

J. Langendorf

O. Paetsch

GRAZIL
(Graphical ZIB Language)

Ein graphisches Anwendungsprogramm zur Darstellung
von Kurven- und Funktionsverläufen im Koordinatensystem

Technical Report 87-7 (Dezember 1987)

KONRAD-ZUSE-ZENTRUM FÜR INFORMATIONSTECHNIK;
Heilbronner Str. 10; D-1000 Berlin 31

INHALTSVERZEICHNIS:	Seite
Einführung	1
Beschreibung der ZUGRIFF-Dateischnittstelle für graphischen Input	2
1. Namensvereinbarung	3
2. Definition der Daten	3
- Bemerkungen zur Syntax	8
3. Daten	8
- Beispiele	9
Beschreibung des Programmpakets GRAZIL und dessen Benutzung . .	19
Graphik Kommandos	21
Beispiele zum Programmlauf	41
Bemerkungen	49
Aufruf und Dateireferenzen des Programmpakets GRAZIL	51
Leistungsmerkmale	53
Literatur	54

EINFÜHRUNG

Das Softwarepaket GRAZIL (GRAphical Zib Language) entstand aus der Anforderung, für das Softwarepaket LARKIN einen graphischen Postprozessor zur Darstellung von Trajektorien gewöhnlicher Differentialgleichungen zu implementieren. Bei dieser Anwendung fallen im allgemeinen Daten mit folgender Struktur an: Eine von vornherein nicht festlegbare Anzahl monoton steigender X-Werte (Stützstellen), sowie zu jedem X-Wert n zugehörige Y-Werte (Lösungsvektor). N ist die Anzahl der in einem LARKIN-Lauf berechneten Spezies (chemische Stoffe, z.B. H_2 , H_2O , C_2H_2 , ...). Der Lösungsvektor beinhaltet deren Konzentration zu einem Zeitpunkt (Stützstelle) /DEU/.

Das aus diesen Anforderungen heraus entwickelte Programm war stark an dieser Struktur orientiert und somit wenig flexibel, um Daten anderer Struktur zu visualisieren.

Eine weitere Anwendung für einen graphischen Postprozessor entstand durch Effizienzuntersuchungen an Programmen. Bei dieser Anwendung wird berechnet, wieviel Zeit ein Programm benötigt, um eine Lösung in vorgegebener Genauigkeit zu erreichen, bzw. wie oft hierfür ein bestimmter Programmteil aufgerufen werden muß, etc. Hier treten die Ergebniswerte immer paarig auf (also je ein X-Wert ("Toleranz") und ein Y-Wert ("Zeit" oder "Anzahl der Aufrufe").

Um bei graphischen Anwendungsprogrammen zur Darstellung von Kurven- und Funktionsverläufen das Einlesen von Daten unterschiedlicher Struktur nicht immer von neuem zu programmieren, boten sich zwei Lösungen an:

1. Einen einheitlichen Aufbau derartiger Dateien zu entwickeln,
oder
2. ein "offenes Lesemodul" zu entwickeln, dem die genaue Struktur der einzulesenden Daten mitgeteilt wird.

Bei GRAZIL wurde der zweite Lösungsansatz implementiert, da er im Gegensatz zu Lösung 1 dem Entwickler der Software, deren Output mittels eines graphischen Postprozessors visualisiert werden soll, die größere Freiheit beim Implementieren läßt.

Beschreibung der ZUGRIFF-Dateischnittstelle für graphischen Input

Aus den oben erwähnten Anforderungen und Gründen wurde ZUGRIFF (Zib Universal GRaphical Input File Format) entwickelt und implementiert. Weitere Anwendungen spezieller Art z.B. CAD oder allgemeinerer Art z.B. Bildspeicherung u. ä. wurden bewußt außer acht gelassen, um das ZUGRIFF-Konzept nicht zu überfrachten.

In einer ZUGRIFF-Datei werden zunächst alle Namen/Spezies deklariert, denen dann die eigentlichen Datenwerte zugewiesen werden sollen. Durch Angabe dieser Namen können dann die darzustellenden Kurven ausgewählt werden. Vor den Daten wird das Format der nachfolgenden Daten definiert, wobei das Paar (Definition, Daten) mehrfach in einer Datei vorkommen kann. Die Reihenfolge Namensvereinbarung, Definition, Daten ist zwingend vorgeschrieben.

In den nachfolgenden Syntaxbeschreibungen bedeutet:

- < ... > => nicht terminales Symbol, d.h. metasprachliche Variable, die noch weiter definiert werden muß.
- [...] => Defaultangabe
- { ... } * => kann mehrfach angegeben werden
- | => logisches "oder"

1. Namensvereinbarung:

&NAME [<name-1>, <name-2>, ..., <name-n>] | [<n>]

name-1 - name-n :: Es werden n Namen vereinbart

-oder-

n :: Es sind n Spezies in den Daten
vorhanden, denen Defaultnamen
zugewiesen werden (V1 - Vn)

Die Namensvereinbarung muß genau einmal vor der ersten Datendefinition vorgenommen werden. Nach einer erneuten Namensvereinbarung ist eine evtl. vorher gegebene Definition aufgehoben.

2. Definition der Daten:

Nach dem Schlüsselwort &DEF wird die Anzahl der X-, Y- und Z-Werte angegeben. Dabei wird der Achsenbezeichner (X, Y oder Z) gefolgt von einem Gleichheitszeichen und der Anzahl der der Achse zuzuordnenen Datenwerte angegeben. Die Reihenfolge der Achsenbezeichner ist frei bzw. wird von der Struktur der Daten vorgegeben. In einer Definition müssen mindestens X und Y definiert sein und jeder Achsenbezeichner darf nur einmal vorkommen.

Z. B. &DEF X=1, Y=6 bedeutet, daß zu je einem X-Wert ein Y-Vektor mit sechs Werten in den Daten vorhanden ist. Daraus folgt eine 2-dimensionale Ausgabe. Wenn sechs Speziesnamen deklariert wurden, dann wird für jede Spezies je ein Y-Wert zugewiesen und es wird für alle der eine X-Wert zugewiesen. Für 3-dimensionale Graphiken muß dementsprechend noch ein Z-Wert angegeben werden.

Für die Anzahl der Datenwerte kann auch ein komplexerer Ausdruck in der Form $X = 3 * 4$, angegeben werden. Das bedeutet, daß insgesamt 12 Datenwerte, die der X-Achse zugeordnet werden sollen, vorhanden sind. Wenn man die Anzahl der Spezies für diese Definition mit 3 annimmt, dann sind die 12 Datenwerte folgendermaßen aufzuteilen: Für jede der drei Spezies ein Wert als ersten X-Wert, dann für jede Spezies der zweite X-Wert usw. Sind dagegen 4 Spezies deklariert, dann sind die ersten 3 Werte drei aufeinanderfolgende X-Werte der ersten Spezies, dann folgen drei X-Werte für die zweite Spezies usw.

Wenn innerhalb der gleichen Datei die Datendefinition geändert werden soll, so darf dabei nicht die Dimension geändert werden. In der Datendefinition sind auch Klammerschachtelungstiefe mit Multiplikationsfaktoren zugelassen. Bei der Datendefinition muß gewährleistet sein, daß allen vereinbarten Namen auch Werte eindeutig zugeordnet werden können. D.h. die Länge eines X- oder Y-Vektors muß mit der Anzahl der Namen in einem ganzzahligen Verhältnis stehen.

Zusätzlich zu den Achsenbezeichnern können in einer Definition auch noch Datenbezeichner in der Form : X-ACHSE = X = 8 angegeben werden. Der Datenbezeichner X-ACHSE wird in GRAZIL als Voreinstellung für den X-Achsentext verwendet und er kann in den Daten verwendet werden. In diesem Fall folgendermaßen :

... X-ACHSE = 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, ...
(hier müssen genau 8 Datenwerte stehen |)

Sind in den Daten irgendwelche Datenbezeichner in der oben genannten Form vorhanden, dann müssen sie in der Definition angegeben werden.

Wenn kein Datenbezeichner vorhanden ist so wird der Achsenbezeichner auch als Datenbezeichner verwendet. Wird kein Achsenbezeichner angegeben, so muß ein Datenbezeichner vorhanden sein und es wird der erste Datenbezeichner gleichzeitig als Achsenbezeichner X angesehen, der zweite Datenbezeichner als Y und bei dreidimensionalen Daten der dritte folgerichtig als Z betrachtet.

Sind in den Daten Werte vorhanden, die nicht einer der Achsen zugeordnet werden sollen und demnach in der Graphik auch nicht dargestellt werden sollen, dann müssen sie als Dummywerte mit z.B. % = 5 deklariert werden. Bei Dummywerten können auch Datenbezeichner mit angegeben werden z.B. DUMMY = % = 5 .

