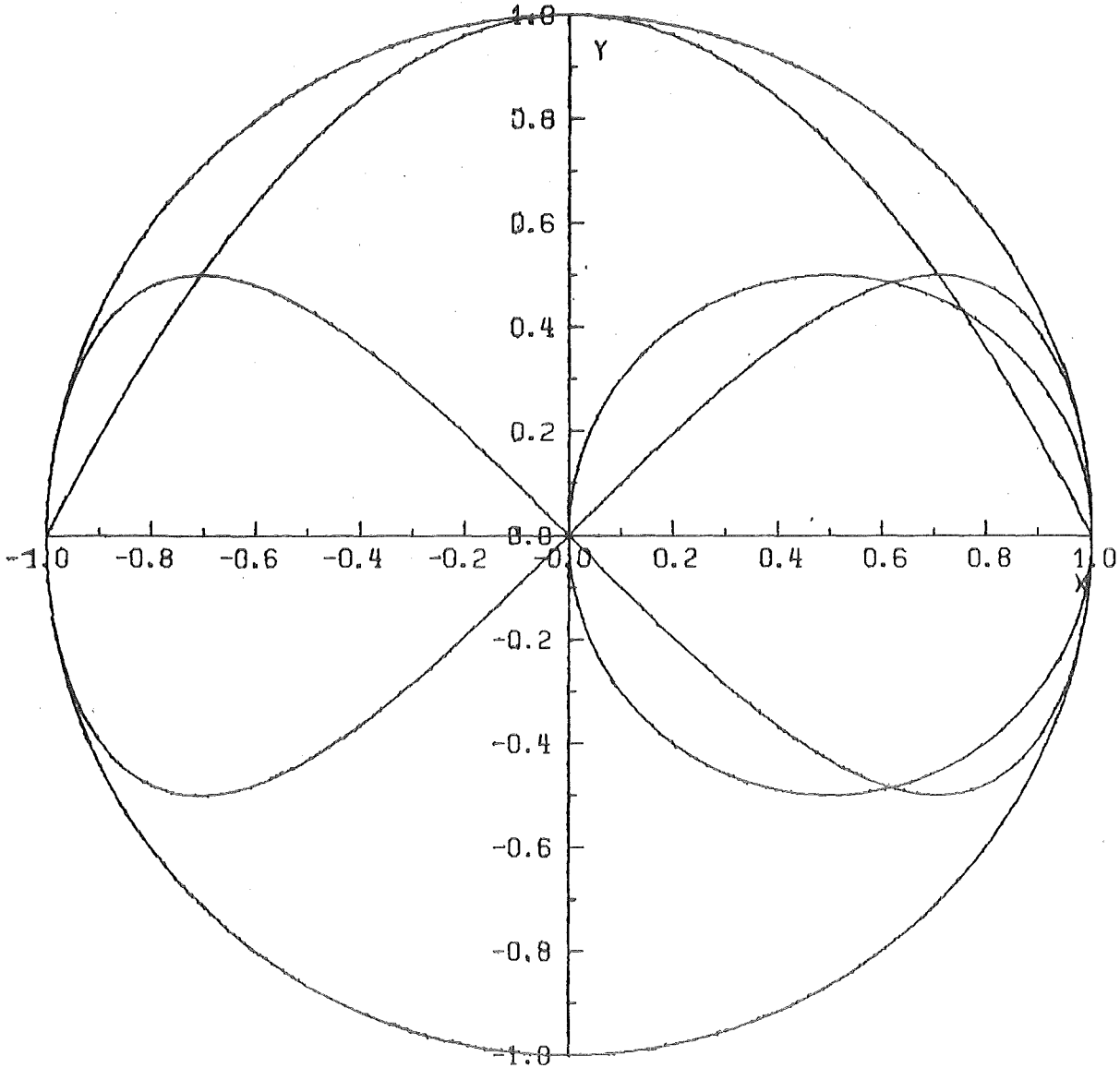
```
C TESTPROGRAMM (KUGEL-ZYLINDER-SCHNITTE - VIVIANI-FENSTER) 0000001
REAL W(722), X(722), Y(722), Z(722), TEXT(11) 0000002
REAL COMX1(5), COMY1(5), COMX2(5), COMY2(5) 0000003
REAL COMX3(5), COMY3(5), COMX4(5), COMY4(5) 0000004
DATA TEXT/'KUGE', 'L/ZY', 'LIND', 'ER-S', 'CHNI', 'TTE ', '- VI', 0000005
* DATA COMX1/'X=SI', 'N**2', '(PHI', ') 0000006
DATA COMX1/'X=SI', 'N**2', '(PHI', ') 0000007
DATA COMY1/'Y=SI', 'N(PH', 'I)*C', 'OS(P', 'HI) 0000008
DATA COMX2/'X=SI', 'N(PH', 'I)*C', 'OS(P', 'HI) 0000009
DATA COMY2/'Y=CO', 'S(PH', 'I) ', '2*' 0000010
DATA COMX3/'X=CO', 'S(PH', 'I) ', '2*' 0000011
DATA COMY3/'Y=SI', 'N(PH', 'I) ', '2*' 0000012
DATA COMX4/'X=SI', 'N**2', '(PHI', ') ', '2*' 0000013
DATA COMY4/'Y=CO', 'S(PH', 'I) ', '2*' 0000014
CONST=3.14159265/180. 0000015
DO 1 I=1, 722 0000016
PHI=(I-1)*.5*CONST 0000017
W(I)=SIN(PHI) 0000018
X(I)=W(I)**2 0000019
Z(I)=COS(PHI) 0000020
1 Y(I)=W(I)*Z(I) 0000021
CALL PLOTI(3,1024) 0000022
CALL CADRE(21.,29.7) 0000023
CALL SYMBP(3.85,2.0,.5,'Z',0.,1) 0000024
