

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

View metadata, citation and similar papers at core.ac.uk

brought to you by  **CORE**
provided by KITopen

Oktober 1971

KFK 1486

Institut für Reaktorentwicklung

- P L O T H L -

Ein Fortran IV Unterprogramm
zur Darstellung von Funktionen von zwei unabhängigen Variablen
durch ihre Höhenlinien auf einem Plotter

U. Schumann



GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

Kernforschungszentrum Karlsruhe

Oktober 1971

KFK 1486

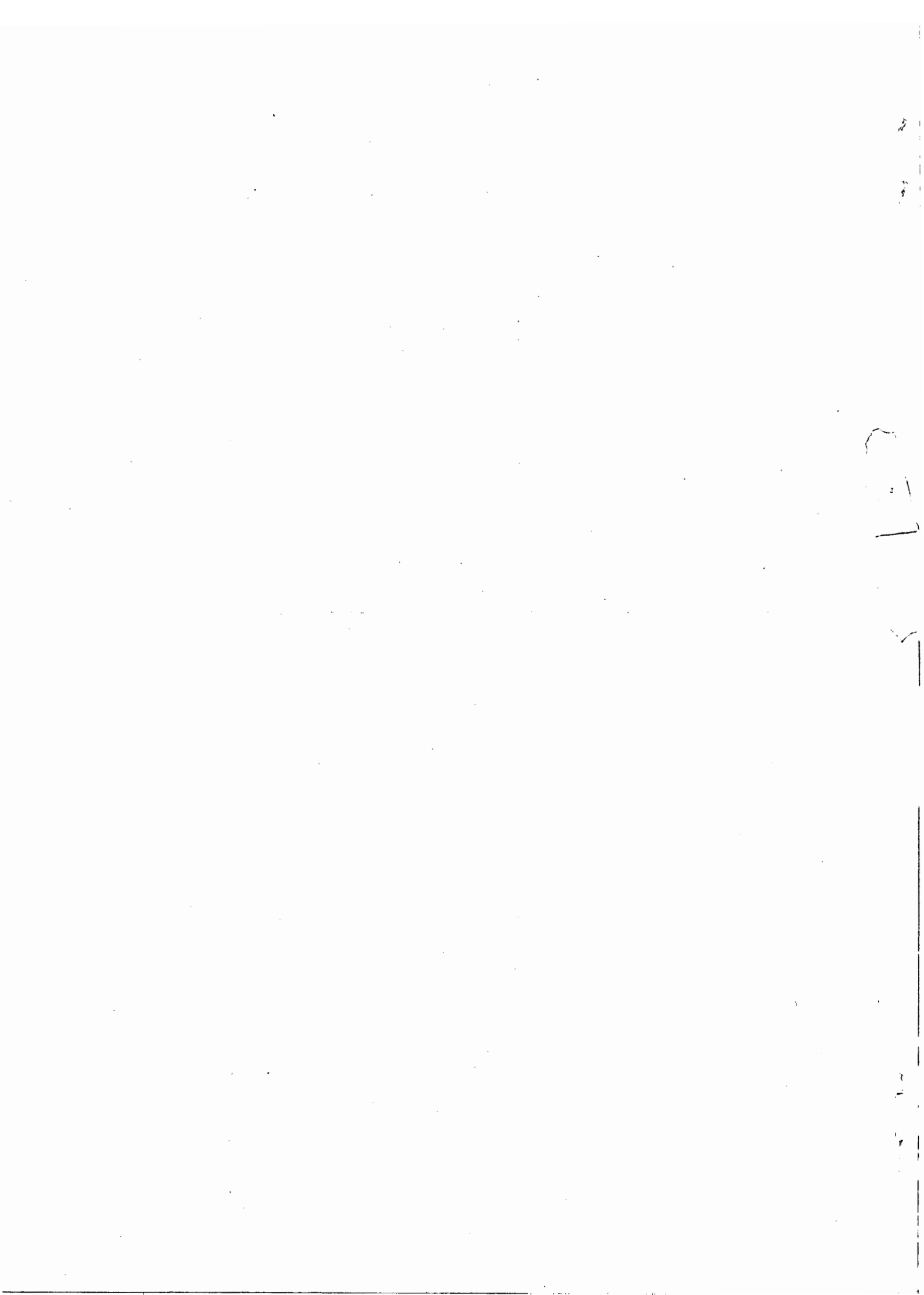
Institut für Reaktorentwicklung

- PLØTHL -

Ein Fortran IV Unterprogramm zur Darstellung von Funktionen
von zwei unabhängigen Variablen durch ihre Höhenlinien auf
einem Plotter

U.Schumann

Gesellschaft für Kernforschung m.b.H., Karlsruhe



Zusammenfassung

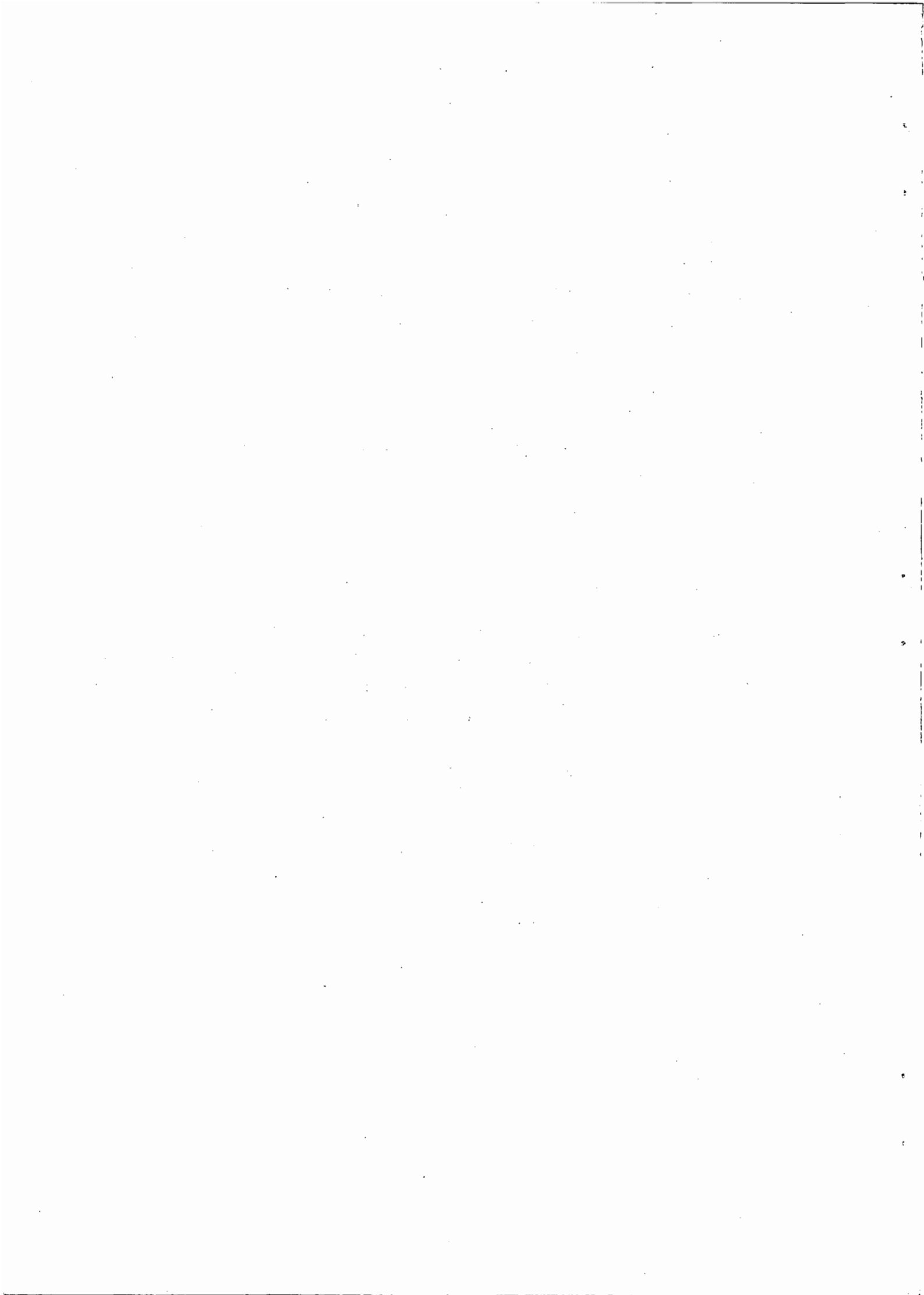
Zum Zeichnen von Höhenlinien einer stetigen und eindeutigen Funktion $F(X,Y)$ von zwei unabhängigen Variablen X und Y , wird ein FORTRAN-IV-Unterprogramm mit Namen PLOTHL beschrieben. PLOTHL erwartet, daß die Funktion F punktweise über einem vier-eckigen Maschennetz gegeben ist. Die Koordination der Maschenpunkte (X,Y) können sowohl karthesisch als auch krummlinig sein. Die Ausgabe der Höhenlinien erfolgt auf einem Plotter.

Dieser Bericht enthält die Beschreibung der Programm Benutzung und der Methode sowie die FORTRAN-Listen. Einige Anwendungsbeispiele demonstrieren die Fähigkeiten von PLOTHL.

Abstract

To create contour plots of steady and unique functions $F(X,Y)$ of two independent variables X and Y , a FORTRAN-IV-subroutine called PLOTHL is described. PLOTHL expects the function F given pointwise on a four-corner mesh grid. The coordinates of the mesh points (X,Y) may be either cartesian or curvilinear. The contours are drawn on a plotter.

This report contains the description of the program call and the method used and the FORTRAN lists. Some examples show the capabilities of PLOTHL.



Inhaltsverzeichnis

Zusammenfassung

1. Zweck, Möglichkeiten und Einschränkungen
2. Anwendungsvorschriften
3. Programmlogik
4. Verbesserungsmöglichkeiten
5. Beispiele
6. Speicherbedarf, Rechenzeit, Unterprogramme

Literatur

Abbildungen

Anhang 1: Fortran-Listen des Unterprogramms PLØTHL

Anhang 2: Kurzbeschreibungen der verwendeten Unterprogramme



1. Zweck, Möglichkeiten und Einschränkungen

Funktionen der Form $F(x,y)$, die gegeben sind für einen Wertebereich (x,y) mit einer Umrandung Γ beschreiben eine Fläche, die ein dreidimensionales Gebilde im Raum (F,x,y) darstellt.

Dreidimensionale Gebilde lassen sich nicht ohne weiteres in einer Zeichnung darstellen, da diese nur zweidimensionale Dinge wiedergeben kann.

Eine Art der Darstellung einer Funktion $F(x,y)$, die sowohl qualitativ als auch quantitativ die Eigenschaften der Funktion gut demonstriert, ist die Darstellung ihrer Höhenlinien

$$F(x,y) = H,$$

mit $H = \text{const.}$

PLOTHL erzeugt unter noch zu erläuternden Voraussetzungen derartige Zeichnungen, in denen eine Funktion $F(x,y)$ durch ihre Höhenlinien dargestellt ist. PLOTHL berücksichtigt hierbei insbesondere die Gegebenheiten, die vorliegen, wenn $F(x,y)$ in einer numerischen Rechnung punktweise über einem viereckigen Netz ermittelt wurde.

Es werden folgende Voraussetzungen angenommen:

- 1) $F(x,y)$ ist eindeutig
- 2) $F(x,y)$ ist stetig
- 3) $F(x,y)$ ist punktweise gegeben mit

$$F(I,J) = F(X(I), Y(J)), \quad I = 1,2,\dots,NX$$
$$J = 1,2,\dots,NY$$

Die Umrandung ist viereckig.

- 4) Die Höhenlinien werden punktweise ermittelt auf den Verbindungslinien zwischen je zwei benachbarten Maschenpunkten sowie (wenn gewünscht) zwischen den Maschenpunkten und den Maschenmittelpunkten (Schnittpunkt F5 der Diagonalen, vergl. Abb. 1) wobei jeweils linear interpoliert wird und die Höhenlinien durch die linearen Verbindungen der ermittelten Punkte dargestellt werden.
- 5) Für die Darstellung des Maschennetzes wird entweder angenommen, daß es sich um ein kartesisches Maschennetz handelt oder aber um ein beliebiges anderes; im letzteren Fall müssen zwei stetige Funktionen bereitgestellt sein, die die Umrechnung der Koordinaten x, y in die Koordinaten ξ, η eines kartesischen Koordinatensystems vornehmen:

$$\xi = \text{XFUN}(X, Y)$$

$$\eta = \text{YFUN}(X, Y)$$

Beispielsweise können den Koordinaten X, Y Zylinderkoordinaten entsprechen, so daß die Transformationen

$$\xi = \text{XFUN}(X, Y) = X \cdot \text{SIN}(Y)$$

$$\eta = \text{YFUN}(X, Y) = X \cdot \text{COS}(Y)$$

anzugeben sind.

Die Ausgabe der Zeichnung erfolgt mit dem Unterprogramm PLOTA [1] auf dem 1130-Plotter.

2. Anwendungsvorschriften

PLOTHL ist eine Fortran-Subroutine mit den Entries NOCHHL und ZUHL.

PLOTHL ist aufzurufen, wenn eine neue Zeichnung eröffnet werden soll mit Ausgabe einer Darstellung der Bereichsumrandung, Erläuterung der Höhenliniensymbole, Beschriftung der Koordinaten und anschließendem Zeichnen von Höhenlinien.

NOCHHL ist aufzurufen, wenn in die vorhergehende Zeichnung noch weitere Höhenlinien gezeichnet werden sollen, wobei das Feld F inzwischen verändert worden sein kann.

ZUHL ist aufzurufen, wenn zusätzlich zu den Höhenlinien noch andere Kurven in die Zeichnung eingezeichnet werden sollen, wie z.B. eine einschränkende Berandungskurve.