&DEF-Syntax :

```
&DEF ::= [ <Namelist> ] <Datadefinition>
Namelist ::= <Varnamelist > :
Varnamelist ::= <Varname> , <Varnamelist> | <Varname>
Varname #1 ::= Character [ Character #]
Datadefinition ::= <Axisdefinitionlist>
Axisdefinitionlist ::= <X-Axisdefinition> , <Y-Axisdefinition>
|
( <X-Axisdefinition> , <Y-Axisdefinition>
) * <NumDV>
|
<X-Axisdefinition> , <Y-Axisdefinition> ,
<Z-Axisdefinition>
|
( <X-Axisdefinition> , <Y-Axisdefinition>
) * <NumDV> , <Z-Axisdefinition>
|
<X-Axisdefinition> , ( <Y-Axisdefinition> ,
<Z-Axisdefinition> ) * <NumDV>
|
( <X-Axisdefinition> , <Y-Axisdefinition> ,
<Z-Axisdefinition> ) * <NumDV>

X-Axisdefinition ::= [ <Dummydefinition> , ] X = <NumDV>
[ , <Dummydefinition> ]
|
[ <Dummydefinition> , ] <Data-Id> = <NumDV>
[ , <Dummydefinition> ]
|
[ <Dummydefinition> , ]
<Data-Id> = X = <NumDV>
[ , <Dummydefinition> ]
```

```

Y-Axisdefinition ::= [ <Dummydefinition> , ] Y = <NumDV>
                  [ , <Dummydefinition> ]
                  |
                  [ <Dummydefinition> , ] <Data-Id> = <NumDV>
                  [ , <Dummydefinition> ]
                  |
                  [ <Dummydefinition> , ]
                  <Data-Id> = Y = <DumDV>
                  [ , <Dummydefinition> ]

Z-Axisdefinition ::= [ <Dummydefinition> , ] Z = <NumDV>
                  [ , <Dummydefinition> ]
                  |
                  [ <Dummydefinition> , ] <Data-Id> = <NumDV>
                  [ , <Dummydefinition> ]
                  |
                  [ <Dummydefinition> , ]
                  <Data-Id> = Z = <DumDV>
                  [ , <Dummydefinition> ]

NumDV              ::= positiv Integer | <Number of Names> #2 |
                  <Axis-Id> #3 | <Expression>

Number of Names #2 ::= N
Axis-Id #3         ::= X | Y | Z
Expression         ::= positiv Integer * <NumDV> |
                  <Number of Names > * <NumDV> |
                  <Axis-Id> * <NumDV>

Data-Id           ::= Character { Character }*
Dummydefinition   ::= [ Data-Id = ] % = <NumDV>

```

Bemerkungen zur Syntax:

- #1 : In einem Speziesnamen (Variablennamen) können "*" und "?" als Wildcards angegeben werden. "*" steht für eine beliebige Anzahl beliebiger Zeichen. "?" steht für ein beliebiges Zeichen. (siehe auch &VARS-Kommando der GRAZIL-Beschreibung)

- #2 : Es wird die Anzahl der für diese Definition gültigen Speziesnamen genommen. Wenn in der Definition eine <Namelist> angegeben wurde, so wird die Anzahl der dort angegebenen Namen genommen, andernfalls wird die Anzahl der vereinbarten Namen des letzten &NAME-Kommandos für N eingesetzt.

- #3 : <Axis-Id> kann nur verwendet werden, wenn der Achsenbezeichner in der Definition schon deklariert wurde (von links nach rechts gesehen). Es wird dann die für diese Achse schon angegebene Anzahl der Achsenwerte eingesetzt.

3. Daten:

Entsprechend der Definition werden die Daten nach dem Schlüsselwort &DATA angegeben. Ein Datenblock geht immer von dem &DATA-Kommando bis zum nächsten mit & beginnenden Kommandoschlüsselwort. Die einzelnen Datenwerte können mit den Datenbezeichnern eingeleitet werden. Wenn Datenbezeichner verwendet werden sollen, so kann immer nur der gesamte Achsenvektor einmal mit einem Achsenbezeichner eingeleitet werden. In einem Datenblock müssen sich immer genau so viele Datenwerte befinden, wie in dem letzten gültigen &DEF-Kommando deklariert wurde. Wenn im &DEF-Kommando für X oder Y nur ein Wert angegeben wurde, dann werden die weiteren Datenwerte für X oder Y in den folgenden Datenblöcken angegeben. Werden für X und Y in der

Definition mehr als ein Wert angegeben, so können weitere Daten für dieselben Spezies nur nach erneuter Definition angegeben werden.

```
&DATA <wert-1>, <wert-2>, ..., <wert-n>
```

```
<wert-1> - <wert-n> :: Datenwerte oder Datenbezeichner
```

Die Vereinbarungs- und Definitionskommandos können über mehrere Zeilen bis zum Beginn des nächsten, mit einem & beginnenden Kommandos (Schlüsselwort) gehen.

Beispiele:

Der bisherige Larkin-Output könnte folgendermaßen definiert sein:

Inputdatei:

```
&HEAD
'          ----- H2-O2 COMBUSTION -----
'
' SEE: Hoppenstaedt,Alfeld,Aitken: Numerical Treatment of rapid
'       chemical Kinetics by Pertubation and Projection Methods.
'       PROCEEDINGS ...
'
'                               DATE OF LATEST CHANGE: OCT. 18, 81
&END
&NAME
H2      ,O2      ,H2O      ,H*      ,OH*
O*      ,HO2
&END
&DEF
T = 1, C = N
&END
```

&DATA

T= 0.000000000000D+00, C=
0.1000000D-06, 0.5000000D-07, 0.0000000D+00, 0.0000000D+00,
0.0000000D+00, 0.0000000D+00, 0.0000000D+00,

&END

&DATA

T= 0.160626572133D-05, C=
0.1000000D-06, 0.5000000D-07, 0.1040122D-23, 0.4817827D-18,
0.8733438D-22, 10689D-22, 0.4818797D-18,

&END

&DATA

T= 0.162232837854D-03, C=
0.1000000D-06, 0.5000000D-07, 0.9189164D-18, 0.4923452D-16,
0.6033600D-18, 0.3420592D-19, 0.4866985D-16,

&END

.
. (insgesamt 23 Data-Blöcke)

&DATA

T= 0.800000000000D-01, C=
0.0000000D+00, 0.4699259D-08, 0.6251866D-07, 0.7642479D-08,
0.1838245D-09, 0.1838147D-09, 0.1688180D-13,

&END

Outputzuordnung der Daten zu den Spezies und den Achsen:

H2:	X	Y
1. :	0.0000000D+00	0.1000002D-06
2. :	0.1606269D-05	0.1000002D-06
3. :	0.1622330D-03	0.1000002D-06
	.	
	.	
	.	
23. :	0.7999998D-01	0.0000000D+00

O2: X Y

1. :	0.0000000D+00	0.5000038D-07
2. :	0.1606269D-05	0.5000038D-07
3. :	0.1622330D-03	0.5000038D-07
	.	
	.	
	.	
23. :	0.7999998D-01	0.4699309D-08

H2O: X Y

1. :	0.0000000D+00	0.0000000D+00
2. :	0.1606269D-05	0.1040137D-23
3. :	0.1622330D-03	0.9189282D-18
	.	
	.	
	.	
23. :	0.7999998D-01	0.6251912D-07

H*: X Y

1. :	0.0000000D+00	0.0000000D+00
2. :	0.1606269D-05	0.4817891D-18
3. :	0.1622330D-03	0.4923486D-16
	.	
	.	
	.	
23. :	0.7999998D-01	0.7642562D-08

OH*: X Y

1. :	0.0000000D+00	0.0000000D+00
2. :	0.1606269D-05	0.8733438D+00
3. :	0.1622330D-03	0.6033600D+00
	.	
	.	
	.	
23. :	0.7999998D-01	0.1838245D+00

```

O*:          X          Y
  1. : 0.0000000D+00  0.0000000D+00
  2. : 0.1606269D-05  0.6710767D-22
  3. : 0.1622330D-03  0.3420650D-19
      .
      .
      .
 23. : 0.7999998D-01  0.1838173D-09

```

```

HO2:         X          Y
  1. : 0.0000000D+00  0.0000000D+00
  2. : 0.1606269D-05  0.4818861D-18
  3. : 0.1622330D-03  0.4867018D-16
      .
      .
      .
 23. : 0.7999998D-01  0.1688200D-13

```

Ein 3-D-Output könnte folgendermaßen definiert sein:

Inputdatei:

```

&NAME SPEZIES_1, SPEZIES_2, SPEZIES_3, SPEZIES_4, SPEZIES_5
&DEF X=44,T=Z=1,VAL=Y=N*X
&DATA
0.00000E+00, 0.25000E+00, 0.37500E+00, 0.50000E+00, 0.56250E+00
0.62500E+00, 0.68750E+00, 0.71875E+00, ... (insgesamt 44 X-Werte)
      .
      .
      .
T= 0.2000000E+01, VAL=

