

Interner Bericht  
DESY F1-75/02  
April 1975

PSA/IPSA  
Programm zur statistischen Analyse  
und graphischen Darstellung physikalischer Daten

von

**DESY-Bibliothek**  
24. JUNI 1975

Volker Blobel



PSA/IPSA  
Programm zur statistischen Analyse  
und graphischen Darstellung physikalischer Daten

von

Volker Blobel  
Deutsches Elektronensynchrotron DESY und  
II. Institut fuer Experimentalphysik, Hamburg

Das mit der Prozedur PSAFORT verfügbare PSA-System erlaubt graphische Darstellungen von FORTRAN-Programmen aus. Durch Aufruf von Unterprogrammen sowie mit Hilfe von Steuerkarten wird ein Datensatz erzeugt, der zur graphischen Darstellung auf Display und Hardcopy-Geraet verwendet wird. Mit dem PSA-System koennen Histogramme und Mittelwerte gebildet sowie Rechnungen mit geordneten Folgen von Koordinatenpaaren durchgefuehrt werden.

INTERNER BERICHT

PSA/IPSA-MANUAL (VERSION 3.0)

DESY F1-75/2

Inhaltsverzeichnis

1. Einleitung . . . . .	5
A. BILDUNG VON DATENSTRINGS IN FORTRAN-PROGRAMMEN . . . . .	7
2. Prozedur und FORTRAN-Programm . . . . .	7
2.1 Die Prozedur PSAFORT . . . . .	7
2.2 FORTRAN-Programm . . . . .	8
2.3 Format der Steuerkarten . . . . .	8
2.4 PSA-Datensätze . . . . .	9
3. Bildung von Datenstrings . . . . .	9
3.1 Unterprogramm GEND . . . . .	9
3.2 Unterprogramm GENU . . . . .	10
3.3.1 Unterprogramm HIST . . . . .	10
3.3.2 Unterprogramm HIST mit ADDC . . . . .	10
3.4.1 Unterprogramm MEAN . . . . .	10
3.4.2 Unterprogramm MEAN mit ADDC . . . . .	10
3.5 Unterprogramm CORR . . . . .	11
3.6 Ergänzungen . . . . .	11
3.6.1 Druckerausgabe . . . . .	11
3.6.2 Speicherplatz-Überlauf . . . . .	11
3.6.3 Mehrfachaufruf . . . . .	11
4. Beispiele fuer die Bildung von Datenstrings . . . . .	12
5. Benutzung von Datenstrings . . . . .	13
5.1 Datenstring-Gruppen . . . . .	13
5.2 Ausstanzen von Datenstrings . . . . .	13
5.3 Einlesen von Datenstrings . . . . .	14
5.4 Zugriff zu einzelnen Datenstrings . . . . .	14
5.5 Standardanpassungen . . . . .	15
5.5.1 Orthogonale Polynome . . . . .	15
5.5.2 Exponential-Funktion . . . . .	15
5.5.3 Glaetten von Daten . . . . .	16
6. Unterprogramm SEQRD . . . . .	16
7. Benutzung des dynamisch organisierten Speichers . . . . .	17
7.1 Einlesen von Daten . . . . .	17
7.2 Einrichten von Blocken . . . . .	17
B. NICHT-INTERAKTIVE GRAPHISCHE AUSGABE . . . . .	19
8. Vorbereitung zur graphischen Ausgabe . . . . .	19
8.1 Darstellungsarten . . . . .	19
8.2 Zusammenstellung der Abbildungen . . . . .	19
8.3 Wahl des Abbildungsformats . . . . .	20
9. Beispiele . . . . .	20
10. IPS-Benutzung . . . . .	21
C. INTERAKTIVE GRAPHISCHE AUSGABE (IPSA) . . . . .	22
11. Starten und Beenden von IPSA . . . . .	22
12. Einfache Kommandos . . . . .	22
13. Kommandos fuer Fortgeschrittene . . . . .	23
14. Skalendefinition . . . . .	24
15. Auswahl von Datenstrings . . . . .	25
16. Definition von Textstrings . . . . .	25

DESY F1-75/2

## 1. EINLEITUNG

Grundlage des PSA-Systems sind Datenstrings. Eine Datenstring, abgekürzt DST, ist eine Folge von n-tupeln ( $n=2,3$  oder  $4$ )  $y_i, x_i [, dy_i [, dx_i ]]$ . Eine Datenstring ist eindeutig gekennzeichnet durch drei ganze Zahlen  $k_a, k_b$  und  $k_c$ . Die Koordinaten einer Datenstring bilden stets eine Einheit, sie koennen nur insgesamt graphisch dargestellt werden.

$(k_a, k_b, k_c)$  = Kennzeichnung einer Datenstring

$(k_a, k_b, *)$  = Menge aller Datenstrings mit gegebener Kennzeichnung  $k_a$  und  $k_b$  und beliebigem  $k_c$ .

$(k_a, *, *)$  = Menge aller Datenstrings mit gegebener Kennzeichnung  $k_a$  und beliebigen  $k_b, k_c$ .

Bei der graphischen Darstellung von Datenstrings sind die Elemente  $dy_i$  die Fehler von  $y_i$ , die Elemente  $dx_i$  die Breite der x-Intervalle zu  $x_i$ . Die Elemente  $dy_i$  und  $dx_i$  koennen in einer Datenstring fehlen, z.B. wenn die Datenstring die Menge der Punktepaare  $y_i, x_i$  einer theoretischen Kurve enthaelt. Wenn eine Datenstring ein Histogramm experimenteller Daten darstellt, sind alle Elemente  $y_i, x_i$  (=Intervallemitte),  $dy_i$  (=Fehler von  $y_i$ ) und  $dx_i$  (= Intervallbreite) vorhanden.

Alle Daten, die graphisch dargestellt werden sollen, muessen zunaechst in die Form von Datenstrings gebracht werden. Dies geschieht in einem normalen Batch-Job mit Hilfe der Prozedur PSAFORT.

Die Benutzung dieser Prozedur ist im Teil A erklart. Zur Bildung von Datenstrings gibt es eine Reihe von Unterprogrammen, unter anderem auch zur Bildung von Histogramm-Datenstrings. Alle wichtigen Regeln werden in den Kapiteln 2 und 3 erklart, in Kapitel 4 folgen Beispiele fuer die Bildung von Datenstrings. Beim ersten Lesen dieses Manuals koennen die Kapitel 5 bis 7 uebergangen werden, da in ihnen Anleitungen fuer fortgeschrittene Benutzer gegeben werden.

Die graphische Darstellung wird in den Teilen B und C erklart. Zur graphischen Darstellung wird einer Datenstring ein Symbol  $nsy$  zugeordnet, einen Ueberblick ueber die moeglichen Darstellungsarten gibt Abb.1. Es gibt eine Standard-Darstellung, bei der alle Datenstrings mit geringem Aufwand seitens der Benutzer graphisch dargestellt werden koennen. Bei der Standard-Darstellung werden in einer Abbildung jeweils alle Datenstrings  $(k_a, k_b, *)$  dargestellt, und zwar jede Datenstring mit Symbol  $nsy=k_c$ .

Die graphische Darstellung kann entweder bereits in dem Batch-Job vorbereitet werden oder aber interaktiv erfolgen. Die Vorbereitung der graphischen Ausgabe im Batch-Job wird in Teil B erklart. Durch Datenkarten gesteuert, wird ein Datensatz erzeugt, der die fertigen Abbildungen enthaelt. Im AUTO-Mode des IPS<sup>1)</sup> koennen die Abbildungen auf dem Display und Handcopy-Geraet dargestellt werden. Alternativ kann die graphische Ausgabe, wie in Teil C erklart, auch interaktiv gesteuert werden. Dazu wird der im Batch-Job erzeugte Datensatz mit den Datenstrings on-line benutzt.

