KERNFORSCHUNGSZENTRUM

KARLSRUHE

Mai 1968

KFK 738

Institut für Neutronenphysik und Reaktortechnik

Ein Verfahren und ein FORTRAN-IV-Programm zur flächentreuen Approximation von Treppenfunktionen durch glatte Kurven

H. Späth

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M.B.H.
KARLSRUHE

K E R N F O R S C H U N G S Z E N T R U M

Mai 1968 KFK 738

Institut für Neutronenphysik und Reaktortechnik

Ein Verfahren und ein FORTRAN-IV-Programm zur flächentreuen Approximation von Treppenfunktionen durch glatte Kurven

H. Späth

Gesellschaft für Kernforschung m.b.H., Karlsruhe

Summary

When smoothing step functions, i.e. multigroup spectra to a weighting function for the calculation of multigroup constants in neutron transport theory, the following problem arises:
To a given step function we search a "smooth" curve thus that the areas below the curves are equal in each of the intervals defined by the discontinuities of the step function.

We give a definition of "smooth" that has been proven to be a good one in practice. We formulate the approximation problem and give a numerical method to calculate such a smooth and area-preserving curve. Propositions about the existence and uniqueness of such a function can only be made for trivial cases.

Inhalt

I. Das Verfahren

1. Definition einer glatten Funktion

Eine Treppenfunktion h kann wie folgt definiert werden:

$$h(x) = \begin{cases} o \text{ für } x \text{ mit } x < x_o \text{ und } x \geq x_n \\ y_i \text{ für } x \text{ mit } x_{i-1} \leq x < x_i \ (i=1,\dots,n) \end{cases} \qquad (1)$$

Hierbei sind $x_o,\dots,x_n$ gegebene Abszissen ($n \geq 1$) mit

$$-\infty < x_o < x_1 < \dots < x_n \infty$$

und $y_1,\dots,y_n$ gegebene Ordinaten mit $/y_i/ < \infty$ für $i=1,\dots,n$.
Eine Funktion F, die in dem Intervall $[x_o,x_n]$ definiert ist und für die bei vorgegebenen Werten y_o, y_{n+1}, (a_i,b_i) $(i=1,\dots,m)$ mit $x_o < a_1 < \dots < a_m < x_n$ und $m \geq 1$ gilt $F(x_o)=y_o$, $F(x_n)=y_{n+1}$, $F(a_i)=b_i$ $(i=1,\dots,m)$, wollen wir "glatt" nennen, wenn sie dort zweimal stetig differenzierbar ist, also wenn

$$F \varepsilon\ C^2[x_o,x_n] \qquad (2)$$

und wenn gilt

$$\int_{x_o}^{x_n} [F''(x)]^2 dx = \min! \qquad (3)$$

Eine solche Funktion F existiert immer und ist eindeutig bestimmt. [1]. Es gilt $F''(x_o)=F''(y_n)=0$.

2. Formulierung des Problems

Eine sinnvolle Forderung an die gesuchte Kurve ist offenbar, daß sie die gegebene Treppenfunktion h in jedem Intervall $[x_{i-1},x_i]$ $(i=1,\dots,n)$ wenigstens einmal schneidet; denn andernfalls müßte in einem Intervall die Kurve ganz oberhalb oder unterhalb der Treppenfunktion verlaufen, was wegen der Flächenbedingungen unmöglich ist.

Daher setzen wir m=n, bezeichnen a_i mit z_i, setzen $b_i=y_i$ und formulieren unsere Approximationsaufgabe folgendermaßen:

Wir suchen Abszissen $z_1,\dots,z_n$ $(n \geq 1)$ mit

$$x_{i-1} < z_i < x_i \quad (i=1,\dots,n) \tag{4}$$

und eine den Bedingungen (2) und (3) genügende "glatte" Funktion $F=F(z;x)=F(z_1,\dots,z_n;x)$ derart, daß die folgenden Bedingungen gelten:

$$F(z;x_0)=y_0,\ F(z;z_i)=y_i(i=1,\dots,n),\ F(z;x_n)=y_{n+1} \tag{5}$$

$$\int_{x_{i-1}}^{x_i} F(z;x)dx=y_i(x_i-x_{i-1}) \quad (i=1,\dots,n) \tag{6}$$

Bei unserer Definition von "glatt" und bei der speziellen Wahl der Unbekannten kann man, wie die Beispiele in Bild 2-4 zeigen, hoffen, daß F in den Intervallen $[x_{i-1},x_i]$ $(i=1,\dots,n)$ bis auf manche Fälle mit $y_{i-1} > y_i$ und $y_i < y_{i+1}$ bzw. $y_{i-1} < y_i$ und $y_i > y_{i+1}$ jeweils nur eine y_i-Stelle z_i besitzt und daß häufig gilt $y_i \leq F(z;x_i) \leq y_{i+1}$ bzw. $y_i \geq F(z;x_i) \geq y_{i+1}$; denn (3) bedeutet ja, daß F im Mittel möglichst wenig gekrümmt sein soll. Das Erfülltsein der Forderung (3) ist wesentlich, denn Differenzierbarkeitsforderungen allein gewährleisten im allgemeinen keine anschaulich glatten Kurven.

