

KfK 4275 B

Juni 1987

# **Rechnerprogramm zur Auswertung einfacher Reaktionsmodelle des Aufbruchs nuklearer Projektile**

J. Oehlschläger, H.-U. Hohn, V. Corcalciuc

Institut für Kernphysik

Hauptabteilung Datenverarbeitung und Instrumentierung

**Kernforschungszentrum Karlsruhe**



KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Kernphysik  
Hauptabteilung Datenverarbeitung und Instrumentierung

KfK 4275 B

RECHNERPROGRAMM ZUR AUSWERTUNG  
EINFACHER REAKTIONSMODELLE  
DES AUFBRUCHS NUKLEARER PROJEKTILE

Jürgen Oehlschläger  
Hans-Ulrich Hohn  
Valentin Corcalciuc<sup>+</sup>

<sup>+</sup> Central Institute of Physics, Bucharest, Romania.

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

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

Kernforschungszentrum Karlsruhe GmbH  
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

## ZUSAMMENFASSUNG

### Rechnerprogramm zur Auswertung einfacher Reaktionsmodelle des Aufbruchs nuklearer Projektile

Es wird ein komfortables Fortran77-Programm zur ausführlichen tabellarischen und grafischen Auswertung von einfachen Reaktionsmodellen (nach Serber) für den Projektilaufbruch von Atomkernen beschrieben. Im Rahmen dieser Modelle, wobei die ein- und auslaufenden Teilchen durch ebene Wellen beschrieben werden, wird der Projektilaufbruch als quasi freier Prozeß behandelt, bei dem das beobachtete der beiden Aufbruchfragmente den Targetkern unbeeinflusst als Spektator passiert, während das unbeobachtete Fragment elastisch oder inelastisch mit dem Targetkern wechselwirkt. Das Modell erlaubt die Berechnung der Winkelverteilung  $d\sigma/d\Omega$ , der Energieverteilung  $d\sigma/dE$  und des doppelt-differentiellen Wirkungsquerschnitts  $d^2\sigma/(d\Omega dE)$  der beobachteten Spektatorfragmente, wobei der Targetkern im ersten Fall dem beobachteten Spektatorteilchen vollkommen transparent, im zweiten Fall opak erscheint. Folgende Modellkorrekturen sind zudem möglich: Coulomb energy differences corrections (CEDC), Coulomb deflection corrections (CDC), CEDC- und CDC-Korrektur.

## ABSTRACT

### Computer program for evaluation of simple models of break-up of nuclear projectiles

A Fortran-77 computer code SERBER is described. The code allows a comfortable treatment of projectile break-up reactions within the framework of a simple reaction model (Serber-model) including graphical plots of the calculated quantities. The Serber-model takes the incoming and outgoing particles to be described by plane waves and treats the projectile break-up as a quasi-free process. The observed break-up fragment is assumed to pass the target nucleus without any interaction as a spectator, whereas the unobserved particle interacts elastically or inelastically with the target nucleus. The model allows the calculation of the angular distribution  $d\sigma/d\Omega$ , the energy distribution  $d\sigma/dE$ , and the double differential cross-section  $d^2\sigma/(d\Omega dE)$  of the observed fragment. The target nucleus can be chosen to appear either transparent or opaque for the observed fragment. Coulomb energy differences corrections or Coulomb deflection corrections are optional.

INHALTSVERZEICHNIS

Zusammenfassung . . . . .	iii
Abstract . . . . .	iv
Inhaltsverzeichnis . . . . .	v
1. Einfache Reaktionsmodelle des Aufbruchs nuklearer Projektile	1
2. Eingabe zum Serber-Programm	
2.1 EINGABEGRÖßEN IN DER GEFORDERTEN REIHENFOLGE . . . . .	3
2.2 TABELLE DER MÖGLICHEN PUNKTSYMBOLS . . . . .	6
2.3 BEISPIELSATZ VON EINGABEKARTEN . . . . .	7
2.4 VOLLSTÄNDIGER BEISPIELJOB . . . . .	9
2.5 AUSGABE DIESES JOBS . . . . .	14
2.6 DIE ZWEI ZEICHNUNGEN DIESES JOBS . . . . .	20
2.7 BEISPIELJOB ZU ALLEN MÖGLICHEN 6 BILDERN . . . . .	22
3. Das Hauptprogramm "SERBER"	
3.1 BESCHREIBUNG DER VARIABLEN . . . . .	25
3.2 ABDRUCK DES HAUPTPROGRAMMS . . . . .	32
4. Beschreibung der Unterprogramme	
4.1 AEQUID . . . . .	71
Kurvenpunkte mit gleichem Abstand	
4.2 AUSGL4 . . . . .	74
Ausgleichspolynom vom Grad 4	
4.3 PLSYMB . . . . .	82
Zentrierte Plotsymbole	

4.4	P12LF . . . . .	85
	Test der Legendre-Funktion $P_{1/2}(s)$ , $s \geq 1$	
	CELINT . . . . .	89
	Vollst. elliptisches Integral 1./2.Art	
	F21HYP . . . . .	91
	Hypergeometrische Reihe ${}_2F_1(a,b;c;x)$	
4.5	SERBXY . . . . .	100
	Plotten von Texten	
4.6	SPLINE . . . . .	105
	Kubische Spline-Interpolation	
4.7	STRIPU . . . . .	110
	Zeichnen von Kurven: gestrichelt etc.	
4.8	SYMBGR . . . . .	118
	Kleine griechische Buchstaben	
4.9	ZABSCH . . . . .	128
	Kurve oben/unten abschneiden	
4.10	ZXYAXN . . . . .	133
	Achsenlinien mit Strichelungen	
4.11	ZXYIBE . . . . .	136
	Beschriftungseinheit	
4.12	ZXYMAS . . . . .	139
	Beschriftungsmaßstab	
4.13	ZXZIFF . . . . .	143
	Abszisse Beziffern	
4.14	ZYZIFF . . . . .	146
	Ordinate Beziffern	

ANHANG A.

PROGRAMMIERTE FORMELN . . . . .	149
PLOTTING SUBROUTINES USED (SYSTEM) . . . . .	157



## 1. EINFACHE REAKTIONSMODELLE DES AUFBRUCHS NUKLEARER PROJEKTILE

Ein einfaches Reaktionsmodell zur Behandlung und Beschreibung des Aufbruchs nuklearer Projektile ist von Serber<sup>1</sup> ursprünglich zur Beschreibung des Deuteronenaufbruchs entwickelt und neuerdings auch zur Deutung des Aufbruchs von <sup>3</sup>He, <sup>4</sup>He, <sup>6</sup>Li und <sup>7</sup>Li herangezogen worden.<sup>2</sup> Im Rahmen dieses Modells, in dem ein- und auslaufende Teilchen durch ebene Wellen beschrieben werden, wird der Projektilaufbruch als quasi freier Prozeß behandelt, bei dem das beobachtete der beiden Aufbruchfragmente den Targetkern unbeeinflusst als Spektator passiert, während das unbeobachtete Fragment elastisch oder inelastisch mit dem Targetkern wechselwirkt. Es wird dabei davon ausgegangen, daß die kinetische Energie des Projektils wesentlich größer als die Fermienergie der Cluster ist. Die Fermibewegung der beiden Clusterfragmente wird durch eine Yukawa-Wellenfunktion beschrieben:

$$\phi_{\text{Yuk}}(r) = N \cdot (e^{-\alpha r}/r) \cdot Y_{00}(\Omega) \quad ,$$

$$\alpha = \sqrt{\frac{2 \mu \varepsilon}{\hbar^2}} \quad ,$$

---

<sup>1</sup> [1] R.Serber, Phys. Rev. 72 (1947) 1008.

<sup>2</sup> [2] N.Matsuoka, A.Shimizu, K.Hosonto, T.Saito, M.Kondo, H.Sakaguchi, Y.Toba, A.Gota, F.Ohtani und N.Nahanishi, Nucl. Phys. A311 (1978) 173.

[3] J.R.Wu, C.C.Chang und H.D.Holmgren, Phys. Rev. C19 (1979) 659.

[4] B.Neumann, KfK-Report 2887, Dez 1979.

[5] H.Jelitto, KfK-Report 4259, in Vorbereitung.

[6] V.Corcalciuc, D.Galeriu, R.Dumitrescu, H.Rebel, H.J. Gils und N.Heide, interner Bericht des KfK (März 1987).

wobei  $\epsilon$  die Bindungsenergie des Projektils,  $N$  die Normierungskonstante der Wellenfunktion,  $\mu$  die reduzierte Masse der Clusterfragmente,  $h$  das Plancksche Wirkungsquantum,  $Y_{00}$  die sphärische Kugelfunktion und  $r$  den Relativabstand der Clusterfragmente bedeuten. Das Modell erlaubt die Berechnung der Winkelverteilung  $d\sigma/d\Omega$ , der Energieverteilung  $d\sigma/dE$  und des doppelt-differentiellen Wirkungsquerschnitts  $d^2\sigma/(d\Omega dE)$  der beobachteten Spektatorfragmente. Hierbei werden zwei Extremfälle unterschieden: Im ersten Fall erscheint der Targetkern dem beobachteten Spektatorteilchen vollkommen transparent, d.h. das beobachtete Teilchen bleibt unbeeinflusst, selbst wenn es den Targetkern durchquert. Im zweiten Fall wird davon ausgegangen, daß das Teilchen beim Durchqueren des Targetkerns absorbiert wird, der Targetkern erscheint opak.

Dieses Programm wurde angeregt durch die experimentellen Arbeiten des Instituts für Kernphysik am  ${}^6\text{Li}$ -Strahl des Zyklotrons und am  ${}^7\text{Li}$ -Strahl des Bukarester Teilchenbeschleunigers. Wir danken Herrn Dipl. Phys. N. Heide für fachliche Beratung in kernphysikalischen Fragen.

## 2. EINGABE ZUM SERBER-PROGRAMM

### Anmerkung:

Alle im Hauptprogramm und in den Unterprogrammen benutzten Variablen genügen der impliziten Fortran-Vereinbarung `IMPLICIT REAL*8 (A-H,O-Y), REAL*4 (Z), INTEGER*4 (I-N)`. Alle Variablen mit dem ersten Buchstaben Z werden beim Plotten verwendet. "Eine Datenkarte" besteht immer aus einer ganzen Kommentarzeile, die ignoriert wird, und aus einer Wertzeile, welche jeweils die richtige Anzahl mit dem richtigen Typ der Eingabegrößen enthalten muß. Es gibt keine Eingabezeilen mit gemischten Typen von Variablen.

### 2.1 EINGABEGRÖßEN IN DER GEFORDERTEN REIHENFOLGE

Name = Bedeutung

- II = Auswahlziffer des zu berechnenden Modells. Gültige Werte sind 1, 2 oder 3.
- 1 : Transparentes Serber-Modell.
  - 2 : Opakes Serber-Modell.
  - 3 : Transparentes und Opakes Serber-Modell.
- Bei ungültigem Wert bricht das Programm sofort ab.
- NPHAS = Auswahlziffer zum Plotten der Phasenraumfaktoren, die bei  $J=1$  bzw.  $J=3$  berechnet und verwendet werden.  $NPHAS > 0$  bedeutet Plotten der Phasenraumfaktoren. Das Maximum dieser Kurven ist dann immer halb so groß wie die zugehörige Energiekurve. Bei  $J=2$  wird diese Eingabe ignoriert.
- JJ(1:3) = Integerfeld des Parameters J. Gültige Werte sind 1, 2, 3. Für jeden anderen eingegebenen Wert erfolgt keine Berechnung und keine Zählung.
- 1 : Energieverteilung.
  - 2 : Winkelverteilung.
  - 3 : Doppelt differentieller Wirkungsquerschnitt.

Das Programm verlangt *immer* drei Eingabewerte für das Feld JJ. Folgende Eingabe ist z.B. sinnvoll: 1 0 2. Der Wert 0 wird ignoriert, 1 und 2 sind sinnvolle Werte. Wie man sieht, ist es nicht notwendig, die maximale Anzahl von Berechnungen voll auszuschöpfen. Die *sortierte* Eingabe der J-Werte und der nachfolgend erwarteten K-Werte wird dringend empfohlen, weil sich die Reihenfolge der Sollwerte YNORM der Kurvenmaxima auf die aufsteigenden J- und K-Werte bezieht.

Nur für jeden *gültigen* J-Wert müssen vier K-Werte *und* vier Normierungsgrößen eingelesen werden:

KK(1:4) = Integerfeld des Parameters K. Gültige Werte sind:

- 0 : Keine Korrektur des Modells.
- 1 : Coulomb energy differences corrections CEDC.
- 2 : Coulomb deflection corrections CDC.
- 3 : CEDC- und CDC- Korrektur.

Für jeden anderen eingegebenen Wert erfolgt keine Berechnung. Bei J=1 und J=3 sind nur die K-Werte 0 und 1 sinnvoll. Nur bei J=2 können alle vier gültigen K-Werte eingegeben werden. *Aber:* Das Programm verlangt *immer* vier Eingabewerte für das Feld KK. Für J=1 und J=3 ist z.B. folgende Eingabe für KK(i) sinnvoll: 0 5 -6 1. Die Werte 5 und -6 werden ignoriert, 0 und 1 sind sinnvolle Werte. Es ist auch möglich, nur einen Teil der Berechnungen ausführen zu lassen. Für J=2 ist dann z.B. folgende Eingabe für KK in Ordnung: 0 3 1 82. Bearbeitet werden dabei nur 0, 1, 3.

YNORM(1:4,1:3) = Maximum-Sollwerte der Kurven bezogen auf die J-Werte. Bei J=1 wird YNORM(1:4,1) gelesen, bei J=2 YNORM(1:4,2) und bei J=3 YNORM(1:4,3).

AMT,AZT = Masse (MeV) und Kernladungszahl des Targets.

AMP,AZP = Masse (MeV) und Kernladungszahl des Projektils.

AMA,AZA = Masse (MeV) und Kernladungszahl des beobachteten Fragments.

AMB,AZB = Masse (MeV) und Kernladungszahl des unbeobachteten Fragments.

EO = Energie (MeV) des Projektils im Laborsystem.

- EPSAB = Bindungsenergie der Cluster A und B im Projektil.  
RO = Konstante zur Berechnung des Kernradius.  
QU, CREAKT\*40 = Q-Wert der Reaktion  $P+T \rightarrow A+B+T$  (in MeV)  
und die Reaktionsformel (werden zusammen in die  
Zeichenkette CZEILE\*80 gelesen, dann aufgeteilt).  
EAMIN, EAMAX = Untere Schranke und obere Schranke der  
der Energie des beobachteten Teilchens (in MeV).  
EADEL = Schrittweite der Energie des beobachteten  
Teilchens (in MeV). EADEL wird gegebenenfalls  
vergrößert, damit nicht mehr als NJ=1000  
Punkte berechnet werden.  
WIMIN = Untere Schranke des Laborwinkels des beobachteten  
Teilchens (in Grad).  
WIMAX = Obere Schranke des Laborwinkels des beobachteten  
Teilchens (in Grad).  
WIDEL = Schrittweite des Laborwinkels des beobachteten  
Teilchens (in Grad). WIDEL wird gegebenenfalls  
vergrößert, damit nicht mehr als NJ=1000  
Punkte berechnet werden.  
WSTREU = Streuwinkel (Laborwinkel) des beobachteten  
Teilchens (in Grad) als Parameter bei der Berechnung  
des doppelt differentiellen Wirkungsquerschnitts.  
ZXI, ZYI, ZIFHO, ZEIHO = Länge der Abszisse und Ordinate, Höhe  
der Ziffern bzw. der Zeichen (in Inches).  
NST(1:4) = Stricharten NS der (maximal) 4 Kurven.  
0 : keine Kurvenlinie eintragen.  
1 : durchgezogene Kurve.  
2 : gestrichelte Kurve.  
3 : punktierte Kurve.  
4 : strichpunktierte Kurve.  
5 : strichdoppelpunktierte Kurve.

Bei positiven Werten der Strichart NS wird gegebenenfalls SPLINE-Interpoliert, um dichter liegende Punkte zu bekommen. (Bei negativen Werten der Strichart NS werden mit der Subroutine AUSGL4 die Koeffizienten eines Ausgleichspolynom vom Grad 4 berechnet, um damit andere und mehr Stützpunkte zu berechnen. Diese Anwendung ist beim SERBER-Programm i. a. nicht

sinnvoll.)

ZST(1:4)= Abstände bei Strichelung der Kurven (in Inches).

NSY(1:4)= Symbolnummern der Kurvensymbole (siehe "Tabelle der möglichen Punktsymbole" unten).

ZSY(1:4)= Größe der Kurvensymbole (in Inches).

NEXTR = Auswahlmöglichkeit, ob externe Datentabellen (eventuell mit Fehlerbalken) in ein Bild eingetragen werden sollen (NEXTR  $\leq$  0 "nein",  $>$  0 "ja").

NEXTR = Wenn überhaupt externe Datentabellen verarbeitet werden sollen, dann ist NEXTR jetzt die Anzahl dieser externen Datentabellen (maximal 10).

NEX(1:NEXTR) = Fortran-Referenz-Nummern der NEXTR externen Datentabellen (maximal 10). Danach folgen dann noch die über die Fortran-Referenz-Nummern in NEX definierten Dateien mit den Datentabellen (formatfrei).

Erste Zeile : I-Wert J-Wert Punkteanzahl  
Weitere Zeilen: x(i)  $\Delta$ x(i) y(i)  $\Delta$ y(i)

## 2.2 TABELLE DER MÖGLICHEN PUNKTSYMBOLS

Werte für das Feldes NSY(1:4). Zahlenwerte NSY=0(1)13 liefern automatisch zentrierte Symbole:

<i>Symbolnummer</i>	<i>Symbol</i>
0	"Quadrat"
1	"Achteck"
2	"Dreieck"
3	"Plus"
4	"Kreuz"
5	"Raute"
6	"Kreuz mit halber Raute"
7	"Kreuz mit Dach"
8	"Zet"
9	"Ypsilon"
10	"Kreuzquadrat"
11	"Stern"
12	"Sanduhr"
13	"Senkrechter Strich"

Zahlenwerte NS=-1(-1)-40 liefern zentrierte Symbole aus der Zeichenkette "ABCDEFGHIJKLMNOPQRSTUVWXYZ\$#@0123456789 ".

<i>Symbolnummer</i>	<i>Symbol</i>
- 1	"A"
.	.
-26	"Z"
-27	"\$"
-28	"#"
-29	"@"
-30	"0"
.	.
-39	"9"
-40	" " (bei unvernünftigem Wert)

### 2.3 BEISPIELSATZ VON EINGABEKARTEN

- \*\*I\*\* 1=TRANSPARENT, 2=OPAKES, 3=TRANSPARENT+OPAKES SERBER-MODELL  
3
- \* PHASENRAUMFAKTOREN PLOTTEN, WENN VORHANDEN ("JA">0) (INTEGER)  
1
- \*\*J\*\* 1=ENERGIE-VERT., 2=WINKEL-VERT., 3=INKLUSIV. WIRK.QUERSCHN.  
1 2 3
- \*\*K\*\* 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR  
0 1 2 3
- \* MAXIMUM-SOLLWERTE DER KURVEN (ERSTER GUELTIGER J-WERT).  
300. -1. -1. -1.
- \*\*K\*\* 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR  
0 1 2 3
- \* MAXIMUM-SOLLWERTE DER KURVEN (NAECHSTER GUELTIGER J-WERT).  
-1. -1. -1. -1.
- \*\*K\*\* 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR  
0 1 2 3
- \* MAXIMUM-SOLLWERTE DER KURVEN (NAECHSTER GUELTIGER J-WERT).  
-1. -1. -1. -1.
- \* MASSE (MEV) UND KERNLADUNGSZAHL DES TARGETS  
25340.798 13.
- \* MASSE (MEV) UND KERNLADUNGSZAHL DES PROJEKTILS

6533.722            3.  
\* MASSE (MEV) UND KERNLADUNG DES BEOBACHTETEN FRAGMENTS  
3727.315            2.  
\* MASSE (MEV) UND KERNLADUNG DES UNBEOBACHTETEN FRAGMENTS  
2808.873            1.  
\* ENERGIE (MEV) DES PROJEKTILS IM LABORSYSTEM  
29.6  
\* BINDUNGSENERGIE DER CLUSTER A UND B IM PROJEKTIL  
2.47  
\* KONSTANTE ZUR BERECHNUNG DES KERNRADIUS  
1.4  
\* Q-WERT DER REAKTION P+T ----> A+B+T  
2.5  
\* UNTERE SCHRANKE ENERGIE (IN MEV)  
11.  
\* OBERE SCHRANKE ENERGIE (IN MEV)  
21.  
\* SCHRITTWEITE DER ENERGIE (IN MEV)  
0.1  
\* UNTERE SCHRANKE DES WINKELS (IN GRAD) (INTEGER)  
-80  
\* OBERE SCHRANKE DES WINKELS (IN GRAD) (INTEGER)  
80  
\* SCHRITTWEITE DES WINKELS (IN GRAD) (INTEGER)  
1  
\* STREUWINKEL ALS PARAMETER DES DOPP-DIFF-WIQU'S (BEI J=3)  
12.5  
\* PLOT: ABSZISSEN- & ORDINATEN-LAENGE, ZIFFERN-, ZEICHEN-HOEHE  
8.            12.            0.2            0.15  
\* PLOT: STRICHART DER KURVE (INTEGER 0-5)  
1            3            5            2  
\* PLOT: ABSTAENDE BEI STRICHELUNG DER KURVE (REAL)  
0.1            0.1            0.1            0.1  
\* PLOT: SYMBOLNUMMERN DER KURVENSYMBOLE (INTEGER)  
-40            -40            -40            -40  
\* PLOT: GROESSE DER KURVENSYMBOLE (REAL)  
0.14            0.14            0.14            0.14  
\* SOLLEN EXTERNE TABELLEN VERARBEITET WERDEN? (INTEGER)  
0



WICHTIG: "Eine Datenkarte" besteht immer aus einer *ganzen* Kommentarzeile die ignoriert wird, und aus einer Wertezeile, welche jeweils die richtige Anzahl mit dem richtigen Typ der Eingabegrößen enthalten muß. Es gibt keine Eingabezeilen mit gemischten Typen von Variablen.

## 2.4 VOLLSTÄNDIGER BEISPIELJOB

```
//ADI930B JOB (0930,545,P0000),HOHN,NOTIFY=ADI930,MSGCLASS=H,
//          MSGLEVEL=(1,1),REGION=2000K
//*MAIN LINES=19
// EXEC F7C,PARM.C='LANGLVL(77),LINECOUNT(63) '
//C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=TSO964.SAMMEL.PLOT(AUSGL4)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PLOT(SPLINE)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(AEQUID)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(PLSYMB)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(STRIPU)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(SYMBGR)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZABSCH)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXYAXN)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXYIBE)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXYMAS)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXZIFF)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZYZIFF)
//          DD DISP=SHR,DSN=ADI930.SERBER2.FORT(SERBXY)
// EXEC F7CLG,PLOT=VERSATEC,
//          PARM.C='LANGLVL(77),LINECOUNT(63) ',PARM.G='SIZE(2000) '
//* C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=ADI930.SERBER2.FORT(SERBER)
//L.SYSPRINT DD DUMMY
//L.SYSIN DD *
ENTRY SERBER
```

//G.SYSIN DD \*

\*\*I\*\* 1=TRANSPARENT, 2=OPAKES, 3=TRANSPARENT+OPAKES SERBER-MODELL

3

\* PHASENRAUMFAKTOREN PLOTTEN, WENN VORHANDEN ("JA">0) (INTEGER)

1

\*\*J\*\* 1=ENERGIE-VERT., 2=WINKEL-VERT., 3=INKLUSIV. WIRK.QUERSCHN.

3 -1 -1

\*\*K\*\* 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR

0 1 0 1

\* MAXIMUM-SOLLWERTE DER KURVEN (ERSTER GUELTIGER J-WERT).

156.35 156.35 -1. -1.

\* MASSE (MEV) UND KERNLADUNGSZAHL DES TARGETS

193683.779 82.

\* MASSE (MEV) UND KERNLADUNGSZAHL DES PROJEKTILS

5601.420 3.

\* MASSE (MEV) UND KERNLADUNG DES BEOBACHTETEN FRAGMENTS

3727.315 2.

\* MASSE (MEV) UND KERNLADUNG DES UNBEOBACHTETEN FRAGMENTS

1875.580 1.

\* ENERGIE (MEV) DES PROJEKTILS IM LABORSYSTEM

156.0

\* BINDUNGSENERGIE DER CLUSTER A UND B IM PROJEKTIL

1.47

\* KONSTANTE ZUR BERECHNUNG DES KERNRADIUS

1.4

\* Q-WERT DER REAKTION P+T ----> A+B+T

1.47

\* UNTERE SCHRANKE ENERGIE (IN MEV)

70.

\* OBERE SCHRANKE ENERGIE (IN MEV)

130.

\* SCHRITTWEITE DER ENERGIE (IN MEV)

0.5

\* UNTERE SCHRANKE DES WINKELS (IN GRAD) (INTEGER)

-80

\* OBERE SCHRANKE DES WINKELS (IN GRAD) (INTEGER)

80

\* SCHRITTWEITE DES WINKELS (IN GRAD) (INTEGER)

1

```
* STREUWINKEL ALS PARAMETER DES DOPP-DIFF-WIQU'S (BEI J=3)
  10.0
* STREUWINKEL ALS PARAMETER DES DOPP-DIFF-WIQU'S (BEI J=3)
  10.0
* PLOT: ABSZISSEN- & ORDINATEN-LAENGE, ZIFFERN-, ZEICHEN-HOEHE
  6.6      8.8      0.2      0.15
* PLOT: STRICHART DER KURVE (INTEGER 0-5)
  1        3        5        2
* PLOT: ABSTAENDE BEI STRICHELUNG DER KURVE (REAL)
  0.1      0.1      0.1      0.1
* PLOT: SYMBOLNUMMERN DER KURVENSYMBOLE (INTEGER)
 -40      -40      -40      -40
* PLOT: GROESSE DER KURVENSYMBOLE (REAL)
  0.14     0.14     0.14     0.14
* SOLLEN EXTERNE TABELLEN VERARBEITET WERDEN? (INTEGER)
  1
* WIEVIELE EXTERNE TABELLEN SOLLEN VERARBEITET WERDEN?
  2
* FORTRAN-REFERENZ-NUMMER(N) DER TABELLEN?
  1  2
//G.FTO1FO01 DD DISP=SHR,DSN=ADI930.SERBER2.FORT($SERDAT1)
//*  erste Zeile:      1      3      59
//*  weitere Z.:      x(i)   Δx(i)   y(i)   Δy(i)
//G.FTO2FO01 DD DISP=SHR,DSN=ADI930.SERBER2.FORT($SERDAT2)
//*  erste Zeile:      2      3      59
//*  weitere Z.:      x(i)   Δx(i)   y(i)   Δy(i)
//G.FTO6FO01 DD SYSOUT=*
//G.SYSLOUT DD DUMMY
//G.PLOTLOG DD DUMMY
//G.PLOTPARM DD *
  &PLOT XMAX=25. SPACE=26. &END
// EXEC SVPLOT
//SORT.SYSOUT DD DUMMY
//PLOT.PLOTLOG DD DUMMY
//PLOT.SYSVECTR DD DUMMY
//
```

INHALT DER DATEIEN "\$SERDATi"

i	3	59		
71.0000000	0.50000000	9.20197773	0.514405906	
72.0000000	0.50000000	16.1322021	0.681102097	
73.0000000	0.50000000	18.2601624	0.724632025	
74.0000000	0.50000000	20.8482208	0.774282932	
75.0000000	0.50000000	22.6598663	0.807223380	
76.0000000	0.50000000	27.8647156	0.895143151	
77.0000000	0.50000000	27.8647156	0.895143151	
78.0000000	0.50000000	29.1299896	0.915241420	
79.0000000	0.50000000	32.0918732	0.960645139	
80.0000000	0.50000000	33.3571625	0.979399681	
81.0000000	0.50000000	36.7791290	1.02840900	
82.0000000	0.50000000	38.2457123	1.04871273	
83.0000000	0.50000000	40.8337555	1.08361435	
84.0000000	0.50000000	44.7733612	1.13468456	
85.0000000	0.50000000	46.0673828	1.15096474	
86.0000000	0.50000000	48.8854980	1.18564701	
87.0000000	0.50000000	50.6683807	1.20707417	
88.0000000	0.50000000	58.5188141	1.29721832	
89.0000000	0.50000000	62.6309509	1.34202290	
90.0000000	0.50000000	67.6632690	1.39489651	
91.0000000	0.50000000	76.3188782	1.48143101	
92.0000000	0.50000000	84.5431519	1.55920982	
93.0000000	0.50000000	92.9112091	1.63455391	
94.0000000	0.50000000	100.819153	1.70269489	
95.0000000	0.50000000	111.200150	1.78820705	
96.0000000	0.50000000	119.683212	1.85516167	
97.0000000	0.50000000	130.926895	1.94034863	
98.0000000	0.50000000	138.518539	1.99581051	
99.0000000	0.50000000	140.358917	2.00902462	
100.000000	0.50000000	151.573792	2.08774471	
101.000000	0.50000000	146.944092	2.05561256	
102.000000	0.50000000	150.394699	2.07960892	
103.000000	0.50000000	156.347290	2.12036514	
104.000000	0.50000000	149.043213	2.07024288	
105.000000	0.50000000	145.189941	2.04330730	
106.000000	0.50000000	136.476837	1.98104572	
107.000000	0.50000000	127.476135	1.91460800	

108.000000	0.50000000	118.389191	1.84510612
109.000000	0.50000000	109.446030	1.77404785
110.000000	0.50000000	98.8349762	1.68585777
111.000000	0.50000000	88.1089325	1.59175205
112.000000	0.50000000	76.8365021	1.48644638
113.000000	0.50000000	68.0946350	1.39933491
114.000000	0.50000000	58.9214020	1.30167198
115.000000	0.50000000	52.0774384	1.22374249
116.000000	0.50000000	49.9782257	1.19882488
117.000000	0.50000000	42.7891846	1.10925770
118.000000	0.50000000	38.9933624	1.05891418
119.000000	0.50000000	35.1112823	1.00482082
120.000000	0.50000000	31.6605377	0.954167247
121.000000	0.50000000	27.6059265	0.890976548
122.000000	0.50000000	23.7813416	0.826958299
123.000000	0.50000000	22.4010620	0.802600741
124.000000	0.50000000	19.2666321	0.744334459
125.000000	0.50000000	17.9150848	0.717752874
126.000000	0.50000000	15.8446484	0.675004780
127.000000	0.50000000	13.7742081	0.629359424
128.000000	0.50000000	12.4514256	0.598376513
129.000000	0.50000000	9.83460617	0.531794965
130.000000	0.50000000	9.37451267	0.519206166
131.000000	0.50000000	7.44785023	0.462786913
132.000000	0.50000000	6.72894669	0.439884901
133.000000	0.50000000	5.11859894	0.383655250
134.000000	0.50000000	5.00357533	0.379320025
135.000000	0.50000000	4.68725586	0.367134273
136.000000	0.50000000	2.58805466	0.272805095
137.000000	0.50000000	2.24298096	0.253967583
138.000000	0.50000000	1.95541859	0.237129450
139.000000	0.50000000	1.35153961	0.197142303
140.000000	0.50000000	1.38029575	0.199228525

## 2.5 AUSGABE DIESES JOBS

MODELL-RECHNUNGEN NACH R.SERBER ZUR

---

BESCHREIBUNG DES PROJEKTIL-AUFBRUCHS VON ATOMKERNEN

---

\*\*\*\*\*  
\* EINGELESENE WERTE, VERWENDETE KONSTANTEN \*  
\*\*\*\*\*

MASSE (MEV) UND KERNLADUNGSZAHL DES TARGETS

193683.7790000000      82.0000000000

MASSE (MEV) UND KERNLADUNGSZAHL DES PROJEKTILS

5601.4200000000      3.0000000000

MASSE (MEV) UND KERNLADUNGSZAHL DES BEOBACHTETEN FRAGMENTS

3727.3150000000      2.0000000000

MASSE (MEV) UND KERNLADUNGSZAHL DES UNBEOBACHTETEN FRAGMENTS

1875.5800000000      1.0000000000

ENERGIE (MEV) DES PROJEKTILS IM LABORSYSTEM

156.0000000000

BINDUNGSENERGIE (MEV) DER CLUSTER A,B IM PROJEKTIL

1.4700000000

Q-WERT DER REAKTION

1.4700000000

KONSTANTE ZUR BERECHNUNG DES KERNRADIUS

1.4000000000

REDUZIERTE MASSE (MEV)

1247.72594662223730

HOEHE DER COULOMBSCHWELLE PRO EINFALLENDER EINHEITSLADUNG (MEV)

10.8931224147767904

KERNRADIUS DES TARGETS

8.2940153089037374

KERNRADIUS DES PROJEKTILS

2.5458518571735986

TOTALER AUFBRUCH-WIRKUNGSQUERSCHNITT

331.678895222986228

```
-----  
|           2 AMP      ( AT ) 1/3 |  
| EO >> ----- EPS ( ---- ) |  
|           AMA       ( AP ) |  
| >>           14.3939859751481152 |  
-----
```

```
*****  
*           BERECHNUNG DER EINZELNEN FAELLE           *  
*****
```

```
*** I=1 , J=3 , K=0 ***  
TRANSPARENTES SERBER MODELL  
INKLUSIVER WIRKUNGSQUERSCHNITT  
KEINE KORREKTUR DES MODELLS
```

ENERGIE	D2(SIGMA)/(D(OMEGA)D(ENERGIE))
70.0000000000000000	1580082.69084132463
75.0000000000000000	2143925.71351708868
80.0000000000000000	2814182.84328016290
85.0000000000000000	3530903.45807727496
90.0000000000000000	4181836.48886292451
95.0000000000000000	4625701.03281264449
100.0000000000000000	4749917.00914043118
105.0000000000000000	4528881.99524526577
110.0000000000000000	4034738.69931008015
115.0000000000000000	3393077.36715622758
120.0000000000000000	2724061.34200074291
125.0000000000000000	2108205.24480887549
130.0000000000000000	1582812.09793901024

```
*** I=1 , J=3 , K=1 ***  
TRANSPARENTES SERBER MODELL  
INKLUSIVER WIRKUNGSQUERSCHNITT  
COULOMB ENERGY DIFFERENCES CORRECTION
```

ENERGIE	D2 (SIGMA)/(D(OMEGA)D(ENERGIE))
70.0000000000000000	1268163.92603347381
75.0000000000000000	1956264.08071134426
80.0000000000000000	2928922.24681688263
85.0000000000000000	4182451.91450924077
90.0000000000000000	5570681.89644857193
95.0000000000000000	6764274.10207926831
100.0000000000000000	7368851.61950520589
105.0000000000000000	7184229.05984312762
110.0000000000000000	6347062.24894663575
115.0000000000000000	5194881.51052104332
120.0000000000000000	4034972.90041624196
125.0000000000000000	3035804.01460529468
130.0000000000000000	2245559.18627644470

\*\*\* I=2 , J=3 , K=0 \*\*\*

OPAKES SERBER MODELL

INKLUSIVER WIRKUNGSQUERSCHNITT

KEINE KORREKTUR DES MODELLS

ENERGIE	D2 (SIGMA)/(D(OMEGA)D(ENERGIE))
70.0000000000000000	31579895.3387949504
75.0000000000000000	45536096.4293080606
80.0000000000000000	65803442.0308494791
85.0000000000000000	95068754.2068621367
90.0000000000000000	136592217.814235587
95.0000000000000000	192302326.154487584
100.0000000000000000	254659942.397585474
105.0000000000000000	292130084.989945650
110.0000000000000000	268526161.238903284
115.0000000000000000	202484558.695724312
120.0000000000000000	137800015.629782524
125.0000000000000000	90856602.8364160620
130.0000000000000000	59697093.3568339832



\*\*\* I=2 , J=3 , K=1 \*\*\*

OPAKES SERBER MODELL

INKLUSIVER WIRKUNGSQUERSCHNITT

COULOMB ENERGY DIFFERENCES CORRECTION

ENERGIE	D2(SIGMA)/(D(OMEGA)D(ENERGIE))
70.0000000000000000	22811701.1711514704
75.0000000000000000	36054692.9979530983
80.0000000000000000	57225632.8007174879
85.0000000000000000	90886846.0877752155
90.0000000000000000	143321862.558317441
95.0000000000000000	220302659.768575460
100.0000000000000000	313910039.122878075
105.0000000000000000	372053230.280599594
110.0000000000000000	333155325.378891349
115.0000000000000000	239150641.847164795
120.0000000000000000	157285874.116962820
125.0000000000000000	102508980.640957620
130.0000000000000000	68001584.1284235790

\*\*\*\*\*  
\* PLOTTEN DER I/J-KOMBINATIONEN \*  
\*\*\*\*\*

BILD-KENNUNG: I=1, J=3.

ANZAHL "NEUER" SOLL-MAXIMA = 2  
KURVEN-NUMMER DES ERSTEN SOLL-MAXIMUMS = 1  
YMIN, YMAX = 1.58008D+06 4.75344D+06 "ALT"  
SOLLWERT VON YMAX = 1.56350D+02  
TRANSF.-FAKTOR = 3.28920D-05  
KURVEN-NUMMER DES NAECHSTEN SOLL-MAXIMUMS = 2  
YMIN, YMAX = 1.26816D+06 7.39644D+06 "ALT"  
SOLLWERT VON YMAX = 1.56350D+02  
TRANSF.-FAKTOR = 2.11386D-05

XMIN, XMAX = 7.00000D+01 1.30000D+02  
ZXMI, ZXMA = 70.0000 130.0000  
IXMI, IXMA = 70 130  
EXMI, EXMA = 1 2  
YMIN, YMAX = 0.0 1.56350D+02  
ZYMI, ZYMA = 0.0 156.3501  
IYMI, IYMA = 0 0  
EYMI, EYMA = 0 0

ZXMI= 70.0000 ZXMA= 130.0000  
ZXIB= 10.0000 ZDXI= 1.1000  
LXL = 7 ZXP = 0.0  
LXB = -1 LXP = -99  
ZYMI= 0.0 ZYMA= 200.0000  
ZYIB= 50.0000 ZDYI= 2.2000  
LYL = 5 ZYP = 0.0  
LYB = -1 LYP = -99

EXTERNE DATEN EINTRAGEN. TABELLE NUMMER 1.  
PHASENRAUMFAKTOREN PLOTTEN

K-WERT = 0

K-WERT = 1

KURVEN EINZEICHNEN

NUMMER = 1 TABELLE = 7

NUMMER = 2 TABELLE = 8

BILD-KENNUNG: I=2, J=3.