```

```

0.50000E-03, 0.99950E+00, 0.12776E-15, 0.41667E+01, 0.00000E+00
0.50000E-03, 0.99950E+00, 0.59660E-09, 0.41667E+01, 0.00000E+00
0.50000E-03, 0.99950E+00, 0.17416E-06, 0.41667E+01, 0.00000E+00
0.49999E-03, 0.99949E+00, 0.97745E-05, 0.41666E+01, 0.00000E+00
0.49992E-03, 0.99945E+00, 0.50840E-04, 0.41662E+01, 0.00000E+00
0.49967E-03, 0.99928E+00, 0.22219E-03, 0.41646E+01, 0.00000E+00
0.49874E-03, 0.99866E+00, 0.84307E-03, 0.41587E+01, 0.00000E+00
0.49765E-03, 0.99793E+00, 0.15694E-02, ... (insgesamt 220 Y-Werte)
.
.
.
&END

```

Outputzuordnung der Daten zu den Spezies und den Achsen;

```

SPEZIES_1:   X           Y           Z
1. : 0.0000000D+00  0.5000005D-03  0.1999998D+01
2. : 0.2500000D+00  0.5000005D-03  0.1999998D+01
3. : 0.3750000D+00  0.5000005D-03  0.1999998D+01
4. : 0.5000000D+00  0.4999905D-03  0.1999998D+01
5. : 0.5625000D+00  0.4999205D-03  0.1999998D+01
6. : 0.6250000D+00  0.4996706D-03  0.1999998D+01
7. : 0.6875000D+00  0.4987405D-03  0.1999998D+01
8. : 0.7187500D+00  0.4976506D-03  0.1999998D+01
.
.
.
44. : 0.2499997D+02  0.1000019D-15  0.1999998D+01

```

SPEZIES_2:	X	Y	Z
1.	: 0.000000D+00	0.9995000D+00	0.1999998D+01
2.	: 0.2500000D+00	0.9995000D+00	0.1999998D+01
3.	: 0.3750000D+00	0.9995000D+00	0.1999998D+01
4.	: 0.5000000D+00	0.9994900D+00	0.1999998D+01
5.	: 0.5625000D+00	0.9994500D+00	0.1999998D+01
6.	: 0.6250000D+00	0.9992800D+00	0.1999998D+01
7.	: 0.6875000D+00	0.9986600D+00	0.1999998D+01
8.	: 0.7187500D+00	0.9979300D+00	0.1999998D+01
	.		
	.		
	.		
44.	: 0.2499997D+02	0.6666700D+00	0.1999998D+01

SPEZIES_3:	X	Y	Z
1.	: 0.0000000D+00	0.1277624D-15	0.1999998D+01
2.	: 0.2500000D+00	0.5966085D-09	0.1999998D+01
3.	: 0.3750000D+00	0.1741604D-06	0.1999998D+01
4.	: 0.5000000D+00	0.9774521D-05	0.1999998D+01
5.	: 0.5625000D+00	0.5084008D-04	0.1999998D+01
6.	: 0.6250000D+00	0.2221903D-03	0.1999998D+01
7.	: 0.6875000D+00	0.8430709D-03	0.1999998D+01
8.	: 0.7187500D+00	0.1569401D-02	0.1999998D+01
	.		
	.		
	.		
44.	: 0.2499997D+02	0.3333300D+00	0.1999998D+01

SPEZIES_4:	X	Y	Z
1.	: 0.000000D+00	0.4166698D+01	0.1999998D+01
2.	: 0.250000D+00	0.4166698D+01	0.1999998D+01
3.	: 0.375000D+00	0.4166698D+01	0.1999998D+01
4.	: 0.500000D+00	0.4166597D+01	0.1999998D+01
5.	: 0.562500D+00	0.4166197D+01	0.1999998D+01
6.	: 0.625000D+00	0.4164598D+01	0.1999998D+01
7.	: 0.687500D+00	0.4158697D+01	0.1999998D+01
8.	: 0.718750D+00	0.4151797D+01	0.1999998D+01
	.		
	.		
	.		
44.	: 0.2499997D+02	0.9999992D+00	0.1999998D+01

SPEZIES_5:	X	Y	Z
1.	: 0.000000D+00	0.000000D+00	0.1999998D+01
2.	: 0.250000D+00	0.000000D+00	0.1999998D+01
3.	: 0.375000D+00	0.000000D+00	0.1999998D+01
4.	: 0.500000D+00	0.000000D+00	0.1999998D+01
5.	: 0.562500D+00	0.000000D+00	0.1999998D+01
6.	: 0.625000D+00	0.000000D+00	0.1999998D+01
7.	: 0.687500D+00	0.000000D+00	0.1999998D+01
8.	: 0.718750D+00	0.000000D+00	0.1999998D+01
	.		
	.		
	.		
18.	: 0.2499997D+02	0.000000D+00	0.1999998D+01

Verschiedene Definitionen für jeweils nur einige der vorhandenen Spezies, wobei für eine Spezies mehrere Definitionen angegeben wurden:

Inputdatei:

```
&HEAD THATS A VERY DIFFICULT PROBLEM
      ISNT IT?
&NAME SPEZIES_1, SPEZIES_2, SPEZIES_3, SPEZIES_4, SPEZIES_5
&DEF SPEZIES_1,SPEZIES_2:(X=1,Y=1)*N
&DATA 1101.0 1201.0 2101.0 2201.0
&DATA 1102.0 1202.0 2102.0 2202.0
&DATA 1103.0 1203.0 2103.0 2203.0
&DATA 1104.0 1204.0 2104.0 2204.0
&DATA 1105.0 1205.0 2105.0 2205.0
&DEF SPEZIES_3:(X=3,Y=3)
&DATA 3101.0 3102.0 3103.0 3201.0 3202.0 3203.0
&DEF SPEZIES_1,SPEZIES_3,SPEZIES_5:(X=N,Y=N)*4
&DATA 1106.0 3104.0 5101.0 1206.0 3204.0 5201.0
      1107.0 3105.0 5102.0 1207.0 3205.0 5202.0
      1108.0 3106.0 5103.0 1208.0 3206.0 5203.0
      1109.0 3107.0 5104.0 1209.0 3207.0 5204.0
&END
&DEF X=1,Y=N
&DATA
      X = 0101.0, Y = 1210.0 2206.0 3208.0 4201.0 5205.0
&END
&DATA
      X = 0102.0, Y = 1211.0 2207.0 3209.0 4202.0 5206.0
&END
&DATA
      X = 0103.0, Y = 1212.0 2208.0 3210.0 4203.0 5207.0
&END
&DATA
      X = 0104.0, Y = 1213.0 2209.0 3211.0 4204.0 5208.0
&END
```

&DATA

X = 0105.0, Y = 1214.0 2210.0 3212.0 4205.0 5209.0

&END

Outputzuordnung der Daten zu den Spezies und den Achsen:

SPEZIES_1:	X	Y
1.	: 0.1101000D+04	0.1201000D+04
2.	: 0.1102000D+04	0.1202000D+04
3.	: 0.1103000D+04	0.1203000D+04
4.	: 0.1104000D+04	0.1204000D+04
5.	: 0.1105000D+04	0.1205000D+04
6.	: 0.1106000D+04	0.1206000D+04
7.	: 0.1107000D+04	0.1207000D+04
8.	: 0.1108000D+04	0.1208000D+04
9.	: 0.1109000D+04	0.1209000D+04
10.	: 0.1010000D+03	0.1210000D+04
11.	: 0.1020000D+03	0.1211000D+04
12.	: 0.1030000D+03	0.1212000D+04
13.	: 0.1040000D+03	0.1213000D+04
14.	: 0.1050000D+03	0.1214000D+04

SPEZIES_2:	X	Y
1.	: 0.2101000D+04	0.2201000D+04
2.	: 0.2102000D+04	0.2202000D+04
3.	: 0.2103000D+04	0.2203000D+04
4.	: 0.2104000D+04	0.2204000D+04
5.	: 0.2105000D+04	0.2205000D+04
6.	: 0.1010000D+03	0.2206000D+04
7.	: 0.1020000D+03	0.2207000D+04
8.	: 0.1030000D+03	0.2208000D+04
9.	: 0.1040000D+03	0.2209000D+04
10.	: 0.1050000D+03	0.2210000D+04

SPEZIES_3: X Y
1. : 0.3101000D+04 0.3201000D+04
2. : 0.3102000D+04 0.3202000D+04
3. : 0.3103000D+04 0.3203000D+04
4. : 0.3104000D+04 0.3204000D+04
5. : 0.3105000D+04 0.3205000D+04
6. : 0.3106000D+04 0.3206000D+04
7. : 0.3107000D+04 0.3207000D+04
8. : 0.1010000D+03 0.3208000D+04
9. : 0.1020000D+03 0.3209000D+04
10. : 0.1030000D+03 0.3210000D+04
11. : 0.1040000D+03 0.3211000D+04
12. : 0.1050000D+03 0.3212000D+04