3. Aufstellung eines nichtlinearen Gleichungssystems

Die Funktion F mit den Eigenschaften (2), (3) und (5) heißt eine natürliche Splinefunktion vom Grad 3 mit den n+2 Knotenstellen $x_0,z_1,\dots,z_n,x_n$. F ist eindeutig bestimmt und, wie in [1] gezeigt wird, in jedem der n+1 Intervalle $[x_0,z_1]$, $[z_i,z_{i+1}]$ $(i=1,\dots,n-1)$ und $[z_n,x_n]$ durch ein Polynom 3. Grades gegeben. Diese kubischen Polynome, also ihre Koeffizienten, sind durch die Bedingungen (5) und die Werte für $F''(z;z_i)$ $(i=1,\dots,n)$ eindeutig festgelegt. Zur Bestimmung der Werte $F''(z;z_i)$ derart, daß (2) und (3) gelten, kann man folgendes tridiagonales Gleichungssystem aufstellen [1]:

$$\Delta z_{i-1} F''(z;z_{i-1}) + 2(\Delta z_i + \Delta z_{i-1}) F''(z;z_i) + \Delta z_i F''(z;z_{i+1})$$

$$= 6 \left(\frac{\Delta y_i}{\Delta z_i} - \frac{\Delta y_{i-1}}{\Delta z_{i-1}} \right) \quad (i=1,\ldots,n) \qquad (7)$$

Hierbei muß $\Delta z_o = z_1 - x_o$ und $\Delta z_n = x_n - z_n$ sowie $F''(z;z_o) = F''(z;z_{n+1}) = 0$ gesetzt werden. Der Strich bedeutet die Ableitung nach dem zweiten Argument. Da die Koeffizientenmatrix symmetrisch und streng diagonal dominant ist, ist sie auch positiv definit und folglich ist die Eliminationsmethode ohne Pivotisierung numerisch stabil.

Wir müssen nun die Punkte $z_i (i=1,\ldots,n)$ so bestimmen, daß die Bedingungen (6) erfüllt sind. Wenn wir die Integrationsintervalle auf den linken Seiten von (6) in $[x_{i-1}, z_i]$ und $[z_i, x_i]$ aufspalten und auf jedes der beiden Teilintervalle die Simpsonsche Integrationsregel anwenden, die die exakten Werte liefert, da Polynome 3. Grades integriert werden, so erhalten wir ein nichtlineares Gleichungssystem

$$Tz = 0 \qquad (8)$$

für die Unbekannten $z=(z_1,\ldots,z_n)$, das in Komponentenschreibweise lautet

$$t_i(z_1,\ldots,z_n) = 0 \quad (i=1,\ldots,n)$$

wobei

$$t_i(z_1,\ldots,z_n) = \frac{x_i - x_{i-1}}{2} y_i - \frac{z_i - x_{i-1}}{2} F(z;x_{i-1}) - \frac{x_i - z_i}{2} F(z;x_i)$$

$$+ \frac{(z_i - x_{i-1})^3}{24} \left[F''(z;z_i) + F''(z;x_{i-1}) \right] \qquad (9)$$

$$+ \frac{(x_i - z_i)^3}{24} \left[F''(z;x_i) + F''(z;z_i) \right]$$

In (9) sind die Bedingungen (5) ausgenutzt.

4. Lösung dieses nichtlinearen Gleichungssystems

Zur Lösung von (8) verwenden wir das gedämpfte Newtonsche Verfahren [2,3]. Wegen (4) sind

$$z_i^{(o)} = \frac{1}{2}(x_{i-1}+x_i) \tag{10}$$

im allgemeinen geeignete Startwerte. Die Iteration wird, damit die Nebenbedingungen (4) gewährleistet bleiben, bedingt [2] durchgeführt: Ist (4) für die Komponente $z_i^{(k)}$ bei der k-ten Iteration nicht erfüllt, so setzt man einfach $z_i^{(k)}=z_i^{(o)}$. Die zur Durchführung der Iteration benötigte Funktionalmatrix $T_z' = (\frac{\partial t_k}{\partial z_i})$ wird, da sie für beliebige $n > 1$ nicht explizit angegeben werden kann, mittels zentraler Differenzen mit den Schrittweiten

$$h_i=z_i^{(o)} \times 5.E-3 \tag{11}$$

approximiert:

$$\frac{\partial t_k}{\partial z_i} = \frac{t_k(z_1,\dots,z_i+h_i,\dots,z_n) - t_k(z_1,\dots,z_i-h_i,\dots,z_n)}{2h_i} \tag{12}$$

Von der benutzten Variante des Newtonschen Verfahrens ist bekannt [2], daß für beliebige $z^{(o)}$ Konvergenz gegen einen stationären Punkt von $\|Tz\|^2$, also einen Punkt mit

$$T_z' \, Tz = 0 \tag{13}$$

gewährleistet ist, falls während der Iteration die Funktionalmatrix T_z' nicht singulär wird. Nun genügen alle Lösungen von (8) der Bedingung (13), und erfahrungsgemäß läßt sich die Konvergenz dieses gedämpften Newtonschen Verfahrens gegen einen stationären Punkt von $\|Tz\|^2$ mit $Tz \neq 0$ durch geeignete Startwertwahl vermeiden, falls eine Lösung von (8) existiert. Die Menge der zulässigen Startwerte, also diejenigen Startwerte, die zu einer Lösung von (12) führen, ist dabei im allgemeinen größer als beim ungedämpften Verfahren.

Da die bedingte Iteration erfahrungsgemäß spätestens nach einigen Schritten nicht mehr notwendig ist, gilt von da an die Konvergenzaussage in unveränderter Form. Mit bedingter Iteration beim Newtonschen Verfahren hat man auch bei anderen Problemen gute Erfahrungen gemacht [2].

5. Existenz- und Eindeutigkeitsaussagen

Bisher haben wir zwar ein Konstruktionsverfahren angegeben, aber weder über die Existenz noch über die Eindeutigkeit der gesuchten Funktion F Aussagen gemacht. Dazu sind wir nur für den Fall n=1 in der Lage. Hier lautet das nichtlineare Gleichungssystem (8) ($z_1=z$)

$$\begin{aligned}&(y_2-y_0)z^3 + [x_1\,(-2y_0+7y_1-5y_2) + x_0\,(5y_0-7y_1+2y_3)]z^2\\ &+ [x_0^2\,(-3y_0+7y_1-4y_2) + x_0x_1\,(-4y_0+4y_2) + x_1^2\,(4y_0-7y_1+3y_2)]z \qquad (14)\\ &+ [x_0^3\,(y_2-y_1) + x_0^2x_1\,(3y_0-4y_1+y_2) + x_0x_1^2\,(-y_0+4y_1-3y_2) + x_1^3\,(y_1-y_0)]=0\end{aligned}$$

Ist $y_0 = y_2$, so erhält man zwei Nullstellen

$$\frac{1}{2}\,(x_0+x_1) \pm \frac{1}{2}\sqrt{\frac{3}{7}}\,(x_0-x_1)$$

und somit zwei Kurven F, die zur Geraden $\frac{1}{2}\,(x_1+x_2)$ symmetrische Lösungen repräsentieren.