Die Aufrufe haben in folgender Weise zu erfolgen:

1) CALL PLOTHL (NX, NY, F, X, Y,
NH, HV, NV, IDNV,
XZ, YZ, IDPLOT,
NTEXT, XTEXT, YTEXT
KMAX, NXMAX, XP, YP, S,
DREI, KARTH,
XFUN, YFUN)

Mit Ausnahme des Parameters IDPLOT sowie der Felder XP, YP, S werden im allgemeinen die Eingabewerte nicht verändert. (Das Feld F kann in PLOTHL geringfügig verändert werden und zwar genau dann, wenn ein Wert F (I,J) exakt gleich einem Höhenniveau ist. F(I,J) wird dann maximal um 1.E-5 mal die maximale Differenz $F(I,J) - F(I-\frac{+1}{0}, J-\frac{+1}{0})$ verändert. Bei genügend stetigen Feldern dürfte diese Änderung für spätere Rechnungen mit F ohne Bedeutung sein. Für die Erläuterung der Notwendigkeit dieser Änderung sei auf die Beschreibung der Programmlogik verwiesen).

Argumenten-Name	Typ	Bedeutung
1) NX	I * 4	Zahl der Netzlinien in X-Richtung
2) NY	I * 4	Zahl der Netzlinien in Y-Richtung
3) F	REAL F(NXMAX,NY)	Funktionswerte F(I,J) I = 1,2, ..., NX; J = 1,2, ..., NY NX ≤ NXMAX
4) X	REAL X(NX)	Feld der X-Koordinaten der Netzlinien
5) Y	REAL Y(NY)	Feld der Y-Koordinaten der Netzlinien
6) NH	I * 4	Zahl der Höhen, für die Höhenlinien gezeichnet werden sollen
7) HV	REAL HV(NH)	Werte der Höhen H für die die Kurven F(X,Y) = H gezeichnet werden sollen
8) NV	INTEGER NV(NH)	0 ≤ NV(I) ≤ 9, I = 1,2, ..., NH NV(i) identifiziert ein Punktzeichen gemäß PLOTA, mit dem die Höhenlinien gekennzeichnet werden, wenn IDNV > 0 ist. Die Höhe der zugehörigen Höhenkurve wird in einer Legende zu dem Punktzeichen in der Zeichnung angegeben.
9) IDNV	I * 4	Bei der Zeichnung des Höhenpolygonzugs wird jeder IDNV-te Knickpunkt mit dem Punktzeichen gekennzeichnet, wenn IDNV > 0 ist. (Empfehlenswert: IDNV ≥ 10)
10) XZ	R * 4	Länge der Zeichnung in waagerechter Richtung in Metern (siehe Bemerkung 1)
11) YZ	R * 4	Länge der Zeichnung in senkrechter Richtung in Metern (siehe Bemerkung 1)
12) IDPLOTT	I * 4	Eingabe: Nummer zur Identifizierung der Zeichnung, 1 ≤ IDPLOTT ≤ 999 999.

Argumenten- Name	Typ	Bedeutung
12) IDPLOT	I * 4	<p>Ausgabe:</p> <p>Wenn kein Fehler festgestellt wurde ist IDPLOT unverändert.</p> <p>Falls IDPLOT beim Aufruf nicht im Bereich $1 \leq \text{IDPLOT} \leq 999\ 999$ liegt, wird IDPLOT wie folgt bestimmt: $\text{IDPLOT} = \text{MINO}(\text{MAXO}(\text{IABS}(\text{IDPLOT}), 1), 999999)$</p> <p>Wird in dem Unterprogramm PLOTA ein Fehler festgestellt, wird IDPLOT negativ gesetzt.</p> <p>Falls $\text{KMAX} < 7$, wird $\text{IDPLOT} = -1000\ 000$ gesetzt.</p> <p>IDPLOT wird zudem vom Unterprogramm PLOTA bei Fehlermeldungen angegeben.</p>
13) NTEXT	DIMENSION NTEXT(15)	<p>Ein aus 60 Zeichen bestehender Text, der als Unterschrift unter die Abbildung gesetzt wird; ist der Text kürzer als 60 Zeichen, so erscheint kein Text in der Zeichnung.</p>
14) XTEXT	beliebig	<p>Ein aus maximal 12 Zeichen bestehender Text, der zur Kennzeichnung der X-Koordinate verwendet wird; der Text muß mit zwei Punkten enden; z.B. 'X-ACHSE..'</p>
15) YTEXT	beliebig	<p>Text zur Kennzeichnung der Y-Koordinate; vergl. XTEXT.</p>
16) KMAX	I * 4	<p>Länge der Arbeitsfelder XP und YP; KMAX muß größer als 6 und sollte größer als $6 * (\text{NX} + \text{NY}) - 2$ sein.</p> <p>Für $\text{KMAX} < 7$ endet PLOTHL mit Fehlermeldung $\text{IDPLOT} = -1000\ 000$; für $\text{KMAX} \leq (6 * (\text{NX} + \text{NY}) - 2)$ wird die Darstellung der Bereichsumrandung stark vereinfacht und im Falle nichtkartesischer Koordinaten zudem sehr ungenau.</p>

Argumenten-Name	Typ	Bedeutung
17) NXMAX	I $\kappa-4$	1. Angabe der DIMENSION-Anweisung für F in dem Programm, in dem F mit den Funktionswerten gefüllt wird.
18) XP	REAL XP(KMAX)	Arbeitsfeld
19) YP	REAL XP(KMAX)	"
20) S	LOGICAL $\kappa 1$ S(NX,NY) oder REAL $S(\frac{NX}{4}, \frac{NY}{4})$	"
21) DREI	LOGICAL (L $\kappa-4$)	= .TRUE., wenn der Mittelpunkt einer Masche mit zur Bestimmung der Höhenkurven herangezogen werden soll. =.FALSE. sonst. Für DREI=.TRUE. ist das Ergebnis genauer, der Zeichenaufwand aber größer.
22) KARTH	LOGICAL (L $\kappa 4$)	=.TRUE., wenn die X- und Y-Koordinaten als karthesische Koordinaten gezeichnet werden sollen; XFUN und YFUN sind dann überflüssig. =.FALSE., wenn die X- und Y-Koordinaten als Krümlinige Koordinaten ξ und η interpretiert werden sollen; XFUN und YFUN müssen definiert sein.
23) XFUN	EXTERNAL	FUNCTION-Unterprogramm zur Berechnung von $\xi = XFUN(X,Y)$
24) YFUN	EXTERNAL	FUNCTION-Unterprogramm zur Berechnung von $\eta = YFUN(X,Y)$.

Bemerkung 1:

XZ und YZ wurden eingeführt, um das Unterprogramm PLOTA leicht gegen andere PLOT-Routinen austauschen zu können. Hier wird die Zeichnung stets mit folgenden Maßen XZ' und YZ' erzeugt.

YZ' = 0.254 m

XZ' = INDZ * YZ' mit

INDZ = (1.01 * XZ)/YZ

INDZ = MAXO (MINO(INDZ,4),1)

Es wird empfohlen YZ = 0.254 zu setzen und

für	INDZ = 1	XZ = 0.254
	INDZ = 2	XZ = 0.508
	INDZ = 3	XZ = 0.762

zu verwenden.

2) CALL NOCHHL

Dieser Aufruf kann nur nach einem Aufruf von PLOTHL erfolgen; seit dessen Aufruf sollten sinnvoll lediglich F und DREI nicht unverändert sein.

3) CALL ZUHL (XP,YP,K,NT,NP,NPG,INT,IDNV)

XP	REAL XP(K)	Feld der X-Koordinaten der Zusatzkurven.
YP	REAL YP(K)	Feld der Y-Koordinaten der Zusatzkurven
K	I * 4	Zahl der Zusatzkoordinaten $K \geq 2$
NT	I * 4	siehe PLOTA Argument 4 (Anhang 2)
NP	I * 4	" " " 5
NPG	I * 4	" " " 6
INT	I * 4	" " " 7
IDNV	I * 4	" " " 8 (NPA)

3. Programmlogik

3.1 Erzeugung der Umrahmung und der Beschriftung

Für $7 \leq KMAX < (6 * (NX+NY)-1)$ wird die Umrahmung als Verbindung der vier Eckpunkte der Umrandung des Maschennetzes gezeichnet; für $(6 * (NX+NY)-1) \leq KMAX$ wird die Umrahmung so gezeichnet, daß sie gleichzeitig Beginn und Ende aller Maschenlinien angibt.

Die Beschriftung mit den Texten NTEXT, XTEXT, YTEXT wird mit PLOTA vorgenommen.

Die Angaben der Höhenwerte zu den Punktzeichen auf den Höhenkurven erfolgt durch Wandeln der Höhenwerte in Texte mit dem Unterprogramm CONVX [3] und Zeichnen mit PLOTA [1]. Da PLOTA verlangt, daß mindestens 2 Punkte in einem Aufruf gezeichnet werden, wird das zu erläuternde Zeichen für die Legende zweimal am gleichen Ort gezeichnet.

3.2 Ermittlung der Koordinaten der Höhenlinien

Die Ermittlung der Höhenlinienkoordinaten erfordert zwei Schritte:

- a) Auffinden eines Punktes der zu der Höhenlinie gehört, zuerst auf dem Rand und dann im Innern.
- b) "Wandern" entlang der Höhenlinie bis diese entweder auf dem Rand endet oder sich zu einem Ring schließt.

Hierbei werden folgende Tatsachen ausgenutzt:

- eine Höhenlinie, die auf dem Rand beginnt, muß auf dem Rand enden.
- alle Höhenlinien, die nicht auf dem Rand enden oder beginnen, bilden Ringe.

Bei der Ermittlung der Höhenlinien müssen einmal durchwanderte Linien als abgearbeitet markiert werden, damit später, wenn an einer anderen Stelle ein Punkt gefunden wird durch den eine Höhenlinie der entsprechenden Höhe führt, entschieden werden kann, ob dieser Punkt zu einer bereits abgearbeiteten oder neuen Höhenlinie gehört. Diese Kennzeichnung geschieht mit den logischen Variablen des Feldes $(S(I,J), I = 1,2, \dots, NX, J=1,2, \dots, NY)$ nach folgendem Verfahren (vergl. Abb. 2):

Das Feld $S(I,J)$ wird zu Beginn mit `.FALSE.` initialisiert.

$S(IM,J)$ mit $1 \leq IM < NX$, $1 \leq J \leq NY$

wird `.TRUE.` gesetzt, wenn ein Höhenlinienpunkt gefunden wird zwischen den Netzpunkten

$$\{X(I), Y(J)\} \quad \text{und} \quad \{X(I+DI), Y(J)\}$$

mit $DI = \pm 1$ und $IM = \text{MINO}(I, I+DI)$,

d.h. immer dann, wenn ein Höhenlinienpunkt gefunden wird auf einer Linie $Y = \text{const}$ ($J = \text{const}$).

Damit können also alle Höhenlinien als abgearbeitet markiert werden, die zumindest einmal eine Linie $J = \text{const}$ schneiden. Alle Höhenlinien, die nicht Linien mit $J = \text{const}$ schneiden, müssen aber die Linie mit $I = 1$ schneiden, da zwischen zwei Linien $J = \text{const}$ und $J = \text{const} + 1$ keine Höhenlinien Ringe bilden kann ohne Linien $J = \text{const}$ zu schneiden. Wenn daher eine Höhenlinie die Maschenlinie $I = 1$ zwischen J und $J+DJ$ ($DJ = \pm 1$) schneidet, wird

$$S(NX, JM) = \text{.TRUE.} \text{ gesetzt, mit } JM = \text{MINO}(J, J+DJ);$$

wie oben angegeben bleiben diese Plätze des Feldes S nach der ersten Bedingung noch ungenutzt.

Diese Kennzeichnung ist ausreichend, da auf einer Maschenbegrenzungslinie nur ein Höhenlinienpunkt liegen kann.

Abgesehen von der Linie $I = 1$ werden bei diesem Verfahren also nur Linien $J = \text{const}$ zur Kennzeichnung abgearbeiteter Höhenlinien berücksichtigt. Dies hat zum Vorteil, daß nicht noch ein zweites Feld logischer Variablen zur Kennzeichnung der Linien $I = \text{const}$ notwendig ist und zum Nachteil, daß bei Höhenlinienpunkten auf Linien $I = \text{const} \neq 1$ erst nach einiger "Wanderung" auf dieser Linie entschieden werden kann, ob diese Linie bereits abgearbeitet wurde oder nicht. Diese Wanderung dauert am längsten bei Höhenlinien, deren Anfangspunkt auf der Linie $I = NX$ liegt und die bei $I = 1$ endet ohne eine Linie $J = \text{const}$ geschnitten zu haben; hier kann erst am Ende entschieden werden, ob diese Höhenlinie bereits abgearbeitet war. Aus diesem Grund beginnt die Suche nach Höhenlinien auf der Maschenlinie $I = NX$, da zu Beginn jede gefundene Höhenlinie noch nicht abgearbeitet ist und die Wanderung auf ihr also nicht unnötig ist.

Der Schritt a), Auffinden eines Höhenlinienpunktes, wird so durchgeführt, daß zunächst auf dem Rand und dann auf Maschenlinien im Inneren ein Intervall gesucht wird, für das gilt:

$$(A-H) * (B-H) < 0 \quad (1)$$

wobei A und B die Funktionswerte an den Intervallgrenzen und H der Höhenwert ist [2].

Wenn A zum Pkt. (I1, J1) und B zum Pkt. (I2, J2) gehört, wird (I,J) mit I=MIN (I1,I2) und J=MIN(J1,J2) als Ausgangspunkt gemerkt; in KS wird notiert ob die Maschenlinie eine Linie I = const. (KS=1) oder J = const. (KS=2) ist. In DI und DJ wird gemäß Abb. 3 die Wanderrichtung mit + 1 oder - 1 angezeigt. Der Zähler K für die Anzahl der Höhenlinienpunkte wird auf 1 gesetzt und in den Arbeitsfeldern XP(1) und YP(1) wird der Höhenlinienpunkt eingetragen sowie im Inneren zusätzlich in XPl und YPl vermerkt. Der Höhenlinienpunkt hat die Koordinaten, die sich zum einen aus der Koordinate der Maschenlinie und zum anderen aus

$$Z = Z1 + (Z2-Z1) * (H-B)/(A-B) \quad (2)$$

berechnen, wobei Z die Koordinate parallel zur Maschenlinie ist mit Z1 dem Wert beim Punkt (I1,J1) und Z2 dem Wert beim Punkt (I2,J2).