SPEZIES_4: X Y
1. : 0.1010000D+03 0.4201000D+04
2. : 0.1020000D+03 0.4202000D+04
3. : 0.1030000D+03 0.4203000D+04
4. : 0.1040000D+03 0.4204000D+04
5. : 0.1050000D+03 0.4205000D+04

SPEZIES_5: X Y
1. : 0.5101000D+04 0.5201000D+04
2. : 0.5102000D+04 0.5202000D+04
3. : 0.5103000D+04 0.5203000D+04
4. : 0.5104000D+04 0.5204000D+04
5. : 0.1010000D+03 0.5205000D+04
6. : 0.1020000D+03 0.5206000D+04
7. : 0.1030000D+03 0.5207000D+04
8. : 0.1040000D+03 0.5208000D+04
9. : 0.1050000D+03 0.5209000D+04

Beschreibung des Programmpakets GRAZIL und dessen Benutzung:

GRAZIL liest eine ZUGRIFF-Datei und kann dann vom Benutzer interaktiv gesteuert werden, um die gewünschte Kurvendarstellung zu zeigen. Der Benutzer hat dabei die Möglichkeit zum:

- Auswählen der darzustellenden Spezies/Variablen
- Auswählen des darzustellenden Datenausschnitts
- Setzen der Achsentexte und Überschriften
- Setzen der Legendenart, Linienart, Markierungsart und Farbe
- Darstellen der Achse in linearer oder logarithmischer Einteilung
- Auswählen einer Interpolationsart
- Setzen der Bildgröße und des Ausgabegerätes
- Erzeugen eines Gitters über der Zeichnung zur Verfügung.

Die Erzeugung und Ausgabe der Zeichnungen geschieht mit Hilfe der Softwarebibliotheken BIZEPS2 /BEC2/ und GKS /BEC1/.

GRAZIL erzeugt zweidimensionale Graphiken. Wenn die von GRAZIL zu lesende ZUGRIFF-Datei dreidimensionale Daten enthält, was erlaubt ist, so werden die Z-Werte von GRAZIL überlesen und haben folglich keine Auswirkung auf die Graphik.

Eine typische Sitzung mit GRAZIL sieht folgendermaßen aus: Der Benutzer ruft GRAZIL mit einer Datei gemäß dem ZUGRIFF-Konzept auf. GRAZIL liest diese Datei ein und geht dann in den interaktiven Modus (meldet sich mit dem Prompt: "PLEASE ENTER COMMAND"). Der Benutzer spezifiziert die Darstellungsoptionen und gibt das Bild aus (z.B. erst als Preview auf einem Sichtgerät und dann nach eventuellen Korrekturen auf einem Plotter).

Zum Aufruf von GRAZIL steht dem Benutzer eine CLIST-Prozedur GRAZIL im Dataset 'GRAPHIK,CLIST' zur Verfügung:

```
GRAZIL [ IND1(<dsname-1>) ] [ IND2(<dsname-2>) ]  
      [ IND3(<dsname-3>) ] [ GKSMMSG(<msg-dsname>) ]  
      [ METAFILE(<metafile>) ] [ BENSONPLOTTER(<benson-out>) ]  
      [ HPLOTTER(<hp-out>) ]
```

CLIST-Prozedur hat folgende optionale Schlüsselwortparameter:

<dsname-1>, :: Drei Datasets zum Einlesen von ZUGRIFF-Dateien
<dsname-2>, und/oder Kommandodateien. GRAZIL liest in der
<dsname-3> Reihenfolge der Numerierung. Eingabedateien
 werden von Spalte 1 bis Spalte 80 gescannt. Man
 beachte also die Numerierung in Spalte 72 - 80.
 Default : - d.h. GRAZIL geht sofort in den
 interaktiven Modus.

<msg-dsname> :: Dataset zur Aufnahme der GKS-Fehlermeldungen.
 Default : * d.h. Ausgabe erfolgt am Terminal.

<metafile> ::: Dataset zur Aufnahme des GKS-Metafiles, falls dieses als Ausgabegerät gewählt wurde.
 Default : METAFILE

<benson-out>, ::: Datasets zur Aufnahme von Plotterbefehlen.
 <hp-out> Default : Ausgabe erfolgt direkt am Plotter, falls der HP-Plotter 7475 oder der BENSON 1230 als Ausgabegerät gewählt wurde.

Graphik-Kommandos

Zur Steuerung der Graphik gibt es zahlreiche, weiter unten aufgeführte Kommandos. Die Kommandos können abgekürzt werden. Die großgeschriebenen Buchstaben eines Kommandonamens müssen mindestens eingegeben werden. Im Kommandofile sind Kommentare erlaubt. Sie werden mit ";" eingeleitet und gelten dann bis zum Zeilenende. Eingabetexte sollen in Apostrophs eingeschlossen werden. Gleitkommazahlen können auch in Exponentendarstellung eingegeben werden. Z.B. der Wert 0,012 in der Form 1,2E-2. In den nachfolgenden Kommandoerklärungen werden einige Begriffe aus der GKS-Welt verwendet, die hier noch einmal kurz erläutert werden sollen:

- Farbe, Damit ist der "Polyline Colour Index" bzw. "Poly-marker Colour Index" des GKS gemeint. Der Colour Index 0 repräsentiert immer die Hintergrundfarbe, alle anderen eine auf der aktuellen Workstation vorhandene Farbe. Der Colour Index 1 ist dabei meist die Standardfarbe schwarz auf Plottern und weiß oder grün auf Sichtgeräten.

- Linientyp ist der "Linientyp" des GKS. Die Linientypen 1-4 sind im GKS genormt und demnach immer vorhanden:
- 1 = durchgezogene Linie
 - 2 = gestrichelte Linie
 - 3 = punktierte Linie
 - 4 = strichpunktierte Linie
- Weitere Linientypen können für bestimmter Workstations vorhanden sein, wobei über ihr konkretes Aussehen keine Aussage getroffen werden kann.
- Markertyp ist der "Marker Type" des GKS. Die Markertypen 1-5 sind im GKS genormt.
- 1 = kleinstmöglicher Punkt .
 - 2 = Kreuz +
 - 3 = Sternchen *
 - 4 = kleiner Kreis o
 - 5 = Kreuz x
- Weitere Markertypen sind implementierungsabhängig.

Im folgenden sind die verfügbaren Kommandos in alphabetischer Reihenfolge aufgeführt:

1. ALIAS

- führt weitere Namen für ein vorhandenes Kommando ein.

Kommandosyntax :

```
&ALIAS <alter Name> [ <neuer Name> [ <iwert> ] ]
```

alter Name :: Ein schon vorhandener Kommandoname. Wird
<alter Name> als einzigster eingegeben,
so werden alle Aliaseinträge für diesen
Kommandonamen gelöscht.

neuer Name :: Neuer, zusätzlicher Kommandoname.

iwert :: legt die Anzahl der mindestens einzuge-
benden Buchstaben des neuen Namens fest.

2. Atext | AXISText

- Liest neue Achsenbeschriftung ein und schaltet zur
nächsten Ausgabe auf diesen um. Der Text darf nicht
länger als 20 Zeichen sein. Ist der Text länger als
20 Zeichen, wird er abgeschnitten.

Kommandosyntax :

&ATEXT <x|y> [<'text'>]

x|y :: Achsenbezeichner

text :: Text oder "*". Die Eingabe eines "*"
bewirkt ein Umschalten zwischen dem mit
ATEXT eingelesenen und dem Default-Text.

Default : Der in der Datendefinition angegebene Achsen-
bezeichner.

3. AUXLinie

- Zeichnet eine gerade Hilfslinie in die Zeichenfläche.
Die Eingabe eines "=" Zeichens läßt den entsprechenden
Positionsparameter unverändert.

Kommandosyntax:

```
&AUXLINE <liniennummer> <richtung> [ <position>  
      [ <linientyp> [ <linienfarbe> [ <linientext>]]]]
```

liniennummer :: Nummer der zu zeichnenden Hilfslinie.

richtung :: hor = Die Linie verläuft waagrecht.

ver = Die Linie verläuft senkrecht.

del = Löscht die Linie.

position :: Gibt die Position der Hilfslinie an, d.h.
bei waagerechten Hilfslinien wird ihr
Schnittpunkt mit der Y-Achse angegeben
und dementsprechend bei senkrechten
Hilfslinien wird der Schnittpunkt mit der
X-Achse angegeben.

linientyp :: >0 = Die Linien werden mit dem
angegebenen Linientyp gezeichnet.
Bei Ausführung des Kommandos &RUN
wird überprüft, ob dieser Linientyp
für die gewählte Workstation
existiert. Wenn nicht, so wird mit
Linientyp 2 gezeichnet.

Default: 2

linienfarbe :: >0 = Die Linien werden in der angegebenen Farbe gezeichnet. Bei Ausführung des Kommandos &RUN wird überprüft, ob diese Farbe für die gewählte Workstation existiert. Wenn nicht, so wird mit Farbe 1 gezeichnet.