Ist $y_0 \neq y_2$, so kann man sich, da, was auch für $n > 1$ gilt, die Nullstellen von (8) invariant gegenüber linearer Transformation gleichzeitig aller Ordinaten $y_i (i=0,\ldots,n+1)$ sind und die Nullstellen sich bei linearer Transformation der Abszissen $x_i (i=0,\ldots,n)$ genauso transformieren, auf den Fall $x_0 = 0$, $x_1 = 1$, $y_0 = 0$, $y_1 = \alpha$, $y_2 = 1$ beschränken. Dann lautet (14)

$$z^3 + (7\alpha - 5)z^2 + (3-7\alpha)z + \alpha = 0 \qquad (15)$$

Zunächst zeigen wir, daß (15) stets drei reelle Nullstellen besitzt. Sei $z_1 = z_1(\alpha)$ die stets existierende reelle Nullstelle. Dann kann man aus den zwischen den Koeffizienten von (15) und den symmetrischen Grundfunktionen der drei Nullstellen z_1, z_2 und z_3 bestehende Gleichungen die Beziehungen

$$z_{2/3} = \frac{1}{2a} (b \pm \sqrt{b^2 - 4ac}\,) \tag{16}$$

$$\text{mit} \quad \begin{aligned} a &= 7z_1^2 - 7z_1 + 1 \\ b &= 7z_1^2 - 15z_1 + 5 \\ c &= z_1^2 - 5z_1 + 3 \end{aligned} \tag{17}$$

herleiten.

Das Polynom in der reellen Variablen z_1

$$b^2 - 4ac = 21z_1^4 - 42z_1^3 + 67z_1^2 - 46z_1 + 13$$

hat nur, wie man berechnen kann, komplexe Nullstellen und nimmt für $z_1 = 0$ einen positiven Wert an. Daher ist $b^2 - 4ac > 0$ für alle z_1 und die Nullstellen z_2 und z_3 sind dann nach (16) reell.

Lösung unseres Problems ist in diesem Fall nach (4) ein Punkt z mit $0 < z < 1$. Die Gleichung (15) liefert für $-\infty < \alpha < 0$ und für $1 < \alpha < \infty$ genau zwei und für $0 \leq \alpha \leq 1$ genau eine Lösung unseres Problems. Dies bedeutet, daß, wenn $y_o \neq y_2$, unabhängig von x_o und x_1 mit $x_o < x_1$ genau für $y_o \leq y_1 - y_2$ und $y_o \geq y_1 \geq y_2$ genau eine Lösung existiert.

Diese Schlußweise läßt sich auf den Fall $n > 1$ nicht verallgemeinern. Doch kann man auch hier, wie Bild 1 zeigt, für spezielle Fälle mehrere Lösungen durch Wahl verschiedener Startwerte für das Iterationsverfahren finden.

6. Beispiele

Um eine Vorstellung von den Ergebnissen zu vermitteln, sind in Bild 2-4 einige Beispiele angegeben. Bild 2 entspricht dem Sample Problem in Teil II. Es wurden für das Sample Problem 6 Iterationen benötigt, damit die Integrale über die glatte Funktion sich von den entsprechenden Rechteckflächen um weniger als 2.E-6 unterscheiden. Die bedingte Iteration war erforderlich beim ersten Schritt für i = 5 und i = 11, sowie beim 3. Schritt für i = 11.