CALL SYMBP1(999.,999.,.5,'EICHNUNG ZU ',0.,12) 0000025
CALL SYMBP(999.,999.,.5,'P',0.,1) 0000026
CALL SYMBP1(999.,999.,.5,'ROGRAMMBEISPIEL ',0.,16) 0000027
CALL SYMBP(999.,999.,.5,'5.',0.,2) 0000028
CALL KFK(41,'IRB ') 0000029
CALL SYMBP(3.8,6.0,.4,TEXT,0.,42) 0000030
CALL PLOTP(3.,15.5,-3) 0000031
CALL AXLINI(15.,-1.,1.,.2,-1.,0.,1.,1.,.2,.25,1,4,' X') 0000032
CALL PLOTP(7.5,-7.5,-3) 0000033
CALL AYLINI(15.,-1.,1.,.2,-1.,0.,1.,1.,.2,.25,1,4,' Y ') 0000034
CALL PLOTP(-7.5,0.,-3) 0000035
CALL SYMBP(9.,18.,.4,COMY3,0.,10) 0000036
CALL SYMBP(9.,18.5,.4,COMX3,0.,10) 0000037
CALL SYMBP(1.5,18.,.4,COMY1,0.,19) 0000038
CALL SYMBP(1.5,18.5,.4,COMX1,0.,13) 0000039
CALL SYMBP(9.,16.5,.4,COMY4,0.,10) 0000040
CALL SYMBP(9.,17.,.4,COMX4,0.,13) 0000041
CALL SYMBP(1.5,16.5,.4,COMY2,0.,10) 0000042
CALL SYMBP(1.5,17.,.4,COMX2,0.,19) 0000043
CALL CURVE(Z,W,722,0.,.15,0) 0000044
CALL CURVE(X,Y,362,0.,.15,0) 0000045
CALL CURVE(Z,X,362,0.,.15,0) 0000046
CALL CURVE(Z,Y,722,0.,.15,0) 0000047
CALL PLOTP(0.,0.,999) 0000048
STOP 0000049
END 0000050
```