Einige Teile des Programm-Systems PSA sind Weiterentwicklungen des Plot-Programms MIRO<sup>2)</sup>.

1) P.-K.Schilling, IPS User's Guide, Interner Bericht DESY R02-75/1.

2) H.Neumann, Plotprogramm MIRO, Interner pp-Memo, 1972.

DESY F1-75/2

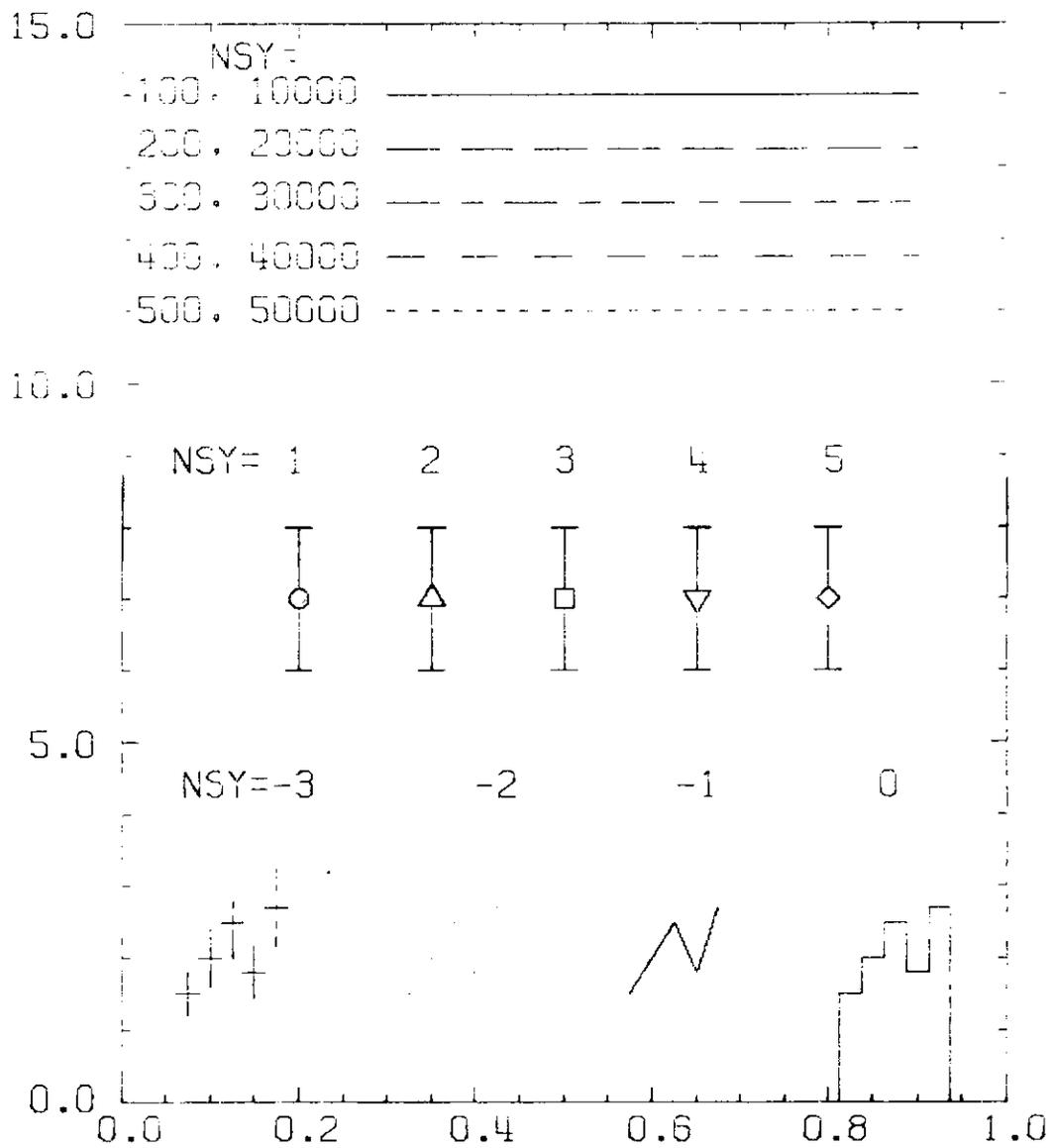


ABB.1 DARSTELLUNGSARTEN

A. BILDUNG VON DATENSTRINGS IN FORTRAN-PROGRAMMEN

2. PROZEDUR UND FORTRAN-PROGRAMM

2.1 Die Prozedur PSAFORT

Die Prozedur PSAFORT besteht aus acht Steps. Die ersten vier Steps FORT, PRINT, LKED und GO sind identisch mit den Steps der Prozedur FCLG. Die weiteren Steps SORT, PP, SS und AA dienen zum Sortieren, Listen und evtl. zur Vorbereitung der graphischen Ausgabe. Wenn die Datenstrings graphisch dargestellt werden sollen, muß entweder bei off-line Vorbereitung der graphischen Ausgabe (siehe Teil B) unter dem DD-Namen AA.PSAIPS ein permanenter Datensatz oder fuer die interaktive graphische Darstellung (siehe Teil C) unter dem DD-Namen PP.PSANEW ein permanenter Datensatz angegeben werden.

In der folgenden Aufstellung wird links die Minimalversion der JCL und rechts moegliche Erweiterungen (siehe auch 2.4.) angegeben. Die unter den DD-Namen PP.PSANEW und AA.PSAIPS angegebenen Datensatze koennen fehlen, wenn keine graphische Darstellung beabsichtigt ist. Der fuer PSA verfügbare Speicherplatz kann durch REGION-Angabe in der EXEC-Karte erweitert werden.

<u>Minimal-Version der JCL</u>	<u>Erweiterungen</u>
//...TOP ... // EXEC PSAFORT //FORT.SYSIN DD *	
FORTRAN-Programm (siehe 2.2.)	//LKED.SYSLIB DD // DD // DD // DD DSN=...,DISP=SHR //GO.PSAOLD DD DSN=... //GO.FIXED001 DD SYSOUT=B //GO.FIXED001 DD DSN=...
//GO.SYSIN DD *	
Steuerkarten (siehe 2.3.)	
//PP.PSANEW DD DSN=...,UNIT=FAST,DISP=(NEW,CATLG), fuer interaktive graphische Ausgabe // SPACE=(TRK,(5,1),RLSE),DCB=R002DCB.VRS	
	//PP.PSAOLD DD DSN=...
//AA.PSAIPS DD DSN=...,UNIT=FAST,DISP=(NEW,CATLG), fuer nicht-interaktive graphische Ausgabe // SPACE=(TRK,(5,1),RLSE),DCB=R002DCB.VRS	

DESY F1-75/2

2.2 FORTRAN-Programme

[CALL ACORE (nb)]	[50000+nb bytes Speicher werden freigelassen]
CALL PSAI	Einlesen der Steuerkarten
Statements zur Bildung von Datenstrings (siehe Kap.3)	
CALL PSAW	Ausschreiben aller gebildeten Datenstrings
STOP	
END	

PSA benutzt allen verfügbaren Speicherplatz bis auf 50000 bytes, die fuer IO-Puffer freigehalten werden. Durch ACORE-Benutzung (s.o.) kann der freigehaltene Speicherplatz vergrössert werden, wenn nach Programmstart CC 80A erfolgt ist. Durch PSAI werden sämtliche Steuerkarten eingelesen. Unterprogramme zur Bildung von Datenstrings sind in Kap. 3 erläutert. Datenstrings werden im Speicher aufgebaut; durch PSAW werden alle gebildeten Datenstrings auf einen internen Datensatz geschrieben und damit fuer das System verfügbar. Durch CALL PSWF (ka) werden alle Datenstrings (ka,\*,\*) ausgeschrieben.