Bild 1

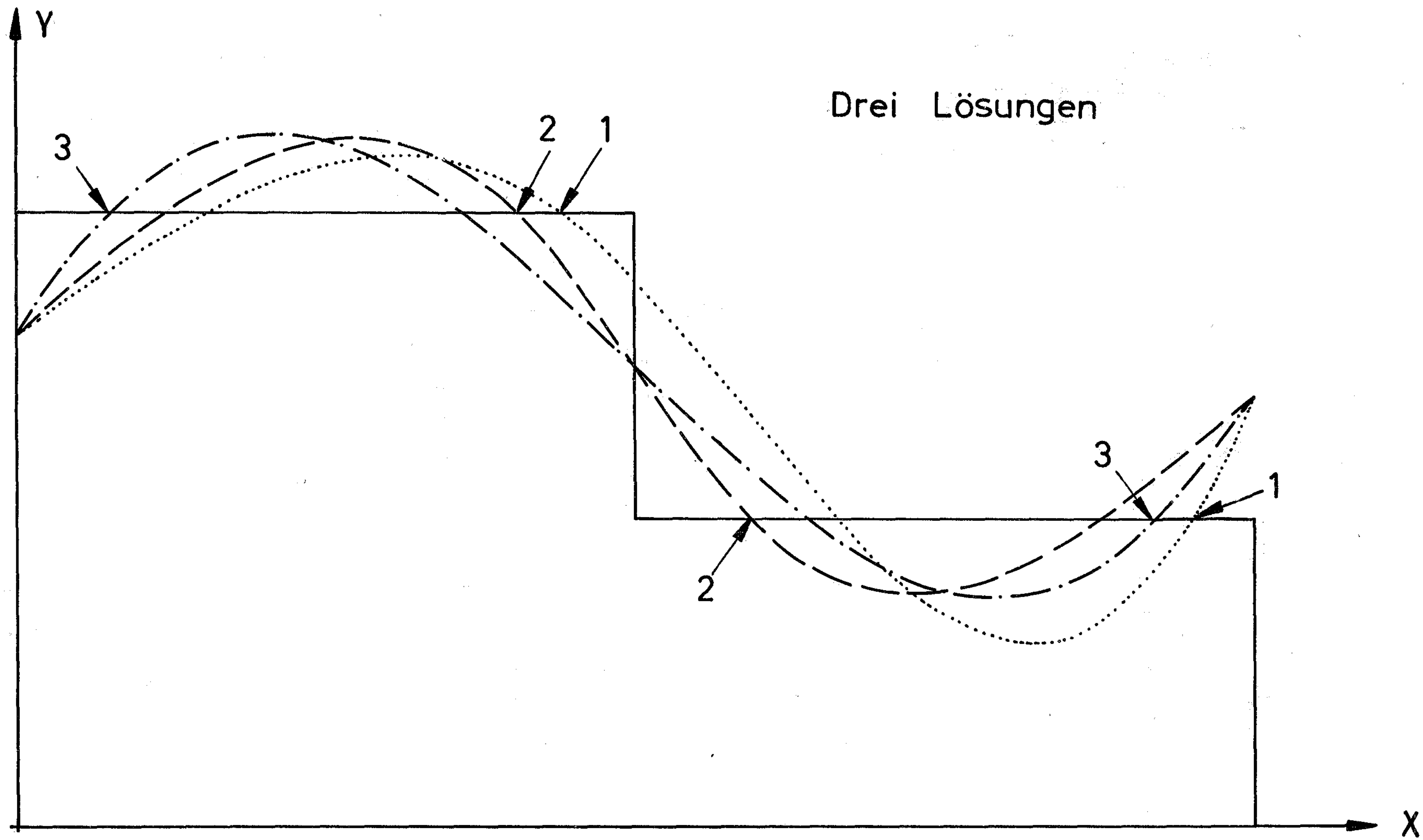


Bild 2

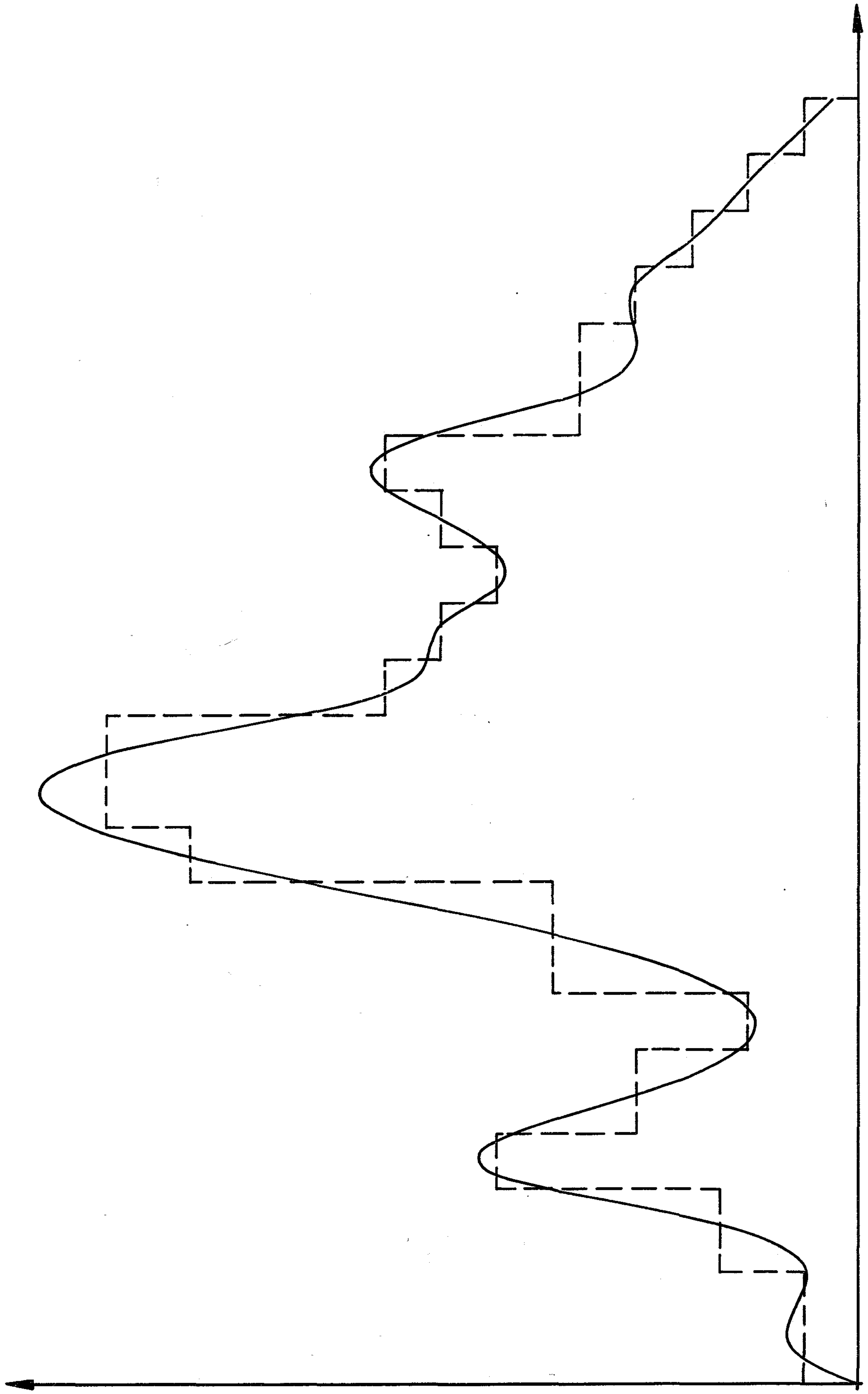