Default : 1

linientext :: Der angegebene Text wird an die Linie geschrieben. Der Text darf maximal 20 Zeichen lang sein.

4. AXISSPEC

- Legt den Schnittpunkt der Achsen, ihre Einteilung und die Anzahl der Ticks fest.

Kommandosyntax:

```
&AXISSPEC <x|y> <einteilung> [ <schnittpunkt>
          [ <hauptticks> [ <unterticks> ]]]
```

x|y :: Achsenbezeichner

einteilung :: lin = Lineare Einteilung der Achse.
log = Logarithmische Einteilung der Achse.

schnittpunkt :: real-wert = Die bezeichnete Achse wird am angegebenen Punkt der anderen Achse gezeichnet. Bei Ausführung des Kommandos &RUN wird überprüft, ob dieser Punkt innerhalb der Zeichenfläche liegt, wenn nicht wird min oder max eingestellt.

min = Es wird der kleinste Wert genommen, der mit dem &BOUNDS Kommando eingestellt wurde oder der durch &VARS implizit eingestellte.

max = Es wird der größte Wert genommen, der mit dem &BOUNDS Kommando eingestellt wurde oder der durch &VARS implizit eingestellte.

mid = Es wird die Mitte der anderen Achse eingestellt.

Default = min

hauptticks :: 0 = Es werden keine Hauptticks gezeichnet und auch keine Beschriftung.

-1 = Nur der Schnittpunkt wird beschriftet.

>=1 = Anzahl der Hauptticks mit Beschriftung.

Default = 10

Kommandosyntax :

```
&COMFILE <userid.dsname>[ <(member)> ]
```

userid.dsname:: vollqualifizierter Datasetname

member :: Membername, falls es sich um einen
 partitioned Dataset handelt.

8. CURve

- stellt für die ausgewählte Kurve die zum Zeichnen der
Kurve erforderlichen Attribute ein.

Kommandosyntax :

```
&Curve <name> <color> [<linientyp>[<markertyp>  
                  [<markerfrequenz>[<interpolation>]]]]
```

name :: Variablenname einer mit dem &VARS-
 Kommando bereits ausgewählten Variablen.
 Wildcards sind analog dem &VARS-Kommando
 erlaubt.

color :: -1 = Die in der Workstation vorhandenen
 Farben werden zur Darstellung der
 Kurven genutzt, d.h. mehrere Kurven
 werden zur Unterscheidung mit ver-
 schiedenen Farben gezeichnet.

0 = Die Kurve wird mit der 1. Farbe
gezeichnet.

>0 = Die Kurve wird in der angegebenen Farbe gezeichnet. Bei Ausführung des Kommandos &RUN wird überprüft, ob diese Farbe für die gewählte Workstation existiert. Wenn nicht, so wird mit Farbe 1 gezeichnet.

Default : -1

linientyp :: -1 = Die im GKS genormten Linientypen werden zur Darstellung der Kurven genutzt, d.h. mehrere Kurven werden zur Unterscheidung mit verschiedenen Linienarten gezeichnet. Die punktierte Linie wird nicht verwendet, da sie für das Gitter reserviert ist.

0 = Es wird keine Linie gezeichnet.

>0 = Die Kurve wird mit dem angegebenen Linientyp gezeichnet. Bei Ausführung des Kommandos &RUN wird überprüft, ob dieser Linientyp für die gewählte Workstation existiert. Wenn nicht, so wird mit Linientyp 1 gezeichnet.

Default : -1

markertyp :: -1 = Die im GKS genormten Markertypen werden zur Markierung der Kurven genutzt, d.h. mehrere Kurven werden zur Unterscheidung mit verschiedenen Markertypen markiert. Der Markertyp 1 (Punkt) wird nicht verwendet, da er auf Linien nicht zu sehen ist.

0 = Es wird kein Marker gezeichnet.
>0 = Die Kurve wird mit dem angegebenen Markertyp markiert. Bei Ausführung des Kommandos &RUN wird überprüft, ob dieser Markertyp für die gewählte Workstation existiert. Wenn nicht, so wird mit Markertyp.2 markiert.

Default : -1

markerfrequenz :: 0 = Es werden keine Marker gezeichnet.
1 = Jeder Stützpunkt wird markiert.
n = Jeder n-te Stützpunkt wird markiert. Es wird mit dem ersten darstellbarem Stützpunkt begonnen und auch der letzte darstellbare Stützpunkt wird markiert.

Default : 1

interpolation :: 0 = Lineare Interpolation
1 = Funktionsinterpolation mit BFLINE
2 = Kurveninterpolation mit GDP
3 = Stückweise kubische Hermite'sche Interpolation mit HERMITE FUNCTION EVALUATOR.

Default : 0

9. COMMANDfile

- siehe Comfile

10. DATA

- Leitet die Eingabe der Daten ein.

Kommandosyntax :

&DATA ... siehe oben

11. DEF

- Legt die Struktur der Eingabedaten fest, es werden die Anzahl der in einem DATA-Block vorhandenen Werte beschrieben. Hierbei können auch einfache Klammerausdrücke und Multiplikationen Vorkommen. Die Dimension der Daten wird durch die spezifizierten Achsen festgelegt. D. h.

definiert man X= ,Y=, Z= erhält man 3D - Daten

definiert man X= ,Y= erhält man 2D - Daten

Kommandosyntax :

&DEF ... siehe Seite 3: Definition der Daten.

12. Dialog

- Schaltet im Kommandofile die Eingabe auf Terminal-eingabe um.

Kommandosyntax :

&DIALOG

13. DUMMY

- Hat keine weitere Funktion.

Kommandosyntax :

&DUMMY

14. END

- Beendet den Eingabestrom eines Kommandos.

Kommandosyntax :

&END

15. EXecute

- siehe Run

16. Grid

- Schalter zum Zeichnen eines Gitters über der Zeichenfläche.

Kommandosyntax :

&GRID <on|off>

on :: Ein Gitter wird gezeichnet.

off :: Kein Gitter wird gezeichnet.

Default : off

17. HEAD

- Leitet den Überschriftenblock im Eingabefile ein.

Kommandosyntax :

```
&HEAD <'text'>
```

18. HEADLine

- Liest eine neue Überschrift ein und schaltet zur nächsten Ausgabe auf diese um. Die Eingabe eines "*" bewirkt ein Umschalten zwischen dem mit HEADLINE und dem mit HEAD eingelesenen Text. Der Text darf nicht länger als 50 Zeichen sein. Ist der Text länger als 50 Zeichen, wird er abgeschnitten.

Kommandosyntax :

```
&HEADLINE <1|2> <'text'>
```

- | | |
|------|---|
| 1 | :: Die eingelesene Textzeile wird als 1. Überschriftszeile ausgegeben |
| 2 | :: Die eingelesene Textzeile wird als 2. Überschriftszeile ausgegeben |
| text | :: Text oder "*". Die Eingabe eines "*" bewirkt ein Umschalten zwischen dem mit HEADLINE eingelesenen und dem Default-Text. |

Default : keine

19. Help

- Ohne Parameter gegeben listet HELP die Kommandos auf. Mit einem Kommandonamen als Parameter wird die Syntax des Kommandos aufgelistet.

Kommandosyntax :

&HELP [<kommando>]

kommando :: Ein gültiger Kommandoname.

20. LEGEndtype

- siehe Legtype

21. LEGtype | LEGEndtype

- Stellt die Legende ein.

Kommandosyntax :

&LEGTTYPE <iwert>

iwert :: 0 = Es wird keine Legende gezeichnet.
 1 = Die Legende wird an den rechten
 oberen oder unteren Rand der Zeich-
 nung gezeichnet.
 2 = Die Kurven werden an ihrem Ende
 beschriftet.

Default : 1

22. LIST <option>

- Listet die Parameter der ausgewählten Option auf.

Kommandosyntax:

&LIST <kommandoname>

kommandoname :: Es werden für das angegebene Kommando die
aktuellen Einstellungen angezeigt.

23. MINratio

- Stellt das Verhältnis des kleinsten zum zweitkleinsten Wert bezogen auf das Verhältnis vom kleinsten zum größten Wert der darzustellenden Variablen ein. Ist das Verhältnis vom kleinsten zum zweitkleinsten Wert größer als angegeben, wird als kleinster Wert der zweitkleinste zum Zeichnen gewählt. Dadurch wird verhindert, daß bei logarithmischer Y-Achse der wichtige Teil der Kurven sich im oberen Teil der Zeichnung "drängelt".

Kommandosyntax :

&MINRATIO <rwert>

rwert :: Realzahl im Intervall 0.0 .. 1.0 .

Default : 0.33

24. NAME

- Legt die Namen der darstellbaren Kurven fest. Es dürfen maximal 128 sein.