Anschließend wird Schritt b), das Wandern auf der Höhenlinie, in Angriff genommen (beginnend bei Statement mit Label l0):

Gemäß Abb. 1 werden zunächst die Größen F1, F2, F3 und F4 bestimmt und, falls DREI =.TRUE. ist, auch F5 = (F1+F2+F3+F4)/4. (Diese Formel ist hinreichend genau, wenn die Maschen genügend klein sind, da eine stetige Fläche vorausgesetzt wird).

Ist DREI =.TRUE., so soll der nächste Höhenlinienpunkt je nach Ausgangslinie zwischen F1 und F5 oder F2 und F5 (KS=2) oder zwischen F4 und F5 oder F1 und F5 (KS=1), also auf einer Diagonalen (vergl. Abb. 1) gesucht werden. Hierbei kann nur eine der beiden Möglichkeiten zutreffen, da es nicht möglich ist, daß eine Höhenlinie alle drei Begrenzungslinien des Dreiecks F2-F2-F5 oder F1-F4-F5 schneidet. Gemäß Formel (1) und (2) kann also leicht entschieden werden, wie die Höhenlinie verläuft und durch welchen Punkt sie zu zeichnen ist. Dieser wird in den Arbeitsfeldern XP(K) und YP(K) eingetragen, wobei K erhöht wird.

Als nächstes ist in dem folgenden Dreieck ähnlich zu verfahren; entweder liegt der nächste Schnittpunkt ebenfalls auf der Diagonalen oder aber auf einem Maschenrand. Im ersten Fall ist noch einmal innerhalb der Masche zu suchen (usw.), im zweiten Fall wird nun die Masche verlassen.

Hierbei sind einige Kontrollvariablen (I,J,DI,DJ,KS) so zu verändern, wie Abb. 4 zeigt, sodaß anschließend fortgefahren werden kann wie wenn die neue Masche die erste gewesen wäre.

Vorher wird jedoch abgefragt, ob nicht der Rand erreicht wurde oder der neue Schnittpunkt auf der Maschengrenzlinie gleich dem Ausgangspunkt ist ($XP(K)=XP1$, $YP(K)=YP1$, Ring geschlossen), in welchen Fällen die Höhenlinie vollständig gefunden ist und gezeichnet werden kann. Hier wird auch überprüft, ob dieser Schnittpunkt bereits früher abgearbeitet wurde (das entsprechende S ist `.TRUE.`), sodaß diese Höhenlinie nicht weiter bearbeitet werden muß und also ein neuer Ausgangspunkt gemäß Schritt a) gesucht werden kann.

Sollte gemäß dem Eingabeparameter DREI=.FALSE. F5 nicht berücksichtigt werden, so fällt die Entscheidung, wo weitergewandert werden soll nicht ganz so leicht. Ausgehend z.B. von einem Schnittpunkt zwischen F1 und F2 (Abb. 1) können folgende Fälle gefunden werden:

- Schnittpunkt liegt entweder
zwischen F1 und F4 oder
zwischen F4 und F3 oder
zwischen F3 und F2

oder

- Schnittpunkte liegen
zwischen F1 und F4 und
zwischen F4 und F3 und
zwischen F3 und F2.

(alle "oder" sind hier exklusiv)

Bei der ersten der beiden Alternativen liegt kein Entscheidungsproblem vor; es wird die Masche auf der Seite verlassen, wo der Schnittpunkt gefunden wurde.

Bei der zweiten der beiden Alternativen gibt es aber die Wahl zwischen den Seiten F1-F4 und F3-F2. Da sich Höhenlinien nicht schneiden dürfen, entfällt F4-F5.

Diese Wahl kann nur unter Betrachtung der Mascheneckpunkte nicht entschieden werden. Es wird daher in diesem Fall so wie bei DREI=.TRUE. zusätzlich der Punkt in der Maschenmitte mit Funktionswert F5 berücksichtigt und wie oben beschrieben vorgegangen.

Alle bisherigen Betrachtungen setzten voraus, daß keiner der Funktionswerte $F(I,J)$ gleich der Höhe H ist (in diesem Fall würde die Bedingung (1) lauten $(A-H) * (B-H) = 0$). In diesem Programm wird dann, wenn festgestellt wird, daß die Höhenlinie exakt durch den Endpunkt einer Maschenseite verläuft, der Funktionswert A an diesem Punkt so verändert, daß

$$A = AEND(A,B) = A + 1.E-5 * (A-B) \quad (3)$$

gesetzt wird, wobei B der Funktionswert auf dem anderen Endpunkt der Maschenseite ist. Dies hat zur Folge, daß, wenn $B \neq H$ ist, anschließend die Höhenlinie die Maschenseite innerhalb der Endpunkte schneidet. War auch $B=H$, so findet das Programm hier keine Höhenlinie, was sinnvoll ist, da $F(x,y) = \text{const.}$ entlang Maschenseite ist.

Die Veränderung des Funktionswertes ist erforderlich, da bei Gleichheit mit dem Höhenwert sonst nicht nur die gerade betrachtete Masche sondern auch noch die benachbarten Maschen untersucht werden müssen, wenn der folgende Punkt der Höhenlinie gefunden werden soll.

Die bleibende Änderung des Funktionswertes im Eingabefeld F ist notwendig, weil sonst ein Funktionswert wechselnd in verschiedener Weise leicht modifiziert wird.

3.3 Reaktion für den Fall, daß die Arbeitsfelder XP und YP nicht ausreichen

Wird bei der Wanderung auf einer Höhenlinie festgestellt, daß diese eine Kurve durch mehr Punkte beschrieben wird als in den KMAX langen Arbeitsfeldern eingetragen werden können, wird die Wanderung unterbrochen und zunächst die Kurve so weit gezeichnet, wie sie gespeichert werden konnte und anschließend an der Unterbrechungsstelle die Wanderung fortgesetzt, und die neuen Punktkoordinaten werden über die alten gespeichert. Für die Abfrage,

die darüber entscheidet, ob sich die Höhenlinie zu einem Ring geschlossen hat, ist daher die gesonderte Speicherung des Anfangspunktes in XPl und YPl erforderlich; dies allerdings nur dann, wenn der Anfangspunkt nicht auf dem Rand, sondern im Innern gefunden wurde.

In der logischen Variablen READY wird vermerkt, ob die Kurve vollständig gezeichnet wurde oder nicht.

4. Verbesserungsmöglichkeiten

Im folgenden werden einige Vorschläge genannt, die die Anwendbarkeit der Routine PLOTHL vergrößern oder die Zeichnung verbessern. Die einzelnen Verbesserungen erfordern einen unterschiedlich großen Aufwand.

- a) Die Umrandung \square könnte durch zusätzliche Angabe einer Begrenzungskurve von den äußeren Grenzen des Maschennetzes abweichen. Soll der Wertebereich nicht einfach geschlossen sein, sind mehrere Begrenzungskurven erforderlich.

Hier treten folgende Probleme auf:

- Bestimmung der Funktionswerte auf den Begrenzungskurven, wenn diese nicht mit Maschengrenzen zusammenfallen.
- Bestimmung des Anfangs- und Endpunktes von Höhenlinien auf dem Rand.

- b) $F(x,y)$ wird nicht auf den Eckpunkten eines viereckigen Netzes, sondern für beliebige Koordinaten x,y angegeben.

Problem: Der Wertebereich muß in Dreiecke zerlegt werden, die als Maschen betrachtet werden. Diese Zerlegung ist nicht ohne weiteres eindeutig.

- c) Die Höhenlinien sind nicht als Polygonzüge, sondern geglättet anzugeben.

Problem: Höhenlinien sind nicht eindeutig Funktionen $x = f(y)$ oder $y = f^*(x)$; eine Interpolation mit Polynomen ist daher nicht möglich.

Eine Höhenlinie darf durch das Glätten nicht so verformt werden, daß sie eine Nachbarhöhenlinie schneidet. Nachbarhöhenlinien können bei beliebig kleinem Höhenunterschied beliebig nahe liegen.

Höhenlinien lassen sich daher wohl nur dadurch glätten, daß man $F(x,y)$ an mehr Stützstellen angibt, wobei die zusätzlichen Werte durch geeignete Interpolationen der Fläche $F(x,y)$ bestimmt werden müssen.

5. Beispiele

Abb. 5

Diese Abbildung zeigt den wohl schwierigsten Testfall für das Programm PLOTHL. Das dargestellte Feld $F(I,J)$ wurde durch den Pseudozufallszahlgenerator RN [4] erzeugt.

$$F(I,J) = RN(O).$$

PLOTHL wurde mit folgenden Parametern aufgerufen:

```
NX = 11; NY = 11; X(I) = 1.+(I-1)*.1;
Y(J) = 1+(I-1)+.1; NH = 7; HV und NV siehe Legende;
IDNV = 30; XZ = .254; YZ = .254; IDPLOT = 5; NTEXT, XTEXT, YTEXT
siehe Abb.; KMAX = 1800, NXMAX = 30, DREI = .TRUE., KARTH = .TRUE..
```

Abb. 6

Diese Abbildung unterscheidet sich von der ersten nur durch IDPLOT sowie DREI=.FALSE..

Man erkennt, daß die Höhenkurven gröber sind aber das Feld etwa gleich gut beschreiben.

Abb. 7

Die Abbildung unterscheidet sich von Abb. 5 durch KARTH=.FALSE. und

$$XFUN(X,Y) = X*SIN(Y)$$

$$YFUN(X,Y) = X*COS(Y)$$

Abb. 8

Diese Abbildung zeigt sieben Höhenlinien einer analytischen Fläche. Die Funktionswerte werden berechnet nach der Formel

$$F(i,J) = 1. - X(i)^2 * Y(J)^2$$

mit $X(i) = -.5 + (i-1) * .1$

$Y(J) = -.5 + (J-1) * .1$

Die übrigen Parameter entsprechen den Angaben zu Abb. 6. Insbesondere ist DREI=.FALSE.

Abb. 9

Abb. 9 zeigt das gleiche Feld wie Abb. 8, jedoch mit DREI=.TRUE. Hier wird offensichtlich, daß die Berücksichtigung des Punktes mit Funktionswert F5 nicht immer ein genaueres Bild liefert. In Abb. 9 sind die Höhenkurven glatter.

Abb. 10

In dieser Abbildung wird ein nichtanalytisches Feld dargestellt. Es sei dem Leser überlassen, anhand der Höhenlinien und der Legende zu erkennen, wie das Feld definiert ist. Dieses Bild zeichnet sich durch extrem lange Höhenkurven aus und diene daher als Testfall.

Abb. 11

Diese Abbildung soll noch einmal veranschaulichen, daß PLOTHL für krummlinige Koordinatensysteme und sehr stark variierende Feldwerte geeignet ist. Gezeichnet ist hier ein randomes Feld

$$F(i,J) = RN(0)$$

mit $NX=20$; $NY=30$; $X(i)=(i-1) * .1 / (NX-1)$;

$Y(J) = \pi * (J-1) * .1 / (NY-1)$; $NH=3$; $DREI=.TRUE$;

$KARTH=.FALSE.$; XFUN und YFUN wie bei Abb. 7

6. Speicherbedarf, Rechenzeit, Unterprogramme

Das Unterprogramm PLOTHL wurde mit dem Fortran-G-Compiler übersetzt.

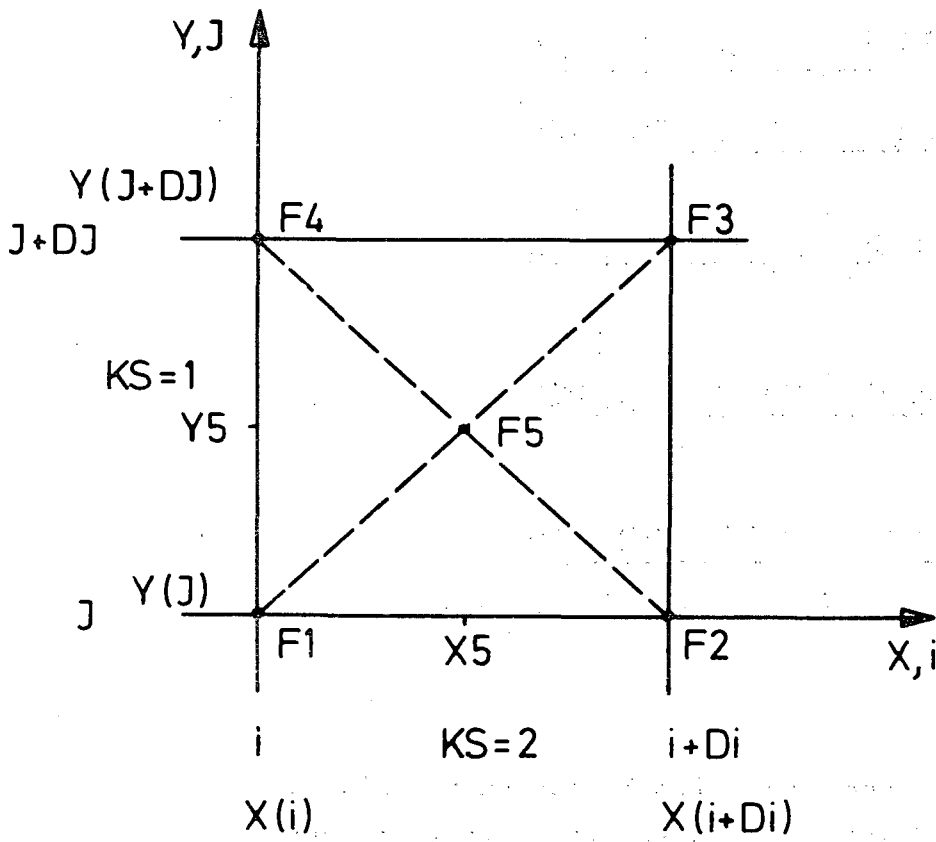
Es benötigt 15840 Byte Speicherplatz.