Bild 3

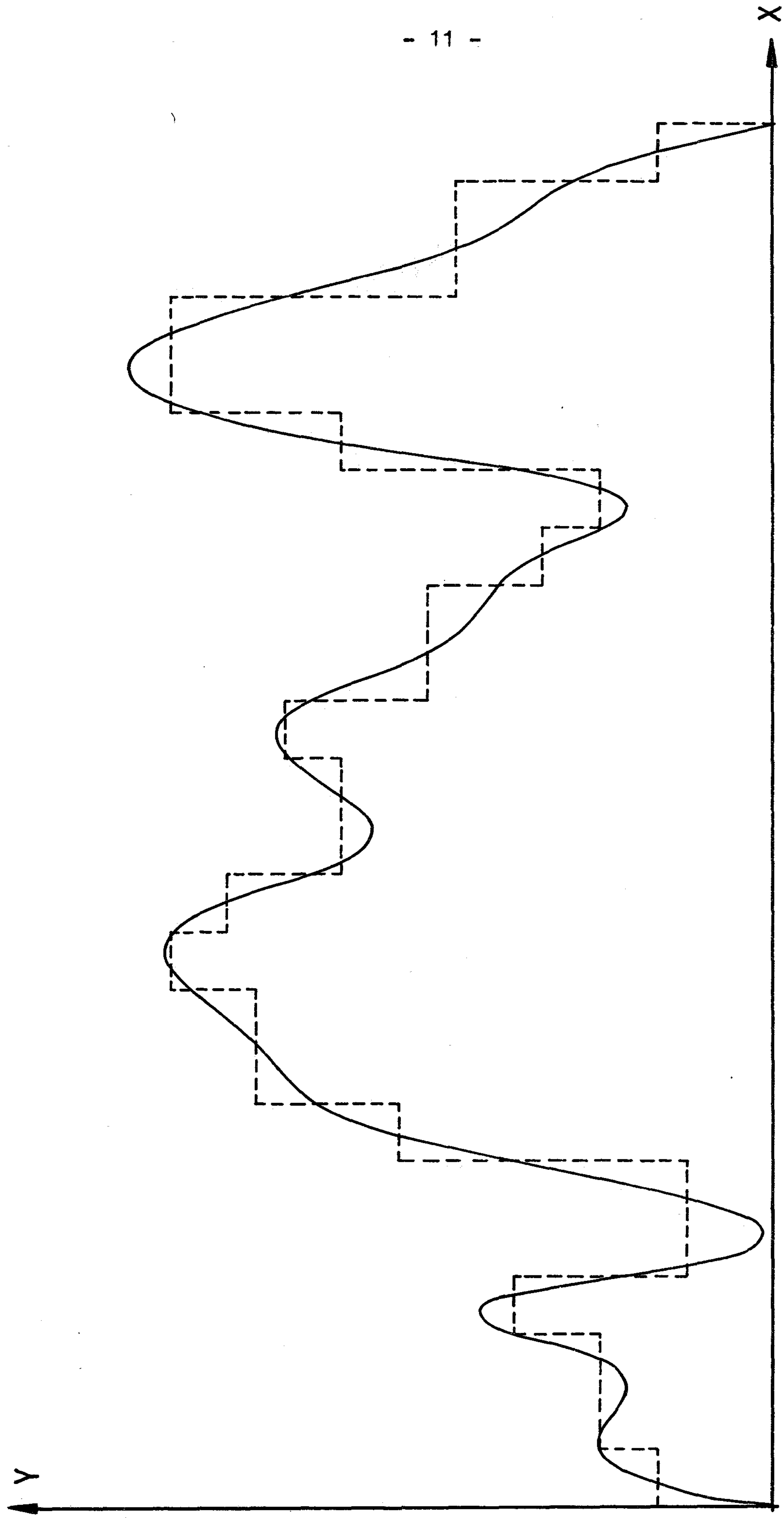
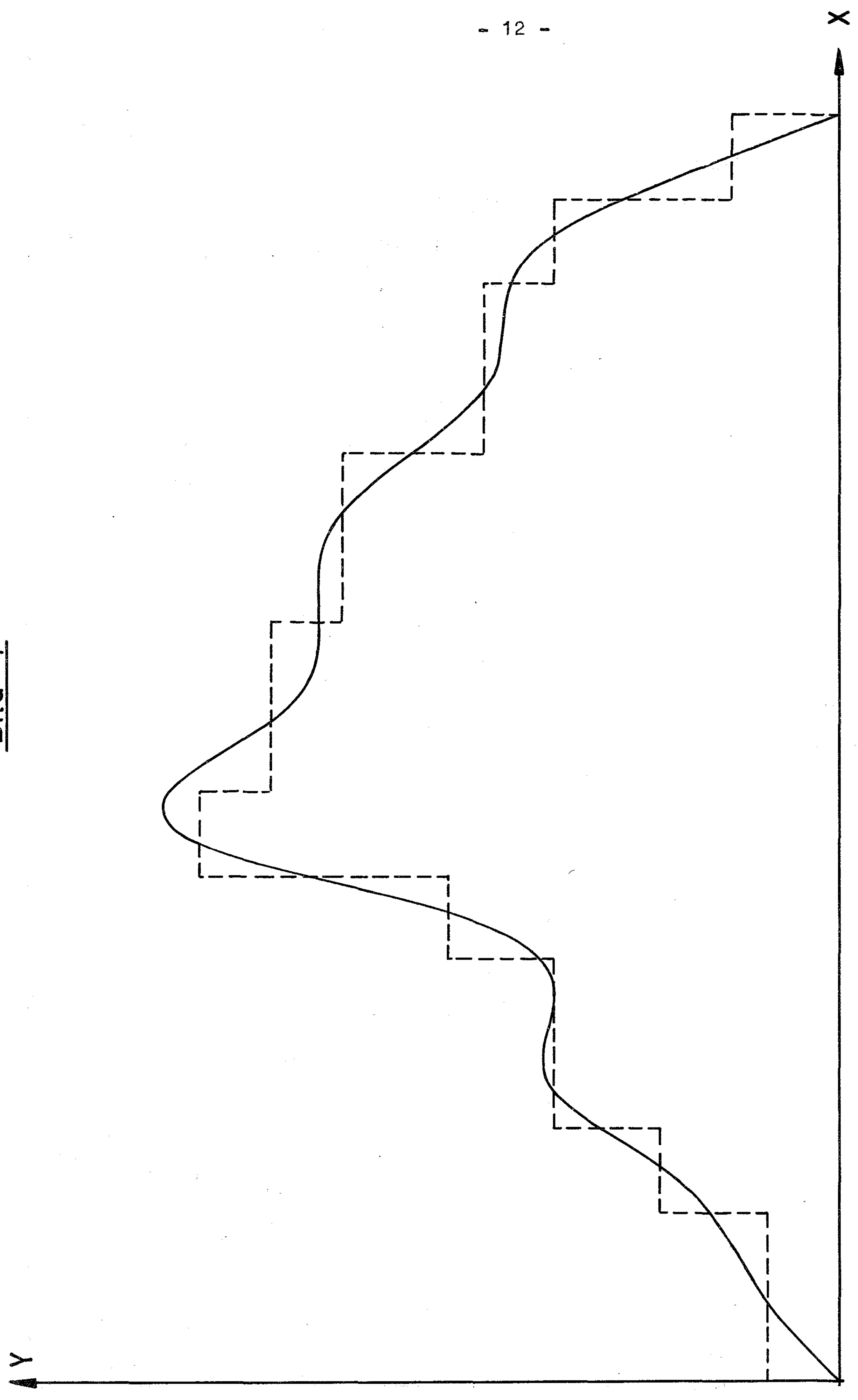