Kommandosyntax (Form 1) :

&NAME <name-1>, <name-1> ..., <name-n> n <= 128

name-1, ... :: Es werden n Namen vereinbart.

name-n ... siehe oben

(Form 2) :

&NAME <n> n <= 128

n :: Es sind n Spezies in den Daten vorhanden,
 denen Defaultnamen zugewiesen werden.

25. Quit

- siehe Stop

26. RENAME

- Verändert den Namen von Kommandos.

Kommandosyntax :

&RENAME <alter Name> <neuer Name> [<iwert>]

alter Name :: Ein schon vorhandener Kommandoname.

neuer Name :: Neuer Kommandoname.

iwert :: legt die Anzahl der mindestens einzugebenden Buchstaben des neuen Namens fest.

27. Run | EXECUTE

- Zeigt das Bild auf der aktuellen Workstation nach Überprüfung der Einstellungen durch &CURVE, &AXISSPEC und &AUXLINE (siehe dort).

Kommandosyntax :

&RUN

28. SAVE

- Sichert den Stand der Kommandoeingabe auf einen File. D. h. die Kommandos, die zur Einstellung eines Bildes gegeben wurden, werden gesichert. Wird das SAVE-Kommando ohne Parameter gegeben, wird vorausgesetzt, daß auf dem Fortrankanal 41 eine Datei allociert ist.

Für eine zu erstellende Kommandodatei gelten folgende DCB-Parameter:

```
RECFM  : FB
LRECL  : 80
BLKSIZ : LRECL * x
DSORG  : PS | PO
```

Kommandosyntax :

```
&SAVE [ <userid.dsname> [ <(member)> ] ]
userid.dsname:: vollqualifizierter Datasetname
member       :: Memername, falls es sich um einen parti-
                tioned Dataset handelt.
```

29. Size

- Stellt die Größe der Zeichenfläche auf der aktuellen Workstation ein. Die Angaben werden in Metern verlangt.

Kommandosyntax :

```
&SIZE <xwert ywert> | <wert> | <Ax>
xwert ywert  :: X,Y-Größe der Zeichenfläche.
wert        :: Stellt eine quadratische Zeichenfläche
                mit der Größe <wert> ein.
Ax          :: A2 - A8 = stellt das entsprechende DIN-
                Format als Zeichenfläche ein. (Quer)
```

Default : A4

30. SPecies

- siehe Vars

31. STATistic

- Zeigt einige Werte an, die zur Programmentwicklung nützlich waren.

Kommandosyntax :

&STATISTIC <on|off>

on :: Statistik wird angezeigt.

off :: Statistik wird nicht angezeigt.

Default : off

32. STOp | Quit

- Beendet das Programm.

Kommandosyntax :

&STOP

33. VARIables

- siehe Vars

34. Vars | VARIables | SPecies

- Wählt die darzustellenden Kurven aus. Es sind Wildcards erlaubt. Die Eingabe eines "?" ersetzt genau einen Buchstaben, die eines "*" ersetzt alle ab dem *. Durch Eingabe von "**" wird nach "*" gesucht (keine Wildcard).

Kommandosyntax :

&VARS <operator> <var1> <var2> ... <varn> n <= 9

operator :: "=" : Trägt die angegebenen Variablen in die Liste der auszugebenen Kurven ein. Eine vorhandene Liste wird überschrieben. Wird nur der Operator "=" gegeben, wird eine vorhandene Liste gelöscht.

"+" : Trägt die angegebenen Variablen zusätzlich in die Liste ein.

"-" : Entfernt die angegebenen Variablen aus der Liste.

var-x :: Spezies/Variablenname der mit &NAME
 vereinbart wurde.

Default : keine

35. Wktype | WOrkstation

- Stellt die aktuelle Workstation ein. Wenn es sich bei der Workstation um ein Sichtgerät handelt, wird dessen maximale Bildschirmgröße als Zeichenfläche eingestellt. Welche Workstations angegeben werden können, ist GKS-Implementationsabhängig. Mit dem Kommando "&LIST WKTYPE" können die aktuell vorhandenen Workstations abgefragt werden.

Kommandosyntax :

&WKTYPE <iwert>

iwert :: Workstation Typ,

1 = Metafile

1230 = Bensonplotter 1230

2648 = HP - Sichtgerät HP2648a

3279 = IBM 3279-Graphikterminal

74753 = HP - Tischplotter 7475

(Zeichenfläche DIN A3)

74754 = HP - Tischplotter 7475

(Zeichenfläche DIN A4)

weitere Workstations sind in Vorbereitung

Default : 1

36. WOrkstation.

- siehe Wktype

2.0 Beispiele zum Programmlauf

2.1 1. Beispiel (unter Ausnutzung der Defaults)

```
1  (in)  grazil ind1(larkin.out(inputdat))
   (out) PLEASE ENTER COMMAND

2  (in)  &wktype 74754
   (out) PLEASE ENTER COMMAND

3  (in)  &vars = H2 O2 H2O H**
   (out) PLEASE ENTER COMMAND
   (in)  &run
   (out) ... -> Bild 1
   (out) PLEASE ENTER COMMAND
   (in)  &stop
```

2.1 2. Beispiel

```
1  (in)  grazil ind1(larkin.out(inputdat))
   (out) PLEASE ENTER COMMAND

2  (in)  &wktype 74754
   (out) PLEASE ENTER COMMAND

3  (in)  &vars = H2 O2 H2O H**
   (out) PLEASE ENTER COMMAND

4  (in)  &size A6
   (out) PLEASE ENTER COMMAND
```

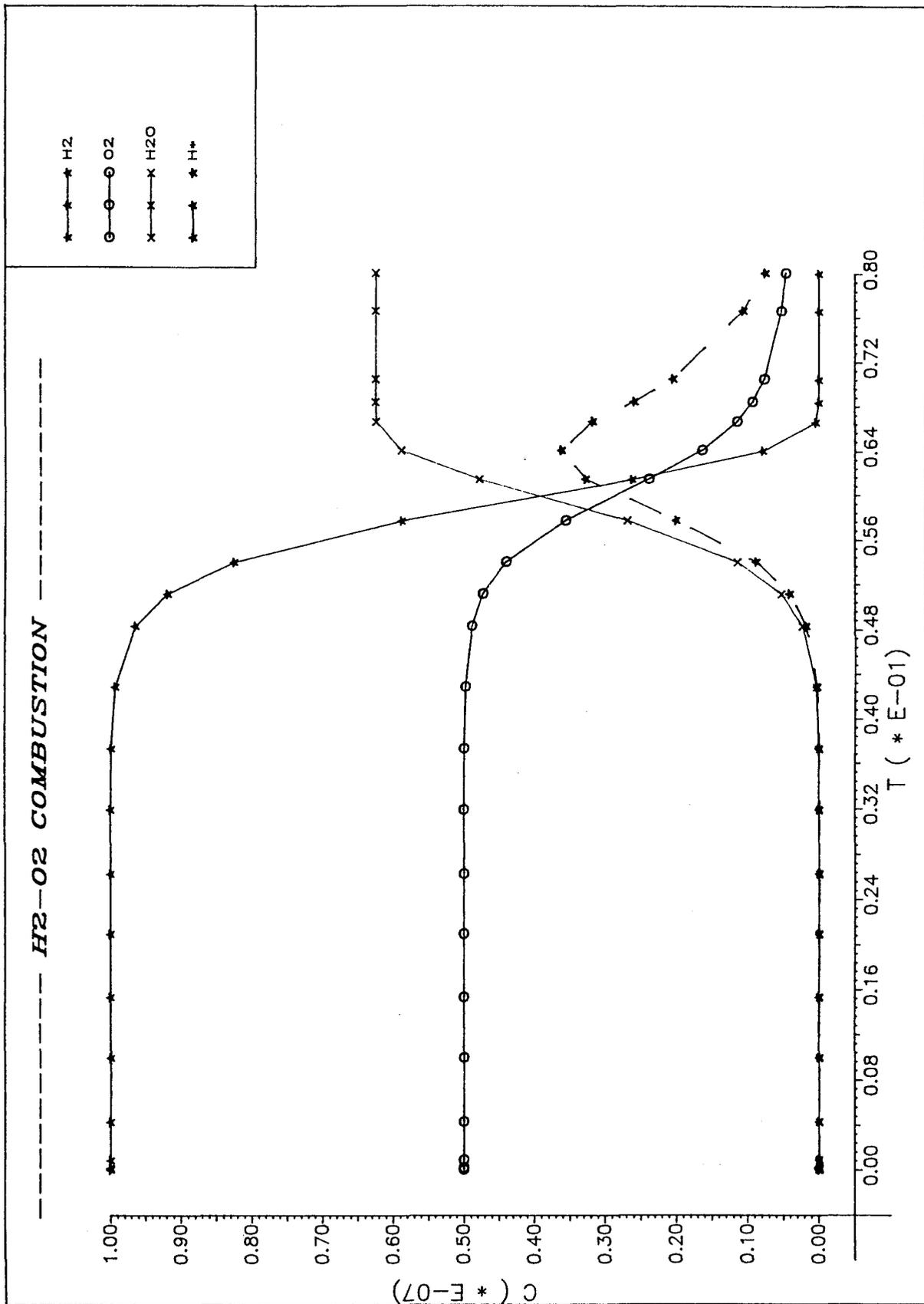


Bild 1

5 (in) &at x 'Zeit'
(out) PLEASE ENTER COMMAND

(in) &atext y 'Konzentration'
(out) PLEASE ENTER COMMAND

6 (in) &headline 1 'Wasserstoff-Sauerstoff-Verbrennung'
(out) PLEASE ENTER COMMAND