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

$$\begin{aligned} X &= \cos(\text{PHI}) \\ Y &= \sin(\text{PHI}) \end{aligned}$$

$$\begin{aligned} X &= \sin(\text{PHI}) * \cos(\text{PHI}) \\ Y &= \cos(\text{PHI}) \end{aligned}$$

$$\begin{aligned} X &= \sin^2(\text{PHI}) \\ Y &= \cos(\text{PHI}) \end{aligned}$$



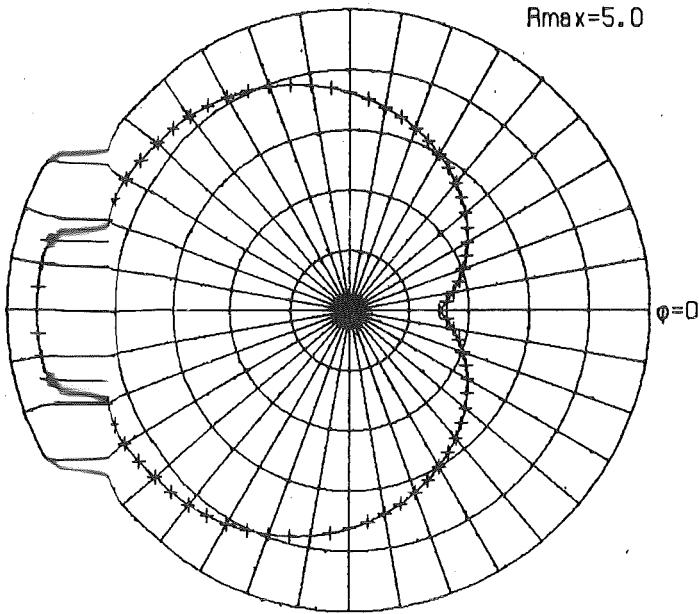
KUGEL/ZYLINDER-SCHNITTE - VIVIANI-FENSTER.



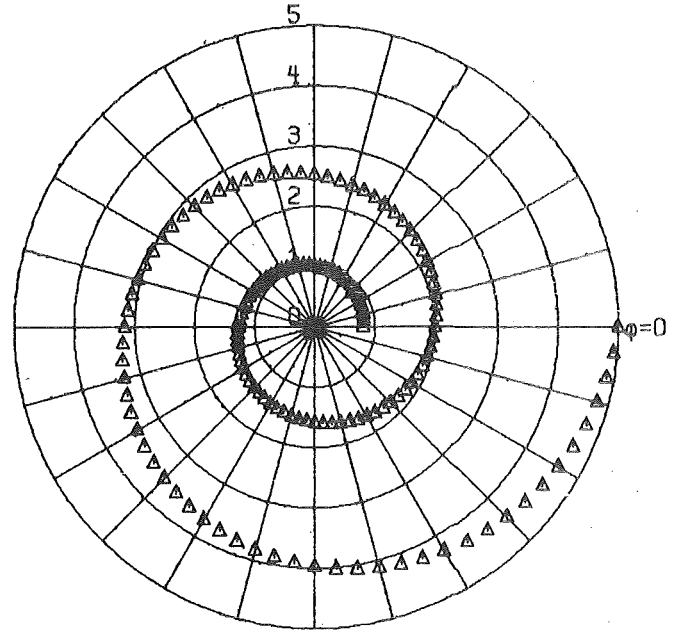
Zeichnung zu Programmbeispiel 5.

```
C TESTPROGRAMM (POLARKOORDINATEN) 0000001
  REAL T1(10)/'ARCH','IMED','ISCH','E SP','IRAL','E: R','=0.1', 0000002
+   '3888','8*PH','I' '/' 0000003
  REAL T2( 7)/'FUNK','TION',': R=','3.*S','IN(P','HI)+','1.5' '/' 0000004
  REAL T3(12)/'LOGA','RITH','MISC','HE S','PIRA','LE: ','R=2.', 0000005
+   '*E**','(0.0','0254','52*P','HI)' '/' 0000006
  REAL T4( 8)/'FUNK','TION',': R=','3.*S','IN(P','HI/2', 0000007
+   '.)+1','.5)' '/' 0000008
  REAL PHI(721), RAD(721) 0000009
  DATA CONST/.0174532/ 0000010
  CALL PLOTI(3,1024) 0000011
  CALL CADRE(21.,29.7) 0000012
  CALL SYMBP(3.85,2.0,.5,'Z',0.,1) 0000013
  CALL SYMBP1(999.,999.,.5,'EICHNUNG ZU ',0.,12) 0000014
  CALL SYMBP(999.,999.,.5,'P',0.,1) 0000015
  CALL SYMBP1(999.,999.,.5,'ROGRAMMBEISPIEL ',0.,16) 0000016
  CALL SYMBP(999.,999.,.5,'6.',0.,2) 0000017
  CALL KFK(41,'IRB ') 0000018
  CALL PLOTP(6.,11.5,-3) 0000019
  CALL POL(4.,5.,1.,.22,-12,0,37,T1) 0000020
  DO 10 I=1,361 0000021
  PHI(I)=I-1 0000022
10 RAD(I)=.0138888*PHI(I) 0000023
  CALL CURVEP(RAD,PHI,361,2.,.2,5) 0000024
  CALL PLOTP(9.5,0.,-3) 0000025
  CALL POL(4.,5.,1.,-.22,-12,1,27,T2) 0000026
  DO 20 I=1,361 0000027
  PHI(I)=I-1 0000028
20 RAD(I)=3.*SIN(PHI(I)*CONST)+1.5 0000029
  CALL CURVEP(RAD,PHI,361,2.,.2,0) 0000030
  CALL PLOTP(0.,11.,-3) 0000031
  CALL POL(4.,5.,1.,.22,-1,3,47,T3) 0000032
  DO 30 I=1,721 0000033
  PHI(I)=I-361 0000034
30 RAD(I)=2.*EXP(.0025452*PHI(I)) 0000035
  CALL CURVEP(RAD,PHI,721,2.,.2,-5) 0000036
  CALL PLOTP(-9.5,0.,-3) 0000037
  CALL POL(4.,5.,1.,-.22,-1,4,31,T4) 0000038
  DO 40 I=1,361 0000039
  PHI(I)=I-1 0000040
40 RAD(I)=3.*SIN(PHI(I)/2.*CONST)+1.5 0000041
  CALL CURVEP(RAD,PHI,361,3.,.2,5) 0000042
  CALL ROT90 0000043
  CALL PLOTP(0.,0.,999) 0000044
  STOP 0000045
  END 0000046
```