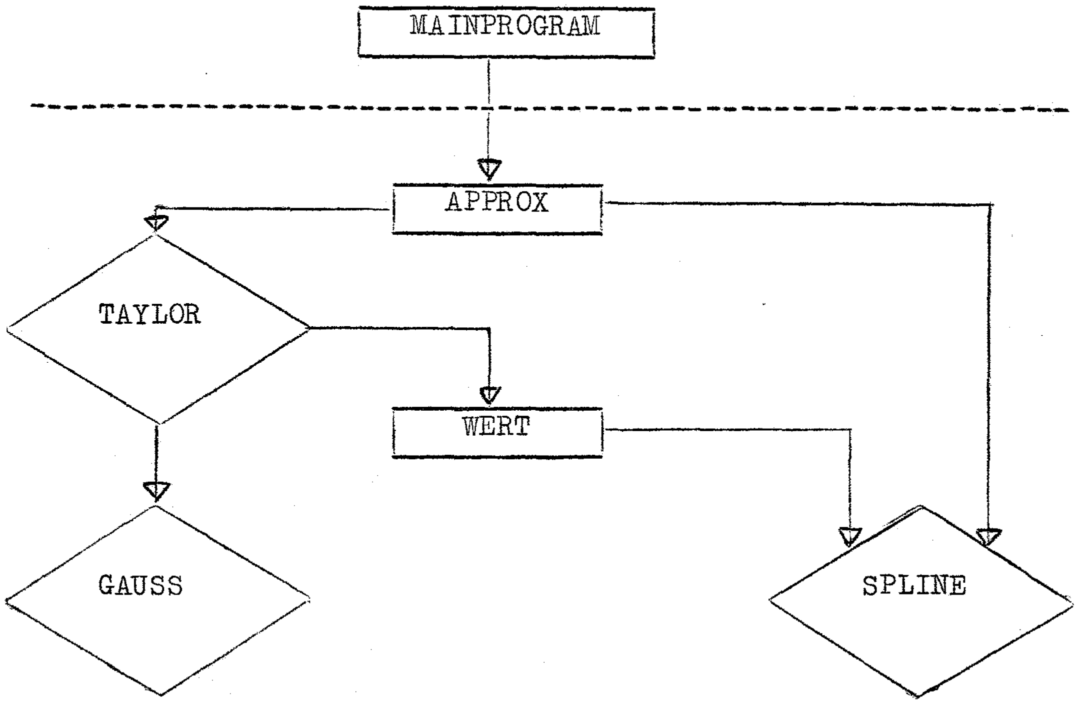
Bild 4



II. Das Programm

1. Struktur

Bei vorgegebener Treppenfunktion kann das in Teil I. beschriebene Iterationsverfahren zur Berechnung der flächentreuen, glatten Funktion durch Aufruf der Subroutine APPROX gestartet und der Abbruch mittels Konvergenztests gesteuert werden. APPROX ruft weitere Subroutinen auf nach folgendem Diagramm:



Die von Rechtecken umrahmten Programmteile besitzen dabei einen gemeinsamen COMMON, dessen Aufbau und Bedeutung unter 2. geschildert wird.

Das MAINPROGRAM ist dabei irgendein Oberprogramm, das APPROX aufruft. Unter 4. geben wir ein Beispiel dafür an, das gleichzeitig dazu diente, die Beispiele in Bild 1-4 und das Sample Problem zu berechnen.

TAYLOR ist eine Subroutine zum Lösen von nichtlinearen Gleichungssystemen, die in ähnlicher Form in ALGOL publiziert ist [3].
SPLINE ist eine Subroutine zur Berechnung einer natürlichen Splinefunktion vom Grad 3 durch vorgegebene Abszissen und Ordinaten [1].

Die Subroutine WERT dient zur Berechnung der Funktionen (9) in unserem speziellen nichtlinearen Gleichungssystem.

Die Subroutine GAUSS schließlich kann eine beliebige Subroutine zur Lösung von linearen Gleichungssystemen sein. Der Aufruf CALL GAUSS (N,A,B,X,K) soll bewirken, daß das lineare Gleichungssystem Ax = b für n Unbekannte gelöst wird. Wenn die Matrix A singulär wird, soll GAUSS mit K = 1 verlassen werden (sonst mit K = 0). Eine solche Subroutine ist elementar und wird daher hier nicht gelistet.

2. Der COMMON-Speicher

Ist N die Anzahl der Treppen und bedeuten Z und F Vektoren der Länge N, die den Unbekannten $z_1,\ldots,z_n$ und den Werten des Funktionensystems (9) an der Stelle Z entsprechen, so muß in dem Programm, das APPROX ruft, das Statement

```
DIMENSION Z(N), F(N)
```

stehen. N muß bekanntlich in einem Hauptprogramm eine feste Zahl sein. Weiter muß im rufenden Programm im COMMON-Block an erster Stelle folgende Liste stehen

```
      COMMON XX,H,XY,Y,S2,S,A1,A2,A3,A4,
     1       TT,ST,SS,EPS1,EPS2,N,M,IPR,
     2       ITMAX,KENN,KI,KO,N1,N2,N11
```

Die Bedeutung der Namen ist in folgender Tabelle angegeben. Darin bedeutet die Kennziffer

1: Eingabegrößen
2: Hilfsgrößen
3: berechnete Größen

Eingabegrößen müssen von dem APPROX rufenden Programm gesetzt sein.