2.3 Format der Steuerkarten

Sämtliche Steuerkarten in freiem Format. Karten beginnen mit einem Kartennamen aus nichtnumerischen Zeichen, es folgen Zahlen; zwischen Zahlen sind nichtnumerische Zeichen erlaubt; Fortsetzungskarten beginnen unmittelbar mit Zahlen.

Ganze Zahlen: ohne Punkt, mit oder ohne Vorzeichen

Reelle Zahlen: mit Punkt, mit oder ohne Vorzeichen und Exponentenfeld

Angabe von n gleichen Zahlen in der Form n\*Zahl möglich. Text hinter Hochkomma (') hinter der letzten Zahl. Steuerkarten werden von FORTRAN-Einheit 5 (SYSIN) gelesen.

Spezielle Steuerkarten:

UNIT iunit	Folgende Karten werden von FORTRAN-Einheit iunit bis Datenende gelesen, dann wieder von FORTRAN-Einheit 5.
POW	Folgende Karten werden gelistet.
POFF	Folgende Karten werden nicht gelistet.
ENDQ	Letzte Karte.

Beispiel:

```
DST 2 103 'COMMENT
BINS 103 1.5 , 0.2 3 *0.1 0.5E-01
ENDQ
```

2.4 PSA-Datensätze

Der unter dem DD-Namen AA.PSAIPS im letzten Step der Prozedur erzeugte Datensatz enthält Abbildungen, er ist ausschliesslich fuer die Ausgabe auf Display oder Plotter unter IPS zu benutzen. Alle gebildeten Datenstrings werden intern in Form von PSA-Records in einheitlichem Format verarbeitet. Datensätze mit diesen Records koennen in spaeteren Jobs wieder verarbeitet werden. Dazu sind DD-Karten notwendig.

Datensatz fuer neu gebildete Datenstrings (+DST aus GO.PSAOLD):

```
//PP.PSANEW DD DISP=(NEW,CATLG),DSN=...
```

Datensatz mit alten Datenstrings ausschliesslich fuer nicht-interaktive Herstellungen von Abbildungen:

```
//PP.PSAOLD DD DISP=OLD,DSN=...
```

Datensatz mit alten Datenstrings zur Benutzung im GO-Step:

```
//GO.PSAOLD DD DISP=OLD,DSN=...
```

Die FORTRAN-Einheiten 44 bis 47 werden im GO-Step vom System benutzt.

3. BILDUNG VON DATENSTRINGS

Datenstrings werden durch den Aufruf von Unterprogrammen im Speicher aufgebaut und durch CALL PSAW bzw. PSAF (siehe 2.3) ausgeschrieben.

ka= erstes Argument der Unterprogramme GEND,GENU,HIST,MEAN,CORR

kb= festgelegt durch CALL DEKB(kb) bis zur Aenderung (zu Beginn ist kb=0)

kc= Argument bei GEND,GENU; intern festgelegt bei HIST,MEAN,CORR

Begrenzungen: 1 ≤ ka ≤ 999  
 kb beliebig, jedoch wird kb > 0 empfohlen.  
 - 10 ≤ kc ≤ 60000

Festlegung der Kennzeichnung kb:

CALL DEKB (kb) legt den Wert kb fest fuer alle folgenden Aufrufe der UPr, bis zur Aenderung durch CALL DEKB (kb). Notwendige Steuerkarten fuer Bildungen von Datenstrings (ka,\*,\*):

```
DST ka ...'text
```

Der Text wird als Kommentar allen Datenstrings (ka,\*,\*) zugeordnet.

3.1 Unterprogramm GEND

Anwendung: Darstellung (eindeutiger) Funktionen, Darstellung von Daten, die in der Form yi, xi, [,dyi [,dyi]] vorliegen.

```
CALL GEND (ka,ko,y,x)           |
CALL GEND (ka,ko,y,x,dy)      |
CALL GEND (ka,ko,y,x,dy,dx)   |
DST ka 'text'                 | Steuerkarte
```

Anzahl Aufrufe (Koordinatenpaare) zu einer DST (ka,kb,ko) nicht begrenzt, jedoch intern in Records zu 120 Koordinatenpaaren, jeweils nach xi (aufsteigend) sortiert, gespeichert; spaetere Zugriff (siehe Kap.5) nur zum ersten der Records moeglich. Ausdruck in Listenform.

DESY F1-75/2

3.2 Unterprogramm GENU

Anwendung: Darstellung nichteindeutiger Funktionen (z.B. Kreisen). Aufrufe und Steuerkarten wie bei GEND. Im Unterschied zu GEND werden die Koordinatenpaare nicht sortiert.

3.3.1 Unterprogramm HIST

Anwendung: Bildung von Histogrammen (gewichtet, w=Gewicht)

```
CALL HIST (ka,x)      | w=1.0
CALL HIST (ka,x,w)   |
DST ka nx 'text'     | nx= Kennzeichnung ns der BINS-Karte
```

Intern festgelegt: ko=0

Definition von Intervall-Einteilungen durch Steuerkarten

```
BINS ns aw bw [nb] | aw=Aufangswert, bw=Intervallbreite
BINS ns aw - ew [nb] | ew=Endwert (ew>0), nb =Anzahl Intervalle (120)
BINS ns aw bw, bw,... | bw= einzelne Intervallbreite
```

Ausdruck als unnormalisierter X-Plot bei den beiden ersten BINS-Karten, als normalisierter X-Plot (Division durch Intervallbreite) bei letzter BINS-Karte.

3.3.2 Unterprogramm HIST mit ADDC

Bildung von Histogrammen fuer Intervalle einer Groesse z.

```
CALL ADDC (nz,z)      | Festlegung des Intervalls der Groesse z
CALL HIST (ka,x) bzw. | zur Intervalleinteilung nz
CALL HIST (ka,x,w)   |
```

Intern festgelegt: ko=Index des z-Intervalls.

Getrenntes Histogramm der Groesse x fuer jedes Intervall der Groesse z, bei ausserhalb der Intervallgrenzen liegender Groesse z wird der HIST-Aufruf ignoriert. Bei Normalisierung auch Division durch z-Intervallbreite.

3.4.1 Unterprogramm MEAN

Anwendung: Bildung von Mittelwerten (gewichtet, w= Gewicht)

```
CALL MEAN (ka,u)     | w=1.0
CALL MEAN (ka,u,w)   |
DST ka 'text'        | Steuerkarte
```

Intern festgelegt: ko=0 (da x nicht definiert, keine graphische Darstellung moeglich). Ausdruck in Listenform.