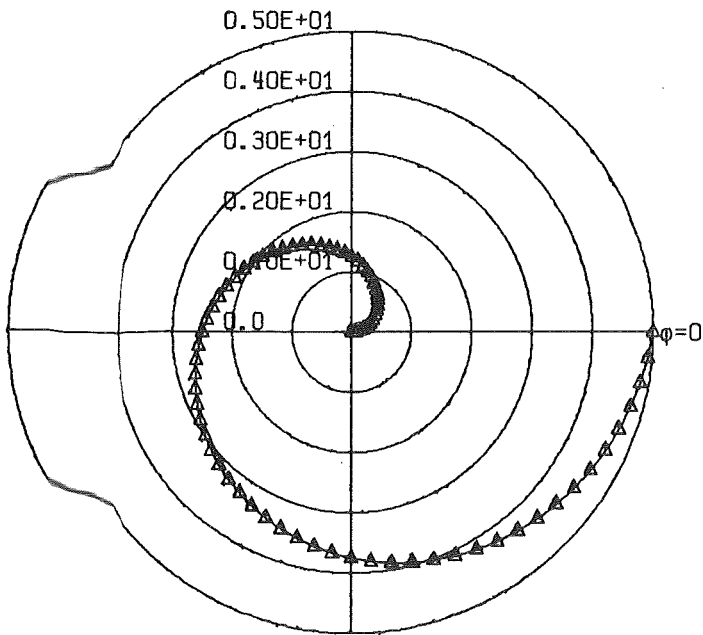
Programmbeispiel 6



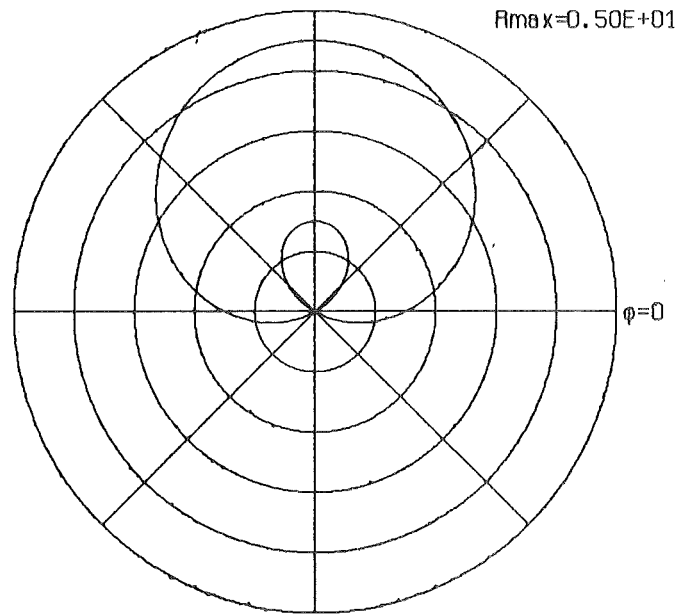
FUNKTION: $R=3 \cdot \sin(\Phi/2) + 1.5$



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



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



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