Liste des COMMON-Speichers		
Name und gegebenenfalls Feldlänge	Typ	Bedeutung
XX(N+1)	1	Gegebene Abszissen XX(1) = x_o,...,XX(N+1) = x_n
H(N)	1	Schrittweiten für die Differenzen-approximation (12)
XY(N+2)	2	
Y(N+2)	1	Gegebene Ordinaten Y(1) = y_o,..., Y(N+2) = y_{n+1}
S2(N+2)	3	$F''(z;x_o)$, $F''(z;z_i)$ (i=1,...,n), $F''(z;x_n)$ charakterisieren die gefundene glatte Kurve zusammen mit den gegebenen Abszissen und Ordinaten
S(N+1)	2	
A1 (N+1) A2 (N+1) A3 (N+1) A4 (N+1)	2,3	Werden zunächst als Hilfsfelder benutzt; nach Verlassen von APPROX stehen darin die Koeffizienten der N+1 Polynome 3. Grades, die die gesuchte Kurve darstellen ($a_1x^3 + a_2x^2 + a_3x + a_4$)
TT(M) ST(M)	1 3	Abszissen mit XX(1) $\leq$ TT(1) $\leq$ ... $\leq$ TT(M) $\leq$ XX(N+1), an denen die Werte ST der gefundenen Kurve berechnet werden sollen
SS	3	Enthält die euklidische Norm des Funktionen-vektors (9) an der gefundenen Stelle Z
EPS1	1	Wenn SS < EPS1, wird die Iteration abgebrochen. (Empfohlen 1.E-20)
EPS2	1	Genauigkeitsforderung für die Komponenten von Z. Empfohlen 1.E-4. Das Iterationsverfahren wird abgebrochen, wenn $\sum_{i=1}^{n} /z_i^{(k+1)}-z_i^{(k)}/ < \mathrm{EPS2} \times \sum_{i=1}^{n} /Z_i^{(k+1)}/$
N	1	Anzahl der Treppen (Die Subroutine TAYLOR ist für 1 $\leq$ N < 27 ausgelegt; für N > 27 ist dort das Dimension-Statement zu ändern)
M	1	Anzahl der zu berechnenden Kurvenpunkte
IPR	1	Wenn IPR > 0 wird während der Iteration gedruckt, sonst nicht
ITMAX	1,3	Muß anfangs vorzugebende Maximalanzahl der auszuführenden Iterationen enthalten (Empfohlen: 20); nach Verlassen von APPROX steht dann die tatsächlich ausgeführte Anzahl darin

KENN	3	Kennzeichen für den Verlauf der Iteration. Wenn KENN = 0 ist, dann ist die Iteration normal verlaufen und man hat eine Lösung des nichtlinearen Gleichungssystems (8) und somit eine glatte Kurve gefunden. Ist KENN = -1, so wurde ein stationärer Punkt von $\|Tz\|^2$ gefunden. Ist KENN = 1, so wurde die Matrix T'_z singulär und die Iteration mußte abgebrochen werden. In diesem Fall empfiehlt es sich, andere Startwerte Z vorzugeben.
KI	1	Nummer des Eingabebandes
KØ	1	Nummer des Ausgabebandes
N1	1,2	N1 = N+1
N2	1,2	N2 = N+2
N11	1,3	N11 = 2 * N1

Die dimensionierten Größen müssen auch im DIMENSION des APPROX rufenden Programms stehen. Das unter 5. gelistete MAINPROGRAM gibt ein Beispiel für das DIMENSION- und COMMON-Statement.
Die Subroutine APPROX wird durch

CALL APPROX (Z,F)

aufgerufen, wobei Z und F die oben erwähnte Bedeutung besitzen. Der Vektor Z muß vor dem Aufruf von APPROX mit Startwerten (z.B. (10)) geladen sein.