3.4.2 Unterprogramm MEAN mit ADDC

Anwendung: Bildung von Mittelwerten fuer Intervalle einer Groesse x.

```
CALL ADDC (nx,x)     | Festlegung des Intervalls der Groesse x
CALL MEAN (ka,u) bzw. | zur Intervalleinteilung nx
CALL MEAN (ka,u,w)   |
```

Intern festgelegt: ko=1 Ausdruck in Listenform

### 3.5 Unterprogramm CORR

Anwendung: Darstellung von Korrelationen zwischen y und x durch Punktdiagramm.

```
CALL CORR (ka,y,x)      | w=1.0
CALL CORR (ka,y,x,w)   |
DST ka [nx] [ny] 'text' | Steuerkarte
```

Intern festgelegt: kc=-2

Anzahl Aufrufe (Koordinatenpaare) zu einer Datenstring nicht begrenzt, intern in Records zu 32 Koordinatenpaaren, nicht sortiert. Bei graphischer Darstellung als Punktdiagramm Gewicht ohne Belang. Ausdruck als Korrelationsplot mit Beruecksichtigung des Gewichts; wenn nx bzw. ny nicht definiert, erfolgt automatische Wahl fuer Ausdruck, in y-Richtung 60 Intervalle. Zu den durch CORR gebildeten Datenstrings kann spaeter (siehe Kap.5) nicht zugegriffen werden.

### 3.6 Ergaenzungen

#### 3.6.1 Druckerausgabe

Steuerkarten fuer Ausdruck:

```
PRINT | gedruckt      werden Datenstrings, deren DST-Karte
NOPRINT | nicht gedruckt hinter dieser Karte liegen
PBINS | Intervalleinteilungen werden gelistet
```

Die durch die PRINT- bzw. NOPRINT-Karte getroffene Anweisung kann durch den folgenden Aufruf geaendert werden:

```
CALL CHLKA (ka,ia)      ia=0 nicht drucken
                        ia=1 drucken
```

Durch den Aufruf

```
CALL RVLKA (ka)
```

wird die alte Anweisung wieder hergestellt.

#### 3.6.2 Speicherplatz-Ueberlauf

Wenn der Speicherplatz zur Bildung der Datenstrings nicht ausreicht, werden die Datenstrings, deren DST-Karten am Ende liegen, nicht bzw. unvollstaendig gebildet.

#### 3.6.3 Mehrfachaufruf

An Stelle des UPr DEKB kann zur Angabe der Kennzeichnung kb fuer bestimmte Anwendungen das UPr MDEKB benutzt werden, das bestimmte logische Funktionen ausfuehrt und durch eine \*MULT-Karte gesteuert wird. Der Aufruf in Zusammenhang mit den UPr GEND, GENU, HIST, MEAN oder CORR, hier am Beispiel von UPr HIST erklart, muss in seiner Struktur dem angegebenen Beispiel folgen:

```
S1 CALL MDEKB (kb,mult,&S2) | kb>0
    CALL HIST (ka,x)
    GOTO S1
S2 ...
*MULT mult Liste von kb-Werten | Steuerkarte
```

DESY F1-75/2

Erste Anwendungsart: Die \*MULT-Karte enthaelt eine Liste von (positiven) kb-Werten. Stimmt der Wert von kb im Argument von MDEKB nicht mit einer der Werte der Liste ueberein, wird Return1 genommen (Sprung nach Statement-Nr. S<sub>1</sub>), andernfalls wird intern CALL DEKB(kb) gerufen und der normale Return genommen; nach Aufruf von HIST im Beispiel muss noch einmal nach Statementnr. S<sub>1</sub> gesprungen werden, da das endgueltige Verlassen der Schleife durch Return 1 aus MDEKB erfolgen muss.

Zweite Anwendungsart: Die \*MULT-Karte enthaelt eine Liste zugelassener positiver kb-Werte, davor eine negative Zahl. Falls der Wert kb im Aufruf von MDEKB in der Liste vorhanden ist, wird intern CALL DEKB mit der negativen Zahl als Argument gerufen und der normale Return genommen. Auf diese Weise kann ein Histogramm mit verschiedenen urspruenglichen kb-Werten gebildet werden.

Die erste und zweite Anwendungsart koennen durch entsprechende Listen kombiniert werden. Dazu muss die \*MULT-Karte als erstes die Liste der positiven kb-Werte zur ersten Anwendungsart enthalten, anschliessend eine oder mehrere der Listen zur zweiten Anwendungsart, die jeweils mit einer negativen Zahl beginnen.

4. BEISPIELE FUER DIE BILDUNG VON DATENSTRINGS

- a) Die sin- und cos-Funktionen sollen im Bereich von 0 bis 2 PI graphisch dargestellt werden. Es werden zwei Datenstrings mit je 101 Punkten jeder Funktion gebildet

```

PI=3.141593
DO 10 I=1,101
X=PI*FLOAT(I-1)/50.0
CALL GEND(1,100,SIN(X),X)
10 CALL GEND(1,200,COS(X),X)
    
```

DST 1 'SIN- UND COS-FUNKTION

- b) Ein Kreis mit Radius R=2.0 soll gezeichnet werden. Eine Datenstring mit 101 Punkten wird gebildet.

```

R=2.0
PI=3.141593
DO 20 I=1,101
PHI=PI*FLOAT(I-1)/50.0
20 CALL GENU(2,100,R*SIN(PHI),R*COS(PHI))
    
```

DST 2 'KREIS MIT RADIUS R=2.0

- c) Ein Benutzer berechnet in seinem Programm zu experimentellen Werten Y(I)=DY(I) bei Abszissen X(I), I=1,N, eine theoretische Kurve durch Aufruf der Funktionen FUNCT, F=FUNCT(X(I)). Die experimentellen Daten und die Kurve sollen in einer Abbildung dargestellt werden. Im folgenden Beispiel werden zwei Datenstrings gebildet, eine mit den experimentellen Daten und eine mit den Punkten der theoretischen Kurve.

```

DO 30 I=1,N
CALL GEND(3,1,Y(I),X(I),DY(I))
F=FUNCT(X(I))
30 CALL GEND(3,100,F,X(I))
    
```

DST 3 ' EXPERIMENTELLE PUNKTE UND THEORETISCHE KURVE

d) Ein Benutzer liest in seinem Programm experimentelle Werte fuer Impulse PA und PB ein. Es sollen Histogramme der Impulse PA und PB und ein Korrelationsplot PA gegen PB hergestellt werden. Drei Datenstrings (zwei Histogramme, ein Korrelationsplot) werden gebildet.

```

40 READ( ) ...,PA,PB,...
C
  CALL HIST(4,PA)
  CALL HIST(5,PB)
  CALL CORR(6,PA,PB)
  GOTO 40

DST 4 101 'HISTOGRAMM VON PA
DST 5 102 'HISTOGRAMM VON PB
DST 6     'PA VERSUS PB
BINS 101 0.0 0.01
BINS 102 0.0 0.02
    
```

e) Erweiterung: Der Benutzer liest zusaeztzlich zu den Impulsen noch eine Run-Nummer IRUN ein. Die Histogramme sollen fuer jeden Run getrennt gebildet werden, der Korrelationsplot jedoch fuer alle Runs gemeinsam.

```

40 READ( ) ..., IRUN,PA,PB,...
C
  CALL DEKB(IRUN)
  CALL HIST(4,PA)
  CALL HIST(5,PB)
  CALL DEKB(0)
  CALL CORR(6,PA,PB)
    
```

## 5. BENUTZUNG VON DATENSTRINGS

### 5.1 Datenstring-Gruppen

Zur Benutzung der in diesem Kapitel beschriebenen Programme muessen Datenstrings durch Steuerkarten zu Datenstring-Gruppen zusammengefasst werden.