ANZAHL "NEUER" SOLL-MAXIMA = 2

KURVEN-NUMMER DES ERSTEN SOLL-MAXIMUMS = 1

YMIN, YMAX = 3.15799D+07 2.92764D+08 "ALT"

SOLLWERT VON YMAX = 1.56350D+02

TRANSF.-FAKTOR = 5.34049D-07

KURVEN-NUMMER DES NAECHSTEN SOLL-MAXIMUMS = 2

YMIN, YMAX = 2.28117D+07 3.72812D+08 "ALT"

SOLLWERT VON YMAX = 1.56350D+02

TRANSF.-FAKTOR = 4.19381D-07

XMIN, XMAX = 7.00000D+01 1.30000D+02  
ZXMI, ZXMA = 70.0000 130.0000  
IXMI, IXMA = 70 130  
EXMI, EXMA = 1 2  
YMIN, YMAX = 0.0 1.56350D+02  
ZYMI, ZYMA = 0.0 156.3501  
IYMI, IYMA = 0 0  
EYMI, EYMA = 0 0

ZXMI= 70.0000 ZXMA= 130.0000  
ZXIB= 10.0000 ZDXI= 1.1000  
LXL = 7 ZXP = 0.0  
LXB = -1 LXP = -99  
ZYMI= 0.0 ZYMA= 200.0000  
ZYIB= 50.0000 ZDYI= 2.2000  
LYL = 5 ZYP = 0.0  
LYB = -1 LYP = -99

EXTERNE DATEN EINTRAGEN. TABELLE NUMMER 2.  
PHASENRAUMFAKTOREN PLOTTEN

K-WERT = 0

K-WERT = 1

KURVEN EINZEICHNEN

NUMMER = 1 TABELLE = 15

NUMMER = 2 TABELLE = 16

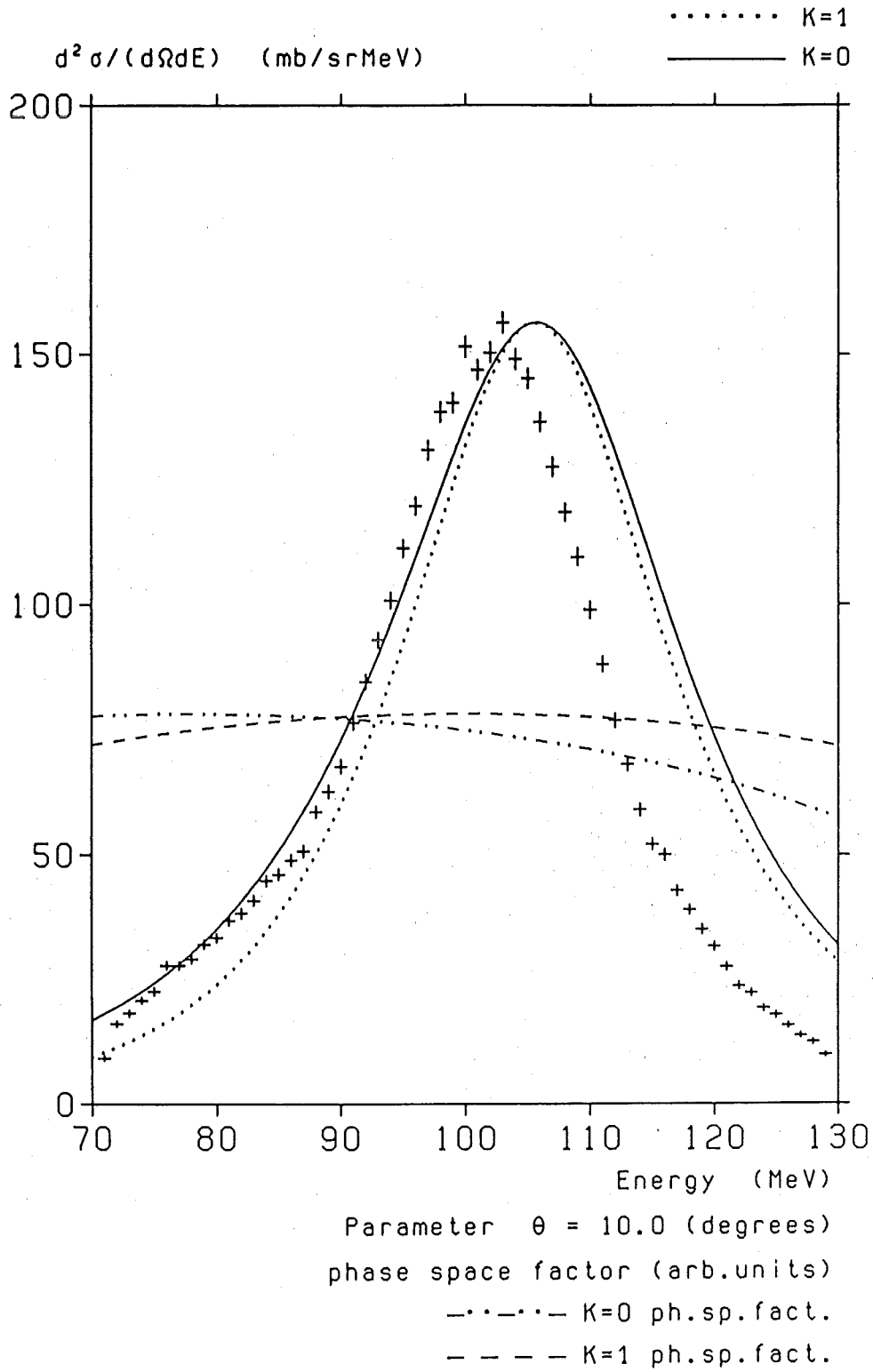
\*\*\*\*\*  
\* E N D E \*  
\*\*\*\*\*

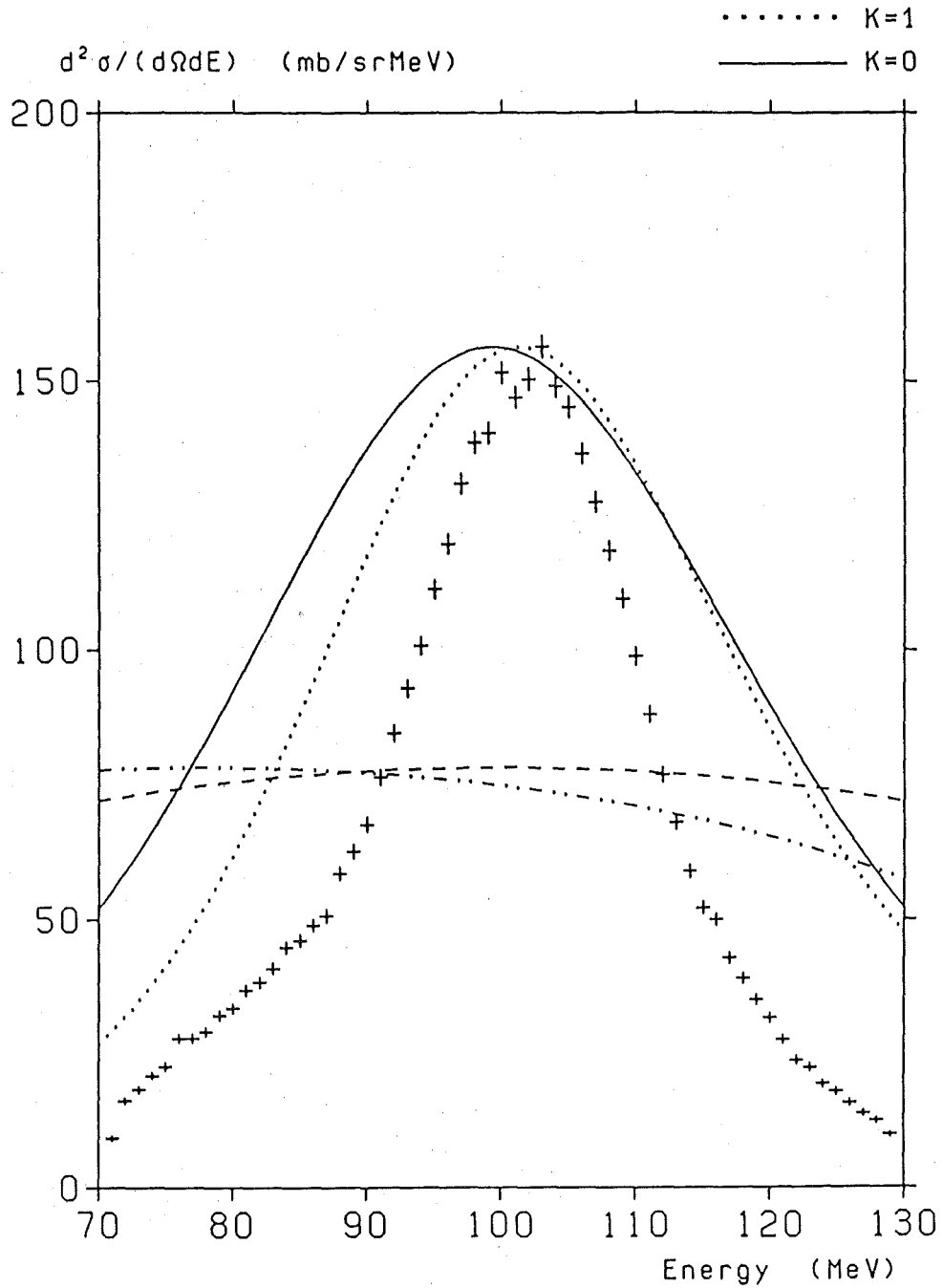
BENOETIGTE PLOTPAPIERLAENGE = 19.20

PROGRAMMENDE SERBER

VERSAPLOT V07.6 13,842 PLOTTABLE VECTORS 87.113 13:55:36

2.6 DIE ZWEI ZEICHNUNGEN DIESES JOBS





Parameter  $\theta = 10.0$  (degrees)  
phase space factor (arb.units)  
- · - · - K=0 ph.sp.fact.  
- - - - K=1 ph.sp.fact.

## 2.7 BEISPIELJOB MIT EINGABE ZU ALLEN MÖGLICHEN 6 BILDERN

```
//ADI930A JOB (0930,545,PO000),HOHN,NOTIFY=ADI930,MSGCLASS=H,
//          MSGLEVEL=(1,1),REGION=2000K
//*MAIN LINES=19
// EXEC F7C,PARM.C='LANGLVL(77),LINECOUNT(63)'
//C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=TSO964.SAMMEL.PLOT(AUSGL4)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PLOT(SPLINE)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(AEQUID)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(PLSYMB)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(STRIPU)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(SYMBGR)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZABSCH)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXYAXN)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXYIBE)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXYMAS)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZXZIFF)
//          DD DISP=SHR,DSN=TSO964.SAMMEL.PC(ZYZIFF)
//          DD DISP=SHR,DSN=ADI930.SERBER2.FORT(SERBXY)
// EXEC F7CLG,PLOT=VERSATEC,
//          PARM.C='LANGLVL(77),LINECOUNT(63)',PARM.G='SIZE(2000)'
//* C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=ADI930.SERBER2.FORT(SERBER)
//L.SYSPRINT DD DUMMY
//L.SYSIN DD *
  ENTRY SERBER
//G.SYSIN DD *
**I** 1=TRANSPARENT, 2=OPAKES, 3=TRANSPARENT+OPAKES SERBER-MODELL
      3
* PHASENRAUMFAKTOREN PLOTTEN, WENN VORHANDEN ("JA">0) (INTEGER)
      1
**J** 1=ENERGIE-VERT., 2=WINKEL-VERT., 3=INKLUSIV. WIRK.QUERSCHN.
      1   2   3
**K** 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR
      0   1   2   3
* MAXIMUM-SOLLWERTE DER KURVEN (ERSTER GUELTIGER J-WERT).
      300.  -1.  -1.  -1.
```

\*\*K\*\* 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR  
0 1 2 3

\* MAXIMUM-SOLLWERTE DER KURVEN (NAECHSTER GUELTIGER J-WERT).  
-1. -1. -1. -1.

\*\*K\*\* 0=KEINE KORREKTUR, 1=CEDC-, 2=CDC-, 3=CEDC+CDC-KORREKTUR  
0 1 2 3

\* MAXIMUM-SOLLWERTE DER KURVEN (NAECHSTER GUELTIGER J-WERT).  
-1. -1. -1. -1.

\* MASSE (MEV) UND KERNLADUNGSZAHL DES TARGETS  
25340.798 13.

\* MASSE (MEV) UND KERNLADUNGSZAHL DES PROJEKTILS  
6533.722 3.

\* MASSE (MEV) UND KERNLADUNG DES BEOBACHTETEN FRAGMENTS  
3727.315 2.

\* MASSE (MEV) UND KERNLADUNG DES UNBEOBACHTETEN FRAGMENTS  
2808.873 1.

\* ENERGIE (MEV) DES PROJEKTILS IM LABORSYSTEM  
29.6

\* BINDUNGSENERGIE DER CLUSTER A UND B IM PROJEKTIL  
2.47

\* KONSTANTE ZUR BERECHNUNG DES KERNRADIUS  
1.4

\* Q-WERT DER REAKTION P+T ----> A+B+T  
2.5

\* UNTERE SCHRANKE ENERGIE (IN MEV)  
11.

\* OBERE SCHRANKE ENERGIE (IN MEV)  
21.

\* SCHRITTWEITE DER ENERGIE (IN MEV)  
0.1

\* UNTERE SCHRANKE DES WINKELS (IN GRAD) (INTEGER)  
-80

\* OBERE SCHRANKE DES WINKELS (IN GRAD) (INTEGER)  
80

\* SCHRITTWEITE DES WINKELS (IN GRAD) (INTEGER)  
1

\* STREUWINKEL ALS PARAMETER DES DOPP-DIFF-WIQU'S (BEI J=3)  
12.5

\* PLOT: ABSZISSEN- & ORDINATEN-LAENGE, ZIFFERN-, ZEICHEN-HOEHE  
8. 12. 0.2 0.15

\* PLOT: STRICHART DER KURVE (INTEGER 0-5)  
1 3 5 2

\* PLOT: ABSTAENDE BEI STRICHELUNG DER KURVE (REAL)  
0.1 0.1 0.1 0.1

\* PLOT: SYMBOLNUMMERN DER KURVENSYMBOLE (INTEGER)  
-40 -40 -40 -40

\* PLOT: GROESSE DER KURVENSYMBOLE (REAL)  
0.14 0.14 0.14 0.14

\* SOLLEN EXTERNE TABELLEN VERARBEITET WERDEN? (INTEGER)  
0

\* WIEVIELE EXTERNE TABELLEN SOLLEN VERARBEITET WERDEN?  
1

\* FORTRAN-REFERENZ-NUMMER(N) DER TABELLEN?  
3

```
//G.FT03F001 DD *  
1 1 2  
13. 0.5 50. 5.  
17. 0.5 70. 5.
```

```
//G.FT06F001 DD SYSOUT=*
```

```
//G.SYSLOUT DD DUMMY
```

```
//G.PLOTLOG DD DUMMY
```

```
//G.PLOTPARM DD *
```

```
&PLOT XMAX=68. SPACE=69. &END
```

```
// EXEC SVPLOT
```

```
//SORT.SYSOUT DD DUMMY
```

```
//PLOT.PLOTLOG DD DUMMY
```

```
//PLOT.SYSVECTR DD DUMMY
```

```
//
```



### 3. DAS HAUPTPROGRAMM "SERBER"

#### 3.1 BESCHREIBUNG DER VARIABLEN

ABS = Fortran builtin function.  
AEQUID = Eigene Subroutine.  
AMA, AMB= Eingabegrößen.  
AMP, AMT= Eingabegrößen.  
AMUE = Reduzierte Masse.  
AP = Konstante (AMP/931.5).  
ASIN = Fortran builtin function.  
AT = Konstante (AMT/931.5).  
ATAN = Fortran builtin function.  
AUSGL4 = Eigene Subroutine.  
AZA = Eingabegröße.  
AZB = Eingabegröße.  
AZP = Eingabegröße.  
AZT = Eingabegröße.  
CKUSYM = Zeichenkette 'ABCDEFGHIJKLMNOPQRSTUVWXYZ\$#@123456789 '  
CREAKT = Zeichenkette (Länge 40) der Reaktionsformel.  
CZEILE = Zeichenkette (Länge 80) als Hilfsvariable.  
CO(0:4)= Koeffizienten eines Ausgleichspolynoms vom Grad vier.  
COS = Fortran builtin function.  
CPI180 =  $\pi/180$   
C180PI =  $180/\pi$   
EADEL = Eingabegröße.  
EAKT = Momentane Teilchenenergie.  
EAMAX = Eingabegröße; wird gegebenenfalls korrigiert, d.h. verkleinert wird.  
EAMIN = Eingabegröße; wird gegebenenfalls korrigiert, d.h. vergrößert wird.  
EAS = EAKT-AZB\*VC  
EC = Konstante (AZP\*VC).  
EPSAB = Eingabegröße.  
EPSTH = Konstante (1.E-5) bei der Berechnung der Winkelverteilung. Die Division durch Null als Tangens des

- Winkels Null wird damit vermieden. Wird der Tangens im Betrag kleiner als EPSTH, so wird die Formel der Winkelverteilung für -EPSTH und +EPSTH ausgewertet.
- ETAKT = Momentaner Winkel des doppeltdifferentiellen Wirkungsquerschnittes.
- EO = Eingabegröße.
- EOF = Hilfsfaktor für Energiewerte.
- EOS = EO-AZP\*VC. Eventuell auftretende negative Werte von EOS werden abgefangen (bei K=1 bzw. K=3).
- I = Laufvariable, Hilfsvariable.
- IA = Schleifenanfangswert.
- IABS = Fortran builtin function.
- IB = Schleifenendwert.
- IOXMA = Exponent des X-Achsenmaximums.
- IOXMI = Exponent des X-Achsenminimums.
- IOYMA = Exponent des Y-Achsenmaximums.
- IOYMI = Exponent des Y-Achsenminimums.
- II = Eingabegröße.
- IIA = Schleifenanfangswert des Auswahlparameters II.
- IIB = Schleifenendwert des Auswahlparameters II.
- IK = Laufvariable (Anzahl der einzutragenden Kurven).
- INT = Fortran builtin function.
- IPL = Steuerparameter fürs Plotten. Bei IPL<1 werden keine Plot-Subroutinen aufgerufen.
- IPR = Parameter, um Druck-Ausgabe zu steuern.  
7≤IPR : Werte-Tabellen werden mit ausgedruckt.  
0<IPR<7 : Weniger Ausgabe.
- IPZ = Positionierung der 10er-Potenz bei der Beschriftung der Y-Achse.
- IRE = Parameter der Fortran-Referenz-Nummer für die Eingabe; steht auf Standardwert = 5.
- IWR = Parameter der Fortran-Referenz-Nummer für die Ausgabe; steht auf Standardwert = 6.
- IXMA = Normiertes X-Achsenmaximum.
- IXMI = Normiertes X-Achsenminimum.
- IYMA = Normiertes Y-Achsenmaximum.
- IYMI = Normiertes Y-Achsenminimum.
- I1 = Laufvariable für den Eingabe-Parameter II beim Plotten.
- J = Laufvariable.
- JJ(1:4) = Feld für alle eingegebenen J-Werte.

JKW(0:4,3) = Feld für alle K-Werte, je J-Wert.  
JM = Zähler für gültige J-Werte.  
JW = Aktueller J-Wert.  
J1 = Momentaner J-Wert beim Plotten.  
K = Laufvariable beim Einlesen der K-Werte ins Feld KK.  
KK(1:4) = Eingabegröße und Speicherplatz für die K-Werte  
der Kurven, die in einem Plot vorkommen.  
KM = Zähler für gültige K-Werte.  
KW = Schleifen- und Zählvariable.  
K1 = Momentaner K-Wert beim Plotten.  
LBILD = Aktuelle Anzahl der ausgeführten Zeichnungen.  
LINE = Plot-Subroutine (System).  
LOG10 = Fortran builtin function.  
LTAB(1:4) = Aktuelle I-,J-,K-abhängige Nummern der Kurven,  
die geplottet werden können.  
LXB = Anzahl der Ziffern hinter dem Dezimalpunkt (X-Achse).  
LXBE = Charakterisierung für den Beschriftungsstartwert bei X.  
LXL = Anzahl der Markierungspositionen auf der X-Achse.  
LXP = Anzahl der Exponentstellen bei Exponentdarstellung  
(X-Achse).  
LYA = Anzahl der Plotpositionen für die 10er-Potenz der  
Y-Achsenbeschriftung (Bezifferung).  
LYB = Anzahl der Ziffern hinter dem Dezimalpunkt (Y-Achse).  
LYBE = Charakterisierung für den Beschriftungsstartwert bei Y.  
LYL = Anzahl der Markierungspositionen auf der Y-Achse.  
LYP = Anzahl der Exponentstellen bei Exponentdarstellung  
(Y-Achse).  
M = Laufvariable.  
MA = Hilfs-, Zählvariable.  
MB = Hilfs-, Zählvariable, z.B.  
I-,J-,K-abhängige Nummer einer errechneten Kurve.  
MD = Hilfs-, Zählvariable.  
MJ = Parameter (Wert=3, Dimensionierung des JJ-Feldes).  
MK = Parameter (Wert=4, Dimensionierung des KK-Feldes).  
ML = Anzahl von neuen Wertepaaren.  
MM = Hilfs-, Zählvariable.  
MOD = Fortran builtin function.  
MP = Zähler für die Anzahl der berechneten Wertepaare  
bzw. aktuelle Anzahl der zu plottenden Wertepaare.  
MQ = Aktuelle Anzahl von (neuen) Wertepaaren.

MX = Parameter  $(NJ+1)*MY$ .  
MY = Parameter (Wert=16).  
NEWPEN = Plot-Subroutine (System).  
NEXT(0:10,6) = Anzahl und laufende Nummern externer Datentabellen je I/J-Kombination. NEXT(0,n) ist die Anzahl der externen Datentabellen mit der Bildkennziffer  $n=3*(I-1)+J$ . NEXT(1:NEXT(0,n),n) sind die laufenden Nummern der externen Datentabelle. XEXT(0,2\*NEXT(m,n)-1) ist die Anzahl der Punkte der externen Datentabelle mit der Nummer m.  
NEXTR = Anzahl externer Datentabellen.  
NI = Parameter  $(NJ+1)*2$ .  
NJ = Parameter (Wert=1000, Dimensionierung der Felder XARR, YARR, ZX, ZY, ZV, ZW).  
NKURV = Nummer der Kurve in einem Plot.  
NPHAS = Eingabegröße.  
NS = Symbolnummer beim Plotten.  
NST(1:4) = Eingabegröße.  
NSY(1:4) = Eingabegröße.  
NT = Aktuelle Strichart beim Plotten.  
NUMBER = Plot-Subroutine (System).  
NX = Maximale Anzahl an Punkten einer externen Datentabelle.  
NY =  $(NX+1)*20$ .  
NYN(-1:4,1:3) = Anzahl und Kennzeichnungen der Sollwerte für YMAX bezüglich der J-Werte. NYN(0,\*) = Anzahl der verschiedenen gewünschten Sollwerte (YNORM-Werte >0). NYN(-1,\*) = Kurvennummer des jeweils ersten gewünschten Sollwertes. NYN(1:4,\*) = Kennzeichnung der Sollwerte (d.h. YNORM-Werte); <0 bedeutet, bisher definiertes Sollmaximum wird übernommen; >0 bedeutet, neues Sollmaximum.  
PHAS = Aktueller Wert des Phasenraumfaktors.  
PI = Konstante  $\pi$ .  
PIRTP = Zwischenfaktor bei der Energieverteilung.  
PLOT = Plot-Subroutine (System).  
PLOTS = Plot-Subroutine (System).  
PLSYMB = Eigene Plot-Subroutine.  
PMAX = Aktuelles Maximum der Werte des Phasenraumfaktors.  
PMIN = Aktuelles Minimum der Werte des Phasenraumfaktors.  
P12 = Halbzahliges Legendre-Polynom.

P12X,P12Y,P12Z = Konstanten bei der P12-Berechnung.  
QU = Eingabegröße.  
RF = Konstante (Totaler Aufbruch-Wirkungsquerschnitt).  
RP = Konstante (Kernradius des Projektils).  
RT = Konstante (Kernradius des Targets).  
RO = Konstante zur Berechnung des Kernradius.  
SERBXY = Eigene Plot-Subroutine.  
SIN = Fortran builtin function.  
SNGL = Fortran builtin function.  
SPLINE = Eigene Subroutine.  
SQRT = Fortran builtin function.  
STRIPU = Eigene Plot-Subroutine.  
SYMBLO = Plot-Subroutine (System).  
TAN = Fortran builtin function.  
TANG = Aktueller Tangens eines Winkels.  
THO = Faktor bei der Berechnung von Verteilungen.  
THOS = Faktor bei der Berechnung von Verteilungen.  
TQ = TANG/THO bzw. Schrittweite bei der Berechnung von Stützpunkten mit einem Ausgleichspolynom.  
TQC = Faktor bei der Berechnung von Verteilungen bzw. aktueller X-Wert bei der Berechnung von Stützpunkten mit einem Ausgleichspolynom.  
TQCS = Faktor bei der Berechnung von Verteilungen.  
TQS = TANG/THOS bzw. aktueller Y-Wert bei der Berechnung von Stützpunkten mit einem Ausgleichspolynom.  
TU = Argument von P12.  
TUS = Faktor bei der Berechnung von P12.  
VC = Konstante (Höhe der Coulombschwelle pro einfallender Einheitsladung in MeV).  
WERT = Wert der Winkelverteilung bzw. Häufigkeitsverteilung für bestimmte Teilchenenergie bzw. doppelt (nach Winkel und Energie) differentieller Wirkungsquerschnitt.  
WIAKT = Momentaner Laufwert des Winkels.  
WIDEL = Eingabegröße.  
WIMAX = Eingabegröße.  
WIMIN = Eingabegröße.  
WSTREU = Eingabegröße.  
XARR(0:NJ,1:MY) = Feld, in dem die X-Werte der Tabellen

gespeichert sind.

XEXT(O:NJ,1:MY) = Feld mit den X-Werten und X-Fehlern von extern vorgegebenen Datenpunkten. In XEXT(O,2\*NEXT(m,n)-1) steht die Anzahl der Punkte der externen Datentabelle mit Nummer m.

XMAX = X-Maximum der Tabelle.

XMIN = X-Minimum der Tabelle.

YARR(O:NJ,1:MY) = Feld, in dem die Y-Werte der Tabellen gespeichert sind.

YEXT(O:NJ,1:MY) = Feld mit den Y-Werten und Y-Fehlern von extern vorgegebenen Datenpunkten.

YFAKT = Hilfsgröße beim Normieren von Kurven.

YMAX = Y-Maximum der Tabelle.

YMIN = Y-Minimum der Tabelle.

YNORM(1:12) = Eingabegröße.

YPHA(O:NJ,0:1) = Feld mit den Werten der Phasenraumfaktoren für K=0 und K=1. YPHA(O,K) enthält den Normierungsfaktor, um in der Zeichnung auf die Hälfte der Höhe der Energie-Kurve zu kommen.

YSOLL = Hilfsgröße beim Normieren von Kurven.

ZABSCH = Eigene Subroutine.

ZABST = Punktabstand beim Stricheln, Strichpunktieren etc.

ZAHL = Zu plottender Zahlenwert.

ZDXI = Abstand zwischen zwei Marken auf der X-Achse (in Inches).

ZDYI = Abstand zwischen zwei Marken auf der Y-Achse (in Inches).

ZEIHO = Zeichengröße (in Inches).

ZIFHO = Zifferngröße (in Inches).

ZST(1:4) = Eingabegröße.

ZSTHO = Strichlänge beim Stricheln der Achse (in Inches).

ZSY(1:4) = Eingabegröße.

ZSYHO = Textgröße beim Beschriften der Zeichnung (in Inches).

ZV(1:NJ) = Abszisse der Wertepaare.

ZW(1:NJ) = Ordinate der Wertepaare.

ZX(1:NJ) = Abszisse der Wertepaare.

ZXA = Benötigte Plotpapierlänge.

ZXD = Zwischengröße beim Plotten von Fehlerbalken.

ZXI = Länge der Abszisse in Inches.

ZXIB = Beschriftungseinheit beim Beziffern der X-Achse.

ZXMA = (Korrigiertes) Maximum auf der X-Achse.

ZXMI = (Korrigiertes) Minimum auf der X-Achse.  
ZXP = Exponent bei der Exponentendarstellung von Ziffern  
an der X-Achse.  
ZXS = Ursprungsverschiebung in X-Richtung (2.00 Inches).  
ZXX = Aktuelle X-Position beim Plotten.  
ZXXALT = Hilfsposition bei der Kurvenkorrektur.  
ZXXE = Hilfsposition bei der Kurvenkorrektur.  
ZXYAXN = Eigene Plot-Subroutine.  
ZXYIBE = Eigene Subroutine.  
ZXYMAS = Eigene Subroutine.  
ZXZIFF = Eigene Plot-Subroutine.  
ZY(1:NJ) = Ordinaten von Wertepaaren.  
ZYD = Zwischengröße beim Plotten von Fehlerbalken.  
ZYI = Länge der Ordinate (in Inches).  
ZYIB = Beschriftungseinheit beim Beziffern der Y-Achse.  
ZYMA = (Korrigiertes) Maximum auf der Y-Achse.  
ZYMI = (Korrigiertes) Minimum auf der Y-Achse.  
ZYP = Exponent bei der Exponentendarstellung von Ziffern  
an der Y-Achse.  
ZYS = Ursprungsverschiebung in Y-Richtung (4.00 Inches).  
ZYY = Aktuelle Y-Position beim Plotten.  
ZYYALT = Hilfsposition bei der Kurvenkorrektur.  
ZYYE = Hilfsposition bei der Kurvenkorrektur.  
ZYZIFF = Eigene Plot-Subroutine.

### 3.2 ABDRUCK DES HAUPTPROGRAMMS

```
*****
* S E R B E R M O D E L L - R E C H N U N G E N  14. 05. 1987 *
*****

PROGRAM SERBER
IMPLICIT REAL*8(A-H,O-Y)
PARAMETER (IRE=5,IWR=6,IPL=1,IPR=7)
PARAMETER (MJ=3,MK=4,NJ=1000,NI=(NJ+1)*2)
PARAMETER (MY=16,MX=(NJ+1)*MY,NX=100,NY=(NX+1)*20)
PARAMETER (PI=3.1415926535897932D0,EPSTH=1.D-5)
PARAMETER (CPI180=0.0174532925199432D0)
PARAMETER (C180PI=57.29577951308232D0)
CHARACTER CKUSYM*40,CREAKT*40,CZEILE*80
DIMENSION XARR(0:NJ,MY),YARR(0:NJ,MY),YNORM(MK,MJ)
DIMENSION XEXT(0:NX,20),YEXT(0:NX,20),YPHA(0:NJ,0:1)
DIMENSION CO(0:4),ZX(NJ),ZY(NJ),ZV(NJ),ZW(NJ)
DIMENSION ZST(MK),ZSY(MK),NEX(10),NEXT(0:10,6)
DIMENSION NST(MK),NSY(MK),LTAB(MK),NYN(-1:MK,MJ)
DIMENSION JJ(MJ),KK(MK),JKW(0:MK,MJ)
DATA CREAKT/'
DATA CKUSYM/'ABCDEFGHIJKLMNOPQRSTUVWXYZ$#@0123456789 '/
DATA P12X/0.75D0/,P12Y/-0.234375D0/,P12Z/0.13671875D0/
DATA XARR/MX*-9999./,YARR/MX*-9999./,YPHA/NI*-9999./
DATA XEXT/NY*-9999./,YEXT/NY*-9999./,YNORM/12*0.DO/
DATA JKW/15*0/,LTAB/4*0/,NYN/18*0/,NEX/10*0/,NEXT/66*0/
DATA ZXA/0./,ZXS/2./,ZYS/4./
*****
* STEUERUNG DES HAUPTPROGRAMMS DURCH I, J, K
*****
READ(IRE,'(A80)')
READ(IRE,*) II
IF (II.LT.1.OR.II.GT.3) THEN
WRITE(IWR,('( ' I-WERT UNGUELTIG. PROGRAMMENDE.'))')
GOTO 399
ENDIF
***** PHASENRAUMFAKTOREN PLOTTEN, WENN VORHANDEN.
* NUR BEI J=1 UND/ODER J=3 MOEGLICH *****
```



```
    READ(IRE, '(A80)')
    READ(IRE,*) NPHAS
***** J-WERTE LESEN *****
    READ(IRE, '(A80)')
    READ(IRE,*) (JJ(J), J=1, MJ)
    JM = 0
    DO 10 J=1, MJ
        IF (JJ(J).GE.1.AND.JJ(J).LE.3) THEN
            JM = JM + 1
        ELSE
            JJ(J) = 9
        ENDIF
10 CONTINUE
    IF (JM.LE.0) THEN
        WRITE(IWR, '(' J-WERT UNGUELTIG. PROGRAMMENDE. ')')
        GOTO 399
    ENDIF
    DO 11 J=1, MJ
    DO 11 K=2, MJ
    IF (JJ(K-1).GT.JJ(K)) THEN
        I=JJ(K)
        JJ(K)=JJ(K-1)
        JJ(K-1)=I
    ENDIF
11 CONTINUE
***** K-WERTE FUER JEDEN GUELTIGEN J-WERT
* UND J-WERTE-BEZOGENE NORMIERUNGSFAKTOREN LESEN *****
    DO 16 J=1, JM
        J1 = JJ(J)
        READ(IRE, '(A80)')
        READ(IRE,*) (KK(K), K=1, MK)
        KW = 3 - 2 * (J1-2)**2
        KM = 0
        DO 12 K=1, MK
            IF (KK(K).GE.0.AND.KK(K).LE.KW) THEN
                KM = KM + 1
            ELSE
                KK(K) = 9
            ENDIF
        DO 12 M=K+1, MK
```

```

          IF (KK(K).EQ.KK(M).AND.
+          (KK(K).GE.O.AND.KK(K).LE.KW)) KK(M) = 9
12  CONTINUE
    IF (KM.LE.O) THEN
      WRITE(IWR, '('' K-WERT UNGUELTIG. PROGRAMMENDE.'')')
      GOTO 399
    ENDIF
    DO 13  K=1,MK
    DO 13  M=2,MK
      IF (KK(M-1).GT.KK(M)) THEN
        I=KK(M)
        KK(M)=KK(M-1)
        KK(M-1)=I
      ENDIF
13  CONTINUE
    JKW(O,J) = KM
    DO 14  K=1,KM
      JKW(K,J) = KK(K)
14  CONTINUE
    READ(IRE, '(A80)')
    READ(IRE,*) (YNORM(K,J1),K=1,MK)
    MA = 0
    DO 15  K=1,KM
      IF (YNORM(K,J1).GT.O.) THEN
        IF (MA.LE.O) NYN(-1,J1) = K
        MA = MA + 1
        NYN(K,J1) = 1
      ELSE
        YNORM(K,J1) = -1.
        NYN(K,J1) = -1
      ENDIF
15  CONTINUE
    NYN(O,J1) = MA
16  CONTINUE
*****
* EINZULESENDE RECHENGRÖSSEN
*****
    READ(IRE, '(A80)')
    READ(IRE,*) AMT,AZT
    READ(IRE, '(A80)')
```

```
READ(IRE,*) AMP,AZP
READ(IRE,'(A80)')
READ(IRE,*) AMA,AZA
READ(IRE,'(A80)')
READ(IRE,*) AMB,AZB
READ(IRE,'(A80)')
READ(IRE,*) EO
READ(IRE,'(A80)')
READ(IRE,*) EPSAB
READ(IRE,'(A80)')
READ(IRE,*) RO
READ(IRE,'(A80)')
READ(IRE,'(A80)') CZEILE
READ(IRE,'(A80)')
READ(IRE,*) EAMIN
READ(IRE,'(A80)')
READ(IRE,*) EAMAX
READ(IRE,'(A80)')
READ(IRE,*) EADEL
READ(IRE,'(A80)')
READ(IRE,*) WIMIN
READ(IRE,'(A80)')
READ(IRE,*) WIMAX
READ(IRE,'(A80)')
READ(IRE,*) WIDEL
READ(IRE,'(A80)')
READ(IRE,*) WSTREU
READ(IRE,'(A80)')
READ(IRE,*) ZXI,ZYI,ZIFHO,ZEIHO
READ(IRE,'(A80)')
READ(IRE,*) (NST(I),I=1,4)
READ(IRE,'(A80)')
READ(IRE,*) (ZST(I),I=1,4)
READ(IRE,'(A80)')
READ(IRE,*) (NSY(I),I=1,4)
READ(IRE,'(A80)')
READ(IRE,*) (ZSY(I),I=1,4)
READ(IRE,'(A80)')
READ(IRE,*) NEXTR
IF (NEXTR.LE.0) GOTO 19
```

```
READ(IRE, '(A80)')
READ(IRE, *) NEXTR
IF (NEXTR.GT.10) NEXTR=0
IF (NEXTR.LE.0) GOTO 18
  READ(IRE, '(A80)')
  READ(IRE, *) (NEX(I), I=1, NEXTR)
  DO 17 MM=1, NEXTR
    MA = NEX(MM)
    READ(MA, *, ERR=18, END=18) I, J, KW
    IF (I.LE.0.OR.I.GT.3) GOTO 19
    IF (J.LE.0.OR.J.GT.3) GOTO 19
    IF (I.EQ.3) I=2-MOD(II,2)
    IF (KW.GT.NX) KW=NX
    IB = 3*(I-1)+J
    NEXT(O, IB) = NEXT(O, IB) + 1
    IA = NEXT(O, IB)
    NEXT(IA, IB) = MM
    IA = 2*MM-1
    XEXT(O, IA) = KW
    IB = 2*MM
    DO 17 K=1, KW
      READ(MA, *, END=18)
+      XEXT(K, IA), XEXT(K, IB), YEXT(K, IA), YEXT(K, IB)
17    CONTINUE
      GOTO 19
18    CONTINUE
      NEXTR = 0
19    CONTINUE
      MB = 81
20    MB = MB - 1
      IF (CZEILE(MB:MB).EQ.' '.AND.MB.GT.1) GOTO 20
      IA = 0
21    IA = IA + 1
      IF (CZEILE(IA:IA).EQ.' '.AND.IA.LT.80) GOTO 21
      IB = INDEX(CZEILE(IA:MB), ' ') + IA - 2
      IF (IB.LE.IA) IB = MB
      CZEILE(75:80) = '(F9.0)'
      WRITE(CZEILE(77:77), '(I1)') MIN(IB-IA+1,9)
      READ(CZEILE(IA:IB), CZEILE(75:80)) QU
      IF (MB.GT.IB) THEN
```

```
MA = IB
22 MA = MA + 1
   IF (CZEILE(MA:MA).EQ.' '.AND.MA.LT.80) GOTO 22
   CREAKT(1:MB-MA+1) = CZEILE(MA:MB)
   ENDIF
```

\*\*\*\*\*

\* DIVERSE KONSTANTEN

\*\*\*\*\*

```
AP = AMP/931.5DO
AT = AMT/931.5DO
VC = AZT*1.44DO/(RO*(AP**(1.DO/3.DO)+AT**(1.DO/3.DO)))
RT = RO*AT**(1.DO/3.DO)
RP = RO*AP**(1.DO/3.DO)
RF = RT*RP*5.DO*PI
EC = AZP*VC
EOS = ABS(E0-AZP*VC)
EOF = 2.DO*AMP/AMA*(AT/AP)**(1.DO/3.DO)*EPSAB
THO = SQRT(AMB/AMA*EPSAB/E0)
THOS = SQRT(AMB/AMA*EPSAB/EOS)
AMUE = AMA*AMB/(AMA+AMB)
EAMIN = MAX(EAMIN,QU,QU+AZA*VC)
EAMAX = MIN(EAMAX,E0-QU,E0-AZA*VC-QU)
IF ((EAMAX-EAMIN)/EADEL.GE.0.980DO*NJ)
+   EADEL=(EAMAX-EAMIN)*(1.02DO/NJ)
IF ((WIMAX-WIMIN)/WIDEL.GE.0.980DO*NJ)
+   WIDEL=(WIMAX-WIMIN)*(1.02DO/NJ)
```

\*\*\*\*\*

\* AUSGABE DER RECHENGROESSEN

\*\*\*\*\*

```
WRITE(IWR,'(//14X,' 'MODELL-RECHNUNGEN NACH R.' ',
+' 'SERBER ZUR' '/14X,35(' '_')//6X,' 'BESCHREI' ',
+' 'BUNG DES PROJEKTIL-AUFBRUCHS VON ATOMKERNEN' '/
+6X,51(' '_'))')
WRITE(IWR,'(//6X,44(' '*'))/
+6X,' '* EINGELESENE WERTE, VERWENDETE KONSTANTEN *')')
WRITE(IWR,'(6X,44(' '*'))/
+/6X,' 'MASSE (MEV) UND KERNLADUNGSZAHL DES TARGETS' ')')
WRITE(IWR,'(6X,2F20.10)') AMT,AZT
WRITE(IWR,'(/6X,' 'MASSE (MEV) UND KERNLADUNGSZAHL ' ',
+' 'DES PROJEKTILS' ')')
```

```
WRITE(IWR, '(6X, 2F20.10)') AMP, AZP
WRITE(IWR, '(/6X, 'MASSE (MEV) UND KERNLADUNGSZAHL ',
+ 'DES BEOBACHTETEN FRAGMENTS')')
WRITE(IWR, '(6X, 2F20.10)') AMA, AZA
WRITE(IWR, '(/6X, 'MASSE (MEV) UND KERNLADUNGSZAHL DES',
+ ' UNBEOBACHTETEN FRAGMENTS')')
WRITE(IWR, '(6X, 2F20.10)') AMB, AZB
WRITE(IWR, '(/6X, 'ENERGIE (MEV) DES PROJEKTILS IM ',
+ 'LABORSYSTEM')')
WRITE(IWR, '(6X, F20.10)') EO
WRITE(IWR, '(/6X, 'BINDUNGSENERGIE (MEV) DER CLUSTER ',
+ 'A, B IM PROJEKTIL' /6X, F20.10)') EPSAB
WRITE(IWR, '(/6X, 'Q-WERT DER REAKTION ', A40/
+ 6X, F20.10)') CREAKT, QU
WRITE(IWR, '(/6X,
+ 'KONSTANTE ZUR BERECHNUNG DES KERNRADIUS')')
WRITE(IWR, '(6X, F20.10)') RO
WRITE(IWR, '(/6X, 'REDUZIERTE MASSE (MEV)')')
WRITE(IWR, '(6X, F35.16)') AMUE
WRITE(IWR, '(/6X, 'HOEHE DER COULOMBSCHWELLE PRO EIN',
+ 'FALLENDER EINHEITSLADUNG (MEV)')')
WRITE(IWR, '(6X, F35.16)') VC
WRITE(IWR, '(/6X, 'KERNRADIUS DES TARGETS')')
WRITE(IWR, '(6X, F35.16)') RT
WRITE(IWR, '(/6X, 'KERNRADIUS DES PROJEKTILS')')
WRITE(IWR, '(6X, F35.16)') RP
WRITE(IWR, '(/6X, 'TOTALER AUFBRUCH-WIRKUNGS',
+ 'QUERSCHNITT' /6X, F35.16//)') RF
WRITE(IWR, '(21X, 33(''-'))')
WRITE(IWR, '(21X, '|          2 AMP      ( AT ) 1/3 |')')
WRITE(IWR, '(21X, '| EO >> ----- EPS ( ---- ) |')')
WRITE(IWR, '(21X, '|          AMA      ( AP ) |')')
WRITE(IWR, '(21X, '|          |')')
WRITE(IWR, '(21X, '| >> ', F21.16, 1X, '|')') EOF
WRITE(IWR, '(21X, 33(''-'))')
```