Es werden außer PLOTA und CØNVX keine Unterprogramme aufgerufen.

Für die Erzeugung der Abb. 6 wurden auf der IBM 360/85 weniger als 1/4 Sekunde CPU-Zeit benötigt.

Literatur

- [1] S.Heine: PLOTA, Ein verallgemeinertes Plot-Programm,
GfK, Programmbeschreibung Nr. 117 (1967)
(unveröffentlicht; die wichtigsten Argumente sind im
Anhang erläutert.)
- [2] J.Kaellis: Unsteady Compled High Flux-Mass, Momentum Transfer
with A Moving Boundary ANL-7797 (March 1971), S. 69
- [3] K.Gogg: CONVX, Fortran-Unterprogramm für die IBM / 360 zur
Umwandlung von in maschineninterner Darstellung vorliegenden
Fest- und Gleitkommazahlen in alphanumerische Darstellung.
GfK; Programmbeschreibung Nr. 243 (1970)
(unveröffentlicht; eine Kurzbeschreibung ist im Anhang ge-
geben.)
- [4] D.S.Seraphin: A Fast Random Number Generator for IBM 360.
Comm. of the ACM 12 (1969), 695



$$\begin{aligned} F1 &= F(i, J) \\ F2 &= F(i+Di, J) \\ F3 &= F(i+Di, J+DJ) \\ F4 &= F(i, J+DJ) \\ F5 &= (F1+F2+F3+F4)/4 \\ X5 &= (X(i)+X(i+Di))/2 \\ Y5 &= (Y(J)+Y(J+DJ))/2 \end{aligned}$$

Abb. 1 Die wichtigsten Variablen und ihre Zuordnung zum Maschennetz

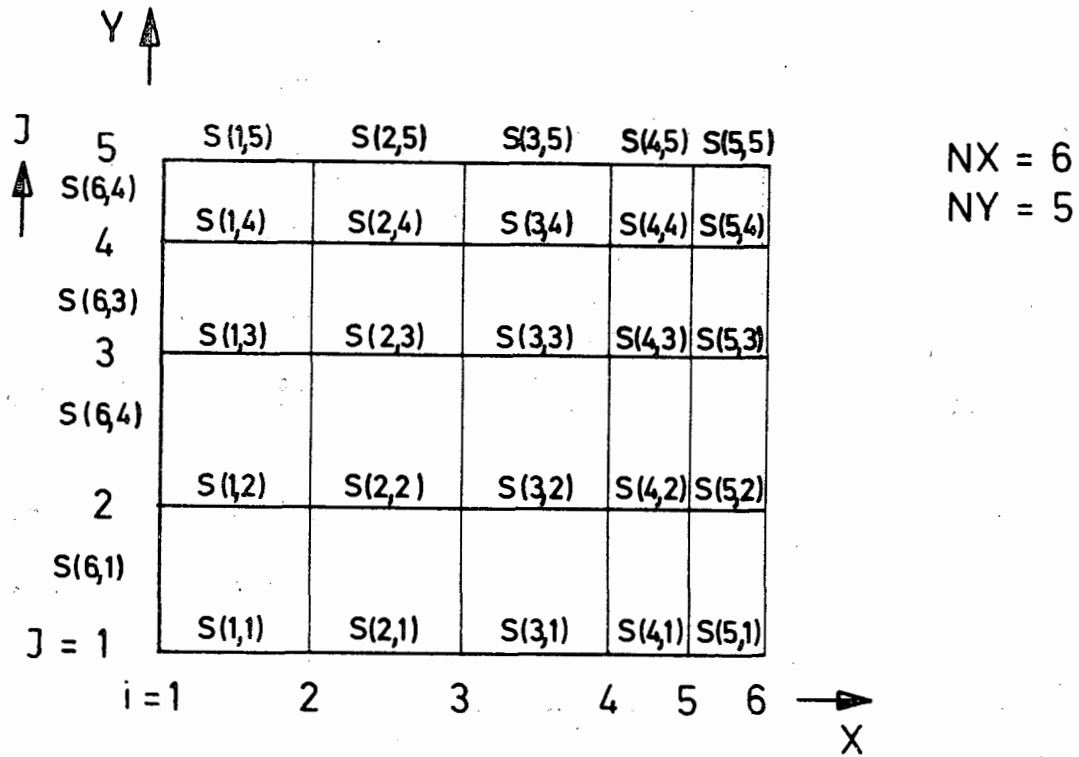


Abb. 2 Kennzeichnung der Maschenbegrenzungslinien mit Höhenlinienpunkten, die bereits abgearbeitet sind

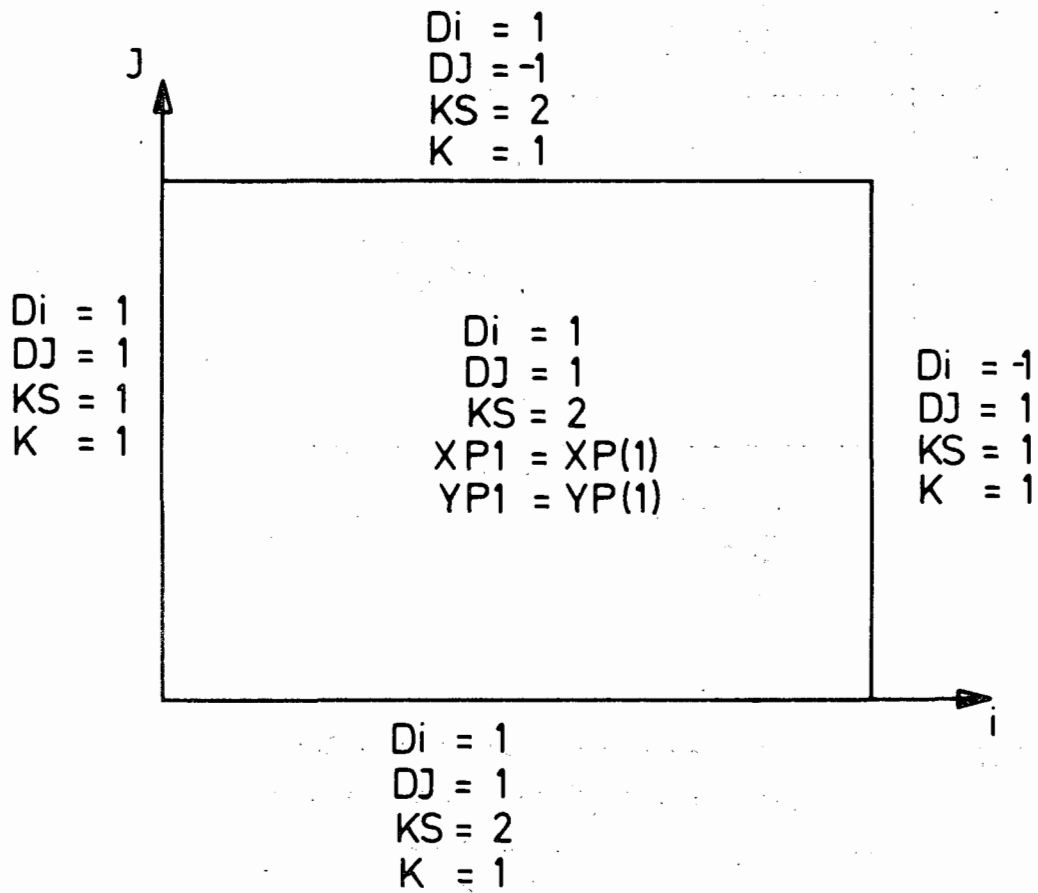


Abb. 3 Anfangswerte für das Abarbeiten einer Höhenlinie

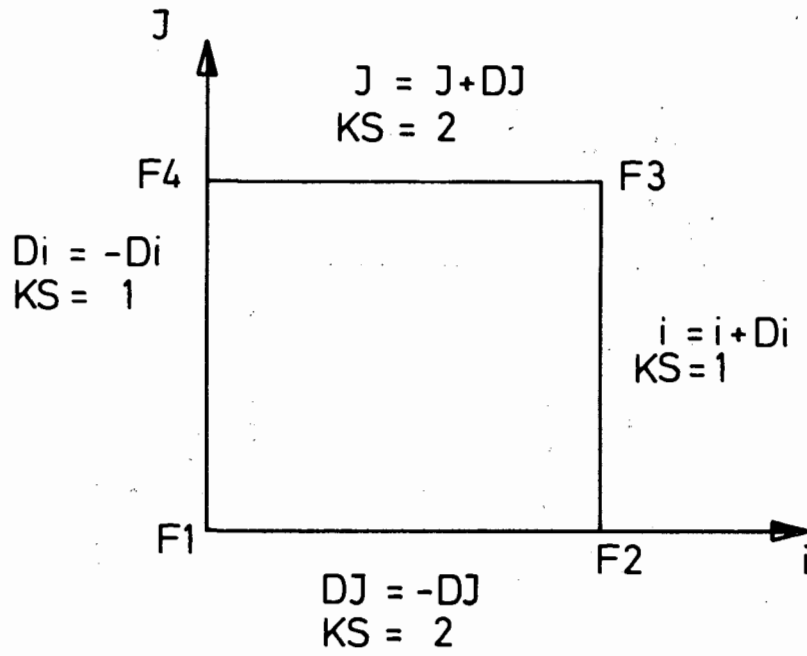
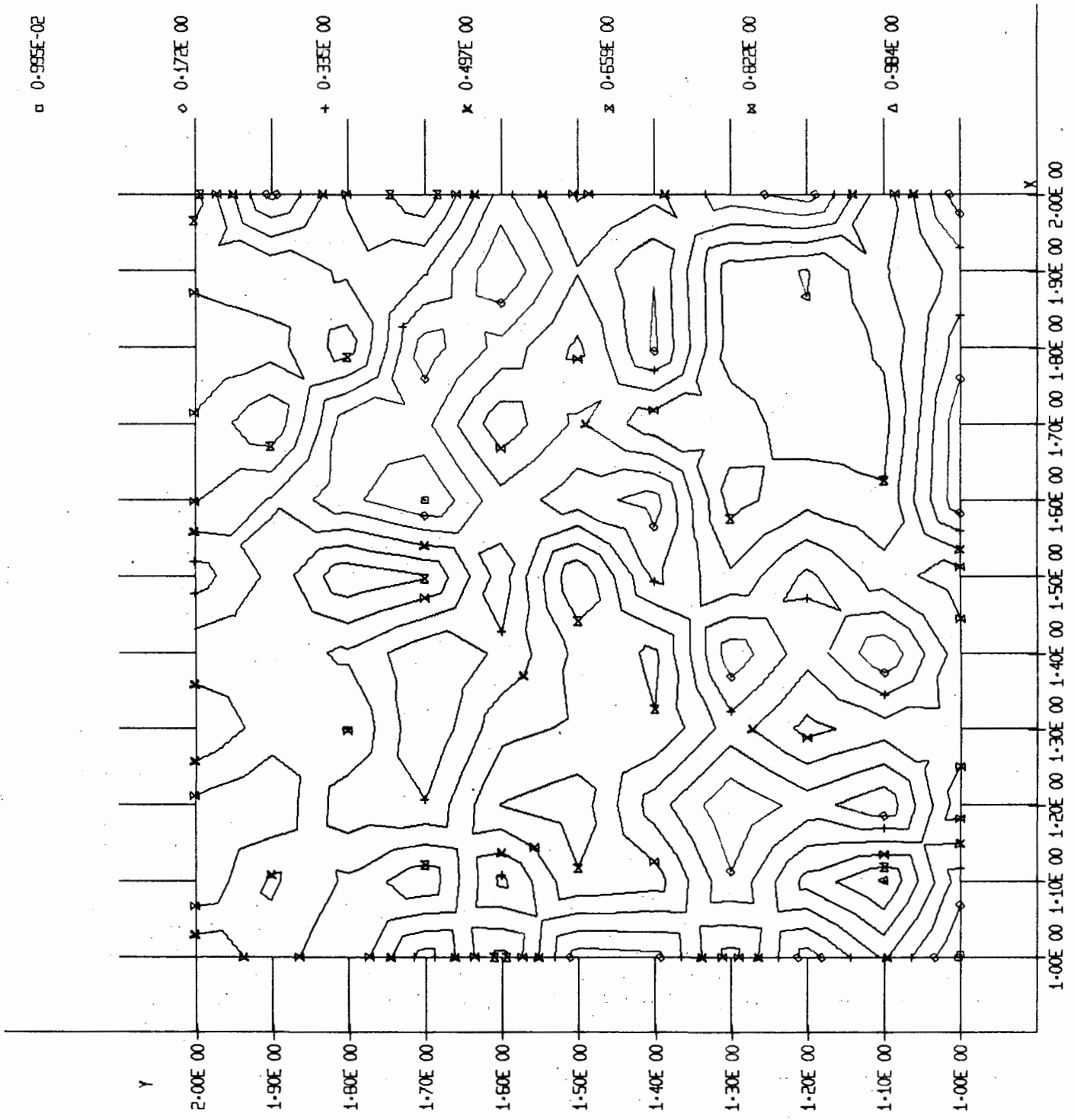


Abb. 4 Änderungen an den Kontrollvariablen i , J , Di , DJ und KS bei Überschreiten der verschiedenen Maschengrenzen