DSTA 1g ka ka ...		(ka,*,*) bilden die
DSTB 1g ka kb, ka kb...		(ka,kb,*) Datenstring-Gruppe 1g
DSTC 1g ka kb kc, ka kb kc ...		(ka,kb,kc)

### 5.2 Ausstanzen von Datenstrings

Alle Arten von Datenstrings ausser durch CORR gebildete Datenstrings koennen ausgestanzt werden, bei den durch GEND oder GENU gebildeten Datenstrings werden nur die jeweils ersten 120 Koordinatenpaare gestanzt. Zum Ausstanzen ist eine Steuerkarte und der Aufruf von RPUNCH notwendig, sowie die Definition des Datensatzes

GO.FT07F001 als SYSOUT=B

PUNCH 1g		Steuerkarten zum Ausstanzen der Datenstringgruppe 1g
CALL RPUNCH		Aufruf

Datenstrings in Lochkartenform koennen durch PSAI in einem folgenden Job wieder gelesen werden.

DESY F1-75/2

5.3 Einlesen von Datenstrings

In einem fruheren Job gebildete Datenstrings koennen eingelese werden, bei Datenstrings in Lochkartenform durch PSAI, bei Datenstrings auf anderen Datent-raegern durch Definition des Datensatzes GO.PSAOLD, siehe Kap. 2.4. Von den eingelesenen Datenstrings koennen bestimmte Datenstrings durch Steuerkarten entfernt bzw. gelistet werden.

DELETE ig | Datenstrings der werden entfernt  
 PRINT ig | Gruppe ig werden gelistet

5.4 Zugriff zu einzelnen Datenstrings

Zum Zugriff zu einzelnen Datenstrings z.B. fuer Anpassungsrechnungen stehen drei Unterprogramme zur Verfuegung. Zunaechst muessen die Datenstrings einer Datenstringgruppe ausgewaehlt werden.

CALL SDST(ig,&S<sub>1</sub>) | Auswahl der Datenstrings der Gruppe ig

Wenn keine Datenstrings der Gruppe vorhanden sind, erfolgt Return 1 (Sprung nach Statement-Nr. S<sub>1</sub>).

Anschliessend kann durch CALL NDST (&S<sub>2</sub>) bzw. durch CALL GDST (ka,kb,kc) eine Datenstring umgespeichert werden in den

COMMON/FDST/N,NF,NL,NKA,NKB,NKC,Y(120),X(120),DY(120),DX(120),Z,DZ

N Anzahl Werte

NF = 1 bzw. erster Index I mit DY(I)≠0 wenn DY im

NL = N bzw. letzter Index I mit DY (I) ≠0 Datenstring enthalten.

NKA,NKB,NKC Kennzeichnung der Datenstring

Y(120),X(120),DY(120),DX(120) Koordinaten der Datenstring

Z,DZ Koordinate und Intervallbreite der zusaetzlichen Koordinafe bei Bildung der Datenstring durch HIST mit ADDC.

Wenn in der Datenstring einzelne Werte DY(I) gleich null sind, werden diese Werte beim Umspeichern auf den mittleren Wert benachbarter Werte gesetzt.

CALL NDST (&S<sub>2</sub>) | Umspeichern der naechsten Datenstring

Nach Umspeichern der Datenstring erfolgt normaler Return; sind keine Datenstrings mehr vorhanden, erfolgt Return 1 (Sprung nach Statementnr. S<sub>2</sub>). Bei Benutzung einer DSTA- oder DSTB-Karte zur Definition der Datenstring-Gruppe erfolgt das Umspeichern in der Reihenfolge der Datenstrings auf dem internen Datensatz. Bei Benutzung der DSTC-Karte erfolgt das Umspeichern in der auf der Karte angegebenen Reihenfolge; wenn eine angegebene String nicht vorhanden ist, erfolgt normaler Return mit N=0.

CALL GDST (ka,kb,kc) | Umspeichern der angegebenen Datenstring

Bei diesem Unterprogramm wird die Kennzeichnung der Datenstring als Argument angegeben. Wenn die angegebene Datenstring nicht vorhanden ist, erfolgt normaler Return mit N=0.

5.5 Standardanpassungen

Durch den Aufruf bestimmter Unterprogramme und die Benutzung von Steuerkarten werden Anpassungen an die Datenstrings von Datenstring-Gruppen durchgefuehrt. Die Anpassungen bilden neue Datenstrings mit gleicher Kennzeichnung ka und kb und einem neuen, durch FUNCTION NEWKC(ko) definierten Wert ko.

ko (new) = 100 fuer ko(old) ≤ 0  
 = n\*100 fuer ko(old) = n (1≤n<100)  
 = k\*100 fuer ko(old) = k\*100 (1≤k<100)

5.5.1 Orthogonale Polynome

POLY ig ipr np xl xh test fac [sfac] | Steuerkarte  
 CALL RPOLY | Aufruf

Parameter der Steuerkarte:

- ig = Nr. der Datenstring-Gruppe
- ipr = 0 kein Ausdruck, =1 Parameterausdruck, =2 Parameter- und Listenausdruck
- np = Anzahl Parameter (1≤np≤21), Grad des Polynoms (np-1)
- xl,xh = untere und obere Grenzen fuer x, wenn xl=xh=0, erfolgt Anpassung im gesamten x-Bereich.
- test = 0 Polynom genau (np-1)-Grades  
 >0 Benutzung des x<sup>2</sup>-Tests zur Reduktion des Polynom-Grades um nichtsignifikante Glieder, z.B. fuer 95% Confidenz-Level test=3.84
- test <0 Benutzung des F-Tests zur Reduktion des Polynom-Grades um nichtsignifikante Glieder fuer 95% Confidenz-Level.
- fac = 0 Datenstring mit Polynom wird gebildet  
 ≠0 Zwei Datenstrings mit Polynom=fac\*Standardabweichungen des Polynoms werden gebildet.
- sfac>0 Datenstring mit ko=-1 (polygonzug) mit Polynom + sfac\*Sexp wird gebildet  
 mit Sexp=SQRT(fit\*DY(I)<sup>2</sup>/Y(I)) fuer SQRT(DY(I)/Y(I)) >3

5.5.2 Exponential-Funktion

EXPF ig ipr np xl xh fac | Steuerkarte  
 CALL REXP | Aufruf

Eine Exponentialfunktion  $f(x)=a_1 \exp(a_2x+a_3x^2)$  wird an die Datenstrings angepasst.

DESY F1-75/2

Parameter der Steuerkarte:

lg = Nr der Datenstringgruppe

ipr = 0 kein Ausdruck, =1 Parameterausdruck,  
= 2 Parameter- und Listenausdruck

np = Anzahl der Parameter, np=2 oder 3.

xl,xh = untere und obere Grenzen fuer x;  
wenn xl=xh=0, erfolgt Anpassung im gesamten x-Bereich.

fac = 0 Datenstring mit Funktion wird gebildet

# 0 Zwei Datenstrings mit Funktion=fac•Standardabweichungen der Funktion werden gebildet.

### 5.5.3 Glätten von Daten

Datenstrings mit den geglätteten Werten der urspruenglichen Datenstrings werden gebildet.