7 (in) &headl 2
(out) PLEASE ENTER COMMAND

8 (in) &legendtype 2
(out) PLEASE ENTER COMMAND

9 (in) &curve * = = = 3
(out) PLEASE ENTER COMMAND

10 (in) &bounds x 2 0.55e-1=
(out) PLEASE ENTER COMMAND

11 (in) &grid on
(out) PLEASE ENTER COMMAND
(in) &run
(out) ... -> Bild 2
(out) PLEASE ENTER COMMAND

12 (in) &grid off
(out) PLEASE ENTER COMMAND

13 (in) &headl 1
(out) PLEASE ENTER COMMAND
(in) &headl 2
(out) PLEASE ENTER COMMAND

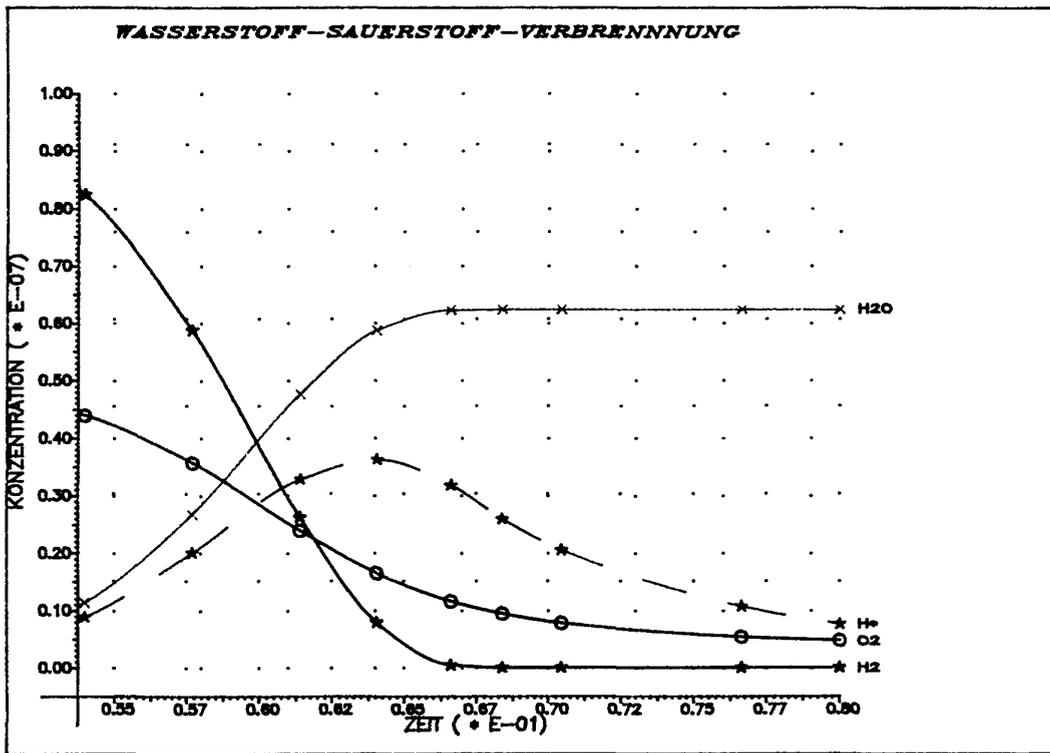


Bild 2

14 (in) &legtype 0
(out) PLEASE ENTER COMMAND

15 (in) &curve * = 1
(out) PLEASE ENTER COMMAND

16 (in) &curve * = = 4
(out) PLEASE ENTER COMMAND

17 (in) &axispec y log
(out) PLEASE ENTER COMMAND
(in) &run
(out) ... -> Bild 3
(out) PLEASE ENTER COMMAND
(in) &stop

2.2 3.Beispiel

18 (in) grazil ind1(formeln.out)
(out) PLEASE ENTER COMMAND

2 (in) &wktype 74754
(out) PLEASE ENTER COMMAND
(in) &size 0,155 0.11
(out) PLEASE ENTER COMMAND

3 (in) &vars = ddf2 ddf3
(out) PLEASE ENTER COMMAND

19 (in) &axispec x = 0.0 0 0
(out) PLEASE ENTER COMMAND

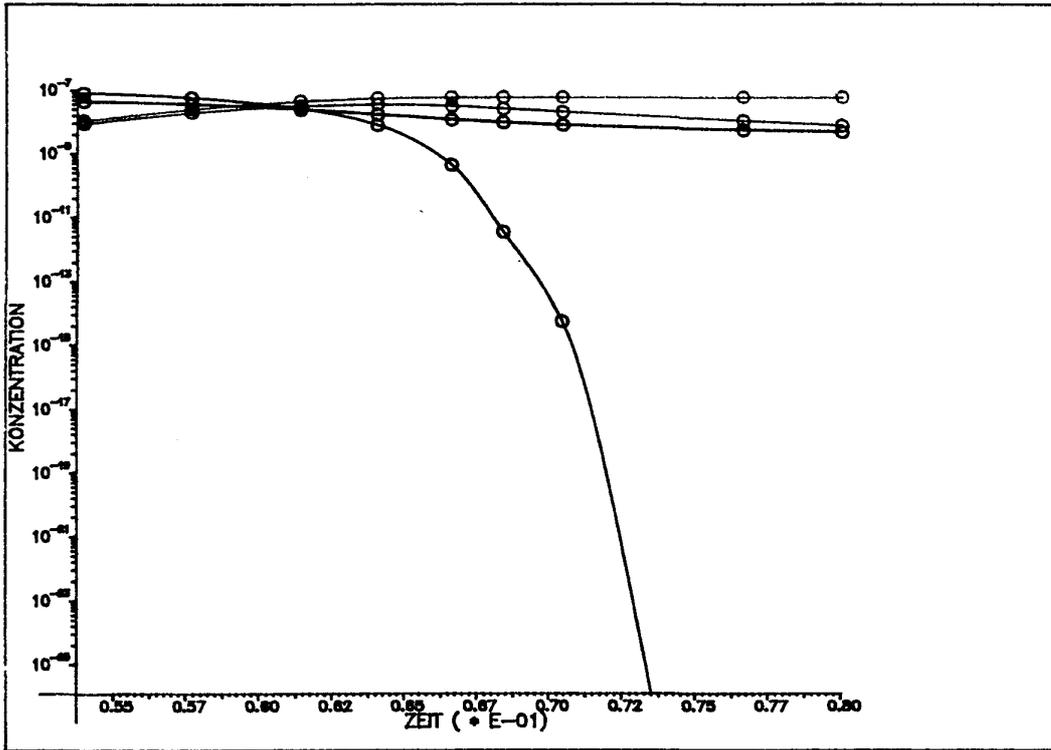


Bild 3

20 (in) &axispec y = 0.0 0 0
(out) PLEASE ENTER COMMAND

21 (in) &curve * = = 0
(out) PLEASE ENTER COMMAND

22 (in) &bounds x 3 -1.2 2.2
(out) PLEASE ENTER COMMAND

23 (in) &bounds y 2 = 0.6e+01
(out) PLEASE ENTER COMMAND

14 (in) &legendtype 0
(out) PLEASE ENTER COMMAND

24 (in) &auxl 1 ver -1.0 = = '-1.0'
(out) PLEASE ENTER COMMAND

25 (in) &auxl 2 ver 1.0 = = '1.0'
(out) PLEASE ENTER COMMAND
(in) &run
(out) ... -> Bild 4
(out) PLEASE ENTER COMMAND