□ 0.955E-02

○ 0.172E 00

+ 0.335E 00

× 0.497E 00

⊗ 0.659E 00

∞ 0.822E 00

△ 0.984E 00

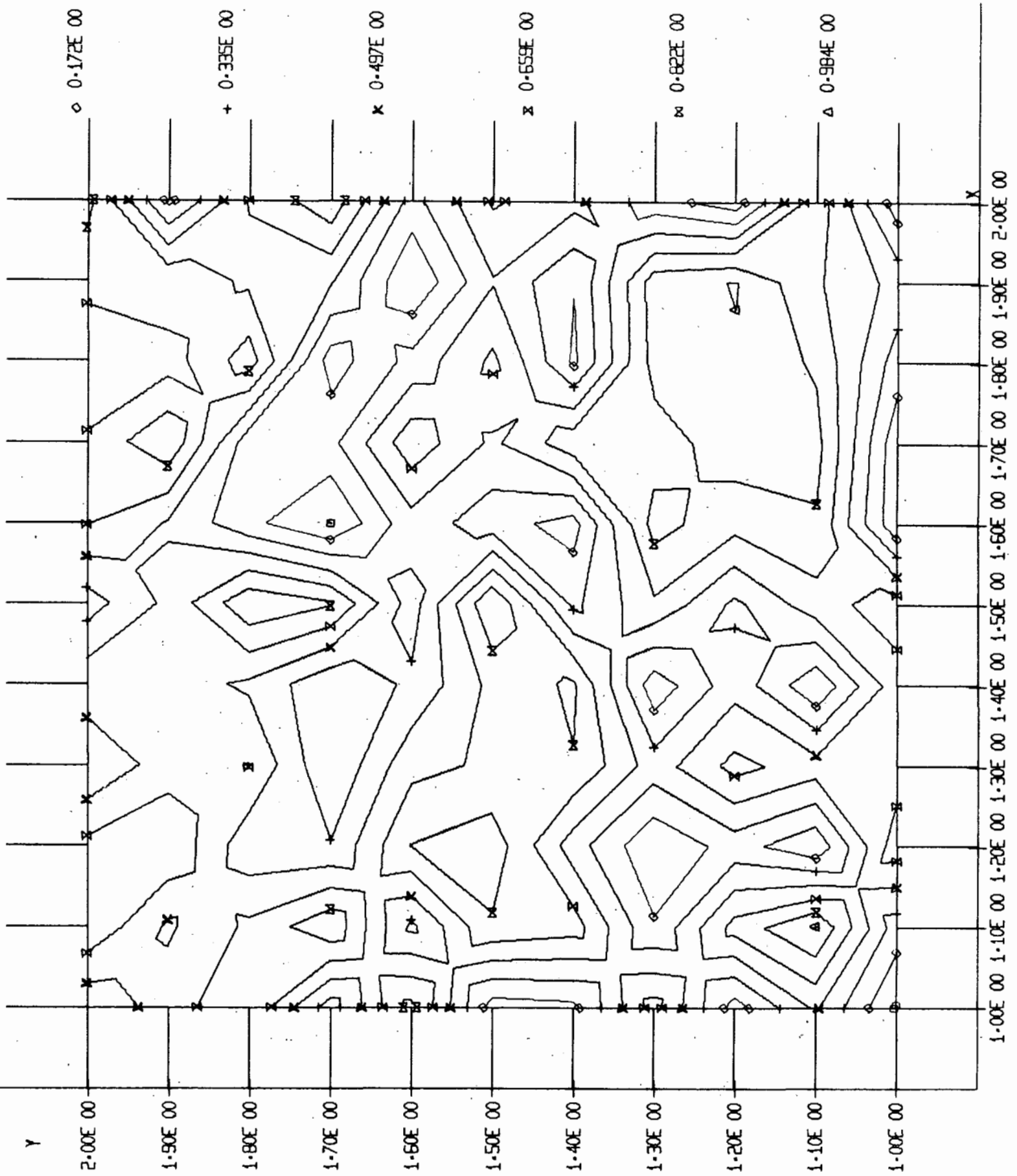


ABB-00006 H.-TEST

□ 0.995E-02

◇ 0.172E 00

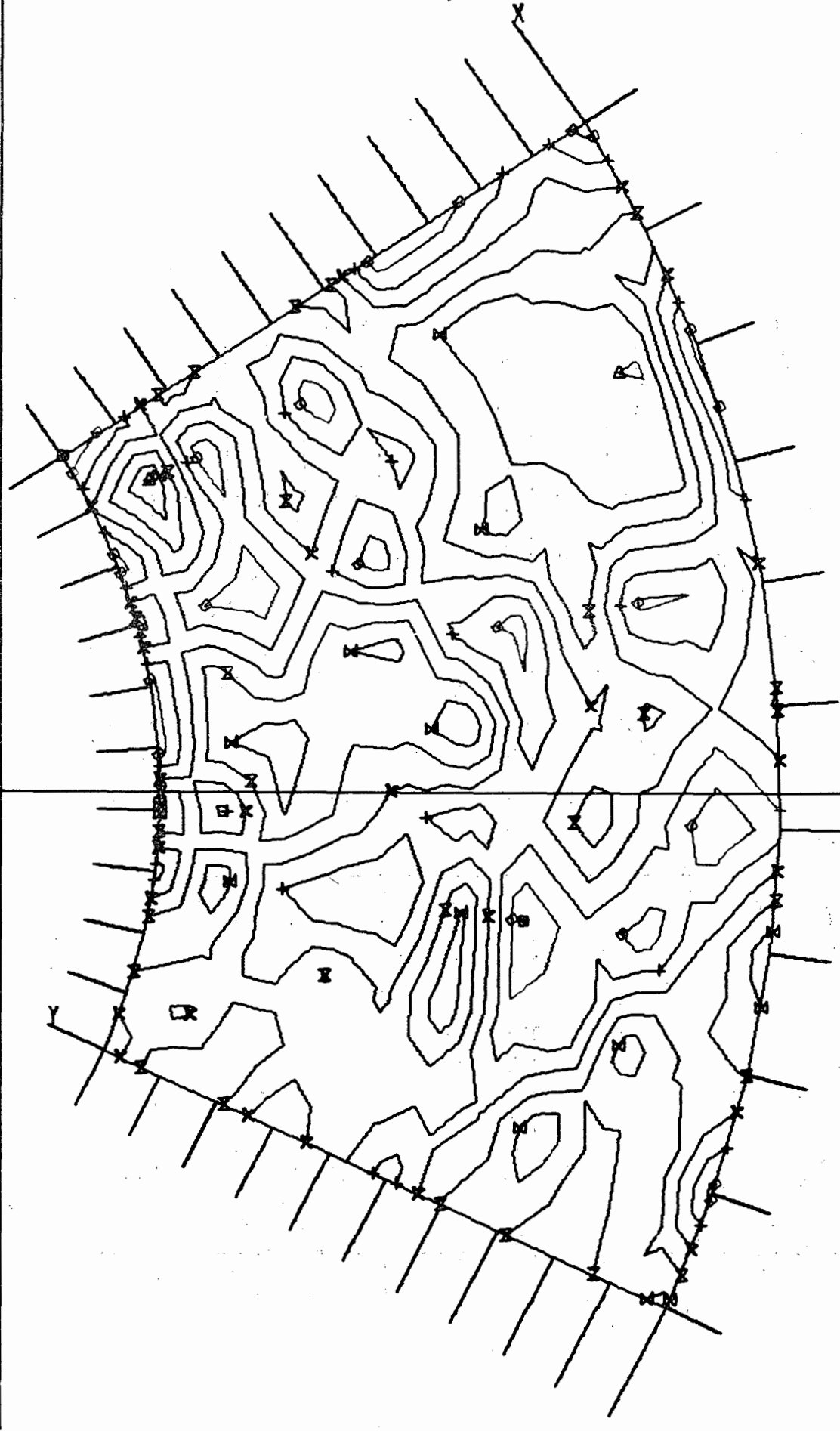
+ 0.335E 00

× 0.497E 00

⊗ 0.659E 00

⊠ 0.822E 00

△ 0.984E 00



□ 0.588E 00

○ 0.948E 00

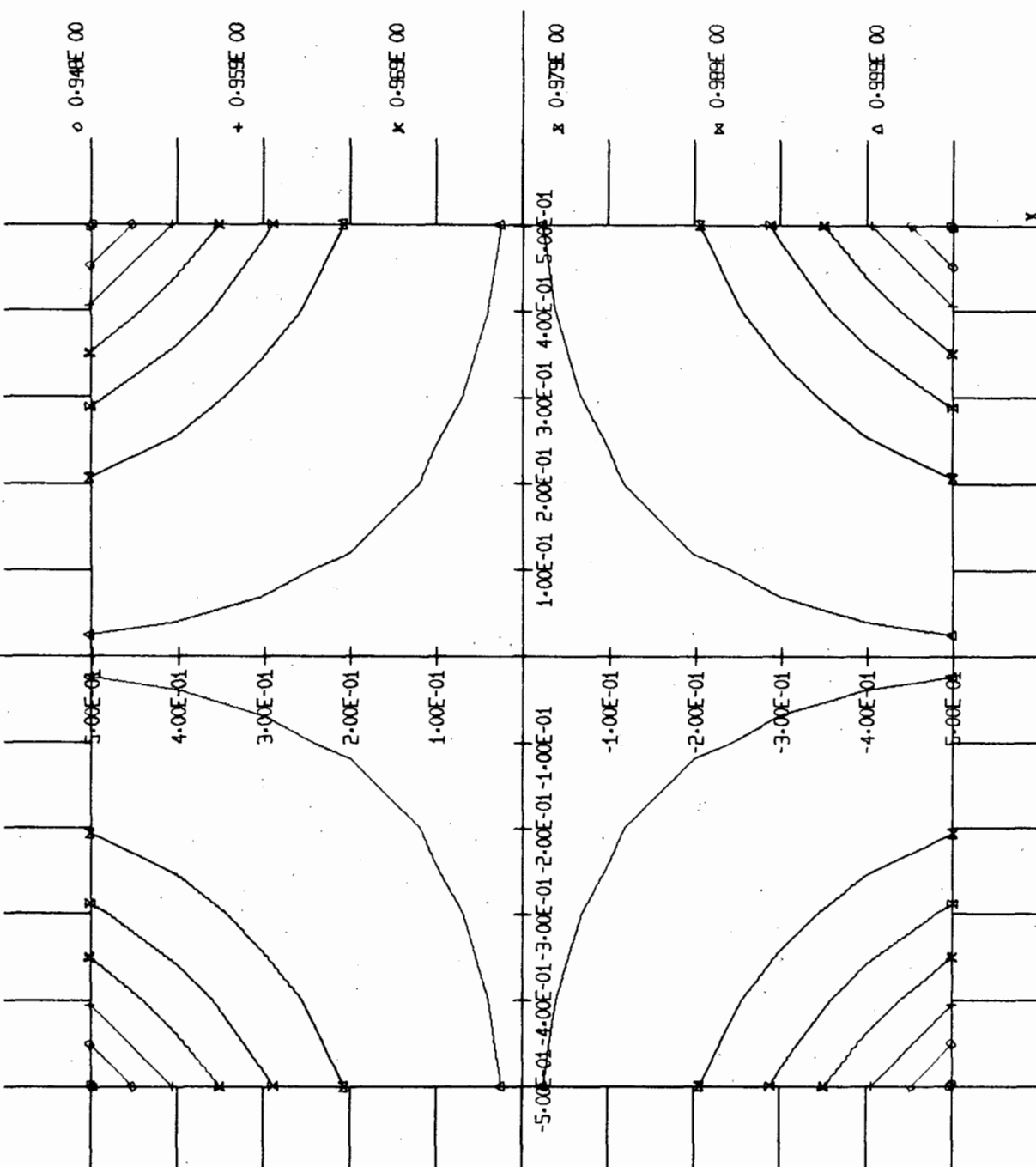
+ 0.955E 00

* 0.965E 00

× 0.975E 00

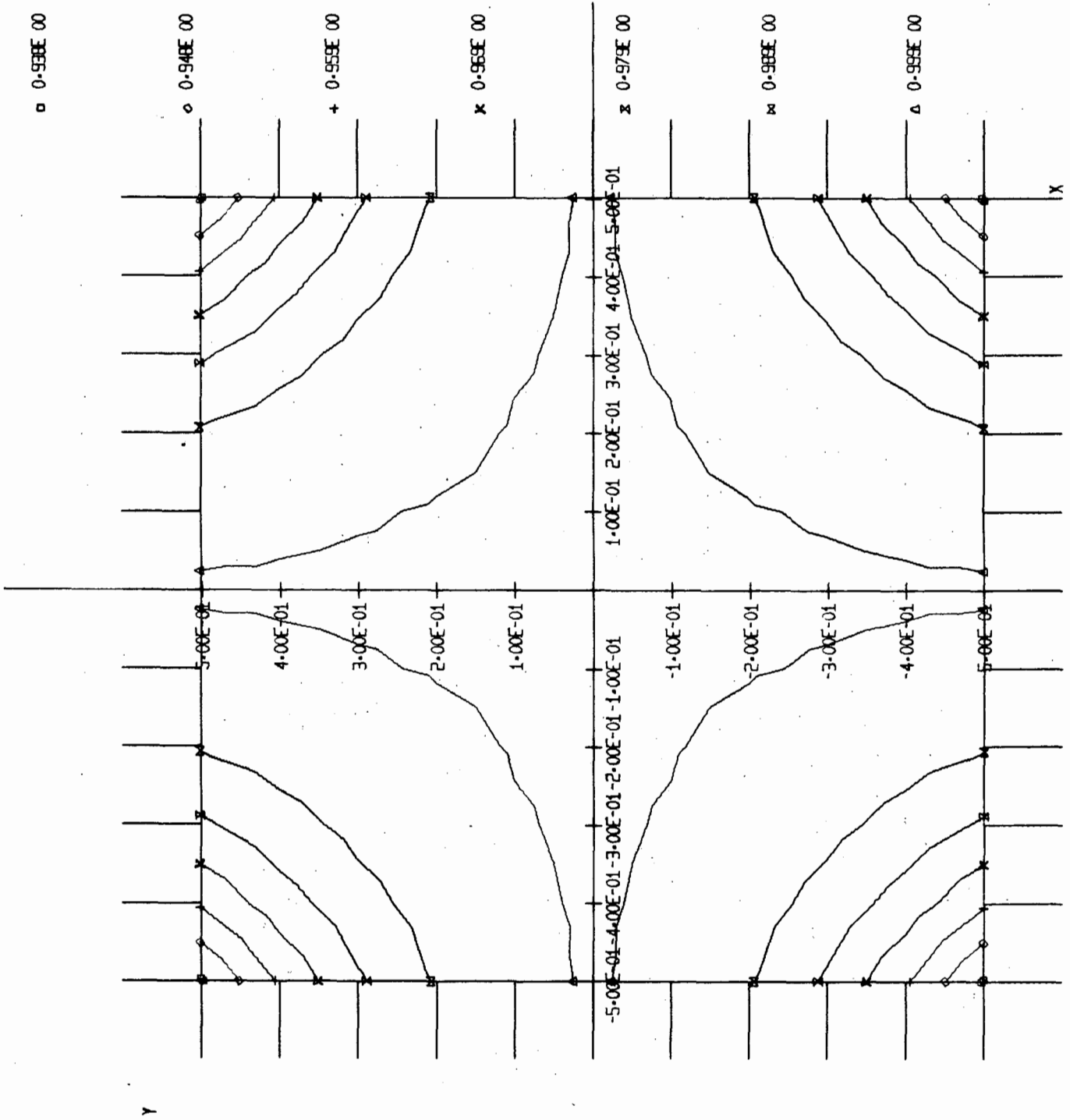