\*\*\*\*\*

\*

BERECHNUNG DER EINZELNEN FAELLE

\*\*\*\*\*

```
WRITE(IWR, '(/6X, 44('*'))/6X, '*      BERECHNUNG ',
+ 'DER EINZELNEN FAELLE      '* /6X, 44('*'))')
```

```
IF (II.EQ.1.OR.II.EQ.3) THEN
DO 142 JW=1,JM
J1 = JJ(JW)
KM = JKW(0,JW)
DO 100 K=1,KM
    KK(K) = JKW(K,JW)
100 CONTINUE
DO 141 KW=1,KM
    K1 = KK(KW)
    IF (J1.EQ.1) THEN
        IF (K1.EQ.0) THEN
            ***** I=1 , J=1 , K=0 *****
            WRITE(IWR,'(//11X, '*** I=1 , J=1 , K=0 ***'//)')
            WRITE(IWR,'(11X, 'TRANSPARENTES SERBER MODELL'//)')
            WRITE(IWR,'(11X, 'ENERGIEVERTEILUNG'//)')
            WRITE(IWR,'(11X, 'KEINE KORREKTUR DES MODELLS'//)')
            IF (IPR.GE.7) WRITE(IWR,'(16X, 'ENERGIE' , 26X,
+ 'D(SIGMA)/D(ENERGIE)'//11X, 60(' - '))')
            MP = 0
            IA = 0
            IB = IA + INT(EAMAX*1.0001/EADEL)
            IB = IB - INT(EAMIN*1.0001/EADEL)
            DO 110 I=IA, IB
                EAKT = EAMIN+EADEL*(I-IA)
                PHAS = 2.DO*AMA*AMB*
+                SQRT(AMA*AMB*(EAKT-QU)*(EO-QU-EAKT))
                WERT = SQRT(2.DO*AMUE*EPSAB)/(PI*(2.DO*AMUE*
+                EPSAB+(SQRT(2.DO*AMA*EAKT)-AMA/AMP*
+                SQRT(2.DO*AMP*EO)**2))*PHAS
                IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+                '(1X, 2F35.16)') EAKT,WERT
                IF (MP.EQ.0) YMAX=WERT
                IF (MP.EQ.0) YMIN=WERT
                IF (MP.EQ.0) PMAX=PHAS
                IF (MP.EQ.0) PMIN=PHAS
                IF (YMAX.LT.WERT) YMAX=WERT
                IF (YMIN.GT.WERT) YMIN=WERT
                IF (PMAX.LT.PHAS) PMAX=PHAS
                IF (PMIN.GT.PHAS) PMIN=PHAS
            MP=MP+1
```

```

XARR(MP,1)=EAKT
YARR(MP,1)=WERT
YPHA(MP,K1)=PHAS
110 CONTINUE
YARR(0,1)=MP
YARR(MP+1,1)=YMAX
YARR(MP+2,1)=YMIN
YPHA(MP+1,K1)=PMAX
YPHA(MP+2,K1)=PMIN
ELSE
IF (K1.EQ.1) THEN
***** I=1 , J=1 , K=1 *****
WRITE(IWR,'(//11X,' '*** I=1 , J=1 , K=1 ***' '/')')
WRITE(IWR,'(11X,' 'TRANSPARENTES SERBER MODELL' ')')
WRITE(IWR,'(11X,' 'ENERGIEVERTEILUNG' ')')
WRITE(IWR,'(11X,' 'COULOMB ENERGY DIFFERENCES' ',
+ ' ' CORRECTION' '/')')
IF (EO-AZP*VC.LE.O.DO) THEN
WRITE(IWR,'(//11X,' 'EOS <= 0. ' ',
+ ' 'CEDC NICHT MOEGlich.' '/')')
GOTO 141
ENDIF
IF (IPR.GE.7) WRITE(IWR,'(16X,' 'ENERGIE' ',26X,
+ ' 'D(SIGMA)/D(ENERGIE)' '/'11X,60(' '-'))')
MP = 0
IA = 0
IB = IA + INT(EAMAX*1.0001/EADEL)
IB = IB - INT(EAMIN*1.0001/EADEL)
DO 111 I=IA,IB
EAKT = EAMIN+EADEL*(I-IA)
EAS = EAKT-AZA*VC
PHAS = 2.DO*AMA*AMB*
+ SQRT(AMA*AMB*(EAS-QU)*(EOS-QU-EAS))
WERT = SQRT(2.DO*AMUE*EPSAB)/(PI*(2.DO*AMUE*
+ EPSAB+(SQRT(2.DO*AMA*EAS)-AMA/AMP*SQRT
+ (2.DO*AMP*EOS)**2))*PHAS
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') EAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
```



```
IF (MP.EQ.0) PMAX=PHAS
IF (MP.EQ.0) PMIN=PHAS
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
IF (PMAX.LT.PHAS) PMAX=PHAS
IF (PMIN.GT.PHAS) PMIN=PHAS
MP=MP+1
XARR(MP,2)=EAKT
YARR(MP,2)=WERT
YPHA(MP,K1)=PHAS
111 CONTINUE
YARR(0,2)=MP
YARR(MP+1,2)=YMAX
YARR(MP+2,2)=YMIN
YPHA(MP+1,K1)=PMAX
YPHA(MP+2,K1)=PMIN
ENDIF
ENDIF
ELSE
IF (J1.EQ.2) THEN
IF (K1.EQ.0) THEN
***** I=1 , J=2 , K=0 *****
WRITE(IWR, '(//11X, '*** I=1 , J=2 , K=0 ***'//)')
WRITE(IWR, '(11X, 'TRANSPARENTES SERBER MODELL'')')
WRITE(IWR, '(11X, 'WINKELVERTEILUNG'')')
WRITE(IWR, '(11X, 'KEINE KORREKTUR DES MODELLS'//)')
IF (IPR.GE.7) WRITE(IWR, '(16X, ' THETA ',26X,
+ ' D(SIGMA)/D(OMEGA) '//11X,60(''-'))')
MP = 0
IA = 0
IB = IA + INT(WIMAX*1.0001/WIDEL)
IB = IB - INT(WIMIN*1.0001/WIDEL)
DO 120 I=IA, IB
WIAKT = WIMIN+WIDEL*(I-IA)
TANG = TAN(CPI180*WIAKT)
WERT = THO/(2.DO*PI*(THO**2+(1.DO+THO**2)*
+ TANG**2)**(1.5DO))
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') WIAKT,WERT
IF (MP.EQ.0) YMAX=WERT
```

```
IF (MP.EQ.0) YMIN=WERT
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
MP=MP+1
XARR(MP,3)=WIAKT
YARR(MP,3)=WERT
120 CONTINUE
YARR(0,3)=MP
YARR(MP+1,3)=YMAX
YARR(MP+2,3)=YMIN
ELSE
IF (K1.EQ.1) THEN
***** I=1 , J=2 , K=1 *****
WRITE(IWR,'(//11X,' '*** I=1 , J=2 , K=1 ***' '/')')
WRITE(IWR,'(11X,' 'TRANSPARENTES SERBER MODELL' ')')
WRITE(IWR,'(11X,' 'WINKELVERTEILUNG' ')')
WRITE(IWR,'(11X,' 'COULOMB ENERGY DIFFERENCES' ,
+ ' ' CORRECTION' '/')')
IF (EO-AZP*VC.LE.O.DO) THEN
WRITE(IWR,'(//11X,' 'EOS <= 0. ' ,
+ ' 'CEDC NICHT MOEGLICH.' '/')')
GOTO 141
ENDIF
IF (IPR.GE.7) WRITE(IWR,'(16X,' ' THETA ' ',26X,
+ ' ' D(SIGMA)/D(OMEGA) ' '/11X,60(' '-''))')
MP = 0
IA = 0
IB = IA + INT(WIMAX*1.0001/WIDEL)
IB = IB - INT(WIMIN*1.0001/WIDEL)
DO 121 I=IA, IB
WIAKT = WIMIN+WIDEL*(I-IA)
TANG = TAN(CPI180*WIAKT)
WERT = THOS/(2.DO*PI*(THOS**2+(1.DO+THOS**2)*
+ TANG**2)**(1.5DO))
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') WIAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
```

```
MP=MP+1
XARR(MP,4)=WIAKT
YARR(MP,4)=WERT
121 CONTINUE
YARR(0,4)=MP
YARR(MP+1,4)=YMAX
YARR(MP+2,4)=YMIN
ELSE
IF (K1.EQ.2) THEN
***** I=1 , J=2 , K=2 *****
WRITE(IWR,'(//11X, '*** I=1 , J=2 , K=2 ***'//)')
WRITE(IWR,'(11X, 'TRANSPARENTES SERBER MODELL')')
WRITE(IWR,'(11X, 'WINKELVERTEILUNG')')
WRITE(IWR,'(11X, 'COULOMB DEFLECTION',
+ ' CORRECTIONS'//)')
IF (IPR.GE.7) WRITE(IWR,'(16X, ' THETA ',26X,
+ ' D(SIGMA)/D(OMEGA) '//11X,60(' - '))')
MP = 0
IA = 0
IB = IA + INT(WIMAX*1.0001/WIDEL)
IB = IB - INT(WIMIN*1.0001/WIDEL)
DO 122 I=IA, IB
WIAKT = WIMIN+WIDEL*(I-IA)
TANG = TAN(CPI180*WIAKT)
TQ = TANG/THO
TQC = 0.5DO*SQRT(AMA/AMB)*EC/SQRT(E0*EPSAB)*
+ (1.DO+AMP/AMA*AZA/AZP)
TU = (1.DO+TQ**2+TQC**2)/(SQRT((1.DO+TQ**2+
+ TQC**2)**2-4.DO*TQ**2*TQC**2))
TUS = 0.5DO*(TU-1.DO)
P12 = (P12Z*TUS+P12Y)*TUS+P12X*TUS+1.DO
WERT=TU*(1.5DO)*P12/(2.DO*PI*(1.DO+TQ**2+TQC**2)
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') WIAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
MP=MP+1
XARR(MP,5)=WIAKT
```

```

      YARR(MP,5)=WERT
122  CONTINUE
      YARR(0,5)=MP
      YARR(MP+1,5)=YMAX
      YARR(MP+2,5)=YMIN
      ELSE
      IF (K1.EQ.3) THEN
***** I=1 , J=2 , K=3 *****
      WRITE(IWR,'(//11X,' '*** I=1 , J=2 , K=3 ***' '/')')
      WRITE(IWR,'(11X,' 'TRANSPARENTES SERBER MODELL' ')')
      WRITE(IWR,'(11X,' 'WINKELVERTEILUNG' ')')
      WRITE(IWR,'(11X,' 'CEDC UND CDC BERUECKSICHTIGT' '/')')
      IF (EO-AZP*VC.LE.0.DO) THEN
        WRITE(IWR,'(//11X,' 'EOS <= 0. ',
+          'CEDC NICHT MOEGlich.' '/')')
        GOTO 141
      ENDIF
      IF (IPR.GE.7) WRITE(IWR,'(16X,' ' THETA ',26X,
+ ' D(SIGMA)/D(OMEGA) ' /11X,60(' - '))')
      MP = 0
      IA = 0
      IB = IA + INT(WIMAX*1.0001/WIDEL)
      IB = IB - INT(WIMIN*1.0001/WIDEL)
      DO 123 I=IA,IB
        WIAKT = WIMIN+WIDEL*(I-IA)
        TANG = TAN(CPI180*I)
        TQS = TANG/THOS
        TQCS = 0.5DO*SQRT(AMA/AMB)*EC/SQRT(EOS*EPSAB)*
+          (1.DO+AMP/AMA*AZA/AZP)
        TU = (1.DO+TQS**2+TQCS**2)/(SQRT((1.DO+TQS**2+
+          TQCS**2)**2-4.DO*TQS**2*TQCS**2))
        TUS = 0.5DO*(TU-1.DO)
        P12 = (P12Z*TUS+P12Y)*TUS+P12X*TUS+1.DO
        WERT=TU**(1.5DO)*P12/(2.DO*PI*
+          (1.DO+TQS**2+TQCS**2))
      IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') WIAKT,WERT
      IF (MP.EQ.0) YMAX=WERT
      IF (MP.EQ.0) YMIN=WERT
      IF (YMAX.LT.WERT) YMAX=WERT

```

```
IF (YMIN.GT.WERT) YMIN=WERT
MP=MP+1
XARR(MP,6)=WIAKT
YARR(MP,6)=WERT
123 CONTINUE
YARR(0,6)=MP
YARR(MP+1,6)=YMAX
YARR(MP+2,6)=YMIN
ENDIF
ENDIF
ENDIF
ENDIF
ELSE
IF (J1.EQ.3) THEN
IF (K1.EQ.0) THEN
***** I=1 , J=3 , K=0 *****
WRITE(IWR,'(//11X,' '*** I=1 , J=3 , K=0 ***'//)')
WRITE(IWR,'(11X,' 'TRANSPARENTES SERBER MODELL'')')
WRITE(IWR,'(11X,' 'INKLUSIVER WIRKUNGS' ,
+ ' 'QUERSCHNITT'')')
WRITE(IWR,'(11X,' 'KEINE KORREKTUR DES MODELLS'//)')
IF (IPR.GE.7) WRITE(IWR,'(16X,' 'ENERGIE' , 18X,
+ ' 'D2(SIGMA)/(D(OMEGA)D(ENERGIE))'//11X,60(' '-'))')
ETAKT = CPI180*WSTREU
MP = 0
IA = 0
IB = IA + INT(EAMAX*1.0001/EADEL)
IB = IB - INT(EAMIN*1.0001/EADEL)
DO 130 I=IA,IB
EAKT = EAMIN+EADEL*(I-IA)
PHAS = 2.DO*AMA*AMB*
+ SQRT(AMA*AMB*(EAKT-QU)*(EO-QU-EAKT))
WERT = RT*RP*SQRT(2.DO*AMUE*EPSAB)/
+ ((2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EO/AMP)-
+ SQRT(2.DO*AMA*EAKT)*COS(ETAKT))**2+(SQRT(2.
+ DO*AMA*EAKT)*SIN(ETAKT))**2)**2)*PHAS
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') EAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
```

```
      IF (MP.EQ.0) PMAX=PHAS
      IF (MP.EQ.0) PMIN=PHAS
      IF (YMAX.LT.WERT) YMAX=WERT
      IF (YMIN.GT.WERT) YMIN=WERT
      IF (PMAX.LT.PHAS) PMAX=PHAS
      IF (PMIN.GT.PHAS) PMIN=PHAS
      MP=MP+1
      XARR(MP,7)=EAKT
      YARR(MP,7)=WERT
      YPHA(MP,K1)=PHAS
130  CONTINUE
      YARR(0,7)=MP
      YARR(MP+1,7)=YMAX
      YARR(MP+2,7)=YMIN
      YPHA(MP+1,K1)=PMAX
      YPHA(MP+2,K1)=PMIN
      ELSE
      IF (K1.EQ.1) THEN
***** I=1 , J=3 , K=1 *****
      WRITE(IWR,'(//11X,' '*** I=1 , J=3 , K=1 ***'//)')
      WRITE(IWR,'(11X,' 'TRANSPARENTES SERBER MODELL'//)')
      WRITE(IWR,'(11X,' 'INKLUSIVER WIRKUNGS' ,
+      ' 'QUERSCHNITT'//)')
      WRITE(IWR,'(11X,' 'COULOMB ENERGY DIFFERENCES' ,
+      ' ' CORRECTION'//)')
      IF (EO-AZP*VC.LE.O.DO) THEN
      WRITE(IWR,'(//11X,' 'EOS <= 0. ' ,
+      ' 'CEDC NICHT MOEGlich.'//)')
      GOTO 141
      ENDIF
      IF (IPR.GE.7) WRITE(IWR,'(16X,' 'ENERGIE' ,18X,
+      ' 'D2(SIGMA)/(D(OMEGA)D(ENERGIE))'//11X,60(' '-'))')
      ETAKT = CPI180*WSTREU
      MP = 0
      IA = 0
      IB = IA + INT(EAMAX*1.0001/EADEL)
      IB = IB - INT(EAMIN*1.0001/EADEL)
      DO 131 I=IA,IB
      EAKT = EAMIN+EADEL*(I-IA)
      EAS = EAKT-AZA*VC
```

```
PHAS = 2.DO*AMA*AMB*
+      SQRT(AMA*AMB*(EAS-QU)*(EO-QU-EAS))
WERT = RT*RP*SQRT(2.DO*AMUE*EPSAB)/
+      ((2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EOS/AMP)-
+      SQRT(2.DO*AMA*EAS)*COS(ETA KT))**2+(SQRT(2.
+      DO*AMA*EAS)*SIN(ETA KT))**2)**2)*PHAS
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+      '(1X,2F35.16)') EAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
IF (MP.EQ.0) PMAX=PHAS
IF (MP.EQ.0) PMIN=PHAS
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
IF (PMAX.LT.PHAS) PMAX=PHAS
IF (PMIN.GT.PHAS) PMIN=PHAS
MP=MP+1
XARR(MP,8)=EAKT
YARR(MP,8)=WERT
YPHA(MP,K1)=PHAS
131  CONTINUE
YARR(0,8)=MP
YARR(MP+1,8)=YMAX
YARR(MP+2,8)=YMIN
YPHA(MP+1,K1)=PMAX
YPHA(MP+2,K1)=PMIN
ENDIF
ENDIF
ENDIF
ENDIF
ENDIF
141 CONTINUE
142 CONTINUE
ENDIF
IF (II.EQ.2.OR.II.EQ.3) THEN
DO 242 JW=1,JM
J1 = JJ(JW)
KM = JKW(0,JW)
DO 143 K=1,KM
KK(K) = JKW(K,JW)
```

143 CONTINUE

```
DO 241 KW=1,KM
  K1 = KK(KW)
  IF (J1.EQ.1) THEN
    PIRTP = PI*0.5DO*RT*RP*AMUE*EPSAB
    IF (K1.EQ.0) THEN
      ***** I=2 , J=1 , K=0 *****
      WRITE(IWR,'(//11X,' '*** I=2 , J=1 , K=0 ***' '/')')
      WRITE(IWR,'(11X,' 'OPAKES SERBER MODELL' ')')
      WRITE(IWR,'(11X,' 'ENERGIEVERTEILUNG' ')')
      WRITE(IWR,'(11X,' 'KEINE KORREKTUR DES MODELLS' '/')')
      IF (IPR.GE.7) WRITE(IWR,'(16X,' 'ENERGIE' ',26X,
+      'D(SIGMA)/D(ENERGIE)' '/11X,60(' '-' '))')
      MP = 0
      IA = 0
      IB = IA + INT(EAMAX*1.0001/EADEL)
      IB = IB - INT(EAMIN*1.0001/EADEL)
      DO 210 I=IA, IB
        EAKT = EAMIN+EADEL*(I-IA)
        PHAS = 2.DO*AMA*AMB*
+        SQRT(AMA*AMB*(EAKT-QU)*(EO-QU-EAKT))
        WERT = PIRTP/((2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA
+        *EAKT)-AMA/AMP*SQRT(2.DO*AMP*EO))**2)**
+        (1.5DO))*PHAS
      IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+      '(1X,2F35.16)') EAKT,WERT
      IF (MP.EQ.0) YMAX=WERT
      IF (MP.EQ.0) YMIN=WERT
      IF (MP.EQ.0) PMAX=PHAS
      IF (MP.EQ.0) PMIN=PHAS
      IF (YMAX.LT.WERT) YMAX=WERT
      IF (YMIN.GT.WERT) YMIN=WERT
      IF (PMAX.LT.PHAS) PMAX=PHAS
      IF (PMIN.GT.PHAS) PMIN=PHAS
      MP=MP+1
      XARR(MP,9)=EAKT
      YARR(MP,9)=WERT
      YPHA(MP,K1)=PHAS
210 CONTINUE
      YARR(0,9)=MP
```



```
YARR(MP+1,9)=YMAX
YARR(MP+2,9)=YMIN
YPHA(MP+1,K1)=PMAX
YPHA(MP+2,K1)=PMIN
ELSE
  IF (K1.EQ.1) THEN
***** I=2 , J=1 , K=1 *****
WRITE(IWR,'(//11X, '*** I=2 , J=1 , K=1 ***'//)')
WRITE(IWR,'(11X, 'OPAKES SERBER MODELL')')
WRITE(IWR,'(11X, 'ENERGIEVERTEILUNG')')
WRITE(IWR,'(11X, 'COULOMB ENERGY DIFFERENCES',
+ ' CORRECTION'//)')
  IF (EO-AZP*VC.LE.O.DO) THEN
    WRITE(IWR,'(//11X, 'EOS <= 0. ',
+ ' CEDC NICHT MOEGlich.'//)')
    GOTO 241
  ENDIF
  IF (IPR.GE.7) WRITE(IWR,'(16X, 'ENERGIE',26X,
+ 'D(SIGMA)/D(ENERGIE)'//11X,60('-''))')
  MP = 0
  IA = 0
  IB = IA + INT(EAMAX*1.0001/EADEL)
  IB = IB - INT(EAMIN*1.0001/EADEL)
  DO 211 I=IA, IB
    EAKT = EAMIN+EADEL*(I-IA)
    EAS = EAKT-AZA*VC
    PHAS = 2.DO*AMA*AMB*
+ SQRT(AMA*AMB*(EAS-QU)*(EO-QU-EAS))
    WERT = PIRTP/((2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA*
+ EAS)-AMA/AMP*SQRT(2.DO*AMP*EOS))**2)**
+ (1.5DO))*PHAS
    IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') EAKT,WERT
    IF (MP.EQ.0) YMAX=WERT
    IF (MP.EQ.0) YMIN=WERT
    IF (MP.EQ.0) PMAX=PHAS
    IF (MP.EQ.0) PMIN=PHAS
    IF (YMAX.LT.WERT) YMAX=WERT
    IF (YMIN.GT.WERT) YMIN=WERT
    IF (PMAX.LT.PHAS) PMAX=PHAS
```

```

      IF (PMIN.GT.PHAS) PMIN=PHAS
      MP=MP+1
      XARR(MP,10)=EAKT
      YARR(MP,10)=WERT
      YPHA(MP,K1)=PHAS
211  CONTINUE
      YARR(0,10)=MP
      YARR(MP+1,10)=YMAX
      YARR(MP+2,10)=YMIN
      YPHA(MP+1,K1)=PMAX
      YPHA(MP+2,K1)=PMIN
      ENDIF
      ENDIF
      ELSE
      IF (J1.EQ.2) THEN
      IF (K1.EQ.0) THEN
***** I=2 , J=2 , K=0 *****
      WRITE(IWR,'(//11X, '*** I=2 , J=2 , K=0 ***'//)')
      WRITE(IWR,'(11X, 'OPAKES SERBER MODELL')')
      WRITE(IWR,'(11X, 'WINKELVERTEILUNG')')
      WRITE(IWR,'(11X, 'KEINE KORREKTUR DES MODELLS')')
      IF (IPR.GE.7) WRITE(IWR,'(16X, ' THETA ',26X,
+ ' D(SIGMA)/D(OMEGA) '//11X,60(' - '))')
      MP = 0
      IA = 0
      IB = IA + INT(WIMAX*1.0001/WIDEL)
      IB = IB - INT(WIMIN*1.0001/WIDEL)
      DO 220 I=IA,IB
      WIAKT = WIMIN+WIDEL*(I-IA)
      TANG = TAN(CPI180*WIAKT)
      IF (ABS(TANG).GT.EPSTH) THEN
      WERT = (RT*RP/PI)*THO/(THO**2+(1.DO+
+ THO**2)*TANG**2)**(1.5DO)*(1.DO-THO**3/
+ (2.DO*(1.DO+3.DO*THO**2)*TANG**3)*((1.DO+
+ (1.DO+1.DO/THO**2)*TANG**2)*ASIN(TANG/(SQRT
+ (THO**2+(1.DO+THO**2)*TANG**2)))-TANG/THO))
      ELSE
      WIAKT = C180PI*ATAN(EPSTH)
      TANG = EPSTH
      WERT = (RT*RP/PI)*THO/
```

```
+          (THO**2+(1.DO+THO**2)*TANG**2)**(1.5DO)*
+          (1.DO-THO**3/(2.DO*(1.DO+3.DO*THO**2))*
+          TANG**3)*((1.DO+(1.DO+1.DO/THO**2)*TANG**2)
+          *ASIN(TANG/(SQRT(THO**2+(1.DO+THO**2)*
+          TANG**2)))-TANG/THO))
+          IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+          '(1X,2F35.16)') WIAKT,WERT
+          IF (MP.EQ.0) YMAX=WERT
+          IF (MP.EQ.0) YMIN=WERT
+          IF (YMAX.LT.WERT) YMAX=WERT
+          IF (YMIN.GT.WERT) YMIN=WERT
+          MP=MP+1
+          XARR(MP,11)=-WIAKT
+          YARR(MP,11)=WERT
+
+          ENDIF
+          IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+          '(1X,2F35.16)') WIAKT,WERT
+          IF (MP.EQ.0) YMAX=WERT
+          IF (MP.EQ.0) YMIN=WERT
+          IF (YMAX.LT.WERT) YMAX=WERT
+          IF (YMIN.GT.WERT) YMIN=WERT
+          MP=MP+1
+          XARR(MP,11)=WIAKT
+          YARR(MP,11)=WERT
220      CONTINUE
+          YARR(0,11)=MP
+          YARR(MP+1,11)=YMAX
+          YARR(MP+2,11)=YMIN
+          ELSE
+          IF (K1.EQ.1) THEN
+          ***** I=2 , J=2 , K=1 *****
+          WRITE(IWR,'(//11X,' '*** I=2 , J=2 , K=1 ***'//)')
+          WRITE(IWR,'(11X,' 'OPAKES SERBER MODELL'//)')
+          WRITE(IWR,'(11X,' 'WINKELVERTEILUNG'//)')
+          WRITE(IWR,'(11X,' 'COULOMB ENERGY DIFFERENCES' ,
+          ' ' CORRECTION'//)')
+          IF (EO-AZP*VC.LE.0.DO) THEN
+          WRITE(IWR,'(//11X,' 'EOS <= 0. ' ,
+          ' 'CEDC NICHT MOEGLICH.'//)')
+          GOTO 241
```

```
      ENDIF
      IF (IPR.GE.7) WRITE(IWR,'(16X,' THETA ',26X,
+      ' ' D(SIGMA)/D(OMEGA) ' '/11X,60(''-'))')
      MP = 0
      IA = 0
      IB = IA + INT(WIMAX*1.0001/WIDEL)
      IB = IB - INT(WIMIN*1.0001/WIDEL)
      DO 221 I=IA, IB
        WIAKT = WIMIN+WIDEL*(I-IA)
        TANG = TAN(CPI180*WIAKT)
        IF (ABS(TANG).GT.EPSTH) THEN
          WERT = (RT*RP/PI)*THOS/
+          (THOS**2+(1.DO+THOS**2)*TANG**2)**(1.5DO)*
+          (1.DO-THOS**3/(2.DO*(1.DO+3.DO*THOS**2)*
+          TANG**3)*((1.DO+(1.DO+1.DO/THOS**2)*TANG**2)
+          *ASIN(TANG/(SQRT(THOS**2+(1.DO+THOS**2)*
+          TANG**2)))-TANG/THOS))
        ELSE
          WIAKT = C180PI*ATAN(EPSTH)
          TANG = EPSTH
          WERT = (RT*RP/PI)*THOS/
+          (THOS**2+(1.DO+THOS**2)*TANG**2)**(1.5DO)*
+          (1.DO-THOS**3/(2.DO*(1.DO+3.DO*THOS**2)*
+          TANG**3)*((1.DO+(1.DO+1.DO/THOS**2)*TANG**2)
+          *ASIN(TANG/(SQRT(THOS**2+(1.DO+THOS**2)*
+          TANG**2)))-TANG/THOS))
          IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+          '(1X,2F35.16)') WIAKT,WERT
          IF (MP.EQ.0) YMAX=WERT
          IF (MP.EQ.0) YMIN=WERT
          IF (YMAX.LT.WERT) YMAX=WERT
          IF (YMIN.GT.WERT) YMIN=WERT
          MP=MP+1
          XARR(MP,12)=-WIAKT
          YARR(MP,12)=WERT
        ENDIF
      IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+      '(1X,2F35.16)') WIAKT,WERT
      IF (MP.EQ.0) YMAX=WERT
      IF (MP.EQ.0) YMIN=WERT
```

```
      IF (YMAX.LT.WERT) YMAX=WERT
      IF (YMIN.GT.WERT) YMIN=WERT
      MP=MP+1
      XARR(MP,12)=WIAKT
      YARR(MP,12)=WERT
221  CONTINUE
      YARR(0,12)=MP
      YARR(MP+1,12)=YMAX
      YARR(MP+2,12)=YMIN
      ELSE
      IF (K1.EQ.2) THEN
***** I=2 , J=2 , K=2 *****
      WRITE(IWR,'(//11X,'''*** I=2 , J=2 , K=2 ***''/)'')
      WRITE(IWR,'(11X,'''OPAKES SERBER MODELL''')')
      WRITE(IWR,'(11X,'''WINKELVERTEILUNG''')')
      WRITE(IWR,'(11X,'''COULOMB DEFLECTION''',
+      '' CORRECTIONS''/)'')
      IF (IPR.GE.7) WRITE(IWR,'(16X,''' THETA ''',26X,
+      '' D(SIGMA)/D(OMEGA) '''/11X,60('''-'''))')
      MP = 0
      IA = 0
      IB = IA + INT(WIMAX*1.0001/WIDEL)
      IB = IB - INT(WIMIN*1.0001/WIDEL)
      DO 222 I=IA,IB
          WIAKT = WIMIN+WIDEL*(I-IA)
          TANG = TAN(CPI180*WIAKT)
          TQ = TANG/THO
          TQC = 0.5DO*SQRT(AMA/AMB)*EC/SQRT(E0*EPSAB)*
+          (1.DO+AMP/AMA*AZA/AZP)
          TU = (1.DO+TQ**2+TQC**2)/(SQRT((1.DO+TQ**2+
+          TQC**2)**2-4.DO*TQ**2*TQC**2))
          TUS = 0.5DO*(TU-1.DO)
          P12 = (P12Z*TUS+P12Y)*TUS+P12X*TUS+1.DO
          IF (ABS(TANG).GT.EPSTH) THEN
+          WERT = TU**(1.5DO)*P12/(2.DO*PI*(1.DO+TQ**2+
+          TQC**2))*(1.DO-THO**3/(2.DO*(1.DO+3.DO*
+          THO**2)*TANG**3)*((1.DO+(1.DO+1.DO/THO**2)*
+          TANG**2)*ASIN(TANG/(SQRT(THO**2+(1.DO+
+          THO**2)*TANG**2)))-TANG/THO))
          ELSE
```

```
WIAKT = C180PI*ATAN(EPSTH)
TANG = EPSTH
WERT = TU**(1.5DO)*P12/(2.DO*PI*(1.DO+TQ**2+
+ TQC**2))*(1.DO-THO**3/(2.DO*(1.DO+3.DO*
+ THO**2)*TANG**3)*((1.DO+(1.DO+1.DO/THO**2)*
+ TANG**2)*ASIN(TANG/(SQRT(THO**2+(1.DO+
+ THO**2)*TANG**2))))-TANG/THO)
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') WIAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
MP=MP+1
XARR(MP,13)=-WIAKT
YARR(MP,13)=WERT
ENDIF
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+ '(1X,2F35.16)') WIAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
MP=MP+1
XARR(MP,13)=WIAKT
YARR(MP,13)=WERT
222 CONTINUE
YARR(0,13)=MP
YARR(MP+1,13)=YMAX
YARR(MP+2,13)=YMIN
ELSE
IF (K1.EQ.3) THEN
***** I=2 , J=2 , K=3 *****
WRITE(IWR,'(//11X, ' '*** I=2 , J=2 , K=3 ***' '/')')
WRITE(IWR,'(11X, ' 'OPAKES SERBER MODELL' ')')
WRITE(IWR,'(11X, ' 'WINKELVERTEILUNG' ')')
WRITE(IWR,'(11X, ' 'CEDC UND CDC BERUECKSICHTIGT' '/')')
IF (EO-AZP*VC.LE.0.DO) THEN
WRITE(IWR,'(//11X, ' 'EOS <= 0. ' ',
+ ' 'CEDC NICHT MOEGLICH.' '/')')
```

```
GOTO 241
ENDIF
IF (IPR.GE.7) WRITE(IWR,'(16X,' THETA ',26X,
+ ' D(SIGMA)/D(OMEGA) '/11X,60(''-'))')
MP = 0
IA = 0
IB = IA + INT(WIMAX*1.0001/WIDEL)
IB = IB - INT(WIMIN*1.0001/WIDEL)
DO 223 I=IA, IB
  WIAKT = WIMIN+WIDEL*(I-IA)
  TANG = TAN(CPI180*WIAKT)
  TQS = TANG/THOS
  TQCS = 0.5DO*SQRT(AMA/AMB)*EC/SQRT(EOS*EPSAB)*
+   (1.DO+AMP/AMA*AZA/AZP)
  TU = (1.DO+TQS**2+TQCS**2)/(SQRT
+   ((1.DO+TQS**2+TQCS**2)**2-4.DO*TQS**2*
+   TQCS**2))
  TUS = 0.5DO*(TU-1.DO)
  P12 = (P12Z*TUS+P12Y)*TUS+P12X*TUS+1.DO
  IF (ABS(TANG).GT.EPSTH) THEN
    WERT = TU**(1.5DO)*P12/(2.DO*PI*(1.DO+TQS**2+
+   TQCS**2))*(1.DO-THOS**3/(2.DO*(1.DO+3.DO*
+   THOS**2)*TANG**3)*((1.DO+(1.DO+1.DO/THOS**2)*
+   TANG**2)*ASIN(TANG/(SQRT(THOS**2+
+   (1.DO+THOS**2)*TANG**2))))-TANG/THOS))
  ELSE
    WIAKT = C180PI*ATAN(EPSTH)
    TANG = EPSTH
    WERT = TU**(1.5DO)*P12/(2.DO*PI*(1.DO+TQS**2+
+   TQCS**2))*(1.DO-THOS**3/(2.DO*(1.DO+3.DO*
+   THOS**2)*TANG**3)*((1.DO+(1.DO+1.DO/THOS**2)*
+   TANG**2)*ASIN(TANG/(SQRT(THOS**2+
+   (1.DO+THOS**2)*TANG**2))))-TANG/THOS))
    IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+   '(1X,2F35.16)') WIAKT,WERT
    IF (MP.EQ.0) YMAX=WERT
    IF (MP.EQ.0) YMIN=WERT
    IF (YMAX.LT.WERT) YMAX=WERT
    IF (YMIN.GT.WERT) YMIN=WERT
    MP=MP+1
```

```

      XARR(MP,14)=-WIAKT
      YARR(MP,14)=WERT
      ENDIF
      IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+      '(1X,2F35.16)') WIAKT,WERT
      IF (MP.EQ.0) YMAX=WERT
      IF (MP.EQ.0) YMIN=WERT
      IF (YMAX.LT.WERT) YMAX=WERT
      IF (YMIN.GT.WERT) YMIN=WERT
      MP=MP+1
      XARR(MP,14)=WIAKT
      YARR(MP,14)=WERT
223  CONTINUE
      YARR(0,14)=MP
      YARR(MP+1,14)=YMAX
      YARR(MP+2,14)=YMIN
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ELSE
      IF (J1.EQ.3) THEN
      IF (K1.EQ.0) THEN
***** I=2 , J=3 , K=0 *****
      WRITE(IWR,'(//11X, ' '*** I=2 , J=3 , K=0 ***' '/')')
      WRITE(IWR,'(11X, ' 'OPAKES SERBER MODELL' ')')
      WRITE(IWR,'(11X, ' 'INKLUSIVER WIRKUNGS' ',
+      ' 'QUERSCHNITT' ')')
      WRITE(IWR,'(11X, ' 'KEINE KORREKTUR DES MODELLS' '/')')
      IF (IPR.GE.7) WRITE(IWR,'(16X, ' 'ENERGIE' ',18X,
+      ' 'D2(SIGMA)/(D(OMEGA)D(ENERGIE))' '/11X,60(' '-' ')')')
      ETAKT = CPI180*WSTREU
      MP = 0
      IA = 0
      IB = IA + INT(EAMAX*1.0001/EADEL)
      IB = IB - INT(EAMIN*1.0001/EADEL)
      DO 230 I=IA,IB
      EAKT = EAMIN+EADEL*(I-IA)
      TU = (2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA*EAKT)*
+      COS(ETAKT))-AMA/AMP*SQRT(2.DO*AMP*EO))**2+0.5DO*
```



```
+      (SQRT(2.DO*AMA*EAKT)*SIN(ETA KT))**2)/
+      SQRT(2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EO/AMP)-
+      SQRT(2.DO*AMA*EAKT)*COS(ETA KT))**2+
+      (SQRT(2.DO*AMA*EAKT)*SIN(ETA KT))**2)/
+      SQRT(2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EO/AMP)-
+      SQRT(2.DO*AMA*EAKT)*COS(ETA KT))**2)
TUS = 0.5DO*(TU-1.DO)
P12 = (P12Z*TUS+P12Y)*TUS+P12X*TUS+1.DO
PHAS = 2.DO*AMA*AMB*
+      SQRT(AMA*AMB*(EAKT-QU)*(EO-QU-EAKT))
WERT = RT*RP*SQRT(2.DO*AMUE*EPSAB)/
+      ((2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EO/AMP)-
+      SQRT(2.DO*AMA*EAKT)*COS(ETA KT))**2+(SQRT(2.DO*
+      AMA*EAKT)*SIN(ETA KT))**2)**2)*PHAS*(2.DO*
+      AMUE*EPSAB+(SQRT(2.DO*AMA**2*EO/AMP)-SQRT(2.DO*
+      AMA*EAKT)*COS(ETA KT))**2+(SQRT(2.DO*AMA*EAKT)*
+      SIN(ETA KT))**2*P12)/(2.DO*AMUE*EPSAB+
+      (SQRT(2.DO*AMA**2*EO/AMP)-SQRT(2.DO*AMA*EAKT))*
+      COS(ETA KT))**2)*(0.75DO)
IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+      '(1X,2F35.16)') EAKT,WERT
IF (MP.EQ.0) YMAX=WERT
IF (MP.EQ.0) YMIN=WERT
IF (MP.EQ.0) PMAX=PHAS
IF (MP.EQ.0) PMIN=PHAS
IF (YMAX.LT.WERT) YMAX=WERT
IF (YMIN.GT.WERT) YMIN=WERT
IF (PMAX.LT.PHAS) PMAX=PHAS
IF (PMIN.GT.PHAS) PMIN=PHAS
MP=MP+1
XARR(MP,15)=EAKT
YARR(MP,15)=WERT
YPHA(MP,K1)=PHAS
230 CONTINUE
YARR(0,15)=MP
YARR(MP+1,15)=YMAX
YARR(MP+2,15)=YMIN
YPHA(MP+1,K1)=PMAX
YPHA(MP+2,K1)=PMIN
ELSE
```

```
IF (K1.EQ.1) THEN
***** I=2 , J=3 , K=1 *****
WRITE(IWR,'(//11X,' '*** I=2 , J=3 , K=1 ***'//)')
WRITE(IWR,'(11X,' 'OPAKES SERBER MODELL'')')
WRITE(IWR,'(11X,' 'INKLUSIVER WIRKUNGS' ,
+ 'QUERSCHNITT'')')
WRITE(IWR,'(11X,' 'COULOMB ENERGY DIFFERENCES' ,
+ 'CORRECTION'//)')
IF (EO-AZP*VC.LE.0.DO) THEN
WRITE(IWR,'(//11X,' 'EOS <= 0. ' ,
+ 'CEDC NICHT MOEGlich.'//)')
GOTO 241
ENDIF
IF (IPR.GE.7) WRITE(IWR,'(16X,' 'ENERGIE' ,18X,
+ 'D2(SIGMA)/(D(OMEGA)D(ENERGIE))'//11X,60(' - '))')
ETA KT = CPI180*WSTREU
MP = 0
IA = 0
IB = IA + INT(EAMAX*1.0001/EADEL)
IB = IB - INT(EAMIN*1.0001/EADEL)
DO 231 I=IA,IB
EAKT = EAMIN+EADEL*(I-IA)
EAS = EAKT-AZA*VC
TU = (2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA*EAS)*
+ COS(ETA KT)-AMA/AMP*SQRT(2.DO*AMP*EOS))**2+
+ 0.5DO*(SQRT(2.DO*AMA*EAS)*SIN(ETA KT))**2)/
+ SQRT(2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EOS/
+ AMP)-SQRT(2.DO*AMA*EAS)*COS(ETA KT))**2+
+ (SQRT(2.DO*AMA*EAS)*SIN(ETA KT))**2)/
+ SQRT(2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EOS/
+ AMP)-SQRT(2.DO*AMA*EAS)*COS(ETA KT))**2)
TUS = 0.5DO*(TU-1.DO)
P12 = (P12Z*TUS+P12Y)*TUS+P12X*TUS+1.DO
PHAS = 2.DO*AMA*AMB*
+ SQRT(AMA*AMB*(EAS-QU)*(EO-QU-EAS))
WERT = RT*RP*SQRT(2.DO*AMUE*EPSAB)/
+ ((2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EOS/AMP)-
+ SQRT(2.DO*AMA*EAS)*COS(ETA KT))**2+
+ (SQRT(2.DO*AMA*EAS)*SIN(ETA KT))**2)**2)*PHAS*
+ (2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EOS/AMP)-
```

```
+      SQRT(2.DO*AMA*EAS)*COS(ETA KT)**2+
+      (SQRT(2.DO*AMA*EAS)*SIN(ETA KT)**2*P12)/
+      (2.DO*AMUE*EPSAB+(SQRT(2.DO*AMA**2*EOS/AMP)-
+      SQRT(2.DO*AMA*EAS)*COS(ETA KT)**2)**(0.75DO)
+      IF (IPR.GE.7.AND.MOD(MP,10).EQ.0) WRITE(IWR,
+      '(1X,2F35.16)') EAKT,WERT
+      IF (MP.EQ.0) YMAX=WERT
+      IF (MP.EQ.0) YMIN=WERT
+      IF (MP.EQ.0) PMAX=PHAS
+      IF (MP.EQ.0) PMIN=PHAS
+      IF (YMAX.LT.WERT) YMAX=WERT
+      IF (YMIN.GT.WERT) YMIN=WERT
+      IF (PMAX.LT.PHAS) PMAX=PHAS
+      IF (PMIN.GT.PHAS) PMIN=PHAS
+      MP=MP+1
+      XARR(MP,16)=EAKT
+      YARR(MP,16)=WERT
+      YPHA(MP,K1)=PHAS
```

```
231      CONTINUE
+      YARR(0,16)=MP
+      YARR(MP+1,16)=YMAX
+      YARR(MP+2,16)=YMIN
+      YPHA(MP+1,K1)=PMAX
+      YPHA(MP+2,K1)=PMIN
+      ENDIF
+      ENDIF
+      ENDIF
+      ENDIF
```