SMTH lg | Steuerkarte mit Angabe der Datenstring-Gruppe  
CALL RSMTH | Aufruf

## 6. UNTERPROGRAMM SEQRD

Fuer Anwendungen, bei denen jeweils ein "event" von einem Datensatz gelesen wird, und fuer jedes "event" gleichartige Rechnungen durchgefuehrt werden, gibt es das Programm SEQRD:

S<sub>1</sub> CALL UREAD  
bei Datenende: Sprung nach S<sub>2</sub>  
CALL DIST  
bei Erfuellung eines Abbruchkriteriums: Sprung nach S<sub>2</sub>  
goto S<sub>1</sub>

S<sub>2</sub> CALL DSTEND  
Return

Benutzer-Programme:

SUBR. UREAD (\*,\*) Return 1 bei Lesefehler, Return 2 bei Datenende.

Eine Standard-Version liest von FT08F001 mit Liste

NTOT, (RV(I), I=1, NTOT) in den Common/CUREAD/NTOT, RV(2000).

SUBROUTINE DIST Rechnungen fuer jedes "event"

SUBROUTINE DSTEND Rechnungen bei Datenende

Steuerkarten fuer vorzeitigen Abbruch:

TIME	nsec		nsec Sekunden vor TIMEOVFL [2]
LIMIT	lim		Abbruch nach lim "events" [unendlich]
ERRORS	nr		nach nr Lesefehlern [5]

7. BENUTZUNG DES DYNAMISCH ORGANISIERTEN SPEICHERS

Vom Programmsystem PSA werden alle eingelesenen Daten sowie die durch Unterprogramm-Aufruf uebermittelten Daten in einem grossen dynamisch organisierten Common blockweise gespeichert.

```
COMMON/DCS/IW(1)
REAL RW(1)
EQUIVALENCE (IW(1),RW(1))
```

Der Benutzer kann diesen Common auch zur blockweisen Speicherung eigener Daten benutzen.

7.1 Einlesen von Daten

Der Benutzer kann eigene Daten in Lochkartenform vom Programm PSAI einlesen lassen.

name a b c ...		name aus nichtnumerischen Zeichen;
\$name n a b c ...		n beliebige ganze Zahl; a,b,c ...
*name n a b c ...		beliebige ganze Zahlen oder Floating-point Zahlen

Die in den Lochkarten stehende Information wird blockweise gespeichert. Den Index I des Blocks erhaelt man durch:

```
CALL BLOK (I, 'name',k,*) | 'name'
CALL BLOK (I, '$name',n,*) | '*name'  enthaelt 4 Zeichen
CALL BLOK (I, '*name',n,*) | '*name'
```

Bei einem nicht mit \$ oder \* beginnenden Namen wird die Information beim ersten Auftreten mit k=0 abgespeichert, bei wiederholtem Auftreten des gleichen Namens mit k=1,2...usw.

iw(I) enthaelt die Anzahl der eingelesenen Werte, in iw(I+1) steht die eingelesene Zahl a, in iw(I+2) steht b usw. Return1 erfolgt, wenn der verlangte Block nicht vorhanden ist. Da die Bloecke u.U. verschoben werden koennen, muss bei jeder Benutzung der Daten der Index I neu beschafft werden.

7.2 Einrichten von Bloecken

Durch den folgenden Aufruf kann der Benutzer einen eigenen durch na und nc gekennzeichneten Block einrichten:

```
CALL BGET (I,na,nc,nwords,&S,IER)
```

Ein Block der Laenge nwords mit der Kennzeichnung na,nc wird eingerichtet, der Index ist I; na='name' nc=ganze Zahl.

Return1 bei folgenden Bedingungen (Sprung nach Statementnr. S):

- IER=0 Block schon vorhanden
- =-1 Block wurde eingerichtet, jedoch mussten dazu bestehende Bloecke verschoben werden. Der Benutzer muss dann CALL REIND rufen, um einwandfreies Arbeiten des PSA-Systems zu erreichen.
- >0 IER Werte fehlen, d.h. Block war nicht einzurichten.

Nach BGET sind alle Worte eines neu eingerichteten Blocks gleich Null gesetzt.

DESY F1-75/2

Weitere Aufrufe:

CALL BFRE (na,nb,nd,IER)

Die Blöcke mit Kennzeichnung na und nb (nb&lt;na&lt;nd) werden entfernt.

CALL BLST(na,nb,nd)

Die Blöcke mit Kennzeichnung na und nb (nb&lt;na&lt;nd) werden gelistet.

CALL BLOK (I,na,nb,&amp;S)

Index I zu durch na und nb gekennzeichneten Block. Return1 (Sprung nach Statementnr. S), wenn Block nicht vorhanden.

B. NICHT-INTERAKTIVE GRAPHISCHE AUSGABE

8. VORBEREITUNG ZUR GRAPHISCHEN AUSGABE

Die graphische Ausgabe wird durch Steuerkarten vorbereitet, die mit den anderen Steuerkarten vom Programm PSAI (siehe 2.2) eingelesen werden. Mit Hilfe der Steuerkarten wird aus den Datenstrings unter dem DD-Namen AA.PSAIPS ein Datensatz mit den fertigen Abbildungen hergestellt.

8.1 Darstellungsarten

Die Art der graphischen Darstellung einer Datenstring wird durch die zugeordnete Groesse nsy bestimmt. Abb.1 gibt einen Ueberblick ueber die Darstellungsarten.

Bei nsy=100,200 etc. werden Kurven exakt durch aufeinanderfolgende Punkte gelegt, bei nsy=10000,20000 etc. entsprechend den Fehlern geglaettet. Positive Werte nsy werden jeweils modulo 5 bzw. 500 bzw. modulo 50000 genommen, d.h. nsy=6 entspricht nsy=1, nsy=600 entspricht nsy=100 und nsy=60000 entspricht nsy=10000. Positive Werte von nsy koennen addiert werden, d.h. bei nsy=101 werden Kreise (nsy=1) und durchgezogene Kurven (nsy=100) gezeichnet. Bei nsy<-3 erfolgt keine Darstellung.

8.2 Zusammenstellung der Abbildungen

Die Zusammenstellung der Datenstrings zu Abbildungen erfolgt durch Steuerkarten. Standard-Darstellung:

ALL ka, kb,

Saemtliche Datenstrings (ka,\*,\*) mit ka,skaska<sub>2</sub> werden dargestellt, in einer Abbildung jeweils alle DST (ka,kb,\*) mit Symbol nsy:=kc.

Nicht-Standard-Darstellung: Steuerkarten jeweils fuer eine Abbildung

PICT ka kb | DST (ka,kb,\*) mit nsy:=kc  
 PICT ka kb ifac | DST (ka,kb,\*) mit nsy:=ifac\*kc

PICT ka,kb (kc,nsy) (kc,nsy)... | DST(ka,kb,kc)  
 | mit nsy, restliche DST (ka,kb,\*) nicht.

PICT ka,kb (kc,nsy) (kc,nsy)...ifac | DST (ka,kb,kc)  
 | mit nsy, restliche DST (ka,kb,\*)  
 | mit nsy=ifac\*kc

Darstellung weiterer DST in einer Abbildung durch der PICT-Karte folgende AND-Karte mit entsprechenden Angaben wie bei PICT-Karten. Datenstrings duerfen fuer mehr als eine Abbildung benutzt werden.