⊠ 0.985E 00

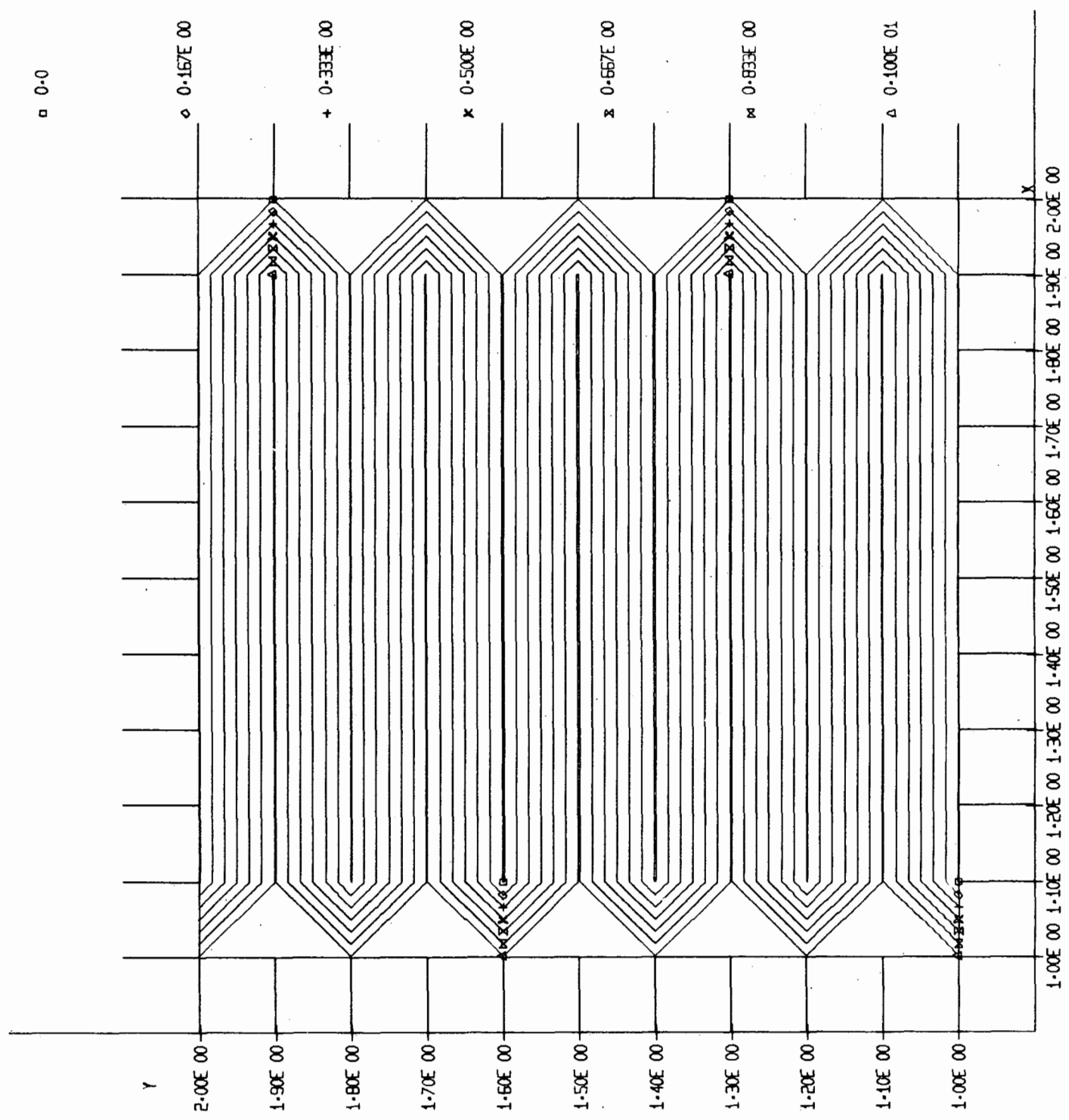
△ 0.995E 00



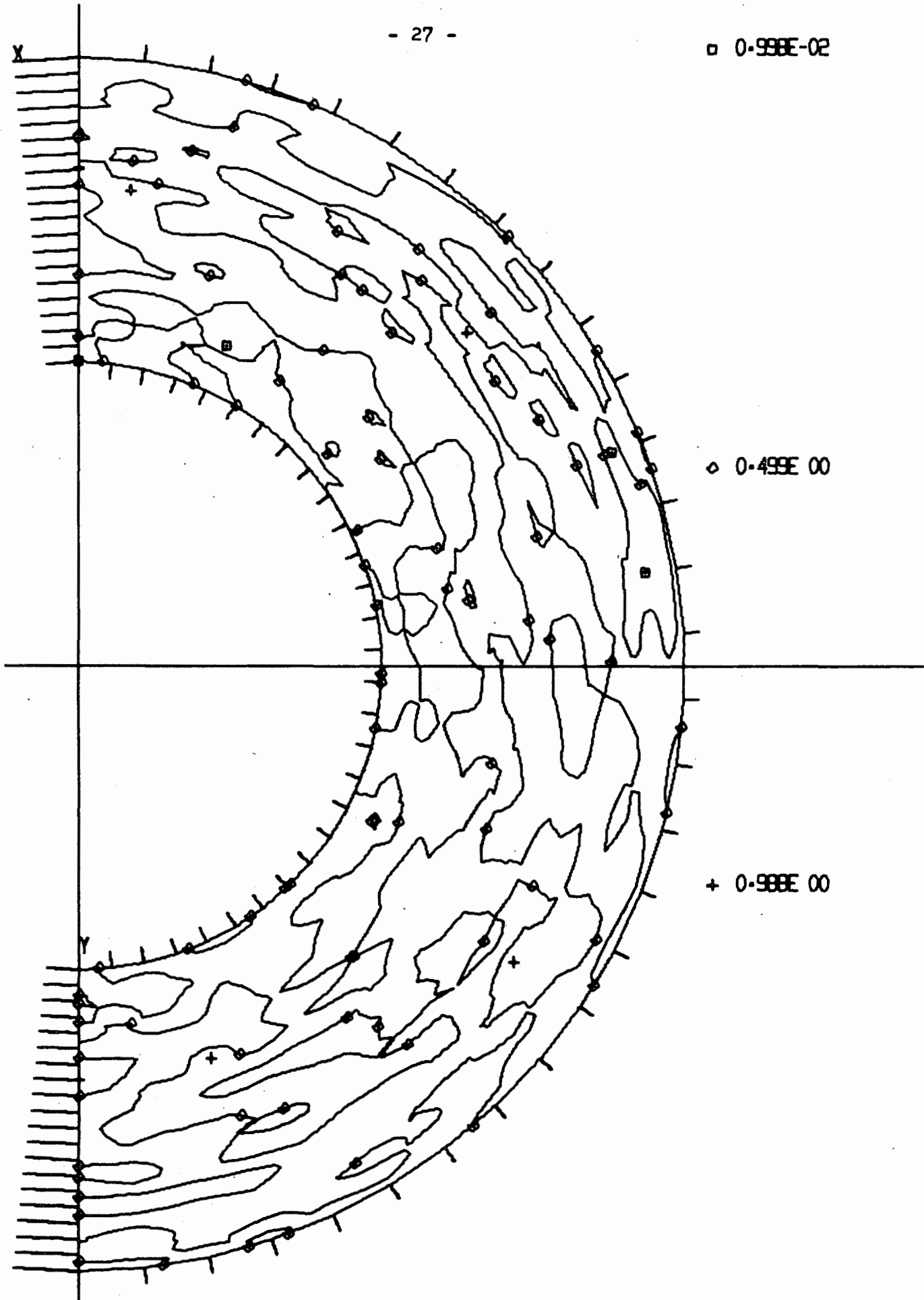
Y

X



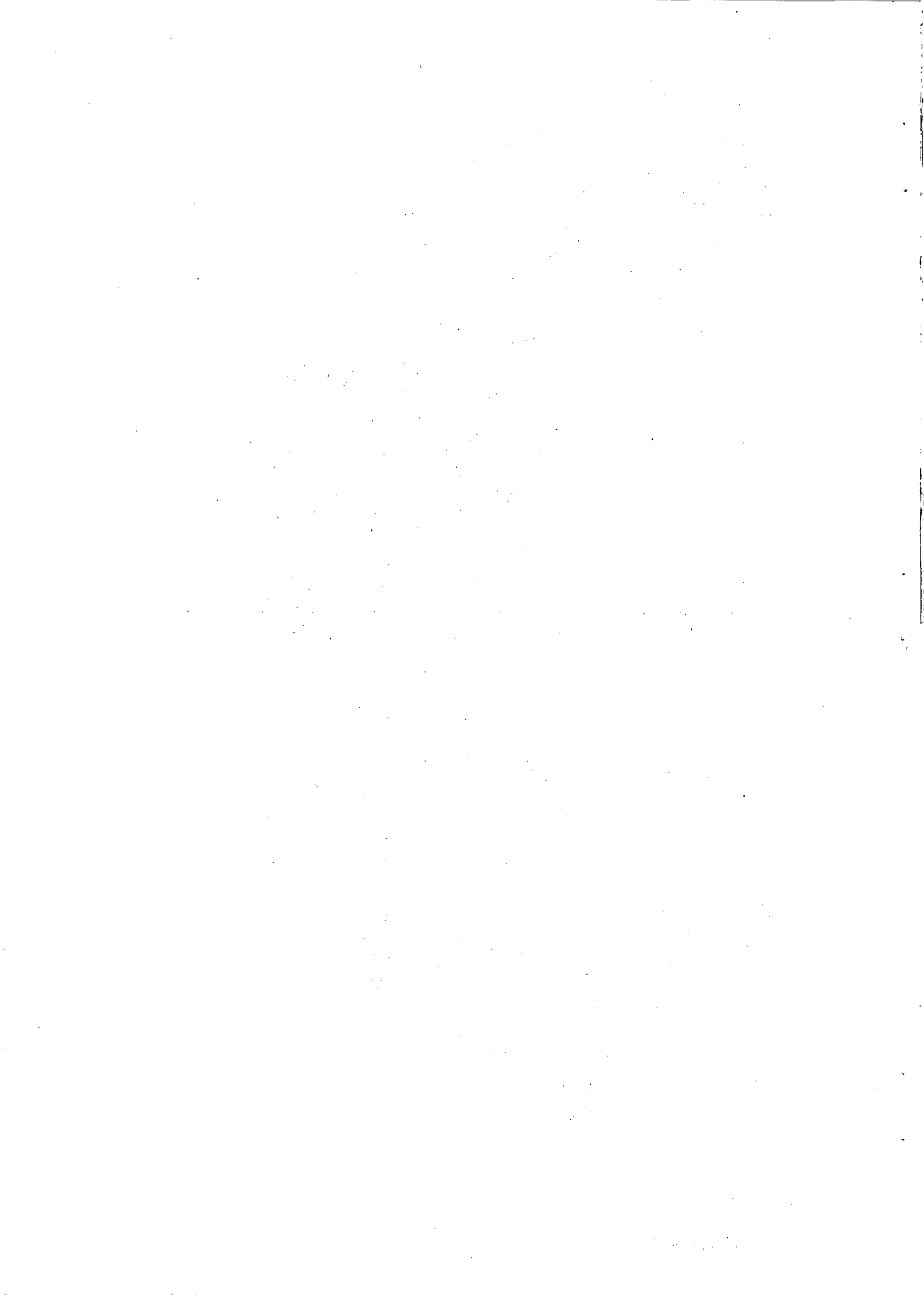


□ 0.988E-02



◇ 0.499E 00

+ 0.988E 00



A N H A N G 1

FORTRAN-LISTEN DES UNTERPROGRAMMS PLOTHL

SUBROUTINE PLOTHL(/NX/,/NY/,F,X,Y,/NH/,HV,NV,/IDNV/,/XZ/,/YZ/,
1/IDPLOT/,NTEXT,XTEXT,YTEXT,KMAX,NXMAX,XP,YP,S
1,/DREI/,/KARTH/,XFUN,YFUN)