241 CONTINUE

242 CONTINUE

ENDIF

\*\*\*\*\*

\* P L O T T E N D E R I/J-KOMBINATIONEN

\*\*\*\*\*

```
IF (IPL.LE.0) GOTO 399
WRITE(IWR, '(//6X,44(''*'')/6X, '*' PLOTTEN DER '' ,
+' 'I/J-KOMBINATIONEN      *''/6X,44(''*'')/)' )
LBILD = 0
IIA = 1
```

```
IIB = 2
IF (II.EQ.1) IIB = 1
IF (II.EQ.2) IIA = 2
DO 392 I1=IIA,IIB
DO 391 JW=1,JM
J1 = JJ(JW)
KM = JKW(0,JW)
NKURV = 0
DO 300 KW=1,KM
  K1 = JKW(KW,JW)
  LTAB(KW) = 1 + K1 + J1*(J1-1) + (I1-1)*8
  NKURV = NKURV + 1
  KK(NKURV) = K1
300 CONTINUE
  IF (NKURV.LE.0) GOTO 391
  LBILD = LBILD + 1
  WRITE(IWR,'(//6X,' 'BILD-KENNUNG: I=' ',
+ I1,' ', J=' ',I1,''.')') I1,J1
***** ACHSEN-MAXIMA UND -MINIMA BESTIMMEN *****
DO 301 K=1,NKURV
  MB = LTAB(K)
  MP = YARR(0,MB)
  IF (K.GT.1) THEN
    IF (YARR(MP+1,MB).GT.YMAX) YMAX=YARR(MP+1,MB)
    IF (YARR(MP+2,MB).LT.YMIN) YMIN=YARR(MP+2,MB)
    IF (XARR(MP,MB).GT.XMAX) XMAX=XARR(MP,MB)
    IF (XARR(1,MB).LT.XMIN) XMIN=XARR(1,MB)
  ELSE
    YMAX = YARR(MP+1,MB)
    YMIN = YARR(MP+2,MB)
    XMAX = XARR(MP,MB)
    XMIN = XARR(1,MB)
  ENDIF
301 CONTINUE
***** SOLLWERTE DER Y-MAXIMA BERUECKSICHTIGEN
YPHA(0,0) = 1.
YPHA(0,1) = 1.
MM = NYN(0,J1)
IF (MM.GE.1) THEN
  MA = NYN(-1,J1)
```

```
MB = LTAB(MA)
MP = YARR(0,MB)
YFAKT = 1.000001 * YNORM(MA,J1)/YARR(MP+1,MB)
WRITE(IWR,'(/11X,' 'ANZAHL "NEUER" SOLL-MAXIMA =',
+      I2)') NYN(0,J1)
WRITE(IWR,'(/11X,' 'KURVEN-NUMMER DES ERSTEN ',
+      'SOLL-MAXIMUMS =',I2)') NYN(-1,J1)
WRITE(IWR,'(/11X,' 'YMIN, YMAX =',1P2E13.5,
+      ' "ALT"')') YARR(MP+2,MB),YARR(MP+1,MB)
WRITE(IWR,'(/17X,' 'SOLLWERT VON YMAX =',
+      1PE13.5)') YNORM(MA,J1)
WRITE(IWR,'(/20X,' 'TRANSF.-FAKTOR =',
+      1PE13.5)') YFAKT
YMAX = YFAKT * YARR(MP+1,MB)
YMIN = YFAKT * YARR(MP+2,MB)
YSOLL = YMAX
IA = 1
IB = MA
DO 304 M=1,MM
302 IF (NYN(IB+1,J1).LT.0) THEN
      IB = IB + 1
      IF (IB.LT.NKURV) GOTO 302
      ENDIF
DO 303 IK=IA,IB
      MB = LTAB(IK)
      IF (NPHAS.GT.0.AND.J1.NE.2) YPHA(0,KK(IK))=YFAKT
      DO 303 I=1,MP
          YARR(I,MB) = YARR(I,MB) * YFAKT
303 CONTINUE
      IF (MM.GE.2.AND.M.LT.MM) THEN
          IA = IB + 1
          IB = IA
          MA = IB
          MB = LTAB(MA)
          MP = YARR(0,MB)
          YFAKT = 1.000001 * YNORM(MA,J1)/YARR(MP+1,MB)
          WRITE(IWR,'(/11X,' 'KURVEN-NUMMER DES NAECHSTEN ',
+          'SOLL-MAXIMUMS =',I2)') MA
          WRITE(IWR,'(/11X,' 'YMIN, YMAX =',1P2E13.5,
+          ' "ALT"')') YARR(MP+2,MB),YARR(MP+1,MB)
```

```
      WRITE(IWR, '( /17X, ''SOLLWERT VON YMAX = '' ,
+         1PE13.5)') YNORM(MA, J1)
      WRITE(IWR, '( /20X, ''TRANSE.-FAKTOR = '' ,
+         1PE13.5)') YFAKT
      YMAX = YFAKT * YARR(MP+1, MB)
      YMIN = YFAKT * YARR(MP+2, MB)
      IF (YMAX.GT.YSOLL) YSOLL = YMAX
      ENDIF
304  CONTINUE
      YMAX = YSOLL
      YMIN = 0.
      ENDIF
***** Y-ACHSEN-MINIMUM UND -MAXIMUM KORRIGIEREN
      ZYIB = 1.
      IYMI = 0
      IYMA = 0
      IEYMI = 0
      IEYMA = 0
      IF (MM.GT.0) THEN
        ZYMI = YMIN
        ZYMA = YMAX
      ELSE
        IF (ABS(YMIN).GT.1.E-4) IEYMI = LOG10(ABS(YMIN))
        IF (ABS(YMAX).GT.1.E-4) IEYMA = LOG10(ABS(YMAX))
        IYMI = ABS(YMIN)/10.DO**IEYMI
        IF (YMIN.GT.0.) THEN
          ZYMI = IYMI
        ELSE
          IF (YMIN.EQ.0.) THEN
            ZYMI = IYMI
          ELSE
            ZYMI = -(1+IYMI)
          ENDIF
        ENDIF
      ENDIF
      IF (ABS(ZYMI).LT.1.E-4) IEYMI = 0
      IYMA = ABS(YMAX)/10.DO**IEYMA
      IF (IEYMA.LT.3) THEN
        ZYMA = 10.DO**IEYMA
        IEYMA = 0
      ELSE
```

```
      ZYMA = 1.
ENDIF
      ZYMI = 0.
      IYMI = 0
      IEYMI = 0
      IF (YMAX.GT.0.) THEN
        IF (ABS(1.DO-ZYMA*IYMA/YMAX).GT.1.E-4) IYMA=IYMA+1
        ZYMA = IYMA*ZYMA
      ELSE
        IF (YMAX.EQ.0.) THEN
          ZYMA = IYMA
        ELSE
          ZYMA = *(IYMA)*ZYMA
        ENDIF
      ENDIF
      IF (ABS(ZYMA).LT.1.E-4) IEYMA = 0
ENDIF
***** X-ACHSEN-MINIMUM UND -MAXIMUM KORRIGIEREN
      IF (MOD(MB,8).GE.3.AND.MOD(MB,8).LE.6) THEN
        ZXIB = 10.
      ELSE
        ZXIB = 1.
      ENDIF
      IEXMI = 0
      IEXMA = 0
      IF (ABS(XMIN).GT.1.E-4) IEXMI = LOG10(ABS(XMIN))
      IF (ABS(XMAX).GT.1.E-4) IEXMA = LOG10(ABS(XMAX))
      IF (ZXIB.NE.1.) THEN
        IXMI = ABS(XMIN)/10.DO**IEXMI
        IF (ABS(XMIN).GT.0.DO.AND.ABS(IXMI*10.DO**IEXMI
+          -ABS(XMIN)).GT.1.E-4*ABS(XMIN)) IXMI=IXMI+1
        IF (XMIN.LT.0.) IXMI=-IXMI
        ZXMI = IXMI
        IF (IEXMI.LE.2) ZXMI=ZXMI*10.**IEXMI
        IXMA = ABS(XMAX)/10.DO**IEXMA
        IF (ABS(XMAX).GT.0.DO.AND.ABS(IXMA*10.DO**IEXMA
+          -ABS(XMAX)).GT.1.E-4*ABS(XMAX)) IXMA=IXMA+1
        IF (XMAX.LT.0.) IXMA=-IXMA
        ZXMA = IXMA
        IF (IEXMA.LE.2) THEN
```

```
ZXMA=ZXMA*10.**IEXMA
ELSE
  ZXMA = 10.DO**(IEXMA-IEXMI)
ENDIF
ELSE
  IXMI = ABS(XMIN)
  IF (XMIN.LT.O.) IXMI=-(1+IXMI)
  ZXMI = IXMI
  IXMA = ABS(XMAX)
  IF (XMAX.LT.O.) IXMA=-IXMA
  IF (XMAX.GT.O..AND.
+   ABS(XMAX-IXMA).GT.1.E-4) IXMA=1+IXMA
  ZXMA = IXMA
ENDIF
IF (IPR.GE.O) THEN
WRITE(IWR,'(/11X,'XMIN, XMAX =',1P2E13.5)') XMIN,XMAX
WRITE(IWR,'( 11X,'ZXMI, ZXMA =',2F11.4)') ZXMI,ZXMA
WRITE(IWR,'( 11X,'IXMI, IXMA =',2I6)') IXMI,IXMA
WRITE(IWR,'( 11X,'EXMI, EXMA =',2I6)') IEXMI,IEXMA
WRITE(IWR,'(/11X,'YMIN, YMAX =',1P2E13.5)') YMIN,YMAX
WRITE(IWR,'( 11X,'ZYMI, ZYMA =',2F11.4)') ZYMI,ZYMA
WRITE(IWR,'( 11X,'IYMI, IYMA =',2I6)') IYMI,IYMA
WRITE(IWR,'( 11X,'EYMI, EYMA =',2I6)') IEYMI,IEYMA
ENDIF
***** BESCHRIFTUNGSEINHEIT UND ABSTAND *****
XMAX = ZXMA
LXBE = -1 + INT(0.04*(ZXMA-ZXMI))
CALL ZXYIBE
+ (ZXI,ZIFHO,SNGL(XMAX),ZXMI,ZXMA,ZXIB,ZDXI,LXBE)
XMAX = ZXMA
XMIN = ZXMI
CALL ZXYMAS
+ (ZIFHO,ZXMI,ZXMA,ZXIB,ZDXI,ZXP,LXP,LXB,LXL,LXBE)
YMAX = ZYMA
LYBE = 0
CALL ZXYIBE
+ (ZYI,ZIFHO,SNGL(YMAX),ZYMI,ZYMA,ZYIB,ZDYI,LYBE)
YMAX = ZYMA
YMIN = ZYMI
CALL ZXYMAS
```



```
+      (ZIFHO,ZYMI,ZYMA,ZYIB,ZDYI,ZYP,LYP,LYB,LYL,LYBE)
***** ACHSEN ZEICHNEN, SRICHELN UND BEZIFFERN *****
WRITE(IWR,'(/11X,'ZXMI='',F11.4,6X,'ZXMA='',
+      F11.4)') ZXMI,ZXMA
WRITE(IWR,'(11X,'ZXIB='',F11.4,6X,'ZDXI='',
+      F11.4)') ZXIB,ZDXI
WRITE(IWR,'(11X,'LXL='',I6,11X,'ZXP='',
+      F11.4)') LXL,ZXP
WRITE(IWR,'(11X,'LXB='',I6,11X,'LXP='',I6)') LXB,LXP
WRITE(IWR,'(/11X,'ZYMI='',F11.4,6X,'ZYMA='',
+      F11.4)') ZYMI,ZYMA
WRITE(IWR,'(11X,'ZYIB='',F11.4,6X,'ZDYI='',
+      F11.4)') ZYIB,ZDYI
WRITE(IWR,'(11X,'LYL='',I6,11X,'ZYP='',
+      F11.4)') LYL,ZYP
WRITE(IWR,'(11X,'LYB='',I6,11X,'LYP='',I6)') LYB,LYP
ZSTHO = ZIFHO*0.5
IF (LBILD.EQ.1) THEN
  CALL PLOTS(0.,0.,0.)
  CALL NEWPEN(3)
  ENDIF
CALL PLOT(ZXS,ZYS,-3)
CALL ZXYAXN
+(LXL,LYL,ZXI,ZYI,ZXMI,ZYMI,ZXIB,ZYIB,ZDXI,ZDYI,ZSTHO)
CALL ZXZIFF(ZIFHO,ZSTHO,ZXMI,ZXIB,ZDXI,ZXP,LXP,LXB,LXL)
CALL ZYZIFF(ZIFHO,ZSTHO,ZYMI,ZYIB,ZDYI,ZYP,LYP,LYB,LYL)
CALL SERBXY(J1,NKURV,KK,NST,
+ZXI,ZYI,ZEIHO,ZIFHO,WSTREU,NPHAS)
***** POTENZ DER Y-ACHSE UND REAKTION *****
ZYY = ZYI + (ZIFHO+ZEIHO) * (NKURV+1)
CALL SYMBLO(0.,ZYY,ZIFHO,CREAKT,0.,40)
IF (.NOT.(IEYMI.EQ.0.AND.IEYMA.EQ.0)
+ .AND.IEYMI.LT.IEYMA) THEN
  IF (IEYMI.EQ.0) IEYMI = IEYMA
  LYA = LOG10(ABS(ZYMA)) + LYB + 4
  IF (ZYMA.LT.0.) LYA=LYA+1
  ZAHL = IEYMI
  IPZ = 1+LOG10(ABS(ZAHL))
  IF (YMIN.LT.0.) ZAHL=-ZAHL
  IF (YMIN.LT.0.) IPZ=IPZ+1
```

```
ZXX = -ZIFHO*LYA
ZYY = (ZYI-0.5*(8.+IPZ)*ZIFHO)*0.5
CALL SYMBLO(ZXX,ZYY,ZIFHO,'* 10',90.,4)
ZXX = ZXX-ZIFHO*2./3.
ZYY = ZYY+ZIFHO*4.
CALL NUMBER(ZXX,ZYY,ZIFHO*0.6,ZAHL,90.,-1)
ENDIF
```

\*\*\*\*\* PUNKTE AUS EXTERNEN DATENTABELLEN EINTRAGEN \*\*\*\*\*

```
IB = 3*(I1-1)+J1
IA = NEXT(0,IB)
IF (NEXTR.GT.0.AND.IA.GT.0) THEN
DO 310 I=1,IA
WRITE(IWR,'(//11X,'EXTERNE DATEN EINTRAGEN.',
+ ' 'TABELLE NUMMER',I3,'.')') NEXT(I,IB)
ZSYHO = ZSY(4-MOD(I,4))
NS = NSY(4-MOD(I,4))
MP = XEXT(0,2*NEXT(I,IB)-1)
```

```
DO 310 M=1,MP
ZXX = (XEXT(M,2*I-1) - ZXMI) * (ZDXI/ZXIB)
ZYY = (YEXT(M,2*I-1) - ZYMI) * (ZDYI/ZYIB)
```

C - - - - FEHLERBALKEN EINZEICHNEN.

```
ZXD = XEXT(M,2*I) * (ZDXI/ZXIB)
ZYD = YEXT(M,2*I) * (ZDYI/ZYIB)
CALL PLOT(ZXX-ZXD,ZYY,3)
CALL PLOT(ZXX+ZXD,ZYY,2)
CALL PLOT(ZXX,ZYY-ZYD,3)
CALL PLOT(ZXX,ZYY+ZYD,2)
```

C - - - - GEBEBENENFALLS PLOTSYMBOL EINTRAGEN.

```
IF (NS.GE.0) THEN
CALL PLSYMB(ZXX,ZYY,ZSYHO,NS)
ELSE
ZXX=ZXX-ZSYHO*(2./7.)
ZYY=ZYY-ZSYHO*0.5
CALL SYMBLO(ZXX,ZYY,ZSYHO,CKUSYM(-NS:-NS),0.,1)
ENDIF
```

310 CONTINUE

ENDIF

\*\*\*\*\* PHASENRAUMFAKTOREN BEI BEDARF ANPASSEN

```
IF (J1.NE.2.AND.NPHAS.GT.0) THEN
WRITE(IWR,'(//11X,'PHASENRAUMFAKTOREN PLOTTEN'))'
```

```
DO 321 IK=1,NKURV
  MB = LTAB(IK)
  MP = YARR(O,MB)
  K = KK(IK)
  WRITE(IWR,'(18X,'K-WERT =',I2)') K
  YPHA(O,K) = YPHA(O,K) * 0.5DO * 10.DO**(-IEYMI) *
+      YARR(MP+1,MB) / YPHA(MP+1,K)
  DO 320 M=1,MP
    ZX(M) = (XARR(M,MB)-ZXMI)*(ZDXI/ZXIB)
    ZY(M) = (YPHA(M,K)*YPHA(O,K)-ZYMI)*(ZDYI/ZYIB)
320  CONTINUE
  NT = NST(NKURV+IK)
  IF (IABS(NT).GT.5.OR.NT.EQ.0) GOTO 321
  ZABST = ZST(NKURV+IK)
  IF (IABS(NT).EQ.1) THEN
C - - - - - KURVENZUG DURCHGEZOGEN MALEN.
    ZX(MP+1) = 0.
    ZY(MP+1) = 0.
    ZX(MP+2) = 1.
    ZY(MP+2) = 1.
    CALL LINE(ZX,ZY,MP,1,0,0)
  ELSE
C - - - - - KURVENZUG GESTRICHELT ETC. MALEN.
    ML = NJ
    CALL AEQUID(MP,ML,ZABST,ZX,ZY,ZV,ZW)
    CALL STRIPU(IABS(NT),ML,ZV,ZW)
  ENDIF
321  CONTINUE
  ENDIF
***** KURVEN EINTRAGEN *****
  WRITE(IWR,'(//11X,'KURVEN EINZEICHNEN')')
  DO 339 IK=1,NKURV
    MB = LTAB(IK)
    MP = YARR(O,MB)
    WRITE(IWR,'(18X,'NUMMER =',I3,5X,
+      'TABELLE =',I4)') IK,MB
    IF (MP.LE.0) GOTO 339
    NS = NSY(IK)
    IF (NS.GT.13.OR.NS.LE.-40) NS=-40
```

```
ZSYHO = ZSY(IK)
C - - - - GEGBENENFALLS PLOTSYMBOLS EINTRAGEN.
DO 330 M=1,MP
  YARR(M,MB) = YARR(M,MB) * 10.DO**(-IEYMI)
  ZX(M) = (XARR(M,MB)-ZXMI)*(ZDXI/ZXIB)
  ZY(M) = (YARR(M,MB)-ZYMI)*(ZDYI/ZYIB)
  IF (NS.LE.-40) GOTO 330
  ZXX = ZX(M)
  ZYY = ZY(M)
  IF (NS.GE.0) THEN
    CALL PLSYMB(ZXX,ZYY,ZSYHO,NS)
  ELSE
    ZXX=ZXX-ZSYHO*(2./7.)
    ZYY=ZYY-ZSYHO*0.5
    CALL SYMBLO(ZXX,ZYY,ZSYHO,CKUSYM(-NS:-NS),0.,1)
  ENDIF
330 CONTINUE
C - - - - KURVEN-LINIE EINTRAGEN.
NT = NST(IK)
IF (IABS(NT).GT.5.OR.NT.EQ.0) GOTO 339
ZABST = ZST(IK)
IF (NT.GT.0) THEN
C - - - - - SPLINE-INTERPOLATION FUER MEHR PUNKTE.
MQ = NJ
CALL SPLINE(MP,MQ,ZX,ZY,ZV,ZW)
ELSE
C - - - - - AUSGLEICHSSPLINE UND NEUE ZWISCHENPUNKTE.
IA = 4
CALL AUSGL4(IA,MP,ZX,ZY,ZY,CO)
IF (IPR.GE.5) WRITE(IWR,'(/9(25X,'CO('',
+ I1,'') = '' ,1PE20.12/))') (I,CO(I),I=0,IA)
MD = (NJ-2) / MP
IF (MD.LE.0) MD = 1
IF (MD.GT.1) THEN
  MQ = 1
  ZV(MQ) = ZX(1)
  ZW(MQ) = ZY(1)
  DO 332 K=2,MP
    TQ = (ZX(K)-ZX(K-1)) / MD
  DO 332 M=1,MD
```

```

TQC = ZX(K-1) + TQ*M
TQS = CO(IA)
DO 331 I=IA-1,0,-1
    TQS = TQS * TQC + CO(I)
331    CONTINUE
    MQ = MQ + 1
    ZV(MQ) = TQC
    ZW(MQ) = TQS
332    CONTINUE
    MQ = MQ + 1
    ZV(MQ) = ZX(MP)
    ZW(MQ) = ZY(MP)
    ELSE
    MQ = MP
    DO 333 K=1,MQ
        ZV(K) = ZX(K)
        ZW(K) = ZY(K)
333    CONTINUE
    ENDIF
ENDIF
C - - - - - EVENTUELL ABSCHNEIDEKORREKTUR.
IF (MQ.GT.MP) THEN
    ZXXALT = -1.
    ZYYALT = 0.
    IF (ZV(1).LT.0.0) ZYYALT = -1.
    IF (ZW(1).GT.ZYI) ZYYALT = ZYI + 1.
    MA = 0
    DO 338 I=1,MQ
        ZXX = ZV(I)
        ZYY = ZW(I)
        ZXXE = ZXX
        ZYYE = ZYY
        IF (MA.LT.NJ-4.AND.(ZXX.GE.0..AND.
+           ZXX.LE.ZXI*1.0001)) CALL ZABSCH
+           (MA,ZYI,ZXX,ZYY,ZXXALT,ZYYALT,ZX,ZY)
        ZXXALT = ZXXE
        ZYYALT = ZYYE
338    CONTINUE
    MP = MA
ENDIF
```

```
      IF (IABS(NT).EQ.1) THEN
C - - - - - KURVENZUG DURCHGEZOGEN MALEN.
      ZX(MP+1) = 0.
      ZY(MP+1) = 0.
      ZX(MP+2) = 1.
      ZY(MP+2) = 1.
      CALL LINE(ZX,ZY,MP,1,0,0)
      ELSE
C - - - - - KURVENZUG GESTRICHELT ETC. MALEN.
      ML = NJ
      CALL AEQUID(MP,ML,ZABST,ZX,ZY,ZV,ZW)
      CALL STRIPU(IABS(NT),ML,ZV,ZW)
      ENDIF
339 CONTINUE
***** WENN GEPLOTTET WURDE, URSPRUNG VERSCHIEBEN *****
      IF (LBILD.GT.0) THEN
      CALL PLOT(ZXS*0.5+ZXI,-ZYS,-3)
      ZXA = ZXA + ZXS*1.5 + ZXI
      ENDIF
391 CONTINUE
392 CONTINUE
***** PROGRAMMENDE *****
399 CONTINUE
***** WENN GEPLOTTET WURDE, PLOTTEN ABSCHLIESSEN *****
      WRITE(IWR,'(//6X,44(''*'))/6X, '*'
+ 'E N D E *'/6X,44(''*'))')
      IF (LBILD.GT.0) THEN
      IF (IPL.GT.0) CALL PLOT(0.,0.,999)
      WRITE(IWR,'(//11X, 'BENOETIGTE PLOTPAPIERLAENGE = ',
+ F10.2)') ZXA
      ENDIF
      WRITE(IWR,'(//11X, 'PROGRAMMENDE SERBER'//)')
      STOP
      END
```

## 4. BESCHREIBUNG DER UNTERPROGRAMME

### 4.1 AEQUID

#### Kurvenpunkte mit gleichem Abstand

Das Unterprogramm AEQUID berechnet aus einer beliebigen Kurve, gegeben durch ZX(i) und ZY(i) in Inches, andere Stützpunkte derart, daß je zwei benachbarte neue Punkte denselben, vorgegebenen Abstand von einander haben. Die erzeugten Punkte können dann dazu verwendet werden, die Kurve zu stricheln, zu punktieren, zu strichpunktieren usw.

```
NX = NPXY
NL = NVWMAX
CALL AEQUID(NX,NL,ZABST,ZX,ZY,ZV,ZW)
```

#### EINGABEPARAMETER:

NX = Anzahl der Eingabepunkte, d.h. die Anzahl der benutzten Werte aus den Feldern ZX(i) und ZY(i).  
NL = Vereinbarte Dimension der Felder ZV(i) und ZW(i) aus dem rufenden Programmsegment.  
ZABST = Gewünschter Abstand zweier benachbarter neuer Punkte (in Inches).  
ZX(1:NX) = Abszissen der Eingabepunkte (in Inches).  
ZY(1:NX) = Ordinaten der Eingabepunkte (in Inches).

#### RÜCKGABEPARAMETER:

NL = Tatsächliche Anzahl neuer Punkte.  
ZV(1:NL) = Abszissen der neuen Punkte (in Inches).  
ZW(1:NL) = Ordinaten der neuen Punkte (in Inches).

RESTLICHE VARIABLEN:

I = Hilfs-, Schleifen-, Zähl-Variable.  
NM = Vereinbarte Dimension der Felder ZV(i) und ZW(i)  
aus dem rufenden Programmsegment.  
SQRT = Fortran builtin function.  
ZFAKT = Aktueller Korrekturfaktor der Koordinaten.  
ZIL = Zuwachs des Abstands bei Hinzunahme des nächsten Punktes.  
ZXDEL = Aktueller Abszissenabstand zweier Eingabepunkte.  
ZYDEL = Aktueller Ordinatenabstand zweier Eingabepunkte.

ABDRUCK DES UNTERPROGRAMMS AEQUID

C - - - - Koordinaten für Kurvenpunkte mit gleichem Abstand

```
SUBROUTINE AEQUID(NX,NL,ZABST,ZX,ZY,ZV,ZW)
DIMENSION ZX(NX),ZY(NX),ZV(NX),ZW(NX)
NM=NL
NL=1
I=1
ZV(NL)=ZX(I)
ZW(NL)=ZY(I)
ZLEN=0.
1 CONTINUE
I=I+1
IF (I.GT.NX) GOTO 3
ZXDEL=ZX(I)-ZX(I-1)
ZYDEL=ZY(I)-ZY(I-1)
ZIL=SQRT(1.E-4+ZXDEL*ZXDEL+ZYDEL*ZYDEL)
IF (ZLEN+ZIL.LE.ZABST) THEN
    ZLEN=ZLEN+ZIL
    GOTO 1
ENDIF
NL=NL+1
IF (NL.GT.NM-2) GOTO 3
ZFAKT=(ZABST-ZLEN)/ZIL
ZV(NL)=ZX(I-1)+ZFAKT*ZXDEL
ZW(NL)=ZY(I-1)+ZFAKT*ZYDEL
ZLEN=ZLEN+ZIL-ZABST
2 IF (ZLEN.LE.ZABST) GOTO 1
ZFAKT=ZABST/ZLEN
```



```
NL=NL+1
IF (NL.GT.NM-2) GOTO 3
ZV(NL)=ZV(NL-1)+ZFAKT*(ZX(I)-ZV(NL-1))
ZW(NL)=ZW(NL-1)+ZFAKT*(ZY(I)-ZW(NL-1))
ZLEN=ZLEN-ZABST
GOTO 2
3 CONTINUE
RETURN
END
```

## 4.2 AUSGL4

### Ausgleichspolynom vom Grad 4

Das Unterprogramm AUSGL4 berechnet zu einem Satz von Ausgangspunkten nach folgender Formel  $CO = A^{*(-1)} * SYX$  oder

$$\begin{pmatrix} CO0 \\ CO1 \\ CO2 \\ CO4 \\ CO4 \end{pmatrix} = \begin{pmatrix} SX0 & SX1 & SX2 & SX3 & SX4 \\ SX1 & SX2 & SX3 & SX4 & SX5 \\ SX2 & SX3 & SX4 & SX5 & SX6 \\ SX3 & SX4 & SX5 & SX6 & SX7 \\ SX4 & SX5 & SX6 & SX7 & SX8 \end{pmatrix}^{-1} * \begin{pmatrix} SYX0 \\ SYX1 \\ SYX2 \\ SYX3 \\ SYX4 \end{pmatrix}$$

die Koeffizienten des Ausgleichspolynoms vom Grad 4. Im rufenden Programmsegment können dann neue Stützpunkte berechnet werden.

```
CALL AUSGL4(KP, NP, ZX, ZY, ZF, CO)
```

#### EINGABEPARAMETER:

KP = Grad des Ausgleichspolynoms.

NP = Anzahl der Ausgangspunkte.

ZX(1:NP), ZY(1:NP) = X- bzw. Y-Koordinaten der Ausgangspunkte.

ZF(1:NP) = Y-Fehler der Ausgangspunkte (hier unberücksichtigt).

#### RÜCKGABEPARAMETER:

CO(0:KP) = Koeffizienten des Ausgleichspolynoms.

#### RESTLICHE VARIABLEN:

ANS1, ANS2, ANS3, XX = Zwischenvariable.

DETA = Determinante der Berechnungsmatrix.

I = Schleifen- und Zaehlvariable.

SXj = Aktuelle Summe der  $ZX(i)**(j)$  , j=0(1)8.

SYXj = Aktuelle Summe der  $ZY(i) * ZX(i)**(j)$  , j=0(1)4.

Xj = Aktueller Ausdruck  $ZX(i)**(j)$  , j=0(1)8.

YXj = Aktuelles Produkt  $ZY(i) * ZX(i)**(j)$  , j=0(1)4.

### ABDRUCK DES UNTERPROGRAMMS AUSGL4

C - - - - - Ausgleichspolynom vom Grad 4 aus 5 Punkten.

C        YY=(( (CO(4)\*XX+CO(3))\*XX+CO(2))\*XX+CO(1))\*XX+CO(0)

      SUBROUTINE AUSGL4(KP,NP,ZX,ZY,ZF,CO)

      IMPLICIT REAL\*8 (A-H,O-Y)

      DIMENSION ZX(NP),ZY(NP),ZF(NP),CO(0:KP)

      SYX0=0.DO

      SYX1=0.DO

      SYX2=0.DO

      SYX3=0.DO

      SYX4=0.DO

      DO 12 I=1,NP

          XX = ZX(I)

          YX0 = ZY(I)

          YX1 = YX0 \* XX

          YX2 = YX1 \* XX

          YX3 = YX2 \* XX

          YX4 = YX3 \* XX

          SYX0 = SYX0 + YX0

          SYX1 = SYX1 + YX1

          SYX2 = SYX2 + YX2

          SYX3 = SYX3 + YX3

          SYX4 = SYX4 + YX4

12 CONTINUE

      SX0=0.DO

      SX1=0.DO

      SX2=0.DO

      SX3=0.DO

      SX4=0.DO

      SX5=0.DO

      SX6=0.DO

      SX7=0.DO

      SX8=0.DO

      DO 13 I=1,NP

          XX = ZX(I)

          X0 = 1.DO

          X1 = X0 \* XX

          X2 = X1 \* XX

          X3 = X2 \* XX

X4 = X3 \* XX  
X5 = X4 \* XX  
X6 = X5 \* XX  
X7 = X6 \* XX  
X8 = X7 \* XX  
SX0 = SX0 + X0  
SX1 = SX1 + X1  
SX2 = SX2 + X2  
SX3 = SX3 + X3  
SX4 = SX4 + X4  
SX5 = SX5 + X5  
SX6 = SX6 + X6  
SX7 = SX7 + X7  
SX8 = SX8 + X8

13 CONTINUE

C - - - - DETERMINANTE.

ANS1=-3.\*SX6\*SX5\*\*2\*SX4\*SX0+2.\*SX6\*SX5\*\*2\*SX3\*SX1+SX6\*SX5  
. \*\*2\*SX2\*\*2+4.\*SX6\*SX5\*SX4\*\*2\*SX1-2.\*SX6\*SX5\*SX4\*SX3\*SX2-  
. 2.\*SX6\*SX5\*SX3\*\*3-3.\*SX6\*SX4\*\*3\*SX2+3.\*SX6\*SX4\*\*2\*SX3\*\*2+  
. SX5\*\*4\*SX0-2.\*SX5\*\*3\*SX4\*SX1-2.\*SX5\*\*3\*SX3\*SX2+3.\*SX5\*\*2\*  
. SX4\*\*2\*SX2+3.\*SX5\*\*2\*SX4\*SX3\*\*2-4.\*SX5\*SX4\*\*3\*SX3+SX4\*\*5  
DETA=SX8\*SX6\*SX4\*SX2\*SX0-SX8\*SX6\*SX4\*SX1\*\*2-SX8\*SX6\*SX3\*\*2  
. \*SX0+2.\*SX8\*SX6\*SX3\*SX2\*SX1-SX8\*SX6\*SX2\*\*3-SX8\*SX5\*\*2\*SX2  
. \*SX0+SX8\*SX5\*\*2\*SX1\*\*2+2.\*SX8\*SX5\*SX4\*SX3\*SX0-2.\*SX8\*SX5\*  
. SX4\*SX2\*SX1-2.\*SX8\*SX5\*SX3\*\*2\*SX1+2.\*SX8\*SX5\*SX3\*SX2\*\*2-  
. SX8\*SX4\*\*3\*SX0+2.\*SX8\*SX4\*\*2\*SX3\*SX1+SX8\*SX4\*\*2\*SX2\*\*2-3.  
. \*SX8\*SX4\*SX3\*\*2\*SX2+SX8\*SX3\*\*4-SX7\*\*2\*SX4\*SX2\*SX0+SX7\*\*2\*  
. SX4\*SX1\*\*2+SX7\*\*2\*SX3\*\*2\*SX0-2.\*SX7\*\*2\*SX3\*SX2\*SX1+SX7\*\*2  
. \*SX2\*\*3+2.\*SX7\*SX6\*SX5\*SX2\*SX0-2.\*SX7\*SX6\*SX5\*SX1\*\*2-2.\*  
. SX7\*SX6\*SX4\*SX3\*SX0+2.\*SX7\*SX6\*SX4\*SX2\*SX1+2.\*SX7\*SX6\*SX3  
. \*\*2\*SX1-2.\*SX7\*SX6\*SX3\*SX2\*\*2-2.\*SX7\*SX5\*\*2\*SX3\*SX0+2.\*  
. SX7\*SX5\*\*2\*SX2\*SX1+2.\*SX7\*SX5\*SX4\*\*2\*SX0-4.\*SX7\*SX5\*SX4\*  
. SX2\*\*2+2.\*SX7\*SX5\*SX3\*\*2\*SX2-2.\*SX7\*SX4\*\*3\*SX1+4.\*SX7\*SX4  
. \*\*2\*SX3\*SX2-2.\*SX7\*SX4\*SX3\*\*3-SX6\*\*3\*SX2\*SX0+SX6\*\*3\*SX1\*\*  
. 2+2.\*SX6\*\*2\*SX5\*SX3\*SX0-2.\*SX6\*\*2\*SX5\*SX2\*SX1+SX6\*\*2\*SX4  
. \*\*2\*SX0-4.\*SX6\*\*2\*SX4\*SX3\*SX1+2.\*SX6\*\*2\*SX4\*SX2\*\*2+SX6\*\*2  
. \*SX3\*\*2\*SX2+ANS1

C - - - - KOEFFIZIENT(0).

ANS2=-SYX2\*SX7\*SX6\*SX3\*SX2+SYX2\*SX7\*SX5\*\*2\*SX1-3.\*SYX2\*SX7  
. \*SX5\*SX4\*SX2+SYX2\*SX7\*SX5\*SX3\*\*2+SYX2\*SX7\*SX4\*\*2\*SX3-SYX2

. \*SX6\*\*2\*SX5\*SX1+SYX2\*SX6\*\*2\*SX4\*SX2+SYX2\*SX6\*SX5\*\*2\*SX2-  
. SYX2\*SX6\*SX4\*\*3-SYX2\*SX5\*\*3\*SX3+SYX2\*SX5\*\*2\*SX4\*\*2-SYX1\*  
. SX8\*SX6\*SX4\*SX1+SYX1\*SX8\*SX6\*SX3\*SX2+SYX1\*SX8\*SX5\*\*2\*SX1-  
. SYX1\*SX8\*SX5\*SX4\*SX2-SYX1\*SX8\*SX5\*SX3\*\*2+SYX1\*SX8\*SX4\*\*2\*  
. SX3+SYX1\*SX7\*\*2\*SX4\*SX1-SYX1\*SX7\*\*2\*SX3\*SX2-2.\*SYX1\*SX7\*  
. SX6\*SX5\*SX1+SYX1\*SX7\*SX6\*SX4\*SX2+SYX1\*SX7\*SX6\*SX3\*\*2+SYX1  
. \*SX7\*SX5\*\*2\*SX2-SYX1\*SX7\*SX4\*\*3+SYX1\*SX6\*\*3\*SX1-SYX1\*SX6  
. \*\*2\*SX5\*SX2-2.\*SYX1\*SX6\*\*2\*SX4\*SX3+SYX1\*SX6\*SX5\*\*2\*SX3+2.  
. \*SYX1\*SX6\*SX5\*SX4\*\*2-SYX1\*SX5\*\*3\*SX4+SYX0\*SX8\*SX6\*SX4\*SX2  
. -SYX0\*SX8\*SX6\*SX3\*\*2-SYX0\*SX8\*SX5\*\*2\*SX2+2.\*SYX0\*SX8\*SX5\*  
. SX4\*SX3-SYX0\*SX8\*SX4\*\*3-SYX0\*SX7\*\*2\*SX4\*SX2+SYX0\*SX7\*\*2\*  
. SX3\*\*2+2.\*SYX0\*SX7\*SX6\*SX5\*SX2-2.\*SYX0\*SX7\*SX6\*SX4\*SX3-2.  
. \*SYX0\*SX7\*SX5\*\*2\*SX3+2.\*SYX0\*SX7\*SX5\*SX4\*\*2-SYX0\*SX6\*\*3\*  
. SX2+2.\*SYX0\*SX6\*\*2\*SX5\*SX3+SYX0\*SX6\*\*2\*SX4\*\*2-3.\*SYX0\*SX6  
. \*SX5\*\*2\*SX4+SYX0\*SX5\*\*4

ANS1=SYX4\*SX7\*SX5\*SX3\*SX1-SYX4\*SX7\*SX5\*SX2\*\*2-SYX4\*SX7\*SX4  
. \*\*2\*SX1+2.\*SYX4\*SX7\*SX4\*SX3\*SX2-SYX4\*SX7\*SX3\*\*3-SYX4\*SX6  
. \*\*2\*SX3\*SX1+SYX4\*SX6\*\*2\*SX2\*\*2+2.\*SYX4\*SX6\*SX5\*SX4\*SX1-2.  
. \*SYX4\*SX6\*SX5\*SX3\*SX2-2.\*SYX4\*SX6\*SX4\*\*2\*SX2+2.\*SYX4\*SX6\*  
. SX4\*SX3\*\*2-SYX4\*SX5\*\*3\*SX1+2.\*SYX4\*SX5\*\*2\*SX4\*SX2+SYX4\*  
. SX5\*\*2\*SX3\*\*2-3.\*SYX4\*SX5\*SX4\*\*2\*SX3+SYX4\*SX4\*\*4-SYX3\*SX8  
. \*SX5\*SX3\*SX1+SYX3\*SX8\*SX5\*SX2\*\*2+SYX3\*SX8\*SX4\*\*2\*SX1-2.\*  
. SYX3\*SX8\*SX4\*SX3\*SX2+SYX3\*SX8\*SX3\*\*3+SYX3\*SX7\*SX6\*SX3\*SX1  
. -SYX3\*SX7\*SX6\*SX2\*\*2-SYX3\*SX7\*SX5\*SX4\*SX1+SYX3\*SX7\*SX5\*  
. SX3\*SX2+SYX3\*SX7\*SX4\*\*2\*SX2-SYX3\*SX7\*SX4\*SX3\*\*2-SYX3\*SX6  
. \*\*2\*SX4\*SX1+SYX3\*SX6\*\*2\*SX3\*SX2+SYX3\*SX6\*SX5\*\*2\*SX1-2.\*  
. SYX3\*SX6\*SX5\*SX3\*\*2+SYX3\*SX6\*SX4\*\*2\*SX3-SYX3\*SX5\*\*3\*SX2+  
. 2.\*SYX3\*SX5\*\*2\*SX4\*SX3-SYX3\*SX5\*SX4\*\*3+SYX2\*SX8\*SX6\*SX3\*  
. SX1-SYX2\*SX8\*SX6\*SX2\*\*2-SYX2\*SX8\*SX5\*SX4\*SX1+SYX2\*SX8\*SX5  
. \*SX3\*SX2+SYX2\*SX8\*SX4\*\*2\*SX2-SYX2\*SX8\*SX4\*SX3\*\*2-SYX2\*SX7  
. \*\*2\*SX3\*SX1+SYX2\*SX7\*\*2\*SX2\*\*2+SYX2\*SX7\*SX6\*SX4\*SX1+ANS2

CO(0)=ANS1/DETA

C - - - - KOEFFIZIENT(1).