DESY F1-75/2

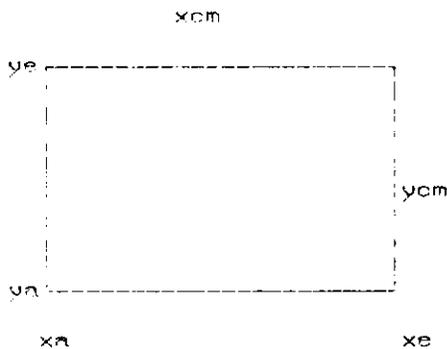
8.3 Wahl des Abbildungsformats

Zuordnung von Text zur folgenden PICT-Karte durch

COMM text\* |

Definition von Grosse und Skalen zu den folgenden PICT-Karten bzw. ALL-Karten bis zur Aenderung durch:

SIZE ycm xcm	wahlweise	COMB ycm xcm ya ye ly xa xe
SC Y ya ye ly		
SC X xa xe		



ly=0 oder 1: lineare y-Skala  
ly=2: logarithmische y-Skala

Bei ya=ye bzw. xa=xe erfolgt automatische Wahl der Skalen.  
Default-Werte:

COMB 10.0 15.0 0.0 0.0 1 0.0 0.0

9. BEISPIELE

In diesem Kapitel wird gezeigt, welche Steuerkarte fuer die graphische Darstellung der in den Beispielen in Kap.4 gebildeten Datenstrings notwendig sind.

a) Alle Datenstrings sollen in der Standard-Darstellung graphisch dargestellt werden.

ALL

Da das Abbildungsformat nicht definiert wurde, werden die Standard-Werte (Grosse 10 mal 15 cm, lineare Skalen, automatische Wahl der Anfangs- und Endwerte) benutzt.

b) Die sin- und cos-Funktion sollen im Format 6 mal 20 cm in getrennten Abbildungen durch durchgezogene Kurven dargestellt werden. Hier werden die Skalenanfangs- und Endwerte fest vorgegeben.

```
SIZE 6.0 20.0
SC Y -1.0 +1.0
SC X 0.0 7.0
COMM SIN-FUNKTION
PICT 1 0 100 100
COMM COS-FUNKTION
PICT 1 0 200 100
```

C) Die in Beispiel c von Kap.4 gebildeten Datenstrings sollen mit logarithmischer y-Skala dargestellt werden, die Wahl der Anfangs- und Endwerte soll automatisch erfolgen.

```
COMB 10.0 15.0 0.0 0.0 2 0.0 0.0
ALL 3 3
```

D) Die theoretische Kurve im vorhergehenden Beispiel soll gestrichelt gezeichnet werden.

```
COMB 10.0 15.0 0.0 0.0 2 0.0 0.0
PICT 3 0 1 1
AND 3 0 100 500
```

E) In einer Abbildung sollen die Histogramme von PA fuer Runs mit IRUN=1 und =2 dargestellt werden, das Histogramm zu IRUN=1 in Histogrammform (nsy=0), das Histogramm zu IRUN=2 durch Balken (nsy=-3).

```
COMB 10.0 15.0 0.0 0.0 0 0.0 0.0
PICT 4 1 0 0
AND 4 2 0 -3
```

10. IPS-BENUTZUNG

An der IPS-Tastatur wird die Benutzung durch folgende Befehle gestartet:

```
!!ON Username
!!GO IPS
AUTO, dsn
```

Als dsn ist ein vorher unter dem DD-Namen AA.PSAIPS erzeugter Datensatz anzugeben; nach AUTO, dsn erfolgt die Anzeige PLOT NR: 1 auf dem Display. Folgende Befehle stehen zur Verfügung:

- D | Darstellung der Abbildungen auf dem Display
- H | Handcopy der Abbildungen auf Plotter (vorher Einschalten)
- S | Skip zur naechsten Abbildung
- X | Darstellung der naechsten Abbildung auf dem Display
- H,n | Handcopy der n folgenden Abbildungen
- S,n | Skip n Abbildungen
- X,n | Skip n Abbildungen und Darstellung der folgenden Abbildung
- L | Beendigung. Durch AUTO kann zur ersten Abbildung auf dem gleichen Datensatz zurueckgegangen werden.

Beendigung der IPS Benutzung durch

```
!! OFF
```

DESY F1-75/2

C. INTERAKTIVE GRAPHISCHE AUSGABE (IPSA)

## 11. STARTEN UND BEENDEN VON IPSA

Beim interaktiven PSA -IPSA- erfolgt die Zusammenstellung der Abbildungen mit Skalenwahl etc. interaktiv. Voraussetzung ist die Existenz eines Datensatzes mit Datenstrings, die in einem Batch-Job unter Benutzung der Prozedur PSAFORT erzeugt wurde. Dazu muss im Batch-Job unter dem DD-Namen PP.PSANEW (nicht AA.PSAIPS) ein permanenter Datensatz angegeben werden. Im Batch-Job sollten alle Steuerkarten, die unmittelbar der Herstellung von Abbildungen dienen (ALL, COMB, SIZE, SCY, SCX, PICT, AND), fehlen.

Die Kommunikation des Benutzers mit dem Programm erfolgt durch Kommandos, die auf der Tastatur geschrieben werden und durch gleichzeitiges Druecken der Tasten CTRL und EOT, in dieser Beschreibung durch (EOT) abgekuerzt, zur IBM geschickt werden.

Das interaktive PSA wird nach dem Kommando

(1) !!ON username (EOT)

gestartet durch

(2) !!GO IPSA (EOT)

Nach dem Start erscheint die Zeile

START OF IPSA,GIVE DSN

auf dem Display. Der Name des Datensatzes mit Datenstrings muss angegeben werden durch das Kommando

(3) DSN 'dsname (EOT) ('=Hochkomma, durch Druecken der Tasten SHIFT und ?)

Danach erscheint auf dem Display eine Fehlermeldung oder die Zeile

WAIT FOR TABLE OF CONTENT

und spaeter ein Inhaltsverzeichnis des Datensatzes, jeweils uebereinander die Kennzeichnungen ka, kb, kc der Datenstrings. Beendet wird IPSA durch die Kommandos

(4) E (EOT)

(5) !!OFF (EOT)

## 12. EINFACHE KOMMANDOS

Durch einfache Kommandos koennen nacheinander alle Datenstrings auf dem Display bzw. Hardcopy-Geraet sichtbar gemacht werden.

(6) X (EOT) | display next datastring(s)  
 (7) R (EOT) | repeat display (identical)  
 (8) H (EOT) | make hardcopy of last display

Die Kapitel 13 bis 16 beziehen sich auf Kommandos fuer Fortgeschrittene, der Anfenger sollte als erste weitere Kommandos die in Kapitel 13 und 14 beschriebenen ausprobieren.

## 13. KOMMANDOS FUER FORTGESCHRITTENE

Durch Kommandos koennen

1. die Skalen definiert werden,
2. Datenstrings fuer die naechste Abbildung gewaehlt werden,
3. Text der Abbildung zugeordnet werden.

Dies wird in den Kapiteln 14 bis 16 erlaeutert.

Es gibt folgende Kurz-Kommandos ohne Parameter:

(9) D (EOT)

Die gewaehnten Datenstrings (s. Kapitel 15 und X-Kommando) werden zusammen mit den definierten Textstrings (s. Kapitel 16) und den definierten Skalen (s. Kapitel 14) dargestellt auf dem Display.

(10) R (EOT)

Der letzte Display wird identisch wiederholt.

(11) H (EOT)

Eine Handcopy des letzten Display wird gemacht.

(12) X (EOT)

Die auf den letzten gewaehnten Datenstring folgende(n) Datenstring(s) werden gewaehlt und auf dem Display dargestellt.

(13) T (EOT)

Ein Inhaltsverzeichnis des Datensatzes mit Angabe der gewaehnten Datenstring wird auf dem Display dargestellt.