C
CN PLOTHL
C
CB EIN FORTRAN IV UNTERPROGRAMM ZUR DARSTELLUNG VON FUNKTIONEN VON
CB ZWEI UNABHAENGIGEN VARIABLEN DURCH IHRE HOEHENLINIEN AUF EINEM
CB PLOTTER
C
CA ULRICH SCHUMANN
C
CD 01.10.71
C
CB KFK 1486
C
CS CONVX, PLOTA (SIEHE ANHANG 2)
C
LOGICAL*1 S(NX,NY),TRUE/.TRUE./,FALSE/.FALSE./,NOTLZW
1,SI,SJ,SIP1,SJP1,LKREIS
1,READY
LOGICAL DREI,KARTH
LOGICAL*1PKT/'.'//
1,LRAND,LZW
NOTLZW(A,H,B)=(A-H)*(B-H).GE.0.
LZW (A,H,B)=(A-H)*(B-H).LT.0.
SCHNIT(X1,H,X2,Y1,Y2)=Y1+(Y2-Y1)*(H-X1)/(X2-X1)
AEND(A1,A2)=A1+1.E-5*(A1-A2)
DIMENSION NTEXT(15)
DIMENSION F(NXMAX,NY),X(NX),Y(NY),HV(NH),NV(NH),XP(1),YP(1)
DIMENSION YTEXT(1),XTEXT(1)
INTEGER NDIR(2)/2,2/,NSC(2)/1,1/
C KS=1 I=CONST.
C KS=2 J=CONST.
INTEGER DI,DJ
IF(KMAX.LT.7)GOTO997
IDPLOT=MAX0(IABS(IDPLOT),1)
IDPLOT=MIN0(IDPLOT,999999)
NX1=NX-1
NY1=NY-1
INDZ=(XZ*1.01)/YZ
INDZ=MIN0(4,INDZ)
INDZ=MAX0(INDZ,1)
XMIN=2.*X(1)-X(2)
XMAX=2.*X(NX)-X(NX1)
YMAX=2.*Y(NY)-Y(NY1)

```

YMIN=2.*Y(1)-Y(2)
SX=(XMAX-XMIN)/(1000.*INDZ-110.)
VX=.01*SX
SY=(YMAX-YMIN)/1000.
VY=.01*SY
C BEGINN DER UMRAHMUNG
IF(KMAX.GE.6*(NX+NY)-1)GOTO310
XP(1)=X(1)
XP(2)=X(NX)
XP(3)=X(NX)
XP(4)=X(1)
XP(5)=X(1)
YP(1)=Y(1)
YP(2)=Y(1)
YP(3)=Y(NY)
YP(4)=Y(NY)
YP(5)=Y(1)
K=5
GOTO 311
310 K=0
J=1
DO 300 I=1,NX
DO 301 L=1,3
XP(K+L)=X(I)
301 YP(K+L)=Y(J)
YP(K+2)=YMIN
K=K+3
300 CONTINUE
K=K-1
I=NX
DO 302 J=1,NY
DO 303 L=1,3
XP(K+L)=X(I)
303 YP(K+L)=Y(J)
XP(K+2)=XMAX
K=K+3
302 CONTINUE
K=K-1
J=NY
DO 307DI=1,NX
I=NX+1-DI
DO 304 L=1,3
XP(K+L)=X(I)
304 YP(K+L)=Y(J)
YP(K+2)=YMAX
K=K+3
307 CONTINUE
I=1
K=K-1
DO 305 DJ=1,NY
J=NY+1-DJ
DO 306 L=1,3
XP(K+L)=X(I)
306 YP(K+L)=Y(J)
XP(K+2)=XMIN
K=K+3
305 CONTINUE

```

```

311 CONTINUE
C   Y-ACHSEN-BESCHRIFTUNG
    YP(K+1)=YMIN+.965*(YMAX-YMIN)
    XP(K+1)=XMIN-55.*SX
C   X-ACHSEN BESCHRIFTUNG
    XP(K+2)=X(NX)+5.*SX
    YP(K+2)=YMIN
    XA=X(1)
    XE=X(NX)
    YA=Y(1)
    YE=Y(NY)
    IF(KARTH)GOTO312
    XMIN=1.E70
    YMIN=1.E70
    XMAX=-1.E70
    YMAX=-1.E70
    DO 313 I=1,K
    A=XFUN(XP(I),YP(I))
    YP(I)=YFUN(XP(I),YP(I))
    XP(I)=A
    XMIN=AMIN1(A,XMIN)
    XMAX=AMAX1(XMAX,XP(I))
    YMIN=AMIN1(YMIN,YP(I))
    YMAX=AMAX1(YMAX,YP(I))
313 CONTINUE
    DYY=YMAX-YMIN
    YMAX=YMAX+0.03*DYY
    YMIN=YMIN-.01*(YMAX-YMIN)
    XMIN=XMIN-.01*(XMAX-XMIN)
    XA=0.
    XE=0.
    YA=0.
    YE=0.
    SX=(XMAX-XMIN)/(1000.*INDZ-110.)
    SY=(YMAX-YMIN)/1000.
    XP(K+1)=XFUN(2.*X(1)-X(2),Y(NY))
    YP(K+1)=YFUN(2.*X(1)-X(2),Y(NY))
    XP(K+2)=XFUN(X(NX),2.*Y(1)-Y(2))
    YP(K+2)=YFUN(X(NX),2.*Y(1)-Y(2))
312 CONTINUE
    DXX=.1*(XE-XA)
    DYY=.1*(YE-YA)
    NTXN=2
    IF(SX-SY)314,315,316
314 DI=1.5*SY/SX
    DXX=DI*DXX
    SX=SY
    GOTO 315
316 DJ=1.5*SX/SY
    DYY=DJ*DYY
    SY=SX
315 CONTINUE
    XMAX0=XMAX
    XMAX=XMIN+1000.*INDZ*SX
    YMAX=YMIN+1000.*SY
    IO=1
    KO=2000

```

```

IN=INDZ
331 IF(KO.GT.K)KO=K
    CALL PLOTA(XP(IO),YP(IO),KO-IO+1,2,0,1,1,1,IN,XMAX,XMIN,SX,
1 YMAX,YMIN,SY,NTEXT,IDPLOT,
2 1,XA,DXX,XE,'E9.2',1,-1,1,1,
3 1,YA,DYY,YE,'E9.2',1,1,-1,1,
4 NTXN,XP(K+1),YP(K+1),NDIR,NSC,YTEXT,XTEXT)
    NTXN=0
    IF(IDPLOT)989,989,332
332 IF(KO.EQ.K) GOTO 330
    IN=0
    IO=KO
    KO=KO+1999
    GOTO 331
330 CONTINUE
    IF(IDNV.LE.0)GOTO800
C   AUSGABE DER LEGENDE
    YP(1)=YMIN+965.*SY
    XP(1)=XMAX0+10.*SX
    XP(2)=XP(1)
    XP(3)=XMAX0+20.*SX
    DY=(YP(1)-YMIN)/NH
    DO 350 K=1,NH
    YP(3)=YP(1)-4.*SY
    YP(2)=YP(1)
    CALL CONVX(HV(K),S,'E10.3 ')
    S(11,1)=PKT
    S(12,1)=PKT
    CALL PLOTA(XP,YP,2,1,NV(K),1,1,1,0,XMAX,XMIN,SX,YMAX,YMIN,SY,
1 NTEXT,IDPLOT,
2 1,XA,DXX,XE,'E9.2',1,-1,1,1,
3 1,YA,DYY,YE,'E9.2',1,1,-1,1,
4 1,XP(3),YP(3),NDIR,NSC,S)
    YP(1)=YP(1)-DY
    IF(IDPLOT)989,989,350
350 CONTINUE
C   ENDE DER UMRAHMUNG
    ENTRY NOCHHL
800 CONTINUE
    NPA1=3
    IF(IDNV.LE.0)NPA1=1
    NPG0=1
    INTO=1
    READY=FALSE
    DO 1000 INH=1,NH
    H=HV(INH)
    NPO=NV(INH)
C   RAND ABSUCHEN
    LRAND=TRUE
    DO 1010 J=1,NY
    DO 1010 I=1,NX
1010 S(I,J)=FALSE
    IO=NX
    IAUS=4
    DO 1004 JO=1,NY1
    IF(F(IO,JO).EQ.H)F(IO,JO)=AEND(F(IO,JO),F(IO,JO+1))
    IF(NOTLZW(F(IO,JO),H,F(IO,JO+1))) GOTO 1004

```



```

I=I0
J=J0
XP(1)=X(NX)
YP(1)=SCHNIT(F(I0,J0),H,F(I0,J0+1),Y(J0),Y(J0+1))
KS=1
DI=-1
DJ=1
K=1
GOTO 10
1004 CONTINUE
J0=1
IAUS=1
DO 1001 I0=1,NX1
IF(S(I0,J0)) GOTO 1001
IF(F(I0,J0).EQ.H)F(I0,J0)=AEND(F(I0,J0),F(I0+1,J0))
IF(NOTLZW(F(I0,J0),H,F(I0+1,J0))) GOTO 1001
KS=2
S(I0,J0)=TRUE
I=I0
J=J0
XP(1)=SCHNIT(F(I,J),H,F(I+1,J),X(I),X(I+1))
YP(1)=Y(J0)
K=1
DI=1
DJ=1
GOTO 10
1001 CONTINUE
I0=1
IAUS=2
DO 1002 J0=1,NY1
IF(S(NX,J0)) GOTO 1002
IF(F(I0,J0).EQ.H)F(I0,J0)=AEND(F(I0,J0),F(I0,J0+1))
IF(NOTLZW(F(I0,J0),H,F(I0,J0+1))) GOTO 1002
KS=1
DI=1
DJ=1
I=I0
J=J0
S(NX,J0)=TRUE
XP(1)=X(I0)
YP(1)=SCHNIT(F(I,J),H,F(I,J+1),Y(J),Y(J+1))
K=1
GOTO 10
1002 CONTINUE
J0=NY
IAUS=3
DO 1003 I0=1,NX1
IF(S(I0,J0))GOTO 1003
IF(F(I0,J0).EQ.H)F(I0,J0)=AEND(F(I0,J0),F(I0+1,J0))
IF(NOTLZW(F(I0,J0),H,F(I0+1,J0))) GOTO 1003
I=I0
J=J0
XP(1)=SCHNIT(F(I,J),H,F(I+1,J),X(I),X(I+1))
YP(1)=Y(J0)
K=1
KS=2
DI=1

```

```

      DJ=-1
      S(IO,J0)=TRUE
      GOTO 10
1003 CONTINUE
      IF((NY1.LT.2).OR.(NX1.LT.2)) GOTO1000
      IAUS=5
C     IM INNEREN SUCHEN
      LRAND=FALSE
      DO 1005 J0=2,NY1
      DO 1005 I0=2,NX1
      IF(S(IO,J0)) GOTO 1005
      IF(NOTLZW(F(IO,J0),H,F(I0+1,J0))) GOTO 1005
      S(IO,J0)=TRUE
      I=I0
      J=J0
      K=1
      XP(1)=SCHNIT(F(IO,J0),H,F(I0+1,J0),X(I0),X(I0+1))
      YP(1)=Y(J0 )
      KS=2
      DI=1
      DJ=1
      XP1=XP(1)
      YP1=YP(1)
      GOTO 10
1005 CONTINUE
1000 CONTINUE
  989 CONTINUE
9000 CONTINUE
      RETURN
C     HOEHENLINIE LANG LAUFEN
  10 CONTINUE
      IF((K+4).GT.KMAX)GOTO999
      K=K+1
      IP=I+DI
      JP=J+DJ
      IF(KS-1)71,71,72
  71 IF(H.EQ.F(IP,J))F(IP,J)=AEND(F(IP,J),F(I,J))
      IF(H.EQ.F(IP,JP))F(IP,JP)=AEND(F(IP,JP),F(I,JP))
      GOTO73
  72 IF(H.EQ.F(IP,JP))F(IP,JP)=AEND(F(IP,JP),F(IP,J))
      IF(H.EQ.F(I,JP))F(I,JP)=AEND(F(I,JP),F(I,J))
  73 CONTINUE
  16 F1=F(I,J)
  15 F2=F(IP,J)
      F3=F(IP,JP)
      F4=F(I,JP)
      IF(DREI)GOTO400
481 CONTINUE
      SJP1=LZW(F4,H,F3)
      SIP1=LZW(F2,H,F3)
      IF(KS-1)11,11,12
  11 CONTINUE
C     AUSGEHEND VON I=CONST.
      SJ =LZW(F1,H,F2)
      IF(SJ.AND.SJP1.AND.SIP1) GOTO 30
      IF(SJ) GOTO 31
      GOTO 13

```

```

12 CONTINUE
C   AUSGEHEND VON J=CONST.
    SI =LZW(F1,H,F4)
    IF(SI.AND.SJP1.AND.SIP1) GOTO 30
    IF(SI) GOTO 32
13  IF(SJP1) GOTO 33
    IF(SIP1) GOTO 34
30  CONTINUE
400 F5=.25*(F1+F2+F3+F4)
    X5=.5*(X(I)+X(IP))
    Y5=.5*(Y(J)+Y(JP))
    IF(F5.EQ.H)F5=AFND(F5,F1)
480 IF(KS-1)421,421,422
422 IF(LZW(F5,H,F2))GOTO401
C   IM UHRZEIGERSINN
    XP(K)=SCHNIT(F1,H,F5,X(I),X5)
    YP(K)=SCHNIT(F1,H,F5,Y(J),Y5)
    K=K+1
    IF(LZW(F1,H,F4))GOTO32
403 XP(K)=SCHNIT(F4,H,F5,X(I),X5)
    YP(K)=SCHNIT(F4,H,F5,Y(JP),Y5)
    K=K+1
    IF(LZW(F3,H,F4))GOTO33
    XP(K)=SCHNIT(F3,H,F5,X(IP),X5)
    YP(K)=SCHNIT(F3,H,F5,Y(JP),Y5)
    K=K+1
    IF(KS.EQ.2)GOTO34
    IF(LZW(F3,H,F2))GOTO34
    XP(K)=SCHNIT(F2,H,F5,X(IP),X5)
    YP(K)=SCHNIT(F2,H,F5,Y(J),Y5)
    K=K+1
    GOTO31
401 CONTINUE
C   GEGEN DEN UHRZEIGERSINN
    XP(K)=SCHNIT(F2,H,F5,X(IP),X5)
    YP(K)=SCHNIT(F2,H,F5,Y(J),Y5)
    K=K+1
    IF(LZW(F2,H,F3))GOTO34
    XP(K)=SCHNIT(F3,H,F5,X(IP),X5)
    YP(K)=SCHNIT(F3,H,F5,Y(JP),Y5)
    K=K+1
    IF(KS.EQ.1)GOTO33
    IF(LZW(F4,H,F3))GOTO33
    XP(K)=SCHNIT(F4,H,F5,X(I),X5)
    YP(K)=SCHNIT(F4,H,F5,Y(JP),Y5)
    K=K+1
    GOTO 32
421 IF(LZW(F5,H,F1))GOTO411
C   IM UHRZEIGERSINN
    GOTO 403
411 CONTINUE
C   GEGEN DEN UHRZEIGERSINN
    XP(K)=SCHNIT(F1,H,F5,X(I),X5)
    YP(K)=SCHNIT(F1,H,F5,Y(J),Y5)
    K=K+1
    IF(LZW(F1,H,F2))GOTO31
    GOTO401