ANS3=SYX0\*SX7\*SX6\*SX4\*SX2+SYX0\*SX7\*SX6\*SX3\*\*2+SYX0\*SX7\*SX5  
. \*\*2\*SX2-SYX0\*SX7\*SX4\*\*3+SYX0\*SX6\*\*3\*SX1-SYX0\*SX6\*\*2\*SX5\*  
. SX2-2.\*SYX0\*SX6\*\*2\*SX4\*SX3+SYX0\*SX6\*SX5\*\*2\*SX3+2.\*SYX0\*  
. SX6\*SX5\*SX4\*\*2-SYX0\*SX5\*\*3\*SX4

ANS2=-SYX2\*SX8\*SX4\*SX3\*SX2+SYX2\*SX8\*SX3\*\*3+SYX2\*SX7\*\*2\*SX3  
. \*SX0-SYX2\*SX7\*\*2\*SX2\*SX1-SYX2\*SX7\*SX6\*SX4\*SX0+SYX2\*SX7\*  
. SX6\*SX3\*SX1-SYX2\*SX7\*SX5\*\*2\*SX0+SYX2\*SX7\*SX5\*SX4\*SX1+SYX2

. \*SX7\*SX5\*SX3\*SX2+SYX2\*SX7\*SX4\*\*2\*SX2-2.\*SYX2\*SX7\*SX4\*SX3  
. \*\*2+SYX2\*SX6\*\*2\*SX5\*SX0-SYX2\*SX6\*\*2\*SX4\*SX1-SYX2\*SX6\*SX5\*  
. SX4\*SX2-SYX2\*SX6\*SX5\*SX3\*\*2+2.\*SYX2\*SX6\*SX4\*\*2\*SX3+SYX2\*  
. SX5\*\*2\*SX4\*SX3-SYX2\*SX5\*SX4\*\*3+SYX1\*SX8\*SX6\*SX4\*SX0-SYX1\*  
. SX8\*SX6\*SX2\*\*2-SYX1\*SX8\*SX5\*\*2\*SX0+2.\*SYX1\*SX8\*SX5\*SX3\*  
. SX2-SYX1\*SX8\*SX4\*SX3\*\*2-SYX1\*SX7\*\*2\*SX4\*SX0+SYX1\*SX7\*\*2\*  
. SX2\*\*2+2.\*SYX1\*SX7\*SX6\*SX5\*SX0-2.\*SYX1\*SX7\*SX6\*SX3\*SX2-2.  
. \*SYX1\*SX7\*SX5\*SX4\*SX2+2.\*SYX1\*SX7\*SX4\*\*2\*SX3-SYX1\*SX6\*\*3\*  
. SX0+2.\*SYX1\*SX6\*\*2\*SX4\*SX2+SYX1\*SX6\*\*2\*SX3\*\*2-2.\*SYX1\*SX6  
. \*SX5\*SX4\*SX3-SYX1\*SX6\*SX4\*\*3+SYX1\*SX5\*\*2\*SX4\*\*2-SYX0\*SX8\*  
. SX6\*SX4\*SX1+SYX0\*SX8\*SX6\*SX3\*SX2+SYX0\*SX8\*SX5\*\*2\*SX1-SYX0  
. \*SX8\*SX5\*SX4\*SX2-SYX0\*SX8\*SX5\*SX3\*\*2+SYX0\*SX8\*SX4\*\*2\*SX3+  
. SYX0\*SX7\*\*2\*SX4\*SX1-SYX0\*SX7\*\*2\*SX3\*SX2-2.\*SYX0\*SX7\*SX6\*  
. SX5\*SX1+ANS3

ANS1=-SYX4\*SX7\*SX5\*SX3\*SX0+SYX4\*SX7\*SX5\*SX2\*SX1+SYX4\*SX7\*  
. SX4\*\*2\*SX0-SYX4\*SX7\*SX4\*SX3\*SX1-SYX4\*SX7\*SX4\*SX2\*\*2+SYX4\*  
. SX7\*SX3\*\*2\*SX2+SYX4\*SX6\*\*2\*SX3\*SX0-SYX4\*SX6\*\*2\*SX2\*SX1-2.  
. \*SYX4\*SX6\*SX5\*SX4\*SX0+SYX4\*SX6\*SX5\*SX3\*SX1+SYX4\*SX6\*SX5\*  
. SX2\*\*2+SYX4\*SX6\*SX4\*\*2\*SX1-SYX4\*SX6\*SX3\*\*3+SYX4\*SX5\*\*3\*  
. SX0-SYX4\*SX5\*\*2\*SX4\*SX1-2.\*SYX4\*SX5\*\*2\*SX3\*SX2+SYX4\*SX5\*  
. SX4\*\*2\*SX2+2.\*SYX4\*SX5\*SX4\*SX3\*\*2-SYX4\*SX4\*\*3\*SX3+SYX3\*  
. SX8\*SX5\*SX3\*SX0-SYX3\*SX8\*SX5\*SX2\*SX1-SYX3\*SX8\*SX4\*\*2\*SX0+  
. SYX3\*SX8\*SX4\*SX3\*SX1+SYX3\*SX8\*SX4\*SX2\*\*2-SYX3\*SX8\*SX3\*\*2\*  
. SX2-SYX3\*SX7\*SX6\*SX3\*SX0+SYX3\*SX7\*SX6\*SX2\*SX1+SYX3\*SX7\*  
. SX5\*SX4\*SX0-SYX3\*SX7\*SX5\*SX2\*\*2-SYX3\*SX7\*SX4\*\*2\*SX1+SYX3\*  
. SX7\*SX4\*SX3\*SX2+SYX3\*SX6\*\*2\*SX4\*SX0-SYX3\*SX6\*\*2\*SX3\*SX1-  
. SYX3\*SX6\*SX5\*\*2\*SX0+SYX3\*SX6\*SX5\*SX4\*SX1+SYX3\*SX6\*SX5\*SX3  
. \*SX2-2.\*SYX3\*SX6\*SX4\*\*2\*SX2+SYX3\*SX6\*SX4\*SX3\*\*2+SYX3\*SX5  
. \*\*2\*SX4\*SX2-2.\*SYX3\*SX5\*SX4\*\*2\*SX3+SYX3\*SX4\*\*4-SYX2\*SX8\*  
. SX6\*SX3\*SX0+SYX2\*SX8\*SX6\*SX2\*SX1+SYX2\*SX8\*SX5\*SX4\*SX0-  
. SYX2\*SX8\*SX5\*SX3\*SX1+ANS2

CO(1)=ANS1/DETA

C - - - - Koeffizient(2).

ANS3=-SYX0\*SX7\*SX6\*SX3\*SX2+SYX0\*SX7\*SX5\*\*2\*SX1-3.\*SYX0\*SX7  
. \*SX5\*SX4\*SX2+SYX0\*SX7\*SX5\*SX3\*\*2+SYX0\*SX7\*SX4\*\*2\*SX3-SYX0  
. \*SX6\*\*2\*SX5\*SX1+SYX0\*SX6\*\*2\*SX4\*SX2+SYX0\*SX6\*SX5\*\*2\*SX2-  
. SYX0\*SX6\*SX4\*\*3-SYX0\*SX5\*\*3\*SX3+SYX0\*SX5\*\*2\*SX4\*\*2  
ANS2=2.\*SYX2\*SX8\*SX4\*SX3\*SX1-SYX2\*SX8\*SX3\*\*2\*SX2-SYX2\*SX7  
. \*\*2\*SX2\*SX0+SYX2\*SX7\*\*2\*SX1\*\*2+2.\*SYX2\*SX7\*SX5\*SX4\*SX0-2.  
. \*SYX2\*SX7\*SX5\*SX3\*SX1-2.\*SYX2\*SX7\*SX4\*\*2\*SX1+2.\*SYX2\*SX7\*

. SX4\*SX3\*SX2-SYX2\*SX6\*SX5\*\*2\*SX0+2.\*SYX2\*SX6\*SX5\*SX4\*SX1-  
. SYX2\*SX6\*SX4\*\*2\*SX2+SYX2\*SX5\*\*2\*SX3\*\*2-2.\*SYX2\*SX5\*SX4\*\*2  
. \*SX3+SYX2\*SX4\*\*4-SYX1\*SX8\*SX6\*SX3\*SX0+SYX1\*SX8\*SX6\*SX2\*  
. SX1+SYX1\*SX8\*SX5\*SX4\*SX0-SYX1\*SX8\*SX5\*SX3\*SX1-SYX1\*SX8\*  
. SX4\*SX3\*SX2+SYX1\*SX8\*SX3\*\*3+SYX1\*SX7\*\*2\*SX3\*SX0-SYX1\*SX7  
. \*\*2\*SX2\*SX1-SYX1\*SX7\*SX6\*SX4\*SX0+SYX1\*SX7\*SX6\*SX3\*SX1-  
. SYX1\*SX7\*SX5\*\*2\*SX0+SYX1\*SX7\*SX5\*SX4\*SX1+SYX1\*SX7\*SX5\*SX3  
. \*SX2+SYX1\*SX7\*SX4\*\*2\*SX2-2.\*SYX1\*SX7\*SX4\*SX3\*\*2+SYX1\*SX6  
. \*\*2\*SX5\*SX0-SYX1\*SX6\*\*2\*SX4\*SX1-SYX1\*SX6\*SX5\*SX4\*SX2-SYX1  
. \*SX6\*SX5\*SX3\*\*2+2.\*SYX1\*SX6\*SX4\*\*2\*SX3+SYX1\*SX5\*\*2\*SX4\*  
. SX3-SYX1\*SX5\*SX4\*\*3+SYX0\*SX8\*SX6\*SX3\*SX1-SYX0\*SX8\*SX6\*SX2  
. \*\*2-SYX0\*SX8\*SX5\*SX4\*SX1+SYX0\*SX8\*SX5\*SX3\*SX2+SYX0\*SX8\*  
. SX4\*\*2\*SX2-SYX0\*SX8\*SX4\*SX3\*\*2-SYX0\*SX7\*\*2\*SX3\*SX1+SYX0\*  
. SX7\*\*2\*SX2\*\*2+SYX0\*SX7\*SX6\*SX4\*SX1+ANS3

ANS1=SYX4\*SX7\*SX5\*SX2\*SX0-SYX4\*SX7\*SX5\*SX1\*\*2-SYX4\*SX7\*SX4  
. \*SX3\*SX0+SYX4\*SX7\*SX4\*SX2\*SX1+SYX4\*SX7\*SX3\*\*2\*SX1-SYX4\*  
. SX7\*SX3\*SX2\*\*2-SYX4\*SX6\*\*2\*SX2\*SX0+SYX4\*SX6\*\*2\*SX1\*\*2+  
. SYX4\*SX6\*SX5\*SX3\*SX0-SYX4\*SX6\*SX5\*SX2\*SX1+SYX4\*SX6\*SX4\*\*2  
. \*SX0-3.\*SYX4\*SX6\*SX4\*SX3\*SX1+SYX4\*SX6\*SX4\*SX2\*\*2+SYX4\*SX6  
. \*SX3\*\*2\*SX2-SYX4\*SX5\*\*2\*SX4\*SX0+SYX4\*SX5\*\*2\*SX3\*SX1+SYX4\*  
. SX5\*SX4\*\*2\*SX1-SYX4\*SX5\*SX3\*\*3-SYX4\*SX4\*\*3\*SX2+SYX4\*SX4\*\*  
. 2\*SX3\*\*2-SYX3\*SX8\*SX5\*SX2\*SX0+SYX3\*SX8\*SX5\*SX1\*\*2+SYX3\*  
. SX8\*SX4\*SX3\*SX0-SYX3\*SX8\*SX4\*SX2\*SX1-SYX3\*SX8\*SX3\*\*2\*SX1+  
. SYX3\*SX8\*SX3\*SX2\*\*2+SYX3\*SX7\*SX6\*SX2\*SX0-SYX3\*SX7\*SX6\*SX1  
. \*\*2-SYX3\*SX7\*SX5\*SX3\*SX0+SYX3\*SX7\*SX5\*SX2\*SX1+SYX3\*SX7\*  
. SX4\*SX3\*SX1-SYX3\*SX7\*SX4\*SX2\*\*2-SYX3\*SX6\*SX5\*SX4\*SX0+SYX3  
. \*SX6\*SX5\*SX3\*SX1+SYX3\*SX6\*SX4\*\*2\*SX1-SYX3\*SX6\*SX4\*SX3\*SX2  
. +SYX3\*SX5\*\*3\*SX0-2.\*SYX3\*SX5\*\*2\*SX4\*SX1-SYX3\*SX5\*\*2\*SX3\*  
. SX2+2.\*SYX3\*SX5\*SX4\*\*2\*SX2+SYX3\*SX5\*SX4\*SX3\*\*2-SYX3\*SX4\*\*  
. 3\*SX3+SYX2\*SX8\*SX6\*SX2\*SX0-SYX2\*SX8\*SX6\*SX1\*\*2-SYX2\*SX8\*  
. SX4\*\*2\*SX0+ANS2

CO(2)=ANS1/DETA

C - - - - - KOEFFIZIENT(3).

ANS3=-SYX0\*SX7\*SX4\*SX3\*\*2-SYX0\*SX6\*\*2\*SX4\*SX1+SYX0\*SX6\*\*2\*  
. SX3\*SX2+SYX0\*SX6\*SX5\*\*2\*SX1-2.\*SYX0\*SX6\*SX5\*SX3\*\*2+SYX0\*  
. SX6\*SX4\*\*2\*SX3-SYX0\*SX5\*\*3\*SX2+2.\*SYX0\*SX5\*\*2\*SX4\*SX3-  
. SYX0\*SX5\*SX4\*\*3

ANS2=-SYX2\*SX7\*SX5\*SX3\*SX0+SYX2\*SX7\*SX5\*SX2\*SX1+SYX2\*SX7\*  
. SX4\*SX3\*SX1-SYX2\*SX7\*SX4\*SX2\*\*2-SYX2\*SX6\*SX5\*SX4\*SX0+SYX2  
. \*SX6\*SX5\*SX3\*SX1+SYX2\*SX6\*SX4\*\*2\*SX1-SYX2\*SX6\*SX4\*SX3\*SX2

. +SYX2\*SX5\*\*3\*SX0-2.\*SYX2\*SX5\*\*2\*SX4\*SX1-SYX2\*SX5\*\*2\*SX3\*  
. SX2+2.\*SYX2\*SX5\*SX4\*\*2\*SX2+SYX2\*SX5\*SX4\*SX3\*\*2-SYX2\*SX4\*\*  
. 3\*SX3+SYX1\*SX8\*SX5\*SX3\*SX0-SYX1\*SX8\*SX5\*SX2\*SX1-SYX1\*SX8\*  
. SX4\*\*2\*SX0+SYX1\*SX8\*SX4\*SX3\*SX1+SYX1\*SX8\*SX4\*SX2\*\*2-SYX1\*  
. SX8\*SX3\*\*2\*SX2-SYX1\*SX7\*SX6\*SX3\*SX0+SYX1\*SX7\*SX6\*SX2\*SX1+  
. SYX1\*SX7\*SX5\*SX4\*SX0-SYX1\*SX7\*SX5\*SX2\*\*2-SYX1\*SX7\*SX4\*\*2\*  
. SX1+SYX1\*SX7\*SX4\*SX3\*SX2+SYX1\*SX6\*\*2\*SX4\*SX0-SYX1\*SX6\*\*2\*  
. SX3\*SX1-SYX1\*SX6\*SX5\*\*2\*SX0+SYX1\*SX6\*SX5\*SX4\*SX1+SYX1\*SX6  
. \*SX5\*SX3\*SX2-2.\*SYX1\*SX6\*SX4\*\*2\*SX2+SYX1\*SX6\*SX4\*SX3\*\*2+  
. SYX1\*SX5\*\*2\*SX4\*SX2-2.\*SYX1\*SX5\*SX4\*\*2\*SX3+SYX1\*SX4\*\*4-  
. SYX0\*SX8\*SX5\*SX3\*SX1+SYX0\*SX8\*SX5\*SX2\*\*2+SYX0\*SX8\*SX4\*\*2\*  
. SX1-2.\*SYX0\*SX8\*SX4\*SX3\*SX2+SYX0\*SX8\*SX3\*\*3+SYX0\*SX7\*SX6\*  
. SX3\*SX1-SYX0\*SX7\*SX6\*SX2\*\*2-SYX0\*SX7\*SX5\*SX4\*SX1+SYX0\*SX7  
. \*SX5\*SX3\*SX2+SYX0\*SX7\*SX4\*\*2\*SX2+ANS3

ANS1=-SYX4\*SX7\*SX4\*SX2\*SX0+SYX4\*SX7\*SX4\*SX1\*\*2+SYX4\*SX7\*  
. SX3\*\*2\*SX0-2.\*SYX4\*SX7\*SX3\*SX2\*SX1+SYX4\*SX7\*SX2\*\*3+SYX4\*  
. SX6\*SX5\*SX2\*SX0-SYX4\*SX6\*SX5\*SX1\*\*2-SYX4\*SX6\*SX4\*SX3\*SX0+  
. SYX4\*SX6\*SX4\*SX2\*SX1+SYX4\*SX6\*SX3\*\*2\*SX1-SYX4\*SX6\*SX3\*SX2  
. \*\*2-SYX4\*SX5\*\*2\*SX3\*SX0+SYX4\*SX5\*\*2\*SX2\*SX1+SYX4\*SX5\*SX4  
. \*\*2\*SX0-2.\*SYX4\*SX5\*SX4\*SX2\*\*2+SYX4\*SX5\*SX3\*\*2\*SX2-SYX4\*  
. SX4\*\*3\*SX1+2.\*SYX4\*SX4\*\*2\*SX3\*SX2-SYX4\*SX4\*SX3\*\*3+SYX3\*  
. SX8\*SX4\*SX2\*SX0-SYX3\*SX8\*SX4\*SX1\*\*2-SYX3\*SX8\*SX3\*\*2\*SX0+  
. 2.\*SYX3\*SX8\*SX3\*SX2\*SX1-SYX3\*SX8\*SX2\*\*3-SYX3\*SX6\*\*2\*SX2\*  
. SX0+SYX3\*SX6\*\*2\*SX1\*\*2+2.\*SYX3\*SX6\*SX5\*SX3\*SX0-2.\*SYX3\*  
. SX6\*SX5\*SX2\*SX1-2.\*SYX3\*SX6\*SX4\*SX3\*SX1+2.\*SYX3\*SX6\*SX4\*  
. SX2\*\*2-SYX3\*SX5\*\*2\*SX4\*SX0+SYX3\*SX5\*\*2\*SX2\*\*2+2.\*SYX3\*SX5  
. \*SX4\*\*2\*SX1-2.\*SYX3\*SX5\*SX4\*SX3\*SX2-SYX3\*SX4\*\*3\*SX2+SYX3\*  
. SX4\*\*2\*SX3\*\*2-SYX2\*SX8\*SX5\*SX2\*SX0+SYX2\*SX8\*SX5\*SX1\*\*2+  
. SYX2\*SX8\*SX4\*SX3\*SX0-SYX2\*SX8\*SX4\*SX2\*SX1-SYX2\*SX8\*SX3\*\*2  
. \*SX1+SYX2\*SX8\*SX3\*SX2\*\*2+SYX2\*SX7\*SX6\*SX2\*SX0-SYX2\*SX7\*  
. SX6\*SX1\*\*2+ANS2

CO(3)=ANS1/DETA

C - - - - KOEFFIZIENT(4).

ANS2=-SYX2\*SX6\*SX5\*SX2\*SX1+SYX2\*SX6\*SX4\*\*2\*SX0-3.\*SYX2\*SX6  
. \*SX4\*SX3\*SX1+SYX2\*SX6\*SX4\*SX2\*\*2+SYX2\*SX6\*SX3\*\*2\*SX2-SYX2  
. \*SX5\*\*2\*SX4\*SX0+SYX2\*SX5\*\*2\*SX3\*SX1+SYX2\*SX5\*SX4\*\*2\*SX1-  
. SYX2\*SX5\*SX3\*\*3-SYX2\*SX4\*\*3\*SX2+SYX2\*SX4\*\*2\*SX3\*\*2-SYX1\*  
. SX7\*SX5\*SX3\*SX0+SYX1\*SX7\*SX5\*SX2\*SX1+SYX1\*SX7\*SX4\*\*2\*SX0-  
. SYX1\*SX7\*SX4\*SX3\*SX1-SYX1\*SX7\*SX4\*SX2\*\*2+SYX1\*SX7\*SX3\*\*2\*  
. SX2+SYX1\*SX6\*\*2\*SX3\*SX0-SYX1\*SX6\*\*2\*SX2\*SX1-2.\*SYX1\*SX6\*



. SX5\*SX4\*SX0+SYX1\*SX6\*SX5\*SX3\*SX1+SYX1\*SX6\*SX5\*SX2\*\*2+SYX1  
. \*SX6\*SX4\*\*2\*SX1-SYX1\*SX6\*SX3\*\*3+SYX1\*SX5\*\*3\*SX0-SYX1\*SX5  
. \*\*2\*SX4\*SX1-2.\*SYX1\*SX5\*\*2\*SX3\*SX2+SYX1\*SX5\*SX4\*\*2\*SX2+2.  
. \*SYX1\*SX5\*SX4\*SX3\*\*2-SYX1\*SX4\*\*3\*SX3+SYX0\*SX7\*SX5\*SX3\*SX1  
. -SYX0\*SX7\*SX5\*SX2\*\*2-SYX0\*SX7\*SX4\*\*2\*SX1+2.\*SYX0\*SX7\*SX4\*  
. SX3\*SX2-SYX0\*SX7\*SX3\*\*3-SYX0\*SX6\*\*2\*SX3\*SX1+SYX0\*SX6\*\*2\*  
. SX2\*\*2+2.\*SYX0\*SX6\*SX5\*SX4\*SX1-2.\*SYX0\*SX6\*SX5\*SX3\*SX2-2.  
. \*SYX0\*SX6\*SX4\*\*2\*SX2+2.\*SYX0\*SX6\*SX4\*SX3\*\*2-SYX0\*SX5\*\*3\*  
. SX1+2.\*SYX0\*SX5\*\*2\*SX4\*SX2+SYX0\*SX5\*\*2\*SX3\*\*2-3.\*SYX0\*SX5  
. \*SX4\*\*2\*SX3+SYX0\*SX4\*\*4

ANS1=SYX4\*SX6\*SX4\*SX2\*SX0-SYX4\*SX6\*SX4\*SX1\*\*2-SYX4\*SX6\*SX3  
. \*\*2\*SX0+2.\*SYX4\*SX6\*SX3\*SX2\*SX1-SYX4\*SX6\*SX2\*\*3-SYX4\*SX5  
. \*\*2\*SX2\*SX0+SYX4\*SX5\*\*2\*SX1\*\*2+2.\*SYX4\*SX5\*SX4\*SX3\*SX0-2.  
. \*SYX4\*SX5\*SX4\*SX2\*SX1-2.\*SYX4\*SX5\*SX3\*\*2\*SX1+2.\*SYX4\*SX5\*  
. SX3\*SX2\*\*2-SYX4\*SX4\*\*3\*SX0+2.\*SYX4\*SX4\*\*2\*SX3\*SX1+SYX4\*  
. SX4\*\*2\*SX2\*\*2-3.\*SYX4\*SX4\*SX3\*\*2\*SX2+SYX4\*SX3\*\*4-SYX3\*SX7  
. \*SX4\*SX2\*SX0+SYX3\*SX7\*SX4\*SX1\*\*2+SYX3\*SX7\*SX3\*\*2\*SX0-2.\*  
. SYX3\*SX7\*SX3\*SX2\*SX1+SYX3\*SX7\*SX2\*\*3+SYX3\*SX6\*SX5\*SX2\*SX0  
. -SYX3\*SX6\*SX5\*SX1\*\*2-SYX3\*SX6\*SX4\*SX3\*SX0+SYX3\*SX6\*SX4\*  
. SX2\*SX1+SYX3\*SX6\*SX3\*\*2\*SX1-SYX3\*SX6\*SX3\*SX2\*\*2-SYX3\*SX5  
. \*\*2\*SX3\*SX0+SYX3\*SX5\*\*2\*SX2\*SX1+SYX3\*SX5\*SX4\*\*2\*SX0-2.\*  
. SYX3\*SX5\*SX4\*SX2\*\*2+SYX3\*SX5\*SX3\*\*2\*SX2-SYX3\*SX4\*\*3\*SX1+  
. 2.\*SYX3\*SX4\*\*2\*SX3\*SX2-SYX3\*SX4\*SX3\*\*3+SYX2\*SX7\*SX5\*SX2\*  
. SX0-SYX2\*SX7\*SX5\*SX1\*\*2-SYX2\*SX7\*SX4\*SX3\*SX0+SYX2\*SX7\*SX4  
. \*SX2\*SX1+SYX2\*SX7\*SX3\*\*2\*SX1-SYX2\*SX7\*SX3\*SX2\*\*2-SYX2\*SX6  
. \*\*2\*SX2\*SX0+SYX2\*SX6\*\*2\*SX1\*\*2+SYX2\*SX6\*SX5\*SX3\*SX0+ANS2

CO(4)=ANS1/DETA

RETURN

END

### 4.3 PLSYMB

#### Zentrierte Plotsymbole

Das Unterprogramm PLSYMB zeichnet einzelne zentrierte Symbole an die aktuelle Position. Vierzehn verschiedene Symbole sind möglich.

```
CALL PLSYMB(ZXX,ZYY,ZSYHO,NS)
```

#### EINGABEPARAMETER:

ZXX, ZYY = Aktuelle X- bzw. Y-Position beim Plotten.

ZSYHO = Größe der Plotsymbole (in Inches).

NS = Nummer des Plotsymbols, mit Werten von 0 bis 13.

0 :	"Quadrat"	7 :	"Kreuz mit Dach"
1 :	"Achteck"	8 :	"Zet"
2 :	"Dreieck"	9 :	"Ypsilon"
3 :	"Plus"	10 :	"Kreuzquadrat"
4 :	"Kreuz"	11 :	"Stern"
5 :	"Raute"	12 :	"Sanduhr"
6 :	"Kreuz & halbe Raute"	13 :	"Pfahl"

RÜCKGABEPARAMETER: keine

#### RESTLICHE VARIABLEN:

L = Hilfs-, Schleifen-, Zähl-Variable.

LINE = Plot-Subroutine (System).

LSY = 13, höchste Symbolnummer.

LXY = 11, maximale Anzahl an Punkten im Linienzug für ein Plotsymbol.

NN = Anzahl der Punkte im Linienzug des Plotsymbols.

ZSYM(X(0:LXY)) = Standard-X-Koordinaten des Plotsymbols.

ZSYM(Y(0:LXY)) = Standard-Y-Koordinaten des Plotsymbols.

ZX(1:LXY+3) = Aktuelle X-Koordinaten zum Plotten des Symbols.

ZY(1:LXY+3) = Aktuelle Y-Koordinaten zum Plotten des Symbols.



```
DATA (ZSYM(X,L),L=1,LXY)/ 2.,-2., 2.,-2., 2.,6*0./
DATA (ZSYMY(L,8),L=1,LXY)/ 2., 2., 2.,-2.,-2.,6*0./
C - - - - - NR. 9 "YPSILON"
DATA (ZSYM(X,L,9),L=1,LXY)/ 2., 0.,-2., 0., 0.,6*0./
DATA (ZSYMY(L,9),L=1,LXY)/ 2., 0., 2., 0.,-2.,6*0./
C - - - - - NR.10 "KREUZQUADRAT"
DATA (ZSYM(X,L,10),L=1,LXY)
$ /-2., 2., 0.,-2., 2., 1.,-1.,-1., 1., 1.,1*0./
DATA (ZSYMY(L,10),L=1,LXY)
$ / 2.,-2., 0.,-2., 2., 1., 1.,-1.,-1., 1.,1*0./
C - - - - - NR.11 "STERN"
DATA (ZSYM(X,L,11),L=1,LXY)
$ /-2., 2., 0.,-2., 2., 0., 0., 0., 0.,-2., 2./
DATA (ZSYMY(L,11),L=1,LXY)
$ / 2.,-2., 0.,-2., 2., 0., 2.,-2., 0., 0., 0./
C - - - - - NR.12 "SANDUHR"
DATA (ZSYM(X,L,12),L=1,LXY)/ 2.,-2., 2.,-2., 0.,6*0./
DATA (ZSYMY(L,12),L=1,LXY)/ 2., 2.,-2.,-2., 0.,6*0./
C - - - - - NR.13 "PFAHL"
DATA (ZSYM(X,L,13),L=1,LXY)/ 0., 0.,9*0./
DATA (ZSYMY(L,13),L=1,LXY)/ 2.,-2.,9*0./
C - - - - ZEICHENGROESSE BERUECKSICHTIGEN - - - - -
NN = ZSYM(X,NS)
DO 1 L=1,NN
    ZX(L) = ZSYM(X,L,NS) * (ZSYHO*0.24) + ZXX
    ZY(L) = ZSYMY(L,NS) * (ZSYHO*0.24) + ZYY
1 ZX(NN+1) = 0.
  ZY(NN+1) = 0.
  ZX(NN+2) = 1.
  ZY(NN+2) = 1.
CALL LINE(ZX,ZY,NN,1,0,0)
RETURN
END
```

#### 4.4 P12LF

Berechnung der Legendre-Funktion  $P_{1/2}(s)$ ,  $s \geq 1$

---

#### FORMELN ZUR BERECHNUNG

In dieser Programmbeschreibung werden drei Berechnungsarten der Legendre-Funktion  $P_{1/2}(s)$ , für  $s \geq 1$ , vorgestellt.

1. Berechnung durch das Polynom

$$P_{1/2}(s) = 1 + \frac{3}{4}S - \frac{15}{64}S^2 + \frac{35}{256}S^3 - \frac{63}{4096}S^4 + \dots$$

$$\text{mit } S = \frac{s-1}{2}$$

2. Berechnung durch eine Reihenentwicklung des vollständigen elliptischen Integrals 2. Art im Unterprogramm CELINT nach der Formel

$$P_{1/2}(s) = \frac{2}{\pi} \cdot \left[ s + (s \cdot s - 1)^{1/2} \right]^{1/2} \cdot \left[ \sum_{i=1}^{10} a(i) X(s)^i + \log\left(\frac{1}{X(s)}\right) \cdot \sum_{i=1}^{10} b(i) X(s)^i \right]$$

$$\text{mit } X(s) = \left[ \frac{2 \cdot (s \cdot s - 1)^{1/2}}{s + (s \cdot s - 1)^{1/2}} \right]^{1/2}$$

3. Berechnung durch eine hypergeometrische Reihe  ${}_2F_1$   
(Unterprogramm F21HYP) als Faktor bei der Auswertung eines  
vollständigen elliptischen Integrals 2. Art nach der Formel

$$P_{1/2}(s) = \frac{2}{\pi} \cdot \left( s + (s \cdot s - 1)^{1/2} \right)^{1/2} \cdot$$

$$\cdot \frac{\pi}{2} \cdot {}_2F_1(-0.5, 0.5; 1.0; X(s))$$

Allgemein ist das vollständige elliptische Integral  
2. Art  $E(X)$  durch das folgende Integral definiert:

$$E(X) = \int_0^{\pi/2} \left( 1 - X^2 \cdot \sin^2 t \right)^{1/2} dt =$$

$$= \frac{\pi}{2} \cdot {}_2F_1(-0.5, 0.5; 1.0; X(s)) \quad \text{für } |X| < 1. ,$$

$$X(s) = \left[ \frac{2 \cdot (s \cdot s - 1)^{1/2}}{s + (s \cdot s - 1)^{1/2}} \right]^{1/2} \quad \text{für } s \geq 1.$$

ABDRUCK DES TESTPROGRAMMS DER FORMELN (MIT BEISPIELEINGABE)

```
//      MSGCLASS=H,MSGLEVEL=(2,0)
//      EXEC F7C, PARM.C='AUTODBL(DBL),LANGLVL(77),LINECOUNT(63)'
//C.SYSPRINT DD DUMMY
//C.SYSIN DD DISP=SHR,DSN=IAK964.HYPERG.DATA(CELINT)
//      DD DISP=SHR,DSN=IAK964.HYPERG.DATA(F21HYP)
//      EXEC F7CLG, PARM.C='AUTODBL(DBL),LANGLVL(77),LINECOUNT(63)'
//* C.SYSPRINT DD DUMMY
//C.SYSIN DD *
```

```
C-----
C      TESTEN DER BERECHNUNG DER LEGENDRE-FUNKTION P12LF(S)
C-----
```

```
PROGRAM TP12LF
IMPLICIT DOUBLE PRECISION (A-H,O-Y)
PARAMETER (PI=3.1415926535897932384626433832795028D0)
PARAMETER (CA=-0.5D0,CB=0.5D0,CC=1.D0)
PARAMETER (PIH=PI*0.5D0,HPI=2.D0/PI)
PARAMETER (CA1=0.44325141463D0,CA2=0.06260601220D0)
PARAMETER (CA3=0.04757383546D0,CA4=0.01736506451D0)
PARAMETER (CB1=0.24998368313D0,CB2=0.09200180037D0)
PARAMETER (CB3=0.04069697526D0,CB4=0.00526449639D0)
1 CONTINUE
READ(5,*,END=3) SS
IF (DABS(SS).LE.CC) GOTO 1
CD=DSQRT(SS*SS-CC)
CE=SS+CD
CX=2.D0*CD/CE
CXX=CC-CX
WRITE(*,*) '      '
WRITE(*,*) '      X(S) = ',CX
WRITE(*,*) '      1-X(S) = ',CXX
WRITE(*,*) '      S = ',SS
C - - - - - 1. Formel
SM = CB * (SS-CC)
WRITE(*,*) '      SM = ',SM
P12A = 1.D0 + 0.75D0*SM - 15.D0/64.D0*SM*SM +
+ 35.D0/256.D0*SM*SM*SM
WRITE(*,*) '      P12A = ',P12A
C - - - - - 2. Formel
```

```
ITYP=3
CALL CELINT(CX, ITYP, CEI1, CEI2)
P12B=HPI*DSQRT(CE)*CEI2
WRITE(*,*) '          P12B = ', P12B, '   EXB = ', CEI2
C - - - - - 3. Formel
EXC=PIH*F21HYP(CA, CB, CC, CX)
P12A=HPI*DSQRT(CE)*EXC
WRITE(*,*) '          P12C = ', P12C, '   EXC = ', EXC
C - - - - - Weitere Formel ähnlich der 2.
EXD=(((CA4*CXX+CA3)*CXX+CA2)*CXX+CA1)*CXX+CC)+
+(((CB4*CXX+CB3)*CXX+CB2)*CXX+CB1)*CXX)*DLOG(CC/CXX)
P12D=HPI*DSQRT(CE)*EXD
WRITE(*,*) '          P12D = ', P12D, '   EXD = ', EXD
GOTO 1
3 CONTINUE
STOP
END
//L.SYSPRINT DD DUMMY
//L.SYSIN DD *
ENTRY TP12LF
//G.SYSIN DD *
1.01
1.04
1.08
1.4
1.8
//
```



## CELINT

Berechnung des vollständigen elliptischen  
Integrals 1. oder/und 2. Art, K(X) bzw. E(X)

---

CALL CELINT(X, ITYP, CEI1, CEI2)

### FORMELN ZUR BERECHNUNG

$$\text{1. Art: } K(X) = \int_0^{\pi/2} (1 - X^2 \cdot \sin^2 t)^{-1/2} dt$$

$$\text{2. Art: } E(X) = \int_0^{\pi/2} (1 - X^2 \cdot \sin^2 t)^{1/2} dt$$

$$CEI1/2 = \sum_{i=0}^{10} a(i) XX^i + \log\left(\frac{1}{XX}\right) \cdot \sum_{i=1}^{10} b(i) XX^i$$

### PARAMETER:

X = Argument ( $0 \leq X < 1$ )  
ITYP = Berechnungsart; 1 : Berechnung von CEI1,  
2 : Berechnung von CEI2, 3 : CEI1 und CEI2.  
CEI1 = compl. ell. int. of the 1st kind  
CEI2 = compl. ell. int. of the 2nd kind

### HILFSVARIABLE:

XX = 1-X  
CE = -LOG(XX)

ABDRUCK DES UNTERPROGRAMMS CELINT

```
C-----
C      VOLLSTAENDIGES ELLIPTISCHES INTEGRAL 1. UND 2. ART
C-----

      SUBROUTINE CELINT(X, ITYP, CEI1, CEI2)
      IMPLICIT DOUBLE PRECISION (A-H, O-Y)
      CEI1 = 0.7237D+76
      CEI2 = 1.DO
      XX = 1.DO - X
      IF (XX.GT.0.DO) CE = -DLOG(XX)
      IF (ITYP.LT.2.OR.X.GE.1.DO) GOTO 1
          CEI2 = CE * ((((((((((3.18591956555015718D-5*XX
$ +.989833284622538479D-3)*XX+.643214658643830177D-2)*XX
$ +.16804023346363385D-1)*XX+.261450147003138789D-1)*XX
$ +.334789436657616262D-1)*XX+.427178905473830956D-1)*XX
$ +.585936612555314917D-1)*XX+.937499997212031407D-1)*XX
$ +.249999999999901772D0)*XX)+((((((((
$ .149466217571813268D-3*XX+.246850333046072273D-2)*XX
$ +.863844217360407443D-2)*XX+.107706350398664555D-1)*XX
$ +.782040406095955417D-2)*XX+.759509342255943228D-2)*XX
$ +.115695957452954022D-1)*XX+.218318116761304816D-1)*XX
$ +.568051945675591566D-1)*XX+.443147180560889526D0)*XX+1.
1  IF (ITYP.EQ.2.OR.X.GE.1.DO) GOTO 2
          CEI1 = CE * ((((((((((.297002809665556121D-4*XX
$ +.921554634963249846D-3)*XX+.597390429915542916D-2)*XX
$ +.155309416319772039D-1)*XX+.239319133231107901D-1)*XX
$ +.301248490128989303D-1)*XX+.373777397586236041D-1)*XX
$ +.48828041906862398D-1)*XX+.703124997390383521D-1)*XX
$ +.124999999999908081D0)*XX+.5D0)+((((((((
$ .139308785700664673D-3*XX+.229663489839695869D-2)*XX
$ +.800300398064998537D-2)*XX+.984892932217689377D-2)*XX
$ +.684790928262450512D-2)*XX+.617962744605331761D-2)*XX
$ +.878980187455506468D-2)*XX+.149380135326871652D-1)*XX
$ +.308851462713051899D-1)*XX+.965735902808562554D-1)*XX
$ +1.38629436111989062D0
2  RETURN
      END
```

## F21HYP

Berechnung eines Funktionswertes der hypergeometrischen Reihe  
 ${}_2F_1(a, b; c; x)$

---

### FORMELN ZUR BERECHNUNG

Die Funktion F21HYP berechnet den Funktionswert einer hypergeometrischen Reihe der Form

$${}_2F_1(a, b; c; x) = \sum_{n=0}^{\infty} \frac{(a)_n \cdot (b)_n}{(c)_n \cdot n!} \cdot x^n$$

Erläuterung der verwendeten Pochhammersymbole:

$$(y)_0 := 1$$

$$(y)_k := y \cdot (y+1) \cdot (y+2) \cdot \dots \cdot (y+k-1)$$

$$\text{z.B. : } y=5 \text{ mit } k=3 \text{ ergibt } (5)_3 = 5 \cdot 6 \cdot 7 = 210$$

Die ersten Glieder der hypergeometrischen Funktion lauten somit:

$$\begin{aligned} {}_2F_1 &= 1 + \frac{a \cdot b}{c} \cdot x \\ &+ \frac{a(a+1) \cdot b(b+1)}{c(c+1) \cdot 2} \cdot x^2 \\ &+ \frac{a(a+1) \cdot (a+2) \cdot b(b+1) \cdot (b+2)}{c(c+1) \cdot (c+2) \cdot 6} \cdot x^3 \\ &+ \frac{a(a+1) \cdot (a+2) \cdot (a+3) \cdot b(b+1) \cdot (b+2) \cdot (b+3)}{c(c+1) \cdot (c+2) \cdot (c+3) \cdot 24} \cdot x^4 + \dots \end{aligned}$$

## AUFRUF DER FUNKTION

$$Y = F21HYP(AA, BB, CX, XX)$$

### PARAMETER:

AA, BB : Zählerparameter  
CX : Nennerparameter  
XX : unabhängige Veränderliche  
F21HYP : berechneter Funktionswert

## EIGENSCHAFTEN DER FUNKTION

### Konvergenzverhalten:

für  $|z| < 1$  absolute Konvergenz,  
für  $|z| = 1$  absolute Konvergenz, wenn  $\text{Re}(a+b-c) < 0$ ,  
für  $|z| = 1, z \neq 1$  bedingte Konvergenz, wenn  $0 \leq \text{Re}(a+b-c) < 1$ ,  
für  $|z| = 1$  Divergenz, wenn  $1 \leq \text{Re}(a+b-c)$ .