Die weiteren Kommandos enthalten z.T. als Parameter Zahlen und Text. Die Kommandos bestehen aus einem aus nicht numerischen Zeichen bestehenden Namen, es folgen Zahlen als zugehoerige Parameter, am Schluss kann Text hinter einem Hochkomma (') stehen. Zahlen koennen mit oder ohne Punkt geschrieben werden, bei Benutzung des E-Formats muss die vor dem E stehende Zahl einen Punkt enthalten, die hinter dem E stehende Zahl darf keinen Punkt enthalten. n gleiche Zahlen koennen in der Form n \* Zahl geschrieben werden. Nicht geschriebene Zahlenparameter werden wie 0 behandelt.

DESY F1-75/2

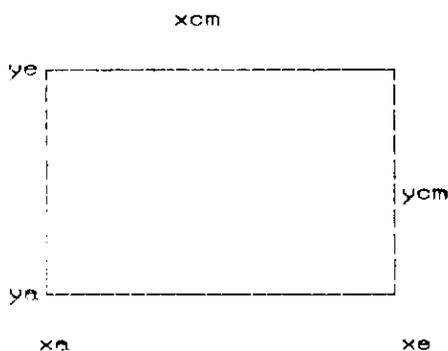
14. SKALENDEFINITION

Die Skalen werden durch eine Reihe von Parametern definiert. Nach jeder Neudefinition durch ein Kommando erscheint auf dem Display eine Parameteruebersicht in der Form:

```

Y ycm ya ye LIN ystep GRID
  LOG          NOGRID

X xcm xa xe LIN Xstep SF=sfac
  LOG
    
```



```

ly=0 oder 1: y-Skala linear
   =2      : y-Skala logarithmisch

lx=0 oder 1: x-Skala linear
   =2      : x-Skala logarithmisch
    
```

NOGRID/GRID: keine Querstriche/Querstriche

sfac: Faktor fuer Veraenderung der Standardsymbolgroesse

ystep, xstep: Schnittweite bei linearer Skala fuer Zahlenwerte, wird bei 0 automatisch gewaehlt.

Kommandos:

- (14) SIZE ycm xcm (EOT)
- (15) S Y ya ye ly ystep (EOT)
- (16) S X xa xe lx xstep (EOT)
- (17) GRID (EOT)
- (18) NOGRID (EOT)
- (19) SF sfac (EOT)

Unsinnig angegebene Werte werden vom Programm, soweit moeglich, korrigiert. Bei ya=ye bzw. xa=xe werden die Skalen entsprechend den Daten automatisch gewaehlt. Ebenso wird ystep und xstep automatisch gewaehlt, wenn ystep=0 bzw. xstep=0 ist.

Beispiele:

```

SIZE 10 10
CY 0 0 2
SX 0 1
    
```

Eine Abbildung der Groesse 10\*10 cm mit automatisch gewaehlter logarithmischer y-Skala und linearer x-Skala von 0 bis 1 wird gewaehlt.

Die Groessenangaben im SIZE-Kommando beziehen sich auf das Handcopy-Geraet, fuer die Display-Darstellung wird verkleinert.

## 15. AUSWAHL VON DATENSTRINGS

Neben dem X-Kommando gibt es die Möglichkeit, Datenstrings auszuwählen und eine bestimmte Darstellungsart zu erhalten. Diese Auswahl von Datenstrings geschieht mit dem SEL- und AND-Kommando. Durch das SEL-Kommando beginnt eine neue Auswahl, durch das AND-Kommando werden zusätzliche Datenstrings gewählt, die Parameter sind bei beiden Kommandos identisch, sie werden fuer das SEL-Kommando beschrieben.

(20) SEL parameter (EOT)  
 (21) AND parameter (EOT)

(20a) SEL ka kb (EOT)

Alle Datenstrings (ka, kb, \*) werden ausgewählt fuer die Darstellung mit nsy=ko.

Hinweis: Das X-Kommando entspricht diesem SEL-Kommando mit den Werten ka, kb der den letzten gewählten Datenstring folgenden Datenstring.

(20b) SEL ka kb (ko, nsy) (ko, nsy)... (EOT)

Alle Datenstrings (ka, kb, \*), deren ko-Werte im Kommando aufgefuehrt werden, werden ausgewählt fuer die Darstellung mit angegebenem nsy. Klammer und Komma koennen fehlen.

(20c) SEL ka kb ifac

Alle Datenstrings (ka, kb, \*) werden ausgewählt fuer die Darstellung mit nsy=ifac\*ko.

(20d) SEL ka kb (ko, nsy) (ko, nsy)...ifac (EOT)

Alle Datenstrings (ka, kb, \*) werden ausgewählt; folgt der ko-Wert in der Liste, wird der gegebene Wert nsy benutzt, folgt er nicht, wird nsy=ifac\*ko benutzt. Eine Datenstring kann nur einmal in einem Kommando gewählt werden, insgesamt maximal 10 mal.

Die Auswahl kann durch das T-Kommando ueberprueft werden. Das jeweils letzte der Kommandos (20) bzw. (21) wird entfernt durch das Kommando:

(22a) DEL (EOT)

Durch das L-Kommando werden die aktiven Kommandos fuer die Auswahl von Datenstrings auf dem Display gelistet:

(22b) L (EOT)

## 16. DEFINITION VON TEXTSTRINGS

Einer Abbildung kann Text zugeordnet werden, der in Form von Textstrings erzeugt, verschoben und entfernt werden kann. Bei jeder Abbildung ist automatisch bereits der evtl. den Datenstrings zugeordnete Text vorhanden. Weitere maximal 9 Textstrings koennen zugefuegt werden. Es gibt folgende Kommandos:

(23) xcm ycm hcm phi 'text  
 (24) MOD xcm ycm hcm phi 'text

Durch das Kommando (23) wird eine neue Textstring erzeugt. Die Parameter xcm, ycm geben in cm auf dem Display die Verschiebung gegen den linken unteren Punkt der Abbildungsumwandung an, hcm ist die Texthoehe in cm (wird gerundet auf vielfache von 0.35cm), phi gibt den Winkel in Grad bzgl. der unteren Skala an. Durch das Kommando (24) wird der gerade erzeugte bzw. durch Kommando (23) gewählte Text modifiziert; eine Modifizierung wird fuer alle #0 angegebenen Parameter vorgenommen. Bei xcm=220 wird die Textstring entfernt. Bei hcm=4 wird 0.7cm benutzt.

DESY F1-75/2

Beispiel:

10	-1.5	'Unterschrift		Neue Textstring
MOD	2			Verschieben um 2cm in x-Richtung
MOD	'TEXT			Änderung des Textes

Durch das Leerkommando

(25) (EOT)

wird jeweils die vorhergehende bzw. die letzte Textstring auf dem Display gezeigt, sie kann anschliessend durch Kommando (24) modifiziert werden. Bei jeder Neudefinition bzw. Modifizierung (ausser Entfernen) wird die letzte Abbildung mit der Änderung auf dem Display gezeigt.

Alle existierenden Textstrings werden in den angegebenen Positionen beim nächsten D-Kommando der Abbildung hinzugefügt. Wenn durch SEL- und AND-Kommandos oder durch das X-Kommando neue Datenstrings gewählt werden, werden die vorher existierenden Textstrings nur vorläufig entfernt. Durch das Kommando

(26) TEXT (EOT)

ist der vorher definierte Text wieder vorhanden.

Wenn jedoch vor dem TEXT-Kommando neue Textstrings durch Kommando (23) definiert werden, sind alle vorläufig entfernten Textstrings endgültig entfernt.