```

```

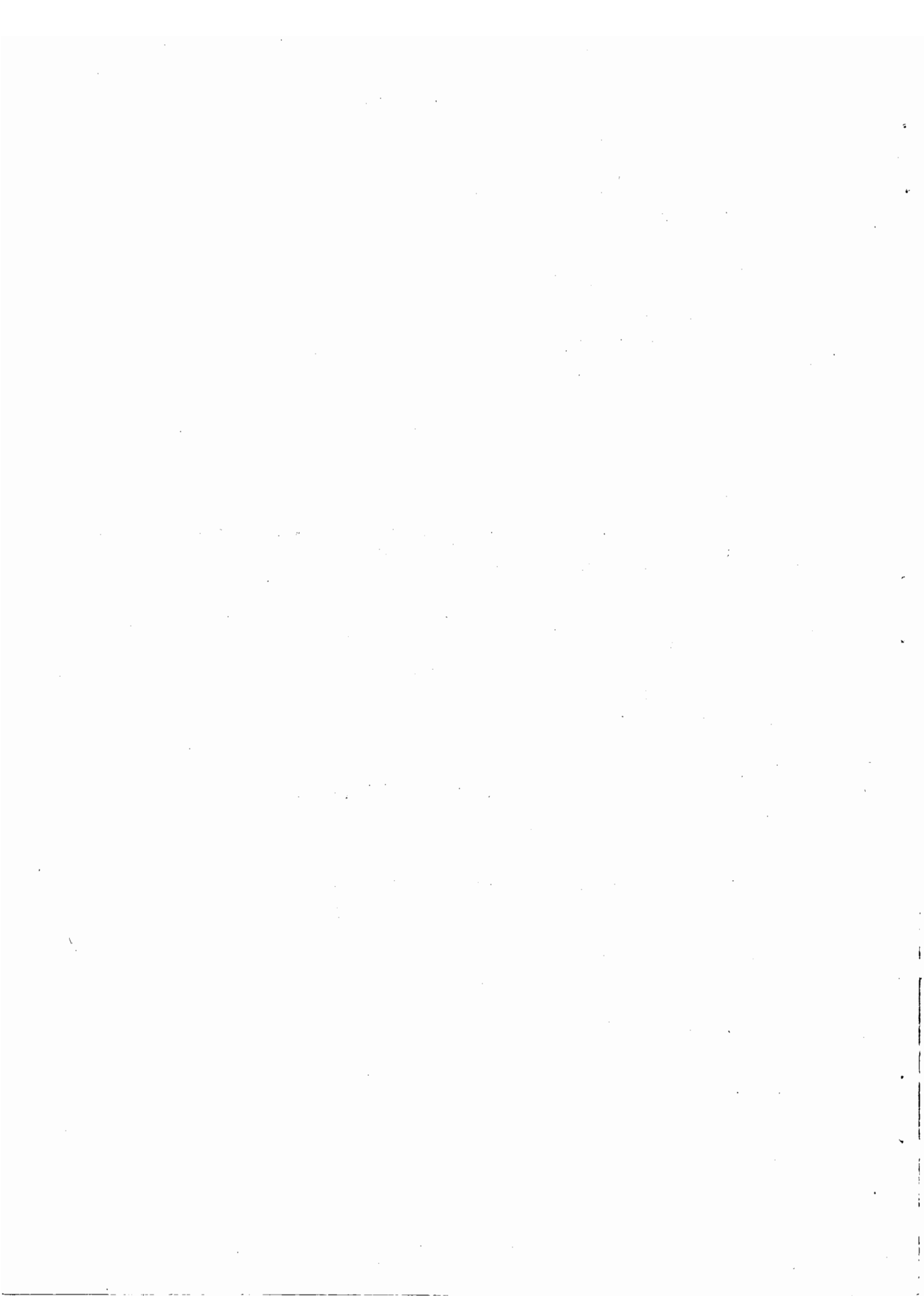
34 CONTINUE
  YP(K)=SCHNIT(F2,H,F3,Y(J),Y(JP))
  XP(K)=X(IP)
  IF(LRAND) GOTO 501
  IF((K.GT.3).AND.(ABS(XP(K)-XP1 ).LT.VX).AND.
5  (ABS(YP(K)-YP1 ).LT.VY))GOTO901
501 CONTINUE
  I=I+DI
  JMIN=MINO(J,JP)
  IF((I.EQ.1).AND.S(NX,JMIN)) GOTO 902
  IF((I.LE.1 ).OR.(I.GE.NX)) GOTO 900
  KS=1
  GOTO 10
33 CONTINUE
  YP(K)=Y(JP)
  XP(K)=SCHNIT(F4,H,F3,X(I),X(IP))
  IF(LRAND) GOTO 502
  IF((K.GT.3).AND.(ABS(XP(K)-XP1 ).LT.VX).AND.
5  (ABS(YP(K)-YP1 ).LT.VY))GOTO901
502 CONTINUE
  IMIN=MINO(I,IP)
  IF(S(IMIN,JP)) GOTO 902
  S(IMIN,JP)=TRUE
  J=J+DJ
  IF((J.LE.1 ).OR.(J.GE.NY)) GOTO 900
  KS=2
  GOTO 10
32 CONTINUE
  YP(K)=SCHNIT(F1,H,F4,Y(J),Y(JP))
  XP(K)=X(I)
  IF(LRAND) GOTO 503
  IF((K.GT.3).AND.(ABS(XP(K)-XP1 ).LT.VX).AND.
5  (ABS(YP(K)-YP1 ).LT.VY))GOTO901
503 CONTINUE
  JMIN=MINO(J,JP)
  IF((I.EQ.1).AND.S(NX,JMIN)) GOTO 902
  IF((I.LE.1 ).OR.I.GE.NX ) GOTO 900
  DI=-DI
  KS=1
  GOTO 10
31 CONTINUE
  YP(K)=Y(J)
  XP(K)=SCHNIT(F1,H,F2,X(I),X(IP))
  IF(LRAND) GOTO 504
  IF((K.GT.3).AND.(ABS(XP(K)-XP1 ).LT.VX).AND.
5  (ABS(YP(K)-YP1 ).LT.VY))GOTO901
504 CONTINUE
  IMIN=MINO(I,IP)
  IF(S(IMIN,J)) GOTO 902
  S(IMIN,J)=TRUE
  DJ=-DJ
  IF(J.LE.1 .OR. J.GE.NY) GOTO 900
  KS=2
  GOTO 10
997 IDPLOT=-1000000
  GOTO 9000
900 CONTINUE

```

```

IF((I.LE.1).OR.(I.GE.NX)) GOTO 910
IMIN=MINO(I,IP)
S(IMIN,J)=TRUE
GOTO 901
910 JMIN=MINO(J,JP)
IF(I.LE.1) S(NX,JMIN)=TRUE
901 CONTINUE
READY=TRUE
999 CONTINUE
IF(K.LT.2) GOTO 902
IF(KARTH)GOTO801
I=1
802 CONTINUE
A=XFUN(XP(I),YP(I))
YP(I)=YFUN(XP(I),YP(I))
XP(I)=A
I=I+1
IF(I.LE.K)GOTO802
801 CONTINUE
I=1
KO=2000
931 IF(KO.GT.K)KO=K
NPA=MINO(KO-I,IDNV)
CALL PLOTA(XP(I),YP(I),KO-I+1,NPA1,NPO,NPGO,INTO,NPA,0,XMAX,
1XMIN,SX,YMAX,YMIN,SY,NTEXT,IDPLOT,
21,XA,DXX,XE,'E9.2',1,-1,1,1,
3 1,YA,DYY,YE,'E9.2',1,1,-1,1,
40)
IF(IDPLOT) 989,989,932
932 IF(KO.GE.K) GOTO 930
IO=KO
KO=KO+1999
GOTO 931
930 CONTINUE
IF(.NOT.READY)GOTO996
902 CONTINUE
READY=FALSE
GOTO(1001,1002,1003,1004,1005,9000),IAUS
996 XP(1)=XP(K)
YP(1)=YP(K)
K=1
GOTO10
ENTRY ZUHL(XP,YP,K,NT,NP,NPG,INT,/IDNV/)
NPO=NP
NPGO=NPG
INTO=INT
NPA1=NT
IAUS=6
GOTO 901
END

```



Anhang 2Kurzbeschreibungen der verwendeten Unterprogramme

Es werden nur die hier maßgeblichen Argumente und Eigenschaften erläutert.

1) Subroutine PLOTA

PLOTA ist wie folgt aufzurufen:

CALL PLOTA (XP, YP, K, NT, NP, NPG, INT, NPA, INDZ,)	Gruppe
XMAX, XMIN, SX, YMAX, YMIN, SY, NTEXT, IDPLOT,)	1
NLGX, XA, DXX, XE, NFX, NSCX, NUØX, NPSX, NX,)	Gruppe
NLGY, YA, DYY, YE, NFY, NSCY, NUØY, NPSY, NY,)	2
NTXT, XB, YB, NDIR, NSC, NTXT1, NTXT2))	Gruppe 3

Bedeutung der Argumente

Gruppe 1

- | | |
|---------|---|
| 1) XP | Feld der Abszissenwerte |
| 2) YP | Feld der Ordinatenwerte |
| 3) K | Zahl der Koordinatenpaare, $2 \leq K \leq 2000$ |
| 4) NT | = 1 punktweise Darstellung
= 2 ausgezogene Linie
= 3 ausgezogene Linie mit Punkten |
| 5) NP | Punktetyp $0 \leq NP \leq 9$
z.B. NP=2 Zeichen = \diamond |
| 6) NPG | Maßstab der Zeichen und Texte; $1 \leq NPG \leq 3$
NPG=1 Zeichenhöhe 8/100 inch,
Alphabethöhe 12/100 inch. |
| 7) INT | Art der Interpolation
1-linear, 2-quadratisch, 3-kubisch |
| 8) NPA | Abstand der zu zeichnenden Punkte für NT=3
$1 \leq NPA \leq K-1$ |
| 9) INDZ | INDZ=0 Kurve in alte Zeichnung
$1 \leq INDZ \leq 4$ $\frac{\text{maximale Ausdehnung in X}}{\text{maximale Ausdehnung in Y}} = \frac{1}{INDZ}$ |

- 10) XMAX maximaler Abszissenwert
- 11) XMIN minimaler Abszissenwert
- 12) SX Maßstab; Abszissendifferenz/(1/100 inch)
- 13-15) entsprechend 10-12 für Ordinate
- 16) NTEXT 60-Zeichen-Text für die Unterschrift
- 17) IDPLOT Zahl zur Identifikation der Zeichnung

Gruppe 2

Diese Angaben dienen zur Definition der Skalen und ihrer Beschriftungen.
Wichtig sind:

- XA bzw. YA Anfang der Skaleneinteilung
 - XE bzw. YE Ende der Skaleneinteilung
 - DXX bzw. DYY Abstand zwischen zwei Skalenstrichen
- DXX bzw. DYY gleich null bewirkt, daß die Skala
ohne irgendwelche Beschriftung gezeichnet wird.

Gruppe 3

Diese letzte Gruppe dient zur Ausgabe weiterer Beschriftungen.

- 36) NTXN Anzahl der zusätzlichen Beschriftungen
 $0 \leq NTXN \leq 10$
- 37) XB Feld der Länge NTXN mit der Abszisse des Anfangs-
 punktes der zusätzlichen Beschriftungen
- 38) YB dito für Ordinate
- 39) NDIR Feld zur Definition der Schreibrichtungen der
 Texte. NDIR(K)=2 bewirkt Schreiben des Textes K
 von links nach rechts.
- 40) NSC Feld zur Definition der Höhe des Beschriftungs-
 textes. NSC(K)=1 bewirkt Alphabethöhe $\frac{12}{100}$ inch.
- 41)-42) NTXTi Beschriftungstexte bestehend aus n Zeichen mit
 $3 \leq n \leq 40$, wobei die letzten beiden Zeichen
 Punkte sein müssen.

Bemerkung: Das Zeichenfeld von PLOTA ist ein Rechteck mit der Höhe
1000/100 inch und der Breite $INDZ \times 1000/100$ inch. Be-
schriftungen müssen $24/100$ inch vom oberen Rand ent-
fernt bleiben und dürfen nicht über den rechten Rand
ragen; andernfalls wird die Beschriftung unterdrückt;

über den linken und unteren Rand darf die Beschriftung maximal 56/100 inch hinausreichen.

2) Subroutine CONVX

Der hier verwendete Aufruf ist von der Form
CALL CONVX (ARG,ALPH,FIELDD)

Die Argumente haben folgende Bedeutung:

- 1) ARG Die umzuwandelnde Zahl
- 2) ALPH Feld in das die Zeichenkette abgelegt werden soll, die der Zahl entspricht.
- 3) FIELDD Field-Descriptor, eine Zeichenkette die das Format der Umwandlung beschreibt. Hier wurde FIELDD='E10.3' verwendet.