Die hypergeometrische Reihe stellt im Falle der gleichmäßigen Konvergenz eine analytische Funktion von  $a, b, c, z$  dar. Als Funktion von  $a, b$  betrachtet, ist  $F(a, b; c; z)$  ( $|z| < 1$ ) eine in der ganzen Ebene mit Ausnahme des Punktes  $\infty$  holomorphe Funktion; als Funktion von  $c$  betrachtet ist sie meromorph und wird in den Punkten  $c=0, -1, -2, \dots$  unendlich von der ersten Ordnung. In beiden Fällen ist  $\infty$  eine wesentlich singuläre Stelle. Ist eine der Zahlen  $a, b$  eine negative ganze Zahl  $-n$  ( $n > 0$ ), so bricht die Reihe ab, und man erhält ein Polynom  $n$ -ten Grades. Für  $c=-m$ , aber  $a \neq -n$  oder  $b \neq -n$  bei  $n < m$  mit  $m=0, 1, 2, \dots$  ist die Funktion nicht mehr definiert, da die Nenner ab einer bestimmten Stelle verschwinden.

Dem exponentiellen Anstieg der Zwischenwerte wird durch Logarithmieren entgegengewirkt. Deshalb sieht die Berechnung folgendermaßen aus:

$$F_2^1 = \sum_{n=0}^{\infty} e^{\text{exponent}}$$

mit  $\text{exponent} = \ln\left(\frac{a}{n}\right) + \ln\left(\frac{b}{n}\right) + n \cdot x - \ln\left(\frac{c}{n}\right) - \ln(n!)$ .

Beim Logarithmieren werden natürlich nur die Beträge verwendet, die Vorzeichen der Argumente des Logarithmus werden getrennt gespeichert und beim nachfolgenden Potenzieren wieder berücksichtigt.

#### TABELLE VON SPEZIELLEN HYPERGEOMETRISCHEN FUNKTIONEN

An der folgenden Aufstellung von Formeln und Beziehungen zwischen hypergeometrischen Funktionen und elementaren, analytischen Funktionen für bestimmte Argumente zeigt sich, daß es nicht immer sinnvoll sein muß, bei der Berechnung eines Funktionswertes die allgemeine Darstellung und Definition der hypergeometrischen Reihe zu verwenden, sondern im konkreten Fall die Verwendung der 'direkten' Formeln sehr viel schneller zum Ziel führen können.

Die Gültigkeit der folgenden Beziehungen beschränkt sich für  $|x| < 1$  nur auf den Konvergenzradius dieser Reihe. Die Funktion  ${}_2F_1$  wird hier - wie häufig in der Literatur zu finden - nur noch kurz mit F bezeichnet:

$$(1) \quad F(a, b; b; x) = (1-x)^{-a}$$

$$(2) \quad F(-a, b; b; -x) = (1+x)^a$$

$$(3) \quad F(a, b; c; x) = (1-x)^{(c-a-b)} \cdot F(c-a, c-b; c; x)$$

$$(4) \quad F(a, b; c; 1) = \frac{\Gamma(c) \cdot \Gamma(c-a-b)}{\Gamma(c-a) \cdot \Gamma(c-b)}, \quad \text{real}(c-a-b) > 0$$

$$(5) \quad F(1, 1; 2; x) = -x^{-1} \cdot \ln(1-x)$$

$$(6) \quad F(1, 1; 2; -x) = x^{-1} \cdot \ln(1+x)$$

$$(7) \quad F\left(\frac{1}{2}, 1; \frac{3}{2}; x^2\right) = \frac{1}{2} \cdot x^{-1} \cdot \ln\left(\frac{1+x}{1-x}\right)$$

$$(8) \quad F\left(\frac{1}{2}, 1; \frac{3}{2}; -x^2\right) = x^{-1} \cdot \arctan(x)$$

$$(9) \quad F\left(\frac{1}{2}, \frac{1}{2}; \frac{3}{2}; x^2\right) = x^{-1} \cdot \arcsin(x) = (1-x^2)^{\frac{1}{2}} \cdot F(1, 1; \frac{3}{2}; x^2)$$

$$(10) \quad F\left(\frac{1}{2}, -\frac{1}{2}; \frac{1}{2}; x^2\right) = (1-x^2)^{\frac{1}{2}}$$

$$(11) \quad F\left(\frac{1}{2} \cdot a, -\frac{1}{2} \cdot a; \frac{1}{2}; \sin^2(x)\right) = \cos(a \cdot x)$$

$$(12) \quad (2 \cdot a - c - x \cdot (a - b)) \cdot F(a, b; c; x) + (c - a) \cdot F(a - 1, b; c; x) + \\ + a \cdot (x - 1) \cdot F(a + 1, b; c; x) = 0$$

### BESCHREIBUNG ALLER VARIABLEN

- AA = 1. Eingabeargument (Zählerparameter).  
BB = 2. Eingabeargument (Zählerparameter).  
CX = 3. Eingabeargument (Nennerparameter).  
CO, C1 = Konstante 0.0, 1.0  
DABS = fortran 77 builtin function  
DEXP = fortran 77 builtin function  
DEXPGR = Konstante (größter zulässiger Exponent = 174).  
DEXPKL = Exponent für die gesetzte Genauigkeitsschranke DGENAU.  
DGENAU = Genauigkeitsschranke der vernachlässigten Reihenglieder.  
DINT = fortran 77 builtin function  
DLOG = fortran 77 builtin function  
DSIGN = fortran 77 builtin function  
EXPON = logarithmierter Wert des aktuellen Reihengliedes.  
FAKLOG = Hilfsfeld, zur Abspeicherung der natürlichen Logarithmen der Fakultäten.  
GROEST = Konstante (größter zulässiger Funktionswert = 1.D+75).  
GUNDEF = Konstante (dieser Wert wird gesetzt, wenn die die Reihe nicht definiert ist. -1.234321234D-75).  
IDINT = fortran 77 builtin function  
IPRINT = Parameter zur Steuerung von Druckausgabe innerhalb der Funktion.  
MA = Schleifenvariable; aktuelle Summe der Potenzen in XX.  
MACX = Hilfsvariable für ganzzahlige negative Werte von CX.  
MAMAX = Konstante (obere Grenze der "unendlichen" Reihe).  
MATERM = aktuelle Anzahl der benötigten Pochhammer-Symbole von AA.  
MA1 = Hilfsvariable für MA-1.  
MINO = fortran 77 builtin function

MOD = fortran 77 builtin function  
POLOAA = Variable für das logarithmierte Pochhammer-Symbol von AA.  
POLOBB = Variable für das logarithmierte Pochhammer-Symbol von BB.  
POLOCX = Variable für das logarithmierte Pochhammer-Symbol von CX.  
SIGAA = Vorzeichen des aktuellen Pochhammer-Symbols von AA.  
SIGBB = Vorzeichen des aktuellen Pochhammer-Symbols von BB.  
SIGCX = Vorzeichen des aktuellen Pochhammer-Symbols von CX.  
SIGXX = Vorzeichen der unabhängigen Variablen XX.  
SUM = aktueller Reihenwert.  
XABS = Betrag von XX.  
XX = 4. Eingabeargument (unabhängige Veränderliche).  
XXLN = natürlicher Logarithmus von XX.

ABDRUCK DES FUNKTIONS-UNTERPROGRAMMS F21HYP

```
C-----
C   BERECHNUNG EINES FUNKTIONSWERTES VON   F (AA,BB;CX;XX)
C                                     2 1
C-----
      DOUBLE PRECISION FUNCTION F21HYP(AA,BB,CX,XX)
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      PARAMETER (CO=0.DO,C1=1.DO,MAMAX=2000,IPRINT=0)
      PARAMETER (DGENAU=1.D-18,GUNDEF=-1.234321234D-75)
      PARAMETER (DEXPGR=174.DO,GROEST=1.D+75)
C-----ABFRAGE, AUF GUELTIGKEIT DER ARGUMENTE-----
      DEXPKL = LOG(DGENAU)
      F21HYP = CO
      MATERM = -2
      MACX   = -1
C-----XX = 0-----
      IF (XX.EQ.CO) THEN
          MATERM = 0
          F21HYP = C1
          GOTO 99
      ENDIF
C-----CX <= 0 & CX GANZZAHLIG-----
      IF (CX.LE.CO.AND.CX.EQ.DINT(CX)) MACX = IDINT(DABS(CX))
C-----AA <= 0 & AA GANZZAHLIG-----
      IF(AA.LE.CO.AND.AA.EQ.DINT(AA)) MATERM = IDINT(DABS(AA))
C-----BB <= 0 & BB GANZZAHLIG-----
      IF (BB.LE.CO.AND.BB.EQ.DINT(BB)) THEN
C C-----AA <= 0 & AA GANZZAHLIG GEWESEN-----
          IF (MATERM.GE.0) THEN
              MATERM = MINO(MATERM, IDINT(DABS(BB)))
          ELSE
              MATERM = IDINT(DABS(BB))
          ENDIF
      ENDIF
C-----CX IST GANZZAHLIG NEGATIV UND GROESSER AA UND BB-----
      IF (MACX.LE.MATERM.AND.MACX.GE.0
$ .OR.MACX.GE.0.AND.MATERM.LT.0) THEN
          IF (IPRINT.GE.1)
$       WRITE(*,*) ' DIE REIHE IST NICHT DEFINIERT. '
```



```
F21HYP = GUNDEF
GOTO 99
C-----REIHE IST EIN POLYNOM IN X-----
ELSE
  IF (MATERM.GE.0) THEN
    IF (IPRINT.GE.1) WRITE(*,*) 'DIE REIHE IST EIN',
$    'POLYNOM VOM GRAD ',MATERM,' IN X.'
  ELSE
C-----UNENDLICHE REIHE LIEGT VOR-----
  IF (DABS(XX).GT.C1) THEN
C C C-----|XX| > 1
  IF (IPRINT.GE.1)
$    WRITE(*,*) ' DIE REIHE DIVERGIERT.'
    F21HYP=GROEST
    GOTO 99
  ELSE
    IF (DABS(XX).LT.C1) THEN
C C C C-----|XX| < 1
  IF (IPRINT.GE.1)
$    WRITE(*,*) ' ABSOLUTE KONVERGENZ.'
    MATERM=MAMAX
  ELSE
C C C C-----|XX| = 1-----
    IF (CX-AA-BB.LE.-C1) THEN
C C C C C-----REAL(CX-AA-BB) =< -1: "DIVERGENZ"
  IF (IPRINT.GE.1) WRITE(*,*)
$    ' DIE REIHE DIVERGIERT.'
    F21HYP=GROEST
    GOTO 99
  ELSE
C C C C C-----REAL(CX-AA-BB) > -1 "?"
  IF (CX-AA-BB.LE.CO) THEN
C C C C C C-----REAL(CX-AA-BB) <= 0 "?"
  IF (XX.EQ.C1) THEN
C C C C C C C-----XX = 1 "REIHE DIVERGENT"
  IF (IPRINT.GE.1) WRITE(*,*)
$    ' DIE REIHE DIVERGIERT.'
    F21HYP=GROEST
    GOTO 99
```

```

                                ELSE
C C C C C C C-----XX := 1 "BEDINGT KONVERGENT"
                                IF (IPRINT.GE.1) WRITE(*,*)
                                ' BEDINGTE KONVERGENZ.'
                                $
                                MATERM=MAMAX
                                ENDIF
                                ELSE
C C C C C C C-----REAL(CX-AA-BB) > 0 "?"
                                IF (IPRINT.GE.1) WRITE(*,*)
                                ' ABSOLUTE KONVERGENZ.'
                                $
                                MATERM=MAMAX
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
                                ENDIF
C-----BERECHNUNG DES REIHENWERTES-----
XXLN = LOG(ABS(XX))
SIGXX = SIGN(C1,XX)
POLOAA = CO
POLOBB = CO
POLOCX = CO
FAKLOG = CO
SIGAA = C1
SIGBB = C1
SIGCX = C1
SUM = C1
DO 91 MA=1,MATERM
MA1 = MA - 1
POLOAA = POLOAA + LOG(ABS(AA + MA1))
POLOBB = POLOBB + LOG(ABS(BB + MA1))
POLOCX = POLOCX + LOG(ABS(CX + MA1))
FAKLOG = FAKLOG + LOG(CO + MA)
SIGAA = SIGAA * SIGN(C1,AA+MA1)
SIGBB = SIGBB * SIGN(C1,BB+MA1)
SIGCX = SIGCX * SIGN(C1,CX+MA1)
EXPON = POLOAA + POLOBB + MA*XXLN - (POLOCX + FAKLOG)
IF (IPRINT.GE.3)
$ WRITE(*,*) ' MA = ',MA,' EXPON = ',EXPON,' SUM = ',SUM
```

```
IF (EXPON.LT.DEXPKL) THEN
  IF (IPRINT.GE.1.AND.MATERM.EQ.MAMAX)
  $ WRITE(*,*) ' DIE REIHE KONVERGIERT BEI MA = ',MA,' .'
  SUM = SUM+SIGAA*SIGBB*SIGCX*SIGXX**MOD(MA,2)*EXP(EXPON)
  IF (IPRINT.GE.1)
  $ WRITE(*,*) ' LETZTER TERM = EXP(',EXPON,')=',EXP(EXPON)
  GOTO 92
ELSE
  IF (EXPON.LE.DEXPGR) THEN
    SUM=SUM+SIGAA*SIGBB*SIGCX*SIGXX**MOD(MA,2)*EXP(EXPON)
  ELSE
    IF (SUM.LE.GROEST) THEN
      IF (IPRINT.GE.1)
      $ WRITE(*,*) ' SUMMAND MIT MA = ',MA,' NICHT ',
      $ ' DARSTELLBAR (EXPONENT OVERFLOW).'
    ELSE
      IF (IPRINT.GE.1)
      $ WRITE(*,*) ' REIHENWERT AB MA = ',MA,' NICHT ',
      $ ' MEHR DARSTELLBAR (EXPONENT OVERFLOW).'
    ENDIF
    GOTO 92
  ENDIF
ENDIF
ENDIF
91 CONTINUE
IF (MATERM.EQ.MAMAX.AND.IPRINT.GE.1) THEN
  WRITE(*,*) ' GEWUENSCHTE RELATIVE GENAUGKEIT ',
  $ DGENAU,' BEI MA = ',MAMAX,' NOCH NICHT ERREICHT.'
  WRITE(*,*) ' LETZTER TERM = EXP(',EXPON,')=',EXP(EXPON)
ENDIF
C-----REIHENBERECHNUNG BIS MAMAX-----
92 CONTINUE
F21HYP = SUM
C-----ENDE DES FUNKTIONS-UNTERPROGRAMMS-----
99 CONTINUE
RETURN
END
```

## 4.5 SERBXY

### Plotten von Texten

Das Unterprogramm SERBXY dient zum Plotten von Texten an den Koordinatenachsen und zum Eintragen der Kurvenkennzeichnungen.

```
CALL SERBXY
      (JWERT, NKURV, KK, NST, ZXI, ZYI, ZSYHO, ZIFHO, WSTREU, NPHAS)
```

### EINGABEPARAMETER:

JWERT = Aktueller Wert des Parameters J aus dem Feld JJ.  
NKURV = Aktuelle Anzahl der Kurven in dem Bild.  
KK(1:4) = Aktuelle K-Werte.  
NST(1:4) = Aktuelle Werte der "Strichelungsart".  
ZXI = Länge der Abszisse in Inches.  
ZYI = Länge der Ordinate in Inches.  
ZSYHO = Textgröße in Inches.  
ZIFHO = Zifferngröße in Inches.  
WSTREU = Streuwinkel (Laborwinkel) des beobachteten Teilchens (in Grad) als Parameter bei der Berechnung des doppeltdifferentiellen Wirkungsquerschnitts.  
NPHAS = Anzahl der zu plottenden Kurvenzüge von Phasenraumfaktoren.

RÜCKGABEPARAMETER: keine

### RESTLICHE VARIABLEN:

CHA\*3 = Hilfsstring beim Zeichnen der Kurvenkennzeichnung.  
CHAXEN(1:27)\*32 = Tabelle der Beschriftungen unter Berücksichtigung der Achsenunterscheidungen und des Text-Typs.  
CHKURV(1:5)\*7 = Zeichensymbole für die fünf Möglichkeiten, eine Kurve zu zeichnen ("Strichelungsart"):  
1: durchgezogen  
2: gestrichelt

- 3: punktiert
- 4: strich-punktiert
- 5: strich-doppelpunktiert

I, IA, IB = Hilfs-, Schleifen-, Zähl-Variable.

NCHTYP(1:27) = Integer-Kennzeichnung der Textart:

- 0: große deutsche Buchstaben
- 1: kleine deutsche Buchstaben
- 2: große griech. Buchstaben
- 3: kleine griech. Buchstaben

NJWERT(1:6) = J-Wert-bezogene Anfangs- und End-Nummer der Elemente aus den Feldern NCHTYP und CHAXEN.

SYMBL0 = Plotsubroutine für große deutsche Buchstaben (System).

SYMBL1 = Plotsubroutine für kleine deutsche Buchstaben (System).

SYMBL2 = Plotsubroutine für große griech. Buchstaben (System).

SYMBL3 = Plotsubroutine für kleine griech. Buchstaben (System).

SYMBGR = Eigene Plotsubroutine für kleine griech. Buchstaben.

ZXX = Aktuelle Plotposition (Abszisse).

ZYY = Aktuelle Plotposition (Ordinate).

#### ABDRUCK DES UNTERPROGRAMMS SERBXY

C - - - Plotten von Text

```
SUBROUTINE SERBXY(JWERT,NKURV, KK, NST,
$   ZXI, ZYI, ZEIHO, ZIFHO, WSTREU, NPHAS)
  IMPLICIT REAL*8(A-H, O-Y)
  CHARACTER CHAXEN(27)*32, CHKURV(5)*7, CHA*3
  DIMENSION NCHTYP(27), NJWERT(6), NST(4), KK(4)
```

C - - - - TEXTART-KENNZEICHNUNG

```
DATA NJWERT/1, 6, 7, 13, 14, 22/
DATA NCHTYP/0, 1, 2, 3, 0, 1,
$           0, 1, 2, 3, 0, 1, 2,
$           0, 1, 2, 3, 0, 1, 0, 1, 2, 0, 0, 1, 0, 1/
```

C - - - - TEXTE

```
DATA (CHAXEN(I), I=1, 6)
$           /' / E ( /M V) ' ,
$           'D D MB E ' ,
$           ' ' ' ,
$           ' S ' ,
$           ' E (M V) ' ,
$           ' NERGY E ' /
```

DATA (CHAXEN(I), I=7, 13)

```

$          /' / ( / )
$          'D D MB SR
$          ' 4
$          ' S
$          '
$          ( )
$          DEGREES
$          '
$          ' 2

```

DATA (CHAXEN(I), I=14, 27)

```

$          /' / ( E) ( / M V)
$          'D D D MB SR E
$          ' 4
$          ' S
$          ' E (M V)
$          ' ENERGY E
$          ' P = ( )
$          ' PARAMETER DEGREES
$          ' 2
$          ' 2
$          ' ( )
$          ' PHASE SPACE FACTOR ARB UNITS
$          '
$          ' PH SP FACT

```

DATA CHKURV/' \_\_\_\_\_', ' - - - - -',

\$ '.....', '.....', '.....' /

IA = NJWERT(JWERT\*2-1)

IB = NJWERT(JWERT\*2)

C - - - - - EINTRAGEN DER KURVEN-KENNZEICHNUNG

DO 40 I=1, NKURV

ZXX = ZXI - ZEIHO\*10.

ZYY = ZYI + (ZIFHO+ZEIHO)\*I

CHA = 'K= '

WRITE(CHA(3:3), '(I1)') KK(I)

CALL SYMBLO(ZXX, ZYY+ZEIHO\*0.5, ZEIHO, CHKURV(NST(I)), 0., 7)

CALL SYMBLO(ZXX+ZEIHO\*8., ZYY, ZEIHO, CHA, 0., 3)

40 CONTINUE

C - - - - - EINTRAGEN DER ORDINATEN-BESCHRIFTUNG.

DO 41 I=IA, IA+3

ZXX = -ZEIHO\*2.

ZYY = ZYI + ZEIHO + ZIFHO

```
      IF (NCHTYP(I).EQ.0)
$      CALL SYMBLO(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
      IF (NCHTYP(I).EQ.1)
$      CALL SYMBL1(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
      IF (NCHTYP(I).EQ.2)
$      CALL SYMBL2(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
      IF (NCHTYP(I).EQ.3)
$      CALL SYMBL3(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
C      oder CALL SYMBGR(ZXX,ZYY,ZSYHO,CHAXEN(I),0.,25)
41 CONTINUE
      IF (JWERT.EQ.3) THEN
          ZXX = -ZEIHO
          ZYY = ZYI + ZEIHO*1.666 + ZIFHO
          CALL SYMBLO(ZXX,ZYY,ZEIHO*0.6,CHAXEN(23)(1:1),0.,1)
      ENDIF
C - - - - EINTRAGEN DER ABSZISSEN-BESCHRIFTUNG.
      DO 42 I=IA+4,IB
          ZXX = ZXI - ZEIHO*31.
          ZYY = -(ZIFHO*3.+ZEIHO)
          IF (I.GE.20) ZYY = ZYY - ZEIHO*2.666
          IF (NCHTYP(I).EQ.0)
$          CALL SYMBLO(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
          IF (NCHTYP(I).EQ.1)
$          CALL SYMBL1(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
          IF (NCHTYP(I).EQ.2)
$          CALL SYMBL2(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
          IF (NCHTYP(I).EQ.3)
$          CALL SYMBL3(ZXX,ZYY,ZEIHO,CHAXEN(I),0.,32)
C      oder CALL SYMBGR(ZXX,ZYY,ZSYHO,CHAXEN(I),0.,25)
42 CONTINUE
      IF (JWERT.EQ.3) THEN
          ZAHL = WSTREU
          ZXX = ZXI - ZEIHO*14.
          CALL NUMBER(ZXX,ZYY,ZEIHO,ZAHL,0.,1)
      ENDIF
C - - - - PHASENRAUMFAKTOR-KENNZEICHNUNG EINTRAGEN.
      IF (NPHAS.GT.0.AND.JWERT.NE.2) THEN
          ZYY = ZYY - ZEIHO*2.666
          ZXX = ZXI - ZEIHO*31.
          CALL SYMBLO(ZXX,ZYY,ZEIHO,CHAXEN(24),0.,32)
```

```
CALL SYMBL1(ZXX,ZYY,ZEIHO,CHAXEN(25),0.,32)
DO 43 I=1,NKURV
  ZYY = ZYY - ZEIHO*2.5
  ZXX = ZXI - ZEIHO*31.
  CALL SYMBLO(ZXX,ZYY,ZEIHO,CHAXEN(26),0.,32)
  CALL SYMBL1(ZXX,ZYY,ZEIHO,CHAXEN(27),0.,32)
  ZXX = ZXX + ZEIHO*8.
  CHA = 'K= '
  WRITE(CH(3:3),'(I1)') KK(I)
  CALL SYMBLO(ZXX,ZYY+ZEIHO*0.5,ZEIHO,
$           CHKURV(NST(I+NKURV)),0.,7)
  CALL SYMBLO(ZXX+ZEIHO*8.,ZYY,ZEIHO,CHA,0.,3)
43 CONTINUE
  ENDIF
RETURN
END
```



## 4.6 SPLINE

### Kubische Spline-Interpolation

Zunächst berechnet das Unterprogramm SPLINE den maximalen Abstand zwischen zwei Eingabe-Stützpunkten. Wenn dieser Wert größer als 0.5 ist (es sollte sich hier ja immer um Koordinaten in Inches handeln), berechnet das Unterprogramm für diesen Satz von Eingabe-Punkten einen Satz von Koeffizienten zur interpolierenden Auswertung von kubischen Splines. Mit diesen Koeffizienten werden dann zwischen je zwei ursprünglichen Punkten ND-1 neue Punkte berechnet. Die X-Koordinaten der Eingabepunkte müssen streng aufsteigend sein.

```
CALL SPLINE(NP,NZ,ZX,ZY,ZV,ZW)
```

#### EINGABEPARAMETER:

NP = Anzahl der Ausgangspunkte.  
NZ = Maximale Anzahl an neuen Punkten, Dimension von ZV und ZW im rufenden Programm.  
ZX(1:NP) = X-Koordinaten der Ausgangspunkte.  
ZY(1:NP) = Y-Koordinaten der Ausgangspunkte.

#### RÜCKGABEPARAMETER:

NZ = Anzahl der neuen Punkte.  
ZV(1:NZ) = X-Koordinaten der neuen Punkte.  
ZW(1:NZ) = Y-Koordinaten der neuen Punkte.

#### RESTLICHE VARIABLEN:

ABS = Fortran builtin function.  
AMAX1 = Fortran builtin function.  
CA = Zwischenvariable.  
CDF1 = Zwischenvariable.  
CDF3 = Zwischenvariable.  
CNP(1:NC) = Koeffizienten-Hilfsfeld.

CO(1:NI,1:NC) = Koeffizienten der Spline-Polynome.  
CTAU = Zwischenvariable.  
LL = Schleifen- und Zählvariable.  
LP = Schleifen- und Zählvariable.  
MINO = Fortran builtin function.  
N = Schleifen- und Zählvariable.  
NC = Konstanter Parameter 3. Zweite Dimension von CO.  
ND = Anzahl der Punkte, die aus einem Eingangspunkt berechnet werden.  
NI = Konstanter Parameter 1000. Erste Dimension von CO.  
NJ = Eingabewert von NZ.  
NP1 = NP - 1  
NP2 = NP - 2  
SQRT = Fortran builtin function.  
ZAB = Aktueller Abstand zwischen zwei Punkten.  
ZAM = Maximum des Abstands zwischen zwei Punkten.  
ZAS = Steigung der Gerade zwischen zwei Punkten.  
ZXD = Aktuelle Zunahme des Abszissenwertes.  
ZXDN = Abszissenschrittweite.

#### ABDRUCK DES UNTERPROGRAMMS SPLINE

```
C - - - Kubische-Spline-Interpolation bei zu wenigen Punkten.  
SUBROUTINE SPLINE(NP,NZ,ZX,ZY,ZV,ZW)  
PARAMETER (NI=1000,NC=3)  
DIMENSION CO(NI,NC),ZX(NP),ZY(NP),ZV(NP),ZW(NP),CNP(NC)  
C - - - - - BERECHNUNG DES ABSTANDSMAXIMUMS.  
ZAM = 5.E-7  
DO 340 N=2,NP  
IF (ZX(N)-ZX(N-1).LT.1.E-4.OR  
$ .ZY(N)-ZY(N-1).GT.1000.) THEN  
ZAM = 3000.  
ELSE  
ZAS = 1.E-4 + (ZY(N)-ZY(N-1))/(ZX(N)-ZX(N-1))  
ZAB = SQRT((ZX(N)-ZX(N-1))**2+(ZY(N)-ZY(N-1))**2)  
ZAM = AMAX1(ZAM,ABS(ZAS*ZAB))  
ENDIF  
340 CONTINUE
```

C - - - - - UMSPEICHERN, FALLS ZAM<1 ODER ZAM>3000.

```
NJ = NZ
NZ = NP
DO 341 N=1, NP
    ZV(N)=ZX(N)
    ZW(N)=ZY(N)
```

341 CONTINUE

```
IF (ZAM.GE.3000.) ZAM = 0.5
IF (NI.LT.NP1) ZAM = 0.5
IF (ZAM.GE.1.) THEN
```

C - - - - - BERECHNUNG DER SPLINE-KOEFFIZIENTEN.

```
NP1 = NP - 1
ND = 1 + ZAM
IF (NP*ND.GE.MINO(NI-2,NJ-2)) ND=MINO(NI-2,NJ-2)/NP
IF (ND.LE.0) ND = 1
IF (ND.EQ.1) GOTO 349
IF (NP.EQ.2) THEN
    IF (ZX(1).GE.ZX(2)) GOTO 346
    CO(1,1) = (ZY(2)-ZY(1)) / (ZX(2)-ZX(1))
    CO(1,2) = 0.
    CO(1,3) = 0.
    GOTO 346
ENDIF
```

```
DO 342 M=2, NP1
    CO(M,2) = ZX(M) - ZX(M-1)
    IF (CO(M,2).LE.0.) GOTO 346
    CO(M,3) = (ZY(M)-ZY(M-1)) / CO(M,2)
```

342 CONTINUE

```
CNP(2) = ZX(NP) - ZX(NP1)
IF (CNP(2).LE.0.) GOTO 346
CNP(3) = (ZY(NP)-ZY(NP1)) / CNP(2)
NP2 = NP - 2
IF (NP.LE.3) THEN
    CO(1,3) = CNP(2)
    CO(1,2) = CO(2,2) + CO(3,2)
    CO(1,1) = ((CO(2,2)+2.*CO(1,2))*CO(2,3)*CNP(2) +
+          CO(2,2)**2*CNP(3)) / CO(1,2)
```

ELSE

```
CO(1,3) = CO(3,2)
CO(1,2) = CO(2,2) + CO(3,2)
```

```
CO(1,1) = ((CO(2,2)+2.*CO(1,2))*CO(2,3)*CO(3,2) +
+ CO(2,2)**2*CO(3,3)) / CO(1,2)
DO 343 M=2,NP2
  CA = -CO(M+1,2) / CO(M-1,3)
  CO(M,1)=CA*CO(M-1,1)+3.*CO(M,2)*CO(M+1,3)+
+ 3.*CO(M+1,2)*CO(M,3)
  CO(M,3)=CA*CO(M-1,2)+2.*CO(M,2)+2.*CO(M+1,2)
343 CONTINUE
ENDIF
CA = -CNP(2) / CO(NP2,3)
CO(NP1,1)=CA*CO(NP2,1)+3.*CO(NP1,2)*CNP(3)+
+ 3.*CNP(2)*CO(NP1,3)
CO(NP1,3)=CA*CO(NP2,2)+2.*CO(NP1,2)+2.*CNP(2)
IF (NP.LE.3) THEN
  CNP(1) = 2. * CNP(3)
  CNP(3) = 1.
  CA = -1. / CO(NP1,3)
ELSE
  CA = CO(NP1,2) + CNP(2)
  CNP(1) = ((CNP(2)+2.*CA)*CNP(3)*CO(NP1,2)+
+ CNP(2)**2*(ZY(NP1)-ZY(NP-2))/CO(NP1,2))/CA
  CA = -CA / CO(NP1,3)
  CNP(3) = CO(NP1,2)
ENDIF
CNP(3) = CA*CO(NP1,2)+CNP(3)
CNP(1) = (CA*CO(NP1,1)+CNP(1)) / CNP(3)
CO(NP1,1) = (CO(NP1,1)-CO(NP1,2)*CNP(1)) / CO(NP1,3)
DO 344 LL=1,NP2
  N = NP1 - LL
  CO(N,1) = (CO(N,1)-CO(N,2)*CO(N+1,1)) / CO(N,3)
344 CONTINUE
DO 345 N=2,NP1
  CTAU = CO(N,2)
  CDF1 = (ZY(N)-ZY(N-1)) / CTAU
  CDF3 = CO(N-1,1) + CO(N,1)-2.*CDF1
  CO(N-1,2) = (CDF1-CO(N-1,1)-CDF3) / CTAU
  CO(N-1,3) = CDF3 / CTAU**2
345 CONTINUE
CTAU = CNP(2)
CDF1 = (ZY(NP)-ZY(NP1)) / CTAU
```

```
CDF3 = CO(NP1,1) + CNP(1)-2.*CDF1
CO(NP1,2) = (CDF1-CO(NP1,1)-CDF3) / CTAU
CO(NP1,3) = CDF3 / CTAU**2
346 CONTINUE
C - - - - ANZAHL DER PUNKTE UM DEN FAKTOR ND>1 ERHOEHEN.
  IF (ND.GE.2) THEN
    DO 348 LX=1,NP1
      LL = 1 + ND * (LX-1)
      ZV(LL) = ZX(LX)
      ZW(LL) = ZY(LX)
      IF (LX.GE.NP) GOTO 348
      ZXDN = (ZX(LX+1)-ZX(LX)) / ND
      DO 347 N=1,ND-1
        ZXD = ZXDN * N
        ZV(LL+N)=ZV(LL)+ZXD
        ZW(LL+N)=((CO(LX,3)*ZXD+CO(LX,2))*
          *
          ZXD+CO(LX,1))*ZXD+ZY(LX)
347 CONTINUE
348 CONTINUE
      NZ = 1 + ND * NP1
      ZV(NZ) = ZX(NP)
      ZW(NZ) = ZY(NP)
    ENDIF
  ENDIF
349 CONTINUE
  RETURN
  END
```

#### 4.7 STRIPU

##### Zeichnen von Kurven - gestrichelt etc.

Das Unterprogramm STRIPU benutzt die neuen Koordinaten eines vorhergehenden AEQUID-Aufrufs zum Zeichnen einer Kurve. Die Kurve kann - zur Zeit - entweder gestrichelt, punktiert, strichpunktiert oder strichdoppelpunktiert gemalt werden.

```
CALL AEQUID(NX,NL,ZABST,ZX,ZY,ZV,ZW)
CALL STRIPU(iabs(NT),NL,ZV,ZW)
```

##### EINGABEPARAMETER:

NT = Kennzeichnung des Darstellungstyps:

2 = stricheln.

3 = punktieren.

4 = strichpunktieren.

5 = strichdoppelpunktieren.

NL = Anzahl der Eingabepunkte, d.h. die Anzahl der benutzten Werte aus den Feldern ZV(i) und ZW(i).

ZV(1:NL) = Abszissen der Eingabepunkte (in Inches).

ZW(1:NL) = Ordinaten der Eingabepunkte (in Inches).

RÜCKGABEPARAMETER: KEINE

RESTLICHE VARIABLEN:

I = Hilfs-, Schleifen-, Zähl-Variable.  
ID = Zählung der Punkte. Maximal 1000-2 können mit einem  
Aufruf von LINE bearbeitet werden.  
LINE = Plotsubroutine (System).  
MOD = Fortran builtin function.  
PLOT = Plotsubroutine (System).  
ZCR(1:ID) = Abszissen der Punkte zum Zeichnen.  
ZDR(1:ID) = Ordinaten der Punkte zum Zeichnen.  
ZLEN = Korrekturgröße der Koordinaten.  
ZXA = Aktuelle Anfangsposition.  
ZXB = Aktuelle Endposition.  
ZXDEL = Korrekturgröße der Koordinaten.  
ZYA = Aktuelle Anfangsposition.  
ZYB = Aktuelle Endposition.  
ZYDEL = Korrekturgröße der Koordinaten.  
ZYY = Aktuelle Ordinatenposition.  
ZYZ = Grenze der Ordinatenposition.

ABDRUCK DES UNTERPROGRAMMS STRIPU

```
C - - - Kurven stricheln, punktieren, strichpunktieren, etc.
      SUBROUTINE STRIPU (NT,NL,ZV,ZW)
      DIMENSION ZV(NL),ZW(NL),ZCR(1000),ZDR(1000)
              ZLEN =0.005
              ZXDEL=0.005
              ZYDEL=0.005
              IF (NT.EQ.2) THEN
C - - - - - STRICHELN VON KURVEN MIT GLEICH GROSSEN STRICHEN.
              NL=NL-MOD(NL,2)
              DO 132 I=1,NL,2
                      CALL PLOT(ZV(I),ZW(I),3)
                      CALL PLOT(ZV(I+1),ZW(I+1),2)
132          CONTINUE
              ENDIF
              IF (NT.EQ.3) THEN
C - - - - - PUNKTIEREN VON KURVEN MIT GLEICHEN PUNKTABSTAENDEN.
              DO 135 I=1,NL
                      ZXA=ZV(I)-ZLEN
```

133

```
ZXB=ZV(I)+ZLEN
ZYY=ZW(I)+ZYDEL
ZYZ=ZW(I)-ZYDEL
ZYB=ZW(I)+ZLEN
ZYA=ZW(I)-ZLEN
ID=1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
CONTINUE
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 134
ZXB=ZXB+ZXDEL
ZXA=ZXA-ZXDEL
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 134
ZXA=ZXA-ZXDEL
ZXB=ZXB+ZXDEL
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
IF (ID.GE.1000-4) THEN
    ZCR(ID+1)=0.
    ZDR(ID+1)=0.
    ZCR(ID+2)=1.
```



```

      ZDR(ID+2)=1.
      CALL LINE(ZCR,ZDR, ID, 1, 0, 0)
      ZCR(1)=ZCR(ID)
      ZDR(1)=ZDR(ID)
      ID=1
      ENDIF
      GOTO 133
134  CONTINUE
      ZCR(ID+1)=0.
      ZDR(ID+1)=0.
      ZCR(ID+2)=1.
      ZDR(ID+2)=1.
      CALL LINE(ZCR,ZDR, ID, 1, 0, 0)
135  CONTINUE
      ENDIF
      IF (NT.EQ.4) THEN
C - - - - STRICH-PUNKTIEREN VON KURVEN MIT GLEICHEN ABSTAENDEN.
      IF (MOD(NL,3).EQ.1) NL=NL-1
      DO 138 I=1,NL,3
      CALL PLOT(ZV(I),ZW(I),3)
      CALL PLOT(ZV(I+1),ZW(I+1),2)
      ZXA=ZV(I+2)-ZLEN
      ZXB=ZV(I+2)+ZLEN
      ZYY=ZW(I+2)+ZYDEL
      ZYZ=ZW(I+2)-ZYDEL
      ZYB=ZW(I+2)+ZLEN
      ZYA=ZW(I+2)-ZLEN
      ID=1
      ZCR(ID)=ZXA
      ZDR(ID)=ZYY
      ID=ID+1
      ZCR(ID)=ZXB
      ZDR(ID)=ZYY
136  CONTINUE
      ZYY=ZYY-0.00999
      IF (ZYY.LT.ZYB) ZXDEL=0.0
      IF (ZYY.LT.ZYA) ZXDEL=-0.01
      IF (ZYY.LT.ZYZ) GOTO 137
      ZXB=ZXB+ZXDEL
      ZXA=ZXA-ZXDEL
```

```

ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 137
ZXA=ZXA-ZXDEL
ZXB=ZXB+ZXDEL
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
IF (ID.GE.1000-4) THEN
    ZCR(ID+1)=0.
    ZDR(ID+1)=0.
    ZCR(ID+2)=1.
    ZDR(ID+2)=1.
    CALL LINE(ZCR,ZDR, ID, 1, 0, 0)
    ZCR(1)=ZCR(ID)
    ZDR(1)=ZDR(ID)
    ID=1
    ENDIF
GOTO 136
137 CONTINUE
    ZCR(ID+1)=0.
    ZDR(ID+1)=0.
    ZCR(ID+2)=1.
    ZDR(ID+2)=1.
    CALL LINE(ZCR,ZDR, ID, 1, 0, 0)
138 CONTINUE
    ENDIF
IF (NT.EQ.5) THEN
C STRICH-DOPPELT-PUNKTIEREN VON KURVEN MIT GLEICHEN ABSTAENDEN.
    IF (MOD(NL,4).EQ.1) NL=NL-1
```

```
DO 144 I=1,NL,4
CALL PLOT(ZV(I),ZW(I),3)
CALL PLOT(ZV(I+1),ZW(I+1),2)
IF (I+2.GT.NL) GOTO 144
ZXA=ZV(I+2)-ZLEN
ZXB=ZV(I+2)+ZLEN
ZYY=ZW(I+2)+ZYDEL
ZYZ=ZW(I+2)-ZYDEL
ZYB=ZW(I+2)+ZLEN
ZYA=ZW(I+2)-ZLEN
ID=1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
CONTINUE
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 141
ZXB=ZXB+ZXDEL
ZXA=ZXA-ZXDEL
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 141
ZXA=ZXA-ZXDEL
ZXB=ZXB+ZXDEL
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
```

140

```
ZDR(ID)=ZYY
IF (ID.GE.1000-4) THEN
  ZCR(ID+1)=0.
  ZDR(ID+1)=0.
  ZCR(ID+2)=1.
  ZDR(ID+2)=1.
  CALL LINE(ZCR,ZDR, ID,1,0,0)
  ZCR(1)=ZCR(ID)
  ZDR(1)=ZDR(ID)
  ID=1
ENDIF
GOTO 140
141 CONTINUE
ZCR(ID+1)=0.
ZDR(ID+1)=0.
ZCR(ID+2)=1.
ZDR(ID+2)=1.
CALL LINE(ZCR,ZDR, ID,1,0,0)
IF (I+3.GT.NL) GOTO 144
ZXA=ZV(I+3)-ZLEN
ZXB=ZV(I+3)+ZLEN
ZYY=ZW(I+3)+ZYDEL
ZYZ=ZW(I+3)-ZYDEL
ZYB=ZW(I+3)+ZLEN
ZYA=ZW(I+3)-ZLEN
ID=1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
142 CONTINUE
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 143
ZXB=ZXB+ZXDEL
ZXA=ZXA-ZXDEL
ID=ID+1
ZCR(ID)=ZXB
```

```
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ZYY=ZYY-0.00999
IF (ZYY.LT.ZYB) ZXDEL=0.0
IF (ZYY.LT.ZYA) ZXDEL=-0.01
IF (ZYY.LT.ZYZ) GOTO 143
ZXA=ZXA-ZXDEL
ZXB=ZXB+ZXDEL
ID=ID+1
ZCR(ID)=ZXA
ZDR(ID)=ZYY
ID=ID+1
ZCR(ID)=ZXB
ZDR(ID)=ZYY
IF (ID.GE.1000-4) THEN
    ZCR(ID+1)=0.
    ZDR(ID+1)=0.
    ZCR(ID+2)=1.
    ZDR(ID+2)=1.
    CALL LINE(ZCR,ZDR,ID,1,0,0)
    ZCR(1)=ZCR(ID)
    ZDR(1)=ZDR(ID)
    ID=1
    ENDIF
GOTO 142
143 CONTINUE
ZCR(ID+1)=0.
ZDR(ID+1)=0.
ZCR(ID+2)=1.
ZDR(ID+2)=1.
CALL LINE(ZCR,ZDR,ID,1,0,0)
144 CONTINUE
ENDIF
RETURN
END
```

## 4.8 SYMBGR

### Kleine griechische Buchstaben

Das Unterprogramm SYMBGR malt Zeichenketten aus griechischen Buchstaben, die mit deutschen Buchstaben nach der folgenden Tabelle umschrieben worden sind:

1. ALPHA	a	9. JOTA	i	17. RHO	r
2. BETA	b	10. KAPPA	k	18. SIGMA	s
3. GAMMA	c	11. LAMBDA	l	19. TAU	t
4. DELTA	d	12. MY	m	20. YPSILON	u
5. EPSILON	e	13. NY	n	21. PHI	v
6. ZETA	f	14. XI	o	22. CHI	w
7. ETA	g	15. OMIKRON	p	23. PSI	x
8. THETA	h	16. PI	q	24. OMEGA	y
				25. THETB	z

Die Zeichenkettenvariable CTEXT darf nur *Leerzeichen* und die *Umschreibungszeichen* der kleinen griechischen Buchstaben enthalten.

```
CALL SYMBGR(ZXZ,ZYZ,ZPLHO,CTEXT,ZWI,LTEXT)
```

#### EINGABEPARAMETER:

ZXZ = Aktuelle Abszissenposition zum Malen des Textes.  
ZYZ = Aktuelle Ordinatenposition zum Malen des Textes.  
ZPLHO = Zeichengröße (in Inches).  
CTEXT\*(\*) = Zeichenkettenvariable mit den Umschreibungszeichen der kleinen griechischen Buchstaben.  
ZWI = Winkel in Grad, um den die Zeichenkette gedreht gemalt werden soll.  
LTEXT = Anzahl der Zeichen aus der Zeichenkette CTEXT, die beim Unterprogrammaufruf bearbeitet werden sollen.