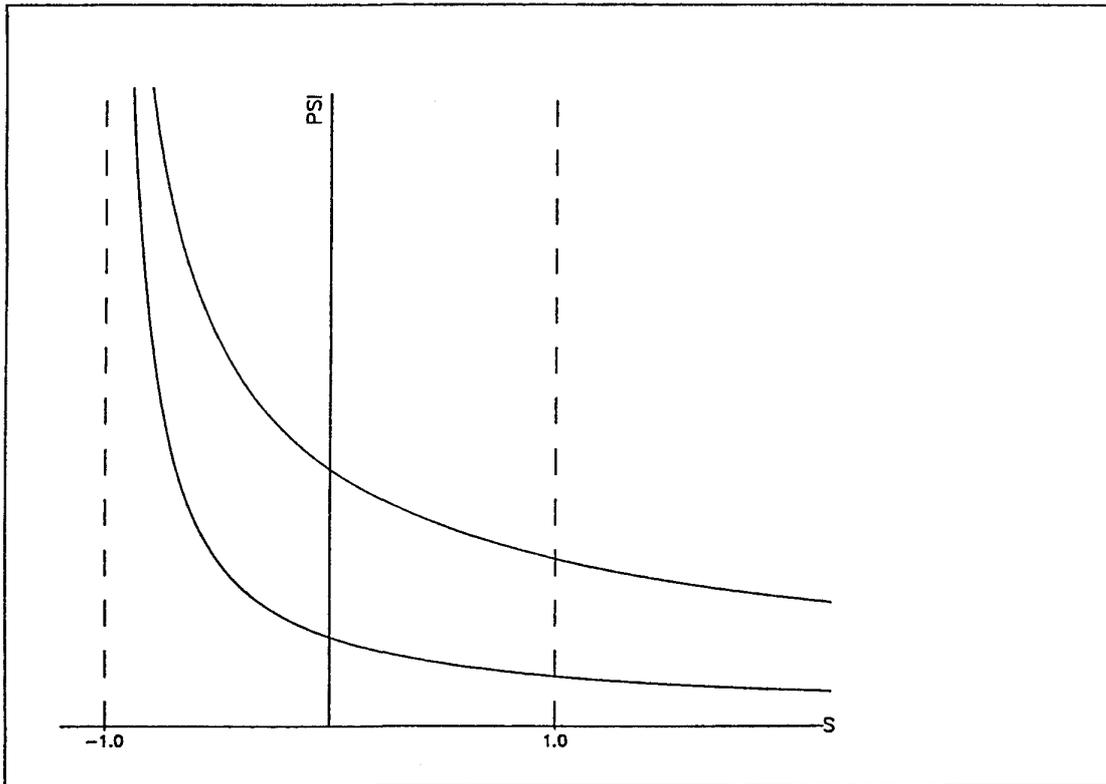


Bild 4

Bemerkungen :

- zu 1. Aufruf des Programms mit der CLIST "GRAZIL", wobei in dem Dataset LARKIN.OUT(INPUTDAT) die Eingabedaten stehen.
- zu 2. Die aktuelle Workstation HP-Tischplotter 74754 (DIN A4) wird eingestellt. Wenn die Ausgabe auf ein Metafile erfolgen soll, wird diese Eingabe nicht benötigt.
- zu 3. Die darzustellenden Kurven werden ausgewählt. "*" schaltet die Suche mit Wildcards aus und sucht nach "*".
- zu 4. Die Zeichenfläche wird auf DIN A6 eingestellt.
- zu 5. Die Achsenbeschriftungen werden geändert.
- zu 6. Die erste Zeile der Überschrift wird geändert.
- zu 7. Die zweite Zeile der Überschrift soll die Voreinstellung der Datendatei zeigen.
- zu 8. Die Kurven sollen an Ihrem Ende beschriftet werden.
- zu 9. Es wird für alle Kurven die Hermite'sche Interpolation ausgewählt.
- zu 10. Die X-Achsendaten werden im Intervall 0.55e-1 bis "=", d.h. die bis dahin gültige obere Grenze wird genommen.
- zu 11. über die Zeichenfläche wird ein Gitter gelegt.
- zu 12. Das Zeichnen des Gitters wird wieder ausgeschaltet.
- zu 13. Die beiden Überschriftenzeilen werden gelöscht.

- zu 14. Es wird keine Legende gezeichnet.
- zu 15. Alle Kurven werden mit Linienart 1 (durchgezogene Linie) gezeichnet.
- zu 16. Alle Kurven werden mit dem Marker 4 (kleiner Kreis) markiert.
- zu 17. Es wird eine logarithmische Einteilung der Y-Achse gewählt.
- zu 18. Aufruf des Programms mit der CLIST "GRAZIL", wobei in dem Dataset FORMELN.OUT die Eingabedaten stehen.
- zu 19. Die Achsenticks für die X-Achse werden ausgeschaltet.
- zu 20. Der Startpunkt der Y-Achse wird auf 0,0 der X-Achse gewählt. Die Achsenticks für die Y-Achse werden ausgeschaltet.
- zu 21. Alle Kurven werden nicht markiert.
- zu 22. Die X-Achsgrenzen werden im Intervall -1,2 bis 2,2 genommen.
- zu 23. Die Y-Achsgrenzen werden im Intervall = bis 6,0, dh. die bis dahin gültige untere Grenze wird genommen.
- zu 24. Die erste Hilfslinie wird senkrecht mit dem Startpunkt -1,0 und der Beschriftung "-1,0" spezifiziert.
- zu 25. Die zweite Hilfslinie wird senkrecht mit dem Startpunkt 1,0 und der Beschriftung "1,0" spezifiziert.

Aufruf und Dateireferenzen des Programmpakets GRAZIL

Folgende DDnames und Datasets werden von GRAZIL verwendet:

- GTEXTFN muß mit dem Dataset 'GRAPHIK.CGM.TEXTFONT' verbunden werden.

- GKSM10 ; hier wird das Metafile geschrieben.

Das Metafile hat folgende DCB-Parameter :

```
RECFM   : FB
BLKSIZ  : LRECL * x
LRECL   : 80
DSORG   : PS | PO
```

- FT16F001 : Auf dem Fortran-Kanal 16 werden die GKS-Meldungen ausgegeben.

- FTxxF001 : Die Fortran-Kanalnr für die Inputdaten wird dem Hauptprogramm über Parameter beim CALL - Aufruf übergeben. Es muß mindestens eine Fortran-Kanalnummer angegeben werden. Maximal können 4 Parameter dem Hauptprogramm übergeben werden. Für die Inputdateien gelten folgende DCB-Parameter:

```
RECFM   : FB
BLKSIZ  : LRECL * x
LRECL   : 80
DSORG   : PS | PO
```

Eingabedateien werden von Spalte 1 bis Spalte 80 gescannt. Man beachte also die Numerierung in Spalte 72 - 80.

- FT21F001 : Auf dem Fortran-Kanal 21 benötigt das Programm eine interne Arbeitsdatei mit folgenden DCB-Parametern

RECFM : F
LRECL : 320
BLKSIZ : 320
DSORG : DA
SPACE : 512,1

- FT30F001 : Auf dem Fortran-Kanal 30 wird der Plotoutput für den Benson - Plotter ausgegeben.

Für eine Ausgabe auf eine Datei gelten folgende DCB-Parameter:

RECFM : FB
LRECL : 80
BLKSIZ : LRECL * x
DSORG : PS | PO

- FT31F001 : Auf dem Fortran-Kanal 31 wird der Plotoutput für den HP7475 - Plotter ausgegeben.

Für eine Ausgabe auf eine Datei gelten folgende DCB-Parameter:

RECFM : FB
LRECL : 80
BLKSIZ : LRECL * x
DSORG : PS | PO

- FT41F001 : Auf dem Fortran-Kanal 41 kann eine Datei zur Sicherung der gegebenen Kommandos angegeben werden.

Es gelten dann folgende DCB-Parameter:

RECFM : FB
LRECL : 80
BLKSIZ : LRECL * x
DSORG : PS | PO

Leistungsmerkmale

Maximale Anzahl der (X,Y)-Punkte pro Variable : 512
Maximale Anzahl von Variablen : 128
Maximale Anzahl von darzustellenden Variablen : 9
Maximale Anzahl von Hilfslinien : 10
Maximallänge der Variablennamen : 12 Buchstaben
Maximallänge des Headlinetextes : 50 Buchstaben
Maximallänge des Achsentextes : 20 Buchstaben
Maximallänge des Hilfslinientextes : 20 Buchstaben
Die maximale Anzahl der Inputdaten ist abhängig von der Struktur
der Daten und der Größe der internen Arbeitsdatei.

Literatur:

- /BEC1/ J. Bechlars, R. Buhtz: **GKS in der Praxis.**
Springer-Verlag 1986.
- /BEC2/ J. Bechlars, R. Buhtz, W. Hecht: **BIZEPS Version 2.1
Benutzerhandbuch.** Freie Universität Berlin, Zentral-
einrichtung für Datenverarbeitung 1984.
- /BUH/ R. Buhtz, J. Langendorf, O. Paetsch, D. A. Buhtz: **ZUGRIFF
Eine vereinheitlichte Datenspezifikation für graphische
Darstellungen und ihre graphische Aufbereitung.** Konrad Zuse-
Zentrum für Informationstechnik Berlin, Technical Report 87-
6 1987.
- /DEU/ P. Deuflhard, G. Bader, U. Nowak: **LARKIN - A Software
Package for the numerical Simulation of large Systems in
chemical Reaction Kinetics.** In: K. H. Ebert, P. Deuflhard,
W. Jaeger (ed): Modeling of chemical Reaction Systems.
Springer Ser. Chem. Phys. 18, 1981.