RÜCKGABEPARAMETER: keine

**RESTLICHE VARIABLEN:**

ZSYM(0:LXY) = Standard-X-Koordinaten des Plotsymbols.  
ZSYM(0:LXY) = Standard-Y-Koordinaten des Plotsymbols.  
ZX(1:LXY+3) = Aktuelle X-Koordinaten zum Plotten des Symbols.  
ZY(1:LXY+3) = Aktuelle Y-Koordinaten zum Plotten des Symbols.  
CGRA\*25 = 'abcdefghijklmnopqrstuvwxy'  
CGRB\*25 = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'  
CGRC\*8(1:25) = Namen der griechischen Buchstaben.  
COS = Fortran builtin function.  
I = Hilfs-, Schleifen-, Zähl-Variable.  
L = Hilfs-, Schleifen-, Zähl-Variable.  
LINE = Plot-Subroutine (System).  
N = Hilfs-, Schleifen-, Zähl-Variable.  
NBU = 25, maximale Anzahl verschiedener Buchstaben.  
NN = Hilfs-, Schleifen-, Zähl-Variable.  
NXBU(1:25) = Anzahl der Punkte für den Linienzug des Buchstabens.  
SIN = Fortran builtin function.  
ZFAKT = Fester Größenfaktor für die richtige Textgröße.  
ZIP180 = Fester Größenfaktor für die richtige Textgröße.  
ZW = Winkelwert ZWI im Bogenmaß.  
ZX(1:27) = aktuelle X-Koordinaten zum Plotten.  
ZXB(0:25,1:25) = X-Koordinaten der Punkte für den Linienzug  
des betreffenden Buchstabens. Im 0. Element steht die  
Anzahl der Punkte für den Linienzug des Buchstabens.  
ZXF = Hilfsvariable für die X-Koordinate.  
ZXX = Aktuelle Abszissenposition zum Plotten.  
ZY(1:27) = Aktuelle Y-Koordinaten zum Plotten.  
ZYB(0:25,1:25) = Y-Koordinaten der Punkte für den Linienzug  
de betreffenden Buchstabens.  
ZYF = Hilfsvariable für die Y-Koordinate.  
ZYY = aktuelle Ordinatenposition zum Plotten.

ABDRUCK DES PLOT-UNTERPROGRAMMS SYMBGR

C - - - Kleine griechische Buchstaben malen.

```
SUBROUTINE SYMBGR(ZXZ,ZYZ,ZPLHO,CTEXT,ZWI,LTEXT)
PARAMETER (NBU=25,ZFAKT=0.13,ZPI180=1.74533E-2)
CHARACTER CTEXT*(*),CGRA*25,CGRB*25,CGRC(25)*8
DIMENSION ZXBU(0:25,25),ZYBU(25,25),ZX(27),ZY(27),NXBU(25)
DATA CGRA/'abcdefghijklmnopqrstuvwxyz'/
DATA CGRB/'ABCDEFGHIJKLMNopqrstuvwxyz'/
DATA CGRC
$ /'ALPHA  ','BETA  ','GAMMA  ','DELTA  ','EPSILON  ',
$ 'ZETA  ','ETA  ','THETA  ','JOTA  ','KAPPA  ',
$ 'LAMBDA  ','MY  ','NY  ','XI  ','OMIKRON  ',
$ 'PI  ','RHO  ','SIGMA  ','TAU  ','YPSILON  ',
$ 'PHI  ','CHI  ','PSI  ','OMEGA  ','THETB  '/
DATA NXBU/22,22,15,23,17,19,18,23, 7,15, 9,15,
$ 12,25,20,14,22,21,10,18,22,14,18,23,25/
```

C - - - - - ALPHA

```
DATA (ZXBU(I, 1),I=0,25)/ 22.02200,
$ 5.00000, 3.91318, 3.04341, 1.95659, 1.08682, 0.86977,
$ 0.43462, 0.21757, 0.0, 0.0, 0.21757, 0.65220,
$ 1.30439, 1.95659, 2.39121, 3.04341, 3.47855, 3.69561,
$ 3.91318, 4.13075, 4.34780, 4.78295, 3*0./
DATA (ZYBU(I, 1),I=1,25)/
$ 4.90000, 2.30585, 1.15313, 0.28828, 0.0, 0.0,
$ 0.28828, 0.57656, 1.44101, 2.30585, 3.17071, 4.03515,
$ 4.61172, 4.90000, 4.90000, 4.61172, 4.03515, 3.45899,
$ 2.30585, 1.44101, 0.86485, 0.0, 3*0./
```

C - - - - - BETA

```
DATA (ZXBU(I, 2),I=0,25)/ 22.02200,
$ 0.0, 0.62485, 1.25010, 1.87495, 2.18757, 2.81242,
$ 3.75030, 4.06252, 4.68777, 5.00000, 4.68777, 4.06252,
$ 3.12505, 2.18757, 3.12505, 3.75030, 4.06252, 4.37515,
$ 4.06252, 3.43767, 2.50020, 0.93748, 3*0./
DATA (ZYBU(I, 2),I=1,25)/
$ 0.0, 2.98667, 4.48000, 5.67466, 6.27200, 6.86933,
$ 7.46666, 7.46666, 7.16800, 6.57066, 5.37600, 4.48000,
$ 3.88267, 3.58400, 3.58400, 3.28533, 2.98667, 2.09067,
$ 0.89600, 0.29867, 0.0, 0.0, 3*0./
```



C - - - - - GAMMA

DATA (ZXBU(I, 3), I=0, 25)/ 15.01501,  
\$ 1.05273, 2.10545, 2.63136, 2.89454, 2.63136, 2.10545,  
\$ 1.05273, 0.52636, 0.26318, 0.0, 0.52636, 1.05273,  
\$ 2.10545, 3.68409, 5.00000, 10\*0./  
DATA (ZYBU(I, 3), I=1, 25)/  
\$ 4.90000, 3.33851, 2.03705, 0.73592, -0.82556, -1.34628,  
\$ -1.86667, -1.86667, -1.60631, -0.82556, 0.21553, 0.99628,  
\$ 2.29741, 3.59887, 4.90000, 10\*0./

C - - - - - DELTA

DATA (ZXBU(I, 4), I=0, 25)/ 23.02299,  
\$ 4.47368, 2.89474, 1.57895, 0.78947, 0.26316, 0.0,  
\$ 0.26316, 0.52632, 1.05263, 2.10526, 3.42105, 4.21053,  
\$ 4.73684, 5.00000, 4.73684, 4.47368, 3.68421, 2.10526,  
\$ 1.84210, 1.84210, 2.63158, 2.89474, 3.94737, 2\*0./  
DATA (ZYBU(I, 4), I=1, 25)/  
\$ 4.13539, 4.02051, 3.56103, 3.10153, 2.41231, 1.83795,  
\$ 0.57436, 0.34461, 0.11487, 0.0, 0.22973, 0.68923,  
\$ 1.37847, 2.41231, 3.67589, 4.13539, 4.82461, 6.20306,  
\$ 6.54769, 7.12205, 7.46666, 7.46666, 7.12205, 2\*0./

C - - - - - EPSILON

DATA (ZXBU(I, 5), I=0, 25)/ 17.01700,  
\$ 5.00000, 3.88889, 2.22222, 1.38889, 0.55556, 0.27777,  
\$ 0.0, 4.70000, 0.0, 0.27777, 0.83332, 1.11111,  
\$ 1.66666, 1.94443, 2.50000, 3.33332, 4.44443, 8\*0./  
DATA (ZYBU(I, 5), I=1, 25)/  
\$ 4.90000, 4.90000, 4.63513, 4.23784, 3.57568, 3.17837,  
\$ 2.62057, 2.62057, 2.62057, 1.45676, 0.79459, 0.52973,  
\$ 0.26487, 0.13243, 0.0, 0.13243, 0.39729, 8\*0./

C - - - - - ZETA

DATA (ZXBU(I, 6), I=0, 25)/ 19.01900,  
\$ 1.66667, 4.16667, 5.00000, 4.16667, 3.33333, 2.50000,  
\$ 1.66666, 0.83333, 0.0, 0.0, 0.41000, 0.83333,  
\$ 3.33333, 3.75432, 4.16667, 3.75432, 3.33333, 2.50000,  
\$ 1.66667, 6\*0./  
DATA (ZYBU(I, 6), I=1, 25)/  
\$ 7.46666, 6.91774, 6.91774, 6.91774, 6.70667, 6.36838,  
\$ 5.46666, 4.17228, 1.97663, 1.42727, 1.04000, 0.87835,  
\$ 0.87835, 0.51111, -0.21948, -0.94815, -1.31775, -1.86667,  
\$ -1.86667, 6\*0./

C - - - - - ETA  
DATA (ZXBU(I, 7), I=0,25)/ 18.01801,  
\$ 0.0 , 0.43478, 0.86957, 1.08696, 1.52174, 1.72222,  
\$ 1.52174, 0.65217, 1.52174, 2.17391, 2.82609, 3.69565,  
\$ 4.13043, 4.34783, 4.78261, 5.00000, 5.00000, 3.47826,7\*0./  
DATA (ZYBU(I, 7), I=1,25)/  
\$ 4.46808, 4.75603, 4.90000, 4.90000, 4.61205, 3.82221,  
\$ 3.02837, 0.29291, 3.02837, 3.89220, 4.46808, 4.75603,  
\$ 4.90000, 4.90000, 4.61205, 4.18015, 3.89220, -1.86667,7\*0./

C - - - - - THETA  
DATA (ZXBU(I, 8), I=0,25)/ 23.02299,  
\$ 5.00000, 0.0 , 0.0 , 0.25000, 1.00000, 1.75000,  
\$ 2.55000, 2.95000, 3.75000, 4.50000, 5.00000, 5.00000,  
\$ 4.75000, 4.25000, 3.75000, 3.25000, 2.50000, 2.00000,  
\$ 1.25000, 0.75000, 0.25000, 0.0 , 0.0 , 2\*0./  
DATA (ZYBU(I, 8), I=1,25)/  
\$ 3.22493, 3.22493, 3.65591, 4.73119, 6.02150, 6.66666,  
\$ 7.00000, 7.00000, 6.66666, 5.80646, 4.30107, 3.44087,  
\$ 2.36559, 1.29032, 0.64516, 0.21505, 0.0 , 0.0 ,  
\$ 0.21505, 0.86021, 1.72043, 2.79571, 3.22493, 2\*0./

C - - - - - JOTA  
DATA (ZXBU(I, 9), I=0,25)/ 7.00700,  
\$ 2.64717, 1.71717, 1.90317, 2.27517, 2.64717, 3.01917,  
\$ 3.39117, 18\*0./  
DATA (ZYBU(I, 9), I=1,25)/  
\$ 4.90000, 0.90263, 0.51579, 0.12895, 0.0 , 0.25789,  
\$ 0.64473, 18\*0./

C - - - - - KAPPA  
DATA (ZXBU(I, 10), I=0,25)/ 15.01501,  
\$ 4.77273, 4.31818, 4.09091, 3.86364, 2.95454, 2.50000,  
\$ 1.81818, 0.90909, 0.0 , 1.59091, 0.90909, 2.27273,  
\$ 2.72727, 3.18182, 5.00000, 10\*0./  
DATA (ZYBU(I, 10), I=1,25)/  
\$ 0.13611, 0.0 , 0.0 , 0.27223, 2.17777, 2.58611,  
\$ 2.85833, 2.85833, 0.0 , 4.90000, 2.85833, 2.85833,  
\$ 2.99444, 3.26667, 4.90000, 10\*0./

C - - - - - LAMBDA  
DATA (ZXBU(I, 11), I=0,25)/ 9.00900,  
\$ 1.00000, 1.50000, 2.00000, 2.00000, 2.50000, 0.0 ,  
\$ 2.50000, 3.00000, 4.20000, 16\*0./

DATA (ZYBU(I,11),I=1,25)/

\$ 7.46666, 7.46666, 7.07368, 7.07368, 4.32280, 0.0 ,  
\$ 4.32280, 1.96491, 0.0 , 16\*0./

C - - - - - MY

DATA (ZXBU(I,12),I=0,25)/ 15.01501,

\$ 0.0 , 0.92593, 1.48148, 0.92593, 1.48148, 2.03704,  
\$ 2.22222, 2.77778, 3.70370, 4.25926, 3.70370, 3.88889,  
\$ 4.25926, 4.44444, 5.00000, 10\*0./

DATA (ZYBU(I,12),I=1,25)/

\$ -1.86667, 1.35555, 4.90000, 1.35555, 0.22779, 0.06667,  
\$ 0.06667, 0.38888, 1.35555, 4.90000, 1.35555, 0.22779,  
\$ 0.06667, 0.06667, 0.38888, 10\*0./

C - - - - - NY

DATA (ZXBU(I,13),I=0,25)/ 12.01200,

\$ 0.0 , 0.71429, 1.42857, 1.42857, 0.71429, 0.0 ,  
\$ 2.14286, 4.00571, 4.28571, 3.57143, 4.28571, 5.00000, 13\*0./

DATA (ZYBU(I,13),I=1,25)/

\$ 4.90000, 4.45455, 3.56364, 2.67272, 1.33636, 0.0 ,  
\$ 1.33636, 3.56364, 4.45455, 4.90000, 4.45455, 4.45455, 13\*0./

C - - - - - XI

DATA (ZXBU(I,14),I=0,25)/ 25.02501,

\$ 1.42857, 2.38095, 5.00000, 3.09524, 1.66667, 0.95238,  
\$ 0.71429, 0.95238, 1.66667, 4.28571, 2.61905, 1.19048,  
\$ 0.23810, 0.0 , 0.23810, 0.95238, 1.90476, 2.85714,  
\$ 3.80952, 4.28571, 4.28571, 3.80952, 3.09524, 2.61905,  
\$ 1.90476/

DATA (ZYBU(I,14),I=1,25)/

\$ 7.46666, 6.95525, 6.69954, 6.57169, 5.80456, 5.03744,  
\$ 4.39817, 4.01461, 3.50320, 3.37535, 3.24749, 2.48036,  
\$ 1.71324, 0.94612, 0.30685, -0.20457, -0.46028, -0.58813,  
\$ -0.58813, -0.84384, -0.97169, -1.48311, -1.73881, -1.86667,  
\$ -1.73881/

C - - - - - OMIKRON

DATA (ZXBU(I,15),I=0,25)/ 20.02000,

\$ 3.75000, 4.50000, 5.00000, 5.00000, 4.75000, 4.25000,  
\$ 3.75000, 3.25000, 2.50000, 2.00000, 1.25000, 0.75000,  
\$ 0.25000, 0.0 , 0.0 , 0.25000, 1.00000, 1.75000,  
\$ 2.75000, 3.75000, 5\*0./

DATA (ZYBU(I,15),I=1,25)/

\$ 4.60303, 4.00909, 2.96969, 2.37576, 1.63333, 0.89091,  
\$ 0.44545, 0.14848, 0.0, 0.0, 0.14848, 0.59393,  
\$ 1.18788, 1.93031, 2.52424, 3.26667, 4.15757, 4.60303,  
\$ 4.90000, 4.60303, 5\*0./

C - - - - - PI

DATA (ZXBU(I,16),I=0,25)/ 14.01401,

\$ 0.0, 0.51724, 1.03448, 1.72414, 1.20690, 1.72414,  
\$ 3.27586, 3.27586, 3.44827, 3.79310, 3.44827, 3.27586,  
\$ 3.27586, 5.00000, 11\*0./

DATA (ZYBU(I,16),I=1,25)/

\$ 5.51108, 6.04441, 6.22218, 6.22218, 0.0, 6.22218,  
\$ 6.22218, 0.53333, 0.17777, 0.0, 0.17777, 0.53333,  
\$ 6.22218, 6.22218, 11\*0./

C - - - - - RHO

DATA (ZXBU(I,17),I=0,25)/ 22.02200,

\$ 0.45450, 1.36349, 1.81798, 2.72752, 3.63651, 3.63651,  
\$ 3.18202, 2.27248, 1.81798, 0.90899, 0.45450, 0.0,  
\$ 0.0, 0.90899, 1.81798, 3.18202, 4.09101, 5.00000,  
\$ 5.00000, 4.09101, 3.27776, 1.81798, 3\*0./

DATA (ZYBU(I,17),I=1,25)/

\$ -1.62559, -1.86667, -1.86667, -1.62559, -0.90237, -0.66545,  
\$ 0.29883, 0.77973, 1.12305, 1.46637, 1.80969, 2.49633,  
\$ 3.52672, 4.55668, 4.90000, 4.90000, 4.55668, 3.52672,  
\$ 2.49633, 1.80969, 1.46637, 1.12305, 3\*0./

C - - - - - SIGMA

DATA (ZXBU(I,18),I=0,25)/ 21.02100,

\$ 3.12500, 3.75000, 4.16667, 4.16667, 3.95833, 3.54167,  
\$ 3.12500, 2.70833, 2.08333, 1.66667, 1.04167, 0.62500,  
\$ 0.20833, 0.0, 0.0, 0.20833, 0.83333, 1.45833,  
\$ 2.29167, 3.95833, 5.00000, 4\*0./

DATA (ZYBU(I,18),I=1,25)/

\$ 4.46764, 3.89117, 2.88236, 2.30588, 1.58529, 0.86471,  
\$ 0.43236, 0.14412, 0.0, 0.0, 0.14412, 0.57647,  
\$ 1.15295, 1.87353, 2.45000, 3.17059, 4.03529, 4.46764,  
\$ 4.75588, 4.61176, 4.90000, 4\*0./

C - - - - - TAU

DATA (ZXBU(I,19),I=0,25)/ 10.01000,  
\$ 0.0 , 0.20833, 0.62500, 4.58333, 5.00000, 4.58333,  
\$ 2.50000, 1.45833, 1.25000, 0.83333, 15\*0./  
DATA (ZYBU(I,19),I=1,25)/  
\$ 4.34000, 4.62000, 4.76000, 4.76000, 4.90000, 4.76000,  
\$ 4.76000, 0.42000, 0.14000, 0.0 , 15\*0./

C - - - - - YPSILON

DATA (ZXBU(I,20),I=0,25)/ 18.01801,  
\$ 0.0 , 0.21739, 1.08696, 1.30435, 0.65217, 0.65217,  
\$ 0.86957, 1.30435, 1.95652, 2.82609, 3.47826, 4.34783,  
\$ 4.78261, 5.00000, 5.00000, 4.78261, 4.56522, 4.13043,7\*0./  
DATA (ZYBU(I,20),I=1,25)/  
\$ 4.90000, 4.76388, 4.90000, 4.35556, 1.63333, 1.36111,  
\$ 0.68056, 0.13611, 0.0 , 0.13611, 0.68056, 1.63333,  
\$ 2.58611, 3.26667, 3.53888, 4.21944, 4.62777, 4.90000,7\*0./

C - - - - - PHI

DATA (ZXBU(I,21),I=0,25)/ 22.02200,  
\$ 1.40625, 0.93750, 0.31250, 0.0 , 0.0 , 0.15625,  
\$ 0.46875, 1.09375, 2.34375, 3.12500, 4.06250, 4.68750,  
\$ 5.00000, 5.00000, 4.84375, 4.53125, 3.75000, 3.12500,  
\$ 2.65625, 2.03125, 1.56250, 1.09375, 3\*0./  
DATA (ZYBU(I,21),I=1,25)/  
\$ 4.63975, 4.37949, 3.59871, 2.81795, 2.29743, 1.64679,  
\$ 1.12628, 0.73589, 0.60576, 0.86603, 1.38653, 2.16731,  
\$ 3.07820, 3.59871, 4.24936, 4.63975, 4.90000, 4.63975,  
\$ 4.11924, 2.55769, 0.60576, -1.86667, 3\*0./

C - - - - - CHI

DATA (ZXBU(I,22),I=0,25)/ 14.01401,  
\$ 0.0 , 0.25000, 0.62500, 0.75000, 1.00000, 2.50000,  
\$ 0.22727, 4.77273, 2.50000, 4.00000, 4.25000, 4.37500,  
\$ 4.75000, 5.00000, 11\*0./  
DATA (ZYBU(I,22),I=1,25)/  
\$ 4.22333, 4.56167, 4.90000, 4.90000, 4.56167, 1.51667,  
\$ -1.55912, 4.59240, 1.51667, -1.52833, -1.86667, -1.86667,  
\$ -1.52833, -1.19000, 11\*0./

C - - - - - PSI

```
DATA (ZXBU(I,23),I=0,25)/ 18.01801,
$ 0.0 , 0.30303, 0.60606, 0.60606, 0.30303, 0.45455,
$ 1.06061, 1.81818, 1.36364, 3.03030, 1.81818, 2.42424,
$ 2.87879, 3.63636, 4.39394, 4.69697, 5.00000, 5.00000,7*0./
DATA (ZYBU(I,23),I=1,25)/
$ 4.62600, 4.49550, 4.10400, 3.84300, 2.14650, 1.10251,
$ 0.45000, 0.18900,-2.16000, 5.67000, 0.18900, 0.18900,
$ 0.45000, 1.23300, 2.53800, 3.19049, 4.10400, 4.49550,7*0./
```

C - - - - - OMEGA

```
DATA (ZXBU(I,24),I=0,25)/ 23.02299,
$ 1.21212, 0.75758, 0.30303, 0.0 , 0.0 , 0.30303,
$ 0.75758, 1.21212, 1.81818, 2.27273, 2.57576, 2.72727,
$ 2.72727, 3.03030, 3.03030, 3.63636, 4.09091, 4.54545,
$ 5.00000, 5.00000, 4.84848, 4.69697, 4.24242, 2*0./
DATA (ZYBU(I,24),I=1,25)/
$ 4.90000, 4.48000, 3.64000, 2.65333, 1.68000, 0.70000,
$ 0.14000, 0.0 , 0.28000, 0.70000, 1.68000, 3.08000,
$ 0.84000, 0.28000, 0.28000, 0.0 , 0.14000, 0.70000,
$ 1.68000, 2.65333, 3.64000, 4.34000, 4.90000, 2*0./
```

C - - - - - THETB

```
DATA (ZXBU(I,25),I=0,25)/ 25.02501,
$ 0.0 , 0.46875, 0.62500, 0.88125, 0.93750, 0.31250,
$ 0.31250, 0.62500, 0.93750, 1.40625, 1.87500, 2.34375,
$ 3.43750, 4.06250, 4.37500, 4.06250, 3.75000, 3.28125,
$ 2.81250, 2.30000, 2.18750, 2.50000, 3.28125, 4.37500,
$ 5.00000/
DATA (ZYBU(I,25),I=1,25)/
$ 3.75000, 3.95833, 3.95833, 3.74107, 3.54167, 1.77084,
$ 1.25000, 0.41667, 0.10417, 0.0 , 0.10417, 0.41667,
$ 1.66667, 2.91667, 4.47917, 5.93750, 6.45833, 6.66667,
$ 6.45833, 5.77142, 5.20655, 4.47917, 3.64584, 3.12500,
$ 3.02084/
```

C - - - - -

```
ZYY = ZYZ
DO 59 L=1,LTEXT
ZXX = ZXZ + ZPLHO*(L-1)
IF (CTEXT(L:L).EQ.' ') GOTO 59
```

```
DO 55 N=1,NBU
  IF (CTEXT(L:L).EQ.CGRB(N:N).OR.
$   CTEXT(L:L).EQ.CGRA(N:N)) THEN
    NN=NXBU(N)
    DO 50 I=1,NN
      IF (ZWI.EQ.0.) THEN
        ZXF = ZXBU(I,N)
        ZYF = ZYBU(I,N)
      ELSE
        ZW = ZPI180 * ZWI
        ZXF = ZXBU(I,N)*COS(ZW) - ZYBU(I,N)*SIN(ZW)
        ZYF = ZXBU(I,N)*SIN(ZW) + ZYBU(I,N)*COS(ZW)
      ENDIF
      ZX(I) = ZXX + ZPLHO * ZFAKT * ZXF
      ZY(I) = ZYY + ZPLHO * ZFAKT * ZYF
50    CONTINUE
      ZX(NN+1)=0.
      ZY(NN+1)=0.
      ZX(NN+2)=1.
      ZY(NN+2)=1.
      CALL LINE(ZX,ZY,NN,1,0,0)
    ENDIF
55  CONTINUE
59  CONTINUE
    RETURN
    END
```

#### 4.9 ZABSCH

##### Kurve oben/unten abschneiden

Das Unterprogramm ZABSCH korrigiert die aus dem vorgegeben Achsenrahmen nach oben und/oder nach unten herauslaufenden Kurventeile. Bei Bedarf werden durch lineare Interpolation Zwischenpunkte auf dem Achsenrahmen berechnet. Die neue Kurve ist dann teilweise mit Achsenlinien identisch.

```
CALL ZABSCH(NP, ZYI, ZXX, ZYY, ZXXA, ZYYA, ZX, ZY)
```

##### EINGABEPARAMETER:

NP = Aktuelle Anzahl an neuen Kurvenpunkten.  
ZYI = Länge der Ordinate (in Inches).  
ZXX = X-Koordinate des aktuellen alten Stützpunktes.  
ZYY = Y-Koordinate des aktuellen alten Stützpunktes.  
ZXXA = X-Koordinate letzten neuen Stützpunktes.  
ZYYA = Y-Koordinate letzten neuen Stützpunktes.  
ZX(1:NP) = X-Koordinaten der bisher bestimmten neuen Punkte.  
ZY(1:NP) = Y-Koordinaten der bisher bestimmten neuen Punkte.

##### RÜCKGABEPARAMETER:

NP = Aktuelle Anzahl an neuen Kurvenpunkten.  
ZX(1:NP) = X-Koordinaten der bisher bestimmten neuen Punkte.  
ZY(1:NP) = Y-Koordinaten der bisher bestimmten neuen Punkte.

RESTLICHE VARIABLEN: keine





```
                ZY(NP)=0.
            ENDIF
        ENDIF
    ENDIF
ELSE
C - - - - - REGULAERE PUNKTE - - - - -
    IF (ZYY.LE.ZYI*1.0001) THEN
        IF (NP.GT.0) THEN
C - - NP > 0
            IF (ZYAA.LT.0.) THEN
C - - - ZYAA < 0
                NP=NP+1
                ZX(NP)=ZXXA-ZYAA*(ZXX-ZXXA)/(ZYY-ZYAA)
                ZY(NP)=0.
                NP=NP+1
                ZX(NP)=ZXX
                ZY(NP)=ZYY
            ELSE
                IF (ZYAA.GT.ZYI*1.0001) THEN
C - - - - ZYAA > ZYI
                    NP=NP+1
                    ZX(NP)=ZXXA+(ZYI-ZYAA)*(ZXX-ZXXA)/(ZYY-ZYAA)
                    ZY(NP)=ZYI
                ENDIF
            C - - - - 0 =< ZYAA <= ZYI ODER ZYAA > ZYI
                NP=NP+1
                ZX(NP)=ZXX
                ZY(NP)=ZYY
            ENDIF
        ELSE
C - - NP = 0
            IF (ZYAA.LT.0.) THEN
C - - - ZYAA < 0
                NP=NP+1
                ZX(NP)=ZXXA+ZYAA*(ZXXA-ZXX)/(ZYAA-ZYY)
                ZY(NP)=0.
                NP=NP+1
                ZX(NP)=ZXX
                ZY(NP)=ZYY
            
```

```

ELSE
C - - - - 0 =< ZYYA
      NP=NP+1
      ZX(NP)=ZXX
      ZY(NP)=ZYY
      ENDIF
ENDIF
ELSE
C - - - - - NACH OBEN ABSCHNEIDEN - - - - -
      IF (NP.GT.O) THEN
C - - NP > 0
      IF (ZYYA.LT.O.) THEN
C - - - ZYYA < 0
      NP=NP+1
      ZX(NP)=ZXXA+ZYYA*(ZXXA-ZXX)/(ZYYA-ZYY)
      ZY(NP)=0.
      NP=NP+1
      ZX(NP)=ZXXA+(ZYI-ZYYA)*(ZXX-ZXXA)/(ZYY-ZYYA)
      ZY(NP)=ZYI
      ELSE
      IF (ZYYA.LE.ZYI*1.0001) THEN
C - - - - 0 =< ZYYA <= ZYI
      NP=NP+1
      ZX(NP)=ZXXA+(ZYI-ZYYA)*(ZXX-ZXXA)/(ZYY-ZYYA)
      ZY(NP)=ZYI
      ELSE
C - - - - ZYYA > ZYI
      NP=NP+1
      ZX(NP)=ZXX
      ZY(NP)=ZYI
      ENDIF
ENDIF
ELSE
C - - NP = 0
      IF (ZYYA.LT.O.) THEN
C - - - ZYYA < 0
      NP=NP+1
      ZX(NP)=ZXXA+ZYYA*(ZXXA-ZXX)/(ZYYA-ZYY)
      ZY(NP)=0.
      NP=NP+1
```

```
ZX(NP)=ZXXA+(ZYI-ZYYA)*(ZXX-ZXXA)/(ZYY-ZYYA)  
ZY(NP)=ZYI
```

```
ELSE
```

```
C - - - ZYYA >= 0
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
RETURN
```

```
END
```

#### 4.10 ZXYAXN

##### Achsenlinien mit Strichelungen

Das Unterprogramm ZXYAXN zeichnet mit den vorhandenen Plotparametern den Achsenrahmen und eventuelle "Null"-Linien, d.h. die Geraden  $X=0$  und/oder  $Y=0$ , die nicht mit einer Linie auf dem Achsenrahmen zusammenfallen. Ferner werden an alle Achsenlinien die entsprechenden Strichelungen gezeichnet.

```
CALL ZXYAXN  
      (LXL,LYL,ZXI,ZYI,ZXMI,ZYMI,ZXIB,ZYIB,ZDXI,ZDYI,ZSTHO)
```

##### EINGABEPARAMETER:

LXL = Anzahl der Markierungspositionen auf der X-Achse.  
LYL = Anzahl der Markierungspositionen auf der Y-Achse.  
ZXI = Länge der Abszisse (Inches).  
ZYI = Länge der Ordinate (Inches).  
ZXMI = Minimum auf der X-Achse.  
ZXIB = Beschriftungseinheit beim Beziffern der X-Achse.  
ZDXI = Abstand zwischen zwei Marken auf der X-Achse (Inches).  
ZYMI = Minimum auf der Y-Achse.  
ZYIB = Beschriftungseinheit beim Beziffern der Y-Achse.  
ZDYI = Abstand zwischen zwei Marken auf der X-Achse (Inches).  
ZSTHO = Strichlänge beim Stricheln der Achse (Inches).

RÜCKGABEPARAMETER: keine

##### RESTLICHE VARIABLEN:

ABS = Fortran builtin function.  
L = Schleifen- und Zählvariable.  
LINE = Plot-Subroutine (System).  
PLOT = Plot-Subroutine (System).  
ZSTH3 =  $2./3.*ZSTHO$  konstant.

ZXR(1:7) = Abszissen für den Achsenrahmen.  
ZXX = Aktuelle X-Position beim Plotten.  
ZXR(1:7) = Ordinaten für den Achsenrahmen.  
ZYY = Aktuelle Y-Position beim Plotten.

ABDRUCK DES UNTERPROGRAMMS ZXYAXN

```
C - - - Achsenlinien und Strichelungen malen - - - - -
  SUBROUTINE ZXYAXN
    $ (LXL,LYL,ZXI,ZYI,ZXMI,ZYMI,ZXIB,ZYIB,ZDXI,ZDYI,ZSTHO)
    DIMENSION ZXR(7),ZXR(7)
    DATA ZXR/6*0.,1./,ZXR/6*0.,1./
    ZSTH3=ZSTHO*0.666
C-----ACHSENRAHMEN ZEICHNEN - - - - -
  ZXR(2) = ZXI
  ZXR(3) = ZXI
  ZYR(3) = ZYI
  ZYR(4) = ZYI
  CALL LINE(ZXR,ZYR,5,1,0,0)
C-----UNTERE ABSZISSE STRICHELN- - - - -
  ZXX=0.0
  DO 323 L=1,LXL
    CALL PLOT(ZXX,0.0,3)
    CALL PLOT(ZXX,-ZSTHO,2)
    ZXX=ZXX+ZDXI
  323 CONTINUE
C-----RECHTE ORDINATE STRICHELN- - - - -
  ZYY=0.0
  DO 324 L=1,LYL
    CALL PLOT(ZXI,ZYY,3)
    CALL PLOT(ZXI+ZSTHO,ZYY,2)
    ZYY=ZYY+ZDYI
  324 CONTINUE
C-----ZUSAETZLICHE ORDINATE STRICHELN- - - - -
  IF (ZXMI.LT.-1.E-4) THEN
    ZXX=ZDXI/ZXIB*ABS(ZXMI)
    CALL PLOT(ZXX,ZYI,3)
    CALL PLOT(ZXX,0.0,2)
    ZYY=ZDYI
  DO 325 L=2,LYL-1
```

```
        CALL PLOT(ZXX-ZSTH3,ZYY,3)
        CALL PLOT(ZXX+ZSTH3,ZYY,2)
        ZYY=ZYY+ZDYI
325     CONTINUE
        ENDIF
C-----LINKE ORDINATE STRICHELN - - - - -
        ZYY=0.0
        DO 326 L=1,LYL
            CALL PLOT(0.0,ZYY,3)
            CALL PLOT(-ZSTHO,ZYY,2)
            ZYY=ZYY+ZDYI
326     CONTINUE
C-----OBERE ABSZISSE STRICHELN - - - - -
        ZXX=0.0
        DO 327 L=1,LXL
            CALL PLOT(ZXX,ZYI,3)
            CALL PLOT(ZXX,ZYI+ZSTHO,2)
            ZXX=ZXX+ZDXI
327     CONTINUE
C-----ZUSAETZLICHE ABSZISSE STRICHELN- - - - -
        IF (ZYMI.LT.-1.E-4) THEN
            ZYY=ZDYI/ZYIB*ABS(ZYMI)
            CALL PLOT(ZXI,ZYY,3)
            CALL PLOT(0.0,ZYY,2)
            ZXX=ZDXI
            DO 328 L=2,LXL-1
                CALL PLOT(ZXX,ZYY-ZSTH3,3)
                CALL PLOT(ZXX,ZYY+ZSTH3,2)
                ZXX=ZXX+ZDXI
328     CONTINUE
        ENDIF
        RETURN
        END
```

#### 4.11 ZXYIBE

##### Beschriftungseinheit

Das Unterprogramm ZXYIBE berechnet zunächst einen Schätzwert für die Beschriftungseinheit der betreffenden Achse, wenn der Parameter LXBE im rufenden Programmsegment nicht negativ gesetzt wurde. Wenn LXBE schon negativ ist, wird der dann ebenfalls im rufenden Programmsegment zu definierende Wert für ZXIB als Startwert verwendet. Nun wird von ZXIB die 10er-Potenz bestimmt und die übrigbleibende "Mantisse" auf 1, 2, oder 5 aufgerundet. Danach wird ZXIB berechnet. ZXMI wird auf den nächst kleineren ohne Rest durch ZXIB teilbaren Wert abgerundet. Entsprechend wird ZXMA auf den nächst größeren ohne Rest durch ZXIB teilbaren Wert aufgerundet. Danach steht der Unterteilungsabstand ZDXI der Achse fest.

```
CALL ZXYIBE(ZXI,ZIFHO,ZMAX,ZXMI,ZXMA,ZXIB,ZDXI,LXBE)
```

##### EINGABEPARAMETER:

ZXI = Länge der Achse (in Inches).  
ZIFHO = Zifferngröße beim Beziffern der Achse (in Inches).  
ZMAX = Ursprüngliches physikalisches Minimum auf der Achse.  
ZXMI = Korrigiertes, physikalisches Minimum auf der Achse.  
ZXMA = Korrigiertes, physikalisches Maximum auf der Achse.  
ZXIB = Eventuell phys. Beschriftungseinheit beim Beziffern der Achse, je nach LXBE-Wert.  
LXBE = Kennung, ob schon ZXIB definiert ist oder nicht.  
Wenn LXBE < 0 ist, soll der schon in ZXIB stehende Wert (als Startwert) verwendet werden. Wenn LXBE ≥ 1 ist, wird der Startwert für ZXIB hier im Unterprogramm berechnet.



**RÜCKGABEPARAMETER:**

ZDXI = Abstand zwischen zwei Marken auf der Achse (in Inches).  
ZXIB = Phys. Beschriftungseinheit beim Beziffern der Achse.  
LXBE = Kennung des ersten möglichen Wertes bei der  
Beschriftung der Achse.  
LXBE = 1 für die "Mantisse" 1 von ZXIB,  
LXBE = 2 für die "Mantisse" 2 von ZXIB,  
LXBE = 3 für die "Mantisse" 5 von ZXIB.

**RESTLICHE VARIABLEN:**

ABS = Fortran builtin function.  
L = Schleifen- und Zählvariable.  
LINE = Plot-Subroutine (System).  
PLOT = Plot-Subroutine (System).  
ZSTH3 = 2./3.\*ZSTHO.  
ZXR(1:7) = Abszissen für den Achsenrahmen.  
ZXX = Aktuelle X-Position beim Plotten.  
ZYR(1:7) = Ordinaten für den Achsenrahmen.  
ZYY = Aktuelle Y-Position beim Plotten.

ABDRUCK DES UNTERPROGRAMMS ZXYIBE

C - - - Beschriftungseinheit berechnen

```
SUBROUTINE ZXYIBE(ZXI,ZIFHO,ZMAX,ZXMI,ZXMA,ZXIB,ZDXI,LXBE)
  IF (LXBE.GE.0) ZXIB=ABS(ZXMA-ZXMI)*1.0001/ZXI*
  *ZIFHO*(2.+2.*AINT(ALOG10(ABS(ZXMA-ZXMI)*1.0001)))
  LXIB=INT(ALOG10(ZXIB*1.0001))
  IF (ALOG10(ZXIB*1.0001).LT.0.) LXIB=LXIB-1
  ZXIB=ZXIB/(10.**LXIB)
  LXBE=1
  IF (ZXIB.LE.1.) THEN
    ZXIB=1.
  ELSE
    IF (ZXIB.LE.2.) THEN
      ZXIB=2.
      LXBE=2
    ELSE
      IF (ZXIB.LE.5.) THEN
        ZXIB=5.
        LXBE=3
      ELSE
        ZXIB=10.
      ENDIF
    ENDIF
  ENDIF
  ZXIB=ZXIB*(10.**LXIB)
  ZXMI=ZXIB*AINT(ZXMI/ZXIB*1.0001)
  ZXMA=ZXIB*AINT(ZXMA/ZXIB*1.0001)+ZXIB
  IF (ABS(1.-(ZXIB+ZMAX)/ZXMA).LT.1.E-4) ZXMA=ZXMA-ZXIB
  ZDXI=ZXI/(ZXMA-ZXMI)*ZXIB
  RETURN
END
```

#### 4.12 ZXYMAS

##### Beschriftungsmaßstab

Das Unterprogramm ZXYMAS verbessert die von ZXYIBE berechneten Werte ZXIB und ZDXI unter Berücksichtigung der Achsenlänge und der Zifferngröße. Ein Überlappen der zu zeichnenden Ziffern wird somit vermieden.

```
CALL ZXYMAS(ZIFHO, ZXMI, ZXMA, ZXIB, ZDXI, ZXP, LXP, LXB, LXL, LXBE)
```

##### EINGABEPARAMETER:

ZIFHO = Zifferngröße (in Inches).  
ZXMI = Beschriftungs-Minimum der Achse.  
ZXMA = Beschriftungs-Maximum der Achse.  
ZXIB = Beschriftungs-Einheit (-Schrittweite) der Achse.  
ZDXI = Abstand zwischen zwei Marken auf der Achse (in Inches).

##### RÜCKGABEPARAMETER:

ZXIB = Beschriftungs-Einheit (-Schrittweite) der Achse.  
ZDXI = Abstand zwischen zwei Marken auf der Achse (in Inches).  
ZXP = Exponent bei der Exponent-Darstellung an der Achse.  
LXP = Anzahl der Exponentstellen bei der Exponent-Darstellung an der Achse.  
LXB = Anzahl der Ziffern hinter dem Dezimalpunkt.  
LXL = Anzahl der Markierungspositionen auf der Achse.  
LXBE = Kennung des ersten möglichen Wertes bei der Beschriftung der Achse (Wert: 1, 2, oder 3).

##### RESTLICHE VARIABLEN:

ABS = Fortran builtin function.  
AINT = Fortran builtin function.  
ALOG10 = Fortran builtin function.  
AMAX1 = Fortran builtin function.

INT = Fortran builtin function.  
IXBE = Kennung der Häufigkeit der Beschriftung mit der  
Einheit LXIB bezogen auf die Zifferngröße ZIFHO.  
IZIF = Anzahl der Stellen für den aktuellen Zahlenwert.  
LXIB = Exponent bei der Berechnung von ZXIB.  
LOB = Anzahl der Markierungspositionen durch das Maximum.  
LUN = Anzahl der Markierungspositionen durch das Minimum.  
ZAHL = Zu plottender Zahlenwert.  
ZFIB(1:10,1:3) = Matrix mit Faktoren für die tatsächliche,  
IXBE-korrigierte Bezifferungs-Einheit ZXIB.  
ZXX = Aktuelle X-Position beim Plotten.  
ZYY = Aktuelle Y-Position beim Plotten.  
ZXZ = Mittel-X-Position beim Plotten.  
ZYZ = Mittel-Y-Position beim Plotten.

ABDRUCK DES UNTERPROGRAMMS ZXYMAS

```
C - - - Verbesserter Maßstab für die Bezifferung
      SUBROUTINE ZXYMAS
      $      (ZIFHO,ZXMI,ZXMA,ZXIB,ZDXI,ZXP,LXP,LXB,LXL,LXBE)
C - - - - FAKTOREN-MATRIX.
      DIMENSION ZFIB(10,3)
      DATA ZFIB/1.,2.0,5.,10.,20.,50.,100.,200.,500.,1000.,
      $      1.,2.5,5.,10.,25.,50.,100.,250.,500.,1000.,
      $      1.,2.0,4.,10.,20.,40.,100.,200.,400.,1000./
C - - - - - ZIFFERNGROESSE FUER ZXIB BERUECKSICHTIGEN.
      ZXX=0.
      IF (ABS(ZXMA).GT.1.E-7)
      $   ZXX=AMAX1(ZXX,ALOG10(ABS(ZXMA)*1.0001))
      IF (ABS(ZXMI).GT.1.E-7)
      $   ZXX=AMAX1(ZXX,ALOG10(ABS(ZXMI)*1.0001))
      ZXX=5./3.*ZIFHO*(1.+AINT(ZXX))
      IXBE=1
115  IF (ZXX.GT.ZDXI*IXBE) THEN
      IXBE=IXBE+1
      GOTO 115
      ENDIF
      ZXIB=ZXIB*ZFIB(IXBE,LXBE)
      ZDXI=ZDXI*ZFIB(IXBE,LXBE)
C - - - - - PLOTSTELLEN HINTER DEM DEZIMALPUNKT.
      LXB=-1
      IF (ZXIB.LE.1.E-4) THEN
C - - - - - - - ZXIB <= 1.E-4 (GLATTE ZEHNERPOTENZ).
      LXB=INT(ALOG10(ABS(ZXIB)*1.0001)*1.0001)
      ZXP=LXB
      LXB=-1
      LXP=1+INT(ALOG10(ABS(ZXP)*1.0001))
      IF (ZXP.LT.-1.E-4) LXP=LXP+1
      ELSE
      LXP=-99
      ZXP=0.
      IF (ZXIB.LE.1.-1.E-4) THEN
C - - - - - - - 1.E-4 < ZXIB < 1.
      LXB=0
      ZXZ=ZXIB*1.0006
```

```
116      CONTINUE
        IF (ABS(ZXZ-AINT(ZXZ)).GT.ZXZ*0.001) THEN
            LXB=LXB+1
            ZXZ=ZXZ*10.
            GOTO 116
        ENDIF

        ELSE
C - - - - - 1 =< ZXIB, GANZZAHLIG.
            LXB=-1
        ENDIF
    ENDIF
C - - - - ANZAHL DER MARKIERUNGSPPOSITIONEN AUF DER ACHSE.
    LOB=INT(ABS(ZXMA)*1.0001/ZXIB)
    LUN=INT(ABS(ZXMI)*1.0001/ZXIB)
    LXL=1
    IF (ZXMA.GT.1.E-4) THEN
        IF (ZXMI.GT.1.E-4) THEN
            LXL=LXL+LOB-LUN
        ELSE
            IF (ABS(ZXMI).LE.1.E-4) THEN
                LXL=LXL+LOB
            ELSE
                LXL=LXL+LOB+LUN
                IF (ABS(1.-ZXIB*INT(ZXMI*1.0001/ZXIB)/ZXMI)
$           .GT.1.E-4) LXL=LXL+1
            ENDIF
        ENDIF
    ENDIF
ELSE
    IF (ABS(ZXMA).LE.1.E-4) THEN
        LXL=LXL+LUN
    ELSE
        LXL=LXL+LUN-LOB
    ENDIF
ENDIF
RETURN
END
```

#### 4.13 ZXZIFF

##### Abszisse Beziffern

Das Unterprogramm ZXZIFF zeichnet mit den vorhandenen Plotparametern die Zahlen unter die vorhandenen Markierungspositionen der X-Achse.

```
CALL ZXZIFF(ZIFHO,ZSTHO,ZXMI,ZXIB,ZDXI,ZXP,LXP,LXB,LXL)
```

##### EINGABEPARAMETER:

ZIFHO = Zifferngröße (in Inches).  
ZSTHO = Länge der Strichelung an der Achse (in Inches).  
ZXMI = Beschriftungs-Minimum der Achse.  
ZXIB = Beschriftungs-Einheit (-Schrittweite) der Achse.  
ZDXI = Abstand zwischen zwei Marken auf der Achse (in Inches).  
ZXP = Exponent bei der Exponent-Darstellung an der Achse.  
LXP = Anzahl der Exponentstellen bei der Exponent-Darstellung an der Achse.  
LXB = Anzahl der Ziffern hinter dem Dezimalpunkt.  
LXL = Anzahl der Markierungspositionen auf der Achse.

RÜCKGABEPARAMETER: keine

##### RESTLICHE VARIABLEN:

ABS, ALOG10 = Fortran builtin function.  
I = Schleifen- und Zählvariable.  
INT = Fortran builtin function.  
IZIF = Anzahl der Stellen für den aktuellen Zahlenwert.  
L = Schleifen- und Zählvariable.  
NUMBER, SYMBOL = Plot-Subroutine (System).  
ZAHL = Zu plottender Zahlenwert.  
ZIFH2 = ZIFHO/2.  
ZXX, ZYY = Aktuelle X- bzw. Y-Position beim Plotten.  
ZXXZ, ZYYZ = Zentrierte X- bzw. Y-Position beim Plotten.

ABDRUCK DES PLOT-UNTERPROGRAMMS ZXZIFF

C- - - - Beziffern der Abszisse.

SUBROUTINE ZXZIFF

\$ (ZIFHO, ZSTHO, ZXMI, ZXIB, ZDXI, ZXP, LXP, LXB, LXL)

ZIFH2=ZIFHO\*0.5

ZYY=-(ZIFHO+ZSTHO\*2.)

ZXX=0.

ZXZ=ZXX

ZAHL=ZXMI

DO 111 L=1, LXL

IF (ABS(ZAHL).LT.ABS(1.E-4\*ZXIB)) THEN

C - - - - - ZAHL = 0.

IF (ABS(ZXP).GT.1.E-4) THEN

C - - - - - BEI EXPONENT-DARSTELLUNG.

IZIF=1

ZXX=ZXZ-(IZIF-5./14.)\*ZIFH2

CALL NUMBER(ZXX, ZYY, ZIFHO, ZAHL, 0., -1)

ELSE

C - - - - - BEI REINER DEZIMAL-DARSTELLUNG.

I=0

IZIF=1

IF (ZXIB.GT.1.E-4.AND.ZXIB.LT.1.-1.E-4)

\$ IZIF=LXB+2

IF (LXB.GT.2) THEN

I=LXB

IZIF=1

LXB=-1

ENDIF

ZXX=ZXZ-(IZIF-5./14.)\*ZIFH2

CALL NUMBER(ZXX, ZYY, ZIFHO, ZAHL, 0., LXB)

IF (I.GT.2) THEN

LXB=I

IF (ZXIB.GT.1.E-4.AND.ZXIB.LT.1.-1.E-4)

\$ IZIF=LXB+2

ENDIF

ENDIF

ELSE

C - - - - - ZAHL = 0.



```
IF (ZXIB.LE.1.E-4*1.0001) THEN
C - - - - - ZXIB <= 1.E-4 (GLATTE ZEHNERPOTENZ).
  ZYZ=ZAHL*10.**(-INT(ZXP))*1.0001
  IZIF=1+INT(ALOG10(ABS(ZYZ)*1.0001))
  IF (ZYZ.LT.-1.E-4*ZXIB) IZIF=IZIF+1
  ZXX=ZXZ-(IZIF+0.5*LXP+18./7.)*ZIFH2
  ZYY=-ZIFHO*2.
  CALL NUMBER(ZXX,ZYY,ZIFHO,ZYZ,0.,-1)
  ZXX=ZXX+ZIFH2+ZIFHO*IZIF
  ZYZ=ZYY+ZIFHO*0.3
  CALL SYMBOL(ZXX,ZYZ,ZIFH2,'.',0.,1)
  ZXX=ZXX+ZIFH2
  CALL SYMBOL(ZXX,ZYY,ZIFHO,'10',0.,2)
  ZXX=ZXX+ZIFHO+ZIFHO
  ZYZ=ZYY+ZIFHO*0.7
  CALL NUMBER(ZXX,ZYZ,ZIFH2,ZXP,0.,-1)
ELSE
  IF (ZXIB.LE.1.-1.E-4) THEN
C - - - - - 1.E-4 < ZXIB < 1.
    IZIF=LXB+2
    IF (ZAHL.GE.10.)
      $ IZIF=IZIF+INT(ALOG10(ZAHL*1.0001))
    ZXX=ZXZ-(IZIF-5./14.)*ZIFH2
    CALL NUMBER(ZXX,ZYY,ZIFHO,ZAHL,0.,LXB)
  ELSE
C - - - - - 1 =< ZXIB GANZZAHLIG.
    IZIF=1+INT(ALOG10(ABS(ZAHL)*1.0001))
    IF (ZAHL+1.E-4*ZXIB.LT.-1.E-4*ZXIB)
      $ IZIF=IZIF+1
    ZXX=ZXZ-(IZIF-5./14.)*ZIFH2
    CALL NUMBER(ZXX,ZYY,ZIFHO,ZAHL,0.,LXB)
  ENDIF
  ENDIF
  ENDIF
C - - - - - PLOTPOSITION UND ZAHLENWERT ERHOEHEN.
  ZXZ=ZXZ+ZDXI
  ZAHL=ZAHL+ZXIB
111 RETURN
END
```

#### 4.14 ZYZIFF

##### Ordinate Beziffern

Das Unterprogramm ZYZIFF zeichnet mit den vorhandenen Plotparametern die Zahlen unter die vorhandenen Markierungspositionen der Y-Achse.

```
CALL ZYZIFF(ZIFHO,ZSTHO,ZYMI,ZYIB,ZDYI,ZYP,LYP,LYB,LYL)
```

##### EINGABEPARAMETER:

ZIFHO = Zifferngröße (in Inches).  
ZSTHO = Länge der Strichelung an der Achse (in Inches).  
ZYMI = Beschriftungs-Minimum der Achse.  
ZYIB = Beschriftungs-Einheit (-Schrittweite) der Achse.  
ZDYI = Abstand zwischen zwei Marken auf der Achse (in Inches).  
ZYP = Exponent bei der Exponent-Darstellung an der Achse.  
LYP = Anzahl der Exponentstellen bei  
der Exponent-Darstellung an der Achse.  
LYB = Anzahl der Ziffern hinter dem Dezimalpunkt.  
LYL = Anzahl der Markierungspositionen auf der Achse.

RÜCKGABEPARAMETER: keine

##### RESTLICHE VARIABLEN:

ABS, ALOG10 = Fortran builtin function.  
I = Schleifen- und Zählvariable.  
INT = Fortran builtin function.  
IZIF = Anzahl der Stellen für den aktuellen Zahlenwert.  
L = Schleifen- und Zählvariable.  
NUMBER, SYMBOL = Plot-Subroutine (System).  
ZAHL = Zu plottender Zahlenwert.  
ZIFH2 = ZIFHO/2.  
ZXX,ZYY = Aktuelle X- bzw. Y-Position beim Plotten.  
ZXZ,ZYZ = Zentrierte X- bzw. Y-Position beim Plotten.

ABDRUCK DES PLOT-UNTERPROGRAMMS ZYZIFF

C- - - - Beziffern der Ordinate.

SUBROUTINE ZYZIFF

\$ (ZIFHO,ZSTHO,ZYMI,ZYIB,ZDYI,ZYP,LYP,LYB,LYL)

ZIFH2=ZIFHO\*0.5

ZYY=-ZIFH2

ZAHL=ZYMI

DO 121 L=1,LYL

CCCC IF (L.EQ.1.AND.ZAHL.EQ.ZYMI) ZYY=ZYY+ZIFH2

IF (ABS(ZAHL).LT.ABS(1.E-4\*ZYIB)) THEN

C - - - - - ZAHL = 0.

IF (ABS(ZYP).GT.1.E-4) THEN

C - - - - - BEI EXPONENT-DARSTELLUNG.

IZIF=1

ZXX=-(ZSTHO+ZIFHO\*IZIF)

CALL NUMBER(ZXX,ZYY,ZIFHO,ZAHL,0.,-1)

ELSE

C - - - - - BEI REINER DEZIMAL-DARSTELLUNG.

I=0

IZIF=1

IF (ZYIB.GT.1.E-4.AND.ZYIB.LT.1.-1.E-4)

\$ IZIF=LYB+2

IF (LYB.GT.2) THEN

I=LYB

IZIF=1

LYB=-1

ENDIF

ZXX=-(ZSTHO+ZIFHO\*IZIF)

CALL NUMBER(ZXX,ZYY,ZIFHO,ZAHL,0.,LYB)

IF (I.GT.2) THEN

LYB=I

IF (ZYIB.GT.1.E-4.AND.ZYIB.LT.1.-1.E-4)

\$ IZIF=LYB+2

ENDIF

ENDIF

ELSE

C - - - - - ZAHL /= 0.

IF (ZYIB.LE.1.E-4) THEN

C - - - - - ZYIB <= 1.E-4 (GLATTE ZEHNERPOTENZ).

```
ZYZ=ZAHL*10.**(-INT(ZYP))*1.0001
IZIF=1+INT(ALOG10(ABS(ZYZ)*1.0001))
IF (ZYZ.LT.-1.E-4*ZYIB) IZIF=IZIF+1
ZXX=-(ZSTHO+ZIFHO*(IZIF+3)+ZIFH2*LYP)
CALL NUMBER(ZXX,ZYY,ZIFHO,ZYZ,0.,-1)
ZXX=ZXX+ZIFH2+ZIFHO*IZIF
ZYZ=ZYY+ZIFHO*0.3
CALL SYMBOL(ZXX,ZYZ,ZIFH2,'.',0.,1)
ZXX=ZXX+ZIFH2
CALL SYMBOL(ZXX,ZYY,ZIFHO,'10',0.,2)
ZXX=ZXX+ZIFHO+ZIFHO
ZYZ=ZYY+ZIFHO*0.7
CALL NUMBER(ZXX,ZYZ,ZIFH2,ZYP,0.,-1)
ELSE
  IF (ZYIB.LE.1.-1.E-4) THEN
C - - - - - 1.E-4 < ZYIB < 1.
    IZIF=LYB+2
    IF (ZAHL.GE.10.)
      $ IZIF=IZIF+INT(ALOG10(ZAHL*1.0001))
      ZXX=-(ZSTHO+ZIFHO*IZIF)
      CALL NUMBER(ZXX,ZYY,ZIFHO,ZAHL,0.,LYB)
    ELSE
C - - - - - 1 =< ZYIB GANZZAHLIG.
      IZIF=1+INT(ALOG10(ABS(ZAHL)*1.0001))
      IF (ZAHL+1.E-4*ZYIB.LT.-1.E-4*ZYIB)
        $ IZIF=IZIF+1
      ZXX=-(ZSTHO+ZIFHO*IZIF)
      CALL NUMBER(ZXX,ZYY,ZIFHO,ZAHL,0.,LYB)
    ENDIF
  ENDIF
ENDIF
ENDIF
C - - - - - PLOTPOSITION UND ZAHLENWERT ERHOEHEN.
CCCC IF (L.EQ.1.AND.ZAHL.EQ.ZYMI) ZYY=ZYY-ZIFH2
ZYY=ZYY+ZDYI
ZAHL=ZAHL+ZYIB
121 CONTINUE
RETURN
END
```

PROGRAMMIERTE FORMELN

————— I=1 , J=1 , K=0 —————

TRANSPARENTES SERBER MODELL  
ENERGIEVERTEILUNG  
KEINE KORREKTUR DES MODELLS

$$\frac{d\sigma}{dE_a}(E_a) = \frac{\sqrt{2\mu C}}{\pi(2\mu C + (\sqrt{2m_a E_a} - \frac{m_a}{m_p} \sqrt{2m_p E_o})^2)} \cdot 2m_a m_b \sqrt{m_a m_b (E_a - Q)(E_o - Q - E_a)}$$

————— I=1 , J=1 , K=1 —————

TRANSPARENTES SERBER MODELL  
ENERGIEVERTEILUNG  
COULOMB ENERGY DIFFERENCES CORRECTION

$$\frac{d\sigma}{dE_a}(E_a) = \frac{\sqrt{2\mu C}}{\pi(2\mu C + (\sqrt{2m_a E_a'} - \frac{m_a}{m_p} \sqrt{2m_p E_o'})^2)} \cdot 2m_a m_b \sqrt{m_a m_b (E_a' - Q)(E_o' - Q - E_a')}$$

————— I=1 , J=2 , K=0 —————

TRANSPARENTES SERBER MODELL  
WINKELVERTEILUNG  
KEINE KORREKTUR DES MODELLS

$$\frac{d\sigma}{d\Omega}(\theta) = \frac{\tan\theta_o}{2\pi \sqrt{(\tan^2\theta_o + (1 + \tan^2\theta_o) \tan^2\theta)^3}}$$

I=1 , J=2 , K=1

TRANSPARENTES SERBER MODELL  
WINKELVERTEILUNG  
COULOMB ENERGY DIFFERENCES CORRECTION

$$\frac{d\delta}{d\Omega}(\theta) = \frac{\tan\theta_0'}{2\pi\sqrt{(\tan^2\theta_0' + (1+\tan^2\theta_0')\tan^2\theta)^3}}$$

I=1 , J=2 , K=2

TRANSPARENTES SERBER MODELL  
WINKELVERTEILUNG  
COULOMB DEFLECTION CORRECTIONS

$$\frac{d\delta}{d\Omega}(\theta) = \frac{\sqrt{u^3} P_{1/2}(u)}{2\pi(1+\zeta^2+\zeta_c^2)}$$

$$u = \frac{1+\zeta^2+\zeta_c^2}{\sqrt{(1+\zeta^2+\zeta_c^2)^2 - 4\zeta^2\zeta_c^2}}$$

$$\zeta_c = \frac{1}{2} \sqrt{\frac{m_a}{m_b}} \frac{Z_p V_c}{\sqrt{E_0 c}} \left(1 + \frac{m_p}{m_a} \frac{Z_a}{Z_p}\right), \quad \zeta = \frac{\tan\theta}{\tan\theta_0}$$

I=1 , J=2 , K=3

TRANSPARENTES SERBER MODELL  
WINKELVERTEILUNG  
CEDC UND CDC BERÜCKSICHTIGT

$$\frac{d\delta}{d\Omega}(\theta) = \frac{\sqrt{u^3} P_{1/2}(u)}{2\pi(1+\zeta^2+\zeta_c^2)}$$

$$u = \frac{1 + \zeta^2 + \zeta_c'^2}{\sqrt{(1 + \zeta^2 + \zeta_c'^2)^2 - 4 \zeta^2 \zeta_c'^2}}$$

$$\zeta_c' = \frac{1}{2} \sqrt{\frac{m_a}{m_b}} \frac{Z_p V_c}{\sqrt{E_0'}} \left( 1 + \frac{m_p}{m_a} \frac{Z_a}{Z_p} \right), \quad \zeta' = \frac{\tan \theta}{\tan \theta_0'}$$

————— I=1 , J=3 , K=0 —————

TRANSPARENTES SERBER MODELL  
 INKLUSIVER WIRKUNGSQUERSCHNITT  
 KEINE KORREKTUR DES MODELLS

$$\frac{d^2 \sigma}{d\Omega dE_a}(\theta, E_a) = \frac{R_T R_p \sqrt{2\mu\epsilon'}}{(2\mu\epsilon' + \sqrt{\frac{2m_a^2 E_0'}{m_p}} - \sqrt{2m_a E_a'} \cos\theta)^2 + (\sqrt{2m_a E_a'} \sin\theta)^2} \cdot f$$

$$f = 2 m_a m_b \sqrt{m_a m_b (E_a - Q)(E_0 - Q - E_a)}$$

————— I=1 , J=3 , K=1 —————

TRANSPARENTES SERBER MODELL  
 INKLUSIVER WIRKUNGSQUERSCHNITT  
 COULOMB ENERGY DIFFERENCES CORRECTION

$$\frac{d^2 \sigma}{d\Omega dE_a}(\theta, E_a) = \frac{R_T R_p \sqrt{2\mu\epsilon'}}{(2\mu\epsilon' + \sqrt{\frac{2m_a^2 E_0'}{m_p}} - \sqrt{2m_a E_a'} \cos\theta)^2 + (\sqrt{2m_a E_a'} \sin\theta)^2} \cdot f'$$

$$f' = 2 m_a m_b \sqrt{m_a m_b (E_a' - Q)(E_0' - Q - E_a')}$$

I=2 , J=1 , K=0

OPAKES SERBER MODELL  
ENERGIEVERTEILUNG  
KEINE KORREKTUR DES MODELLS

$$\frac{d\sigma}{dE_a}(E_a) = \frac{\frac{\pi}{4} R_T R_P 2\mu\epsilon}{\sqrt{(2\mu\epsilon + (\sqrt{2m_a E_a} - \frac{m_a}{m_p} \sqrt{2m_p E_o})^2)^3}} \cdot 2m_a m_b \sqrt{m_a m_b (E_a - Q)(E_o - Q - E_a)}$$

I=2 , J=1 , K=1

OPAKES SERBER MODELL  
ENERGIEVERTEILUNG  
COULOMB ENERGY DIFFERENCES CORRECTION

$$\frac{d\sigma}{dE_a}(E_a) = \frac{\frac{\pi}{4} R_T R_P 2\mu\epsilon}{\sqrt{(2\mu\epsilon + (\sqrt{2m_a E'_a} - \frac{m_a}{m_p} \sqrt{2m_p E'_o})^2)^3}} \cdot 2m_a m_b \sqrt{m_a m_b (E'_a - Q)(E'_o - Q - E'_a)}$$

I=2 , J=2 , K=0

OPAKES SERBER MODELL  
WINKELVERTEILUNG  
KEINE KORREKTUR DES MODELLS

$$\frac{d\sigma}{d\Omega}(\theta) = \frac{R_T R_P}{\pi} \frac{\tan\theta_o}{\sqrt{(\tan^2\theta_o + (1 + \tan^2\theta_o) \tan^2\theta)^3}}$$



$$\cdot \left( 1 - \frac{\tan^3 \theta_0}{2(1+3 \tan^2 \theta_0) \tan^3 \theta} \left( \left( 1 + \left( 1 + \frac{1}{\tan^2 \theta_0} \right) \tan^2 \theta \right) \right) \right) \cdot$$

$$\cdot \operatorname{asin} \left( \frac{\tan \theta}{\sqrt{\tan^2 \theta_0 + (1 + \tan^2 \theta_0) \tan^2 \theta}} - \frac{\tan \theta}{\tan \theta_0} \right)$$

———— I=2 , J=2 , K=1 ————

OPAKES SERBER MODELL  
WINKELVERTEILUNG  
COULOMB ENERGY DIFFERENCES CORRECTION

$$\frac{d\sigma}{d\Omega}(\theta) = \frac{R_T R_P}{\pi} \frac{\tan \theta'_0}{\sqrt{(\tan^2 \theta'_0 + (1 + \tan^2 \theta'_0) \tan^2 \theta)^3}} \cdot$$

$$\cdot \left( 1 - \frac{\tan^3 \theta'_0}{2(1+3 \tan^2 \theta'_0) \tan^3 \theta} \left( \left( 1 + \left( 1 + \frac{1}{\tan^2 \theta'_0} \right) \tan^2 \theta \right) \right) \right) \cdot$$

$$\cdot \operatorname{asin} \left( \frac{\tan \theta}{\sqrt{\tan^2 \theta'_0 + (1 + \tan^2 \theta'_0) \tan^2 \theta}} - \frac{\tan \theta}{\tan \theta'_0} \right)$$

I=2 , J=2 , K=2

OPAKES SERBER MODELL  
WINKELVERTEILUNG  
COULOMB DEFLECTION CORRECTIONS

$$\frac{d\delta}{d\Omega}(\theta) = \left( 1 - \frac{\tan^3 \theta_0}{2(1+3\tan^2 \theta_0)\tan^3 \theta} \left( \tan^2 \theta_0 + (1+\tan^2 \theta_0)\tan^2 \theta \right) \cdot \right.$$

$$\left. \cdot \operatorname{asin} \left( \frac{\tan \theta}{\sqrt{\tan^2 \theta_0 + (1+\tan^2 \theta_0)\tan^2 \theta}} - \frac{\tan \theta}{\tan \theta_0} \right) \right) \frac{\sqrt{u^3} P_{1/2}(u)}{2\pi(1+\zeta^2+\zeta_c^2)}$$

$$u = \frac{1+\zeta^2+\zeta_c^2}{\sqrt{(1+\zeta^2+\zeta_c^2)^2 - 4\zeta^2\zeta_c^2}}$$

$$\zeta_c = \frac{1}{2} \sqrt{\frac{m_a}{m_b}} \frac{Z_p V_c}{\sqrt{E_0 \mathcal{L}'}} \left( 1 + \frac{m_p}{m_a} \frac{Z_a}{Z_p} \right) , \quad \zeta = \frac{\tan \theta}{\tan \theta_0}$$

I=2 , J=2 , K=3

OPAKES SERBER MODELL  
WINKELVERTEILUNG  
CEDC UND CDC BERÜCKSICHTIGT

$$\frac{d\delta}{d\Omega}(\theta) = \left( 1 - \frac{\tan^3 \theta'_0}{2(1+3\tan^2 \theta'_0)\tan^3 \theta} \left( \tan^2 \theta'_0 + (1+\tan^2 \theta'_0)\tan^2 \theta \right) \cdot \right.$$

$$\left. \cdot \operatorname{asin} \left( \frac{\tan \theta}{\sqrt{\tan^2 \theta'_0 + (1+\tan^2 \theta'_0)\tan^2 \theta}} - \frac{\tan \theta}{\tan \theta'_0} \right) \right) \frac{\sqrt{u^3} P_{1/2}(u)}{2\pi(1+\zeta^2+\zeta_c^2)}$$

$$u' = \frac{1 + \zeta^2 + \zeta_c^2}{\sqrt{(1 + \zeta^2 + \zeta_c^2)^2 - 4 \zeta^2 \zeta_c^2}}$$

$$\zeta_c^2 = \frac{1}{2} \sqrt{\frac{m_a}{m_b}} \frac{Z_p V_c}{\sqrt{E_0 C}} \left(1 + \frac{m_p}{m_a} \frac{Z_a}{Z_p}\right), \quad \zeta^2 = \frac{\tan \theta}{\tan \theta_0}$$

I=2, J=3, K=0

OPAKES SERBER MODELL  
 INKLUSIVER WIRKUNGSQUERSCHNITT  
 KEINE KORREKTUR DES MODELLS

$$\frac{d^2 \sigma}{d\Omega dE_a}(\theta, E_a) = \frac{R_T R_p \sqrt{2\mu C}}{(2\mu C + \sqrt{\frac{2m_a^2 E_0}{m_p}} - \sqrt{2m_a E_a} \cos \theta)^2 + (\sqrt{2m_a E_a} \sin \theta)^2} \cdot$$

$$\cdot \left[ \frac{2\mu C + \left(\sqrt{\frac{2m_a^2 E_0}{m_p}} - \sqrt{2m_a E_a} \cos \theta\right)^2 + (\sqrt{2m_a E_a} \sin \theta)^2 P_{1/2}(u)}{4 \sqrt{(2\mu C + \left(\sqrt{\frac{2m_a^2 E_0}{m_p}} - \sqrt{2m_a E_a} \cos \theta\right)^2)^3}} \right] \cdot f$$

$$u = \frac{2\mu C + \left(\sqrt{2m_a E_a} \cos \theta - \frac{m_a}{m_p} \sqrt{m_p E_0}\right)^2 + \frac{1}{2} (\sqrt{2m_a E_a} \sin \theta)^2}{\sqrt{2\mu C + \left(\sqrt{\frac{2m_a^2 E_0}{m_p}} - \sqrt{2m_a E_a} \cos \theta\right)^2 + (\sqrt{2m_a E_a} \sin \theta)^2}}$$

$$\cdot \frac{1}{\sqrt{2\mu C + \left(\sqrt{\frac{2m_a^2 E_0}{m_p}} - \sqrt{2m_a E_a} \cos \theta\right)^2}}$$

$$f = 2 m_a m_b \sqrt{m_a m_b (E_a - Q)(E_0 - Q - E_a)}$$

I=2 , J=3 , K=1

OPAKES SERBER MODELL  
 INKLUSIVER WIRKUNGSQUERSCHNITT  
 COULOMB ENERGY DIFFERENCES CORRECTION

$$\frac{d^2 \sigma}{d\Omega dE_a}(\theta, E_a) = \frac{R_T R_p \sqrt{2\mu\epsilon}}{(2\mu\epsilon + \sqrt{\frac{2m_a^2 E_o'}{m_p} - \sqrt{2m_a E_a'} \cos\theta})^2 + (\sqrt{2m_a E_a'} \sin\theta)^2} \cdot$$

$$\cdot \left[ \frac{2\mu\epsilon + \left( \sqrt{\frac{2m_a^2 E_o'}{m_p} - \sqrt{2m_a E_a'} \cos\theta} \right)^2 + (\sqrt{2m_a E_a'} \sin\theta)^2 P_{1/2}(u)}{4 \sqrt{(2\mu\epsilon + \left( \sqrt{\frac{2m_a^2 E_o'}{m_p} - \sqrt{2m_a E_a'} \cos\theta} \right)^2)^3}} \right] \cdot f'$$

$$u = \frac{2\mu\epsilon + \left( \sqrt{2m_a E_a'} \cos\theta - \frac{m_a - \sqrt{m_p E_o'}}{m_a} \right)^2 + \frac{1}{2} (\sqrt{2m_a E_a'} \sin\theta)^2}{\sqrt{2\mu\epsilon + \left( \sqrt{\frac{2m_a^2 E_o'}{m_p} - \sqrt{2m_a E_a'} \cos\theta} \right)^2 + (\sqrt{2m_a E_a'} \sin\theta)^2}}$$

$$\cdot \frac{1}{\sqrt{2\mu\epsilon + \left( \sqrt{\frac{2m_a^2 E_o'}{m_p} - \sqrt{2m_a E_a'} \cos\theta} \right)^2}}$$

$$f' = 2m_a m_b \sqrt{m_a m_b (E_a' - Q)(E_o' - Q - E_a')}$$

## PLOTTING SUBROUTINES USED (SYSTEM)

CALL PLOTS(0.,0.,0.) Initialisation for plotting  
on a (monochrom) Versatec plotter.

CALL PLOT(ZXX,ZYY,-3) Defines a new origin (0,0) shifted  
by (ZXX,ZYY) from the old one.

CALL PLOT(ZXX,ZYY,+3) Move to the position (ZXX,ZYY) without  
drawing.

CALL PLOT(ZXX,ZYY,+2) Draws the line to the position (ZXX,ZYY).

CALL LINE(ZX,ZY,NXY,1,0,0) Draws the line between the NXY  
points (ZX(i),ZY(i)) without symbols.

CALL LINE(ZX,ZY,NXY,1,2,4) Draws the line between the NXY  
points including symbol number 4  
at each second point.

CALL NEWPEN(IPEN) In monochrom plots: Thickness of plotted  
lines. In color plots: Number of a color.

CALL NUMBER(ZXX,ZYY,ZHIGH,ZAHL,ZANGLE,NZ)  
Draws the real number ZAHL with angle  
ZANGLE beginning at the position (ZXX,ZYY)  
with the height ZHIGH and NZ digits after  
the decimal point (NZ=-1 gives integer  
numbers).

CALL SYMBOL(ZXX,ZYY,ZHIGH,CHEXT,ZANGLE,NCH)  
Draws the NCH characters of CHEXT with  
angle ZANGLE beginning at the position  
(ZXX,ZYY) with the height ZHIGH, capital  
latin letters;

CALL SYMBLO(.....) same as SYMBOL;

CALL SYMBL1(.....) same as SYMBLO for small latin letters;

CALL SYMBL2(.....) same as SYMBLO for capital greek letters;

CALL SYMBL3(.....) same as SYMBLO for small greek letters  
(see "Korrespondenztafel").

- α) SYMBLO (plottet einen String aus lateinischen Großbuchstaben,  
Ziffern und Sonderzeichen)  
SYMBLS (wie SYMBLO, jedoch in Spiegelschrift)  
SYMBL1 (plottet einen String aus lateinischen Kleinbuchstaben)  
SYMBL2 ( " " " griechischen Großbuchstaben)  
SYMBL3 ( " " " " Kleinbuchstaben)

CALL SYMBLi (XCOR,YCOR,HEI,ITEXT,ANG,NCH) mit  $i = 0, 1, 2, 3$  und  
NCH  $\geq 0$ .

Die Argumente haben die gleiche Bedeutung wie in SYMBOL mit folgenden Ausnahmen:

zu XCOR,YCOR: Durch Angabe des speziellen Wertes 999.0 für XCOR und/oder YCOR im Aufruf von SYMBLi läßt sich eine mit einer der 5 Subroutinen SYMBLi begonnene Zeile positionsrichtig fortsetzen (siehe Programmbeispiel und Abbildung A1), nicht aber eine mit SYMBOL begonnene Zeile. Der Fortsetzungspunkt einer mit SYMBLi begonnenen Zeile läßt sich auch mit der Subroutine WHERE (S. 21, 111, 155) erfragen. Soll einunddieselbe Zeile Zeichen aus verschiedenen SYMBLi-Subroutinen oder sowohl SYMBOL- als auch SYMBLi-Zeichen, z.B. Groß- und Klein-Buchstaben, enthalten, dann plote man entsprechend maskierte Zeilen mit gleichen Anfangskoordinaten übereinander (siehe Programmbeispiel und Abbildung A1).

zu HEI: Zur Festlegung der Höhe der Zeichen-Grundraster (in inch) benutzen die Subroutinen SYMBLi den Absolutbetrag von HEI. Zur Festlegung der Breite der Zeichen-Grundraster dient das Vorzeichen von HEI, u.zw. bedeutet:  
positives Vorzeichen: Breite =  $7/7 * \text{Höhe}$  (wie die SYMBOL-Zeichen des CALCOMP-, STATOS- und VERSATEC-Plotters),  
negatives Vorzeichen: Breite =  $6/7 * \text{Höhe}$  (wie die SYMBOL-Zeichen des XYNETICS-Plotters).

zu ITEXT: Die Subroutinen SYMBLi plotten nicht die Zeichen gemäß den Symboltabellen, z.B. S. 120 für den STATOS- oder S. 166 - 170 für den XYNETICS-Plotter, sondern deren Entsprechungen gemäß der Korrespondenztabelle in Abbildung A1.

zu NCH: In den Subroutinen SYMBLi bewirkt NCH=0 sofortigen Rücksprung ins aufrufende Programm (no operation).

Als Beispiele für den Aufruf der Subroutinen SYMBLi sei der Programmausschnitt wiedergegeben, mit dem der Mittelteil von Abbildung A1 geplottet wurde:

```
INTEGER*4 IBUF(1000)
CALL PLOTS(IBUF,1000,0)
:
C-----DREHPUNKT MARKIEREN
CALL SYMBOL(4.3,4.0,0.14,3,0.0,-1)
C-----ZEILEN FORTSETZEN
CALL SYMBL0(4.3,4.0,-0.21,' S',45.0,2)
CALL SYMBL1(999.0,999.0,-0.21,'S',45.0,1)
CALL SYMBL2(999.0,999.0,-0.21,'S',45.0,1)
CALL SYMBL3(999.0,999.0,-0.21,'S',45.0,1)
C-----ZEILEN SCHACHTELN
CALL SYMBL0(4.3,4.0,-0.21,' K K',-45.0,5)
CALL SYMBL1(4.3,4.0,-0.21,' f ',-45.0,5)
:
CALL PLOT(0.0,0.0,999)
STOP
END
```

β) SYMBL4 (plottet gefüllte zentrierte Punktsymbole)

CALL SYMBL4(XCOR,YCOR,HEI,ITEQ,ANG,NCH) mit NCH < 0.

Die Argumente haben dieselbe Bedeutung wie in SYMBOL mit folgenden Ausnahmen:

zu XCOR,YCOR: Im Gegensatz zu SYMBOL bewirken in SYMBL4 Werte = 999.0 nicht Ermittlung des letzten Koordinaten des "Plotterstifts", sondern eine Fehlermeldung und Rücksprung ins aufrufende Programm. (Don't use with 999.0)

zu ITEQ: Es sind nur die Code-Nummern 0; 1; 2; 5; 12 zulässig (valid) (siehe Abbildung A1); andernfalls erfolgt eine Fehlermeldung und Rücksprung ins aufrufende Programm.

8) Korrespondenztafel

SYMBOL: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  
#,\$,-/+\_)ϕ| &>:, -'?"=!(

SYMBLO: ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789  
#,\$,-/+\_)ϕ| &>:, -'?"=!(

SYMBLS: e8zδz#ES10SYXWVUTzR090IMJKLIHGfE0CBA

SYMBL1: abcdefghijklmnopqrstuvwxyz äöüß

SYMBL2: ΧΔ ΦΓ Λ Π ΡΣ ΘΨΩ

SYMBL3: αβχδεφζ ι κλμνοπ ρστ ξυζ ηθψω

SSZO  
+  
KFK

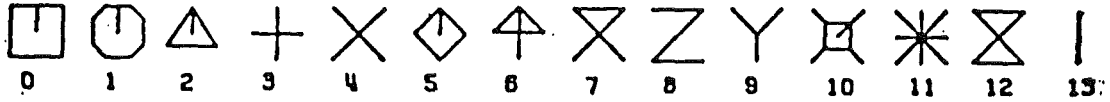
SYMBL4: ITEO: 0 1 2 5 12  
■ ● ▲ ◆ ✕

ANG: 45.0 180.0 45.0 90.0  
◆ ▼ ■ ✕



8)

A: Zentrierte Symbole (0-13)



B: Einzeichnen von Werten in Diagramme:

CALL SYMBOL (1.5, 0.5, 0.14, 3, 0.0, -1)  
 CALL SYMBOL (3.25, 0.0, 0.14, 0, 0.0, -2)  
 CALL SYMBOL (4.5, 0.0, 0.14, 5, 0.0, -2)

Feder ist abgehoben    - - -  
 Feder ist abgesenkt    \_\_\_\_\_



C: Gleichzeitige Verwendung von SYMBOL und NUMBER und Zeichnen eines Exponenten:

CALL SYMBOL (X, Y, 0.14, 10HVALUE OF X, 0.0, 10)  
 CALL SYMBOL (999.0, Y + 0.1, 0.07, 2H2, 0.0, 2)  
 CALL SYMBOL (999.0, Y, 0.14, 2H=, 0.0, 2)  
 CALL NUMBER (999.0, 999.0, 0.14, VALUE, 0.0, 3)

VALUE OF X<sup>2</sup> = 12.123

D: Zeichnen von Text und Zahlen unter verschiedenen Winkeln:

DO I = 0, 315, 45

ANGLE = I

CALL SYMBOL (X, Y, 0.1, 7H ANGLE=, ANGLE, 7)

10 CALL NUMBER (999.0, 999.0, 0.1, ANGLE, ANGLE, -1)