3. Liste der Subroutinen

```
      SUBROUTINE APPROX(Z,F)
C
C     CALLS SUBROUTINES TAYLOR AND SPLINE, TAYLOR CALLS WERT
C
      DIMENSION XX(27),Z(26),F(26),H(26),XY(28),Y(28),S2(28),S(28),
     1          A1(28),A2(28),A3(28),A4(28),TT(1000),ST(1000)
      COMMON XX,H,XY,Y,S2,S,A1,A2,A3,A4,TT,ST,
     1       SS,EPS1,EPS2,N,M,IPR,ITMAX,KENN,
     2       KI,KO,N1,N2,N11
      EXTERNAL WERT
C
      XY(1)=XX(1)
      XY(N2)=XX(N1)
      CALL TAYLOR(N,Z,H,F,WERT,ITMAX,EPS1,EPS2,SS,IPR,KENN)
      CALL SPLINE(N2,XY,Y,S2,1,0,M,TT,ST,NR,A1,A2,A3,A4)
      IF(KENN) 1,1,6
    1 X1=XX(1)
      DO 5 I=1,N1
      IF(I-N1) 3,2,3
    2 X2=XX(N1)
      GOTO 4
    3 X2=Z(I)
    4 H1=1./(X1-X2)
      A1(I)=H1/6.*(S2(I)-S2(I+1))
      A2(I)=H1*.5*(S2(I+1)*X1-S2(I)*X2)
      H3=X1*X1
      H2=X2*X2
      R1=Y(I)-H3*(A1(I)*X1+A2(I))
      R2=Y(I+1)-H2*(A1(I)*X2+A2(I))
      A3(I)=H1*(R1-R2)
      A4(I)=H1*(R2*X1-R1*X2)
      X1=X2
    5 CONTINUE
    6 RETURN
      END
```

```
      SUBROUTINE TAYLOR(N,X,H,F,VALUES,ITMAX,EPS1,EPS2,S,IPR,KENN)
C     EXTERNAL VALUES
      DIMENSION X(27),H(27),F(27),FP(27),FM(27),DX(27),DF(27,27)
      HS=1.E38
      KENN=0
      IZ=0
 1000 IZ=IZ+1
      IF(ITMAX-IZ)18,1,1
   18 KENN=1
      GO TO 9999
    1 L=0
      HL=1.
 2000 L=L+1
      IF(L-16)2,3,3
    3 KENN=-1
      GO TO 9999
    2 CALL VALUES(X,F)
      IF(IPR)9,9,13
   13 WRITE(9,14)
   14 FORMAT(1H0,42H     I           X(I)                          FX(I)/1H )
      DO 15 I=1,N
   15 WRITE(9,16) I,X(I),F(I)
   16 FORMAT(1X,I5,2E20.8)
    9 HF=0.
      DO 4 I=1,N
    4 HF=HF+F(I)*F(I)
      IF(HF-(1.-.2*HL)*HS)5,5,19
   19 HL=HL*.5
      DO 20 I=1,N
   20 X(I)=X(I)+HL*DX(I)
      GO TO 2000
    5 HS=HF
      IF(HS-EPS1)12,12,7
    7 DO 10 I=1,N
      HF=H(I)
      HZ=2.*HF
      X(I)=X(I)+HF
      CALL VALUES(X,FP)
      X(I)=X(I)-HZ
      CALL VALUES(X,FM)
      X(I)=X(I)+HF
      HZ=1./HZ
      DO 10 K=1,N
      DF(K,I)=(FP(K)-FM(K))*HZ
   10 CONTINUE
      CALL GAUSS(N,DF,F,DX,KENN)
      IF(KENN-1) 8,9999,8
    8 HZ=0.
      HM=0.
      DO 11 I=1,N
      X(I)=X(I)-DX(I)
      HM=HM+ABS(X(I))
   11 HZ=HZ+ABS(DX(I))
      IF(HZ-EPS2*HM)12,12,1000
   12 KENN=0
 9999 CALL VALUES(X,F)
      S=0.
      DO 21 I=1,N
   21 S=S+F(I)*F(I)
      ITMAX=IZ
      RETURN
      END
```

```
      SUBROUTINE WERT(Z,F)
C
C     CALLS SUBROUTINE SPLINE
C
      DIMENSION XX(27),Z(26),F(26),H(26),XY(28),Y(28),S2(28),S(28),
     1           A1(28),A2(28),A3(28),A4(28),TT(1000),ST(1000)
      COMMON XX,H,XY,Y,S2,S,A1,A2,A3,A4,TT,ST,
     1       SS,EPS1,EPS2,N,M,IPR,ITMAX,KENN,
     2       KI,KO,N1,N2,N11
      DO 4 I=1,N
      IF(Z(I)-XX(I))2,1,1
    1 IF(XX(I+1)-Z(I))2,3,3
    2 Z(I)=.5*(XX(I)+XX(I+1))
      IF(IPR) 3,3,9
    9 WRITE(KO,8) I
    3 XY(I+1)=Z(I)
    4 CONTINUE
C
      CALL SPLINE(N2,XY,Y,S2,0,0,N1,XX,S,NR,A1,A2,A3,A4)
C
      I=0
      S20=0.
    7 I=I+1
      KK=I+1
      H1=Z(I)-XX(I)
      H2=XX(KK)-Z(I)
      S21=S2(KK)+H2/(XY(KK+1)-Z(I))*(S2(KK+1)-S2(KK))
      F(I)=.5*((XX(KK)-XX(I))*Y(KK)-H1*S(I)-H2*S(KK) )
     1      +(H1**3*(S20+S2(KK))+H2**3*(S2(KK)+S21))/24.
      S20=S21
      IF(I-N)7,10,7
    8 FORMAT(1H0,1X,13HZWANG BEI I =,I3)
   10 RETURN
      END
```

```
      SUBROUTINE SPLINE(N,X,Y,F2,KENN,NAB,M,T,F,NR,H,DY,S,E)
C     DIMENSION X(N),Y(N),F2(N),T(M),F(2*M),H(N),DY(N),S(N),E(N)
      DIMENSION X(2),Y(2),F2(2),T(2),F(2),H(2),DY(2),S(2),E(2)
      IF(N-3) 16,1,1
    1 NR=0
      N1=N-1
      N2=N-2
      DO 2 I=1,N1
      H(I)=X(I+1)-X(I)
    2 DY(I)=(Y(I+1)-Y(I))/H(I)
      IF(KENN) 3,3,7
    3 F2(N)=0.
      F2(1)=0.
      DO 4 I=2,N1
    4 F2(I)=6.*(DY(I)-DY(I-1))
      Z=.5/(H(1)+H(2))
      S(1)=-H(2)*Z
      E(1)=F2(2)*Z
      K=1
      DO 5 I=2,N2
      J=I+1
      Z=1./(2.*(H(I)+H(J))+H(I)*S(K))
      S(I)=-H(J)*Z
      E(I)=(F2(J)-H(I)*E(K))*Z
    5 K=I
      F2(N1)=E(N2)
      DO 6 I=2,N2
      J=N2-I+2
      K=J-1
    6 F2(J)=S(K)*F2(J+1)+E(K)
      IF(KENN) 17,7,7
    7 IF(T(1)-X(1)) 16,8,8
    8 IF(T(M)-X(N)) 9,9,16
    9 DO 10 I=1,N1
   10 S(I)=(F2(I+1)-F2(I))/H(I)
      I=2
      K=1
      DO 15 J=1,M
   11 H2=T(J)-X(I)
      IF(H2) 13,13,12
   12 K=I
      I=I+1
      GOTO 11
   13 H1=T(J)-X(K)
      H3=H1*H2
      H4=F2(K)+H1*S(K)
      Z=(F2(I)+F2(K)+H4)/6.
      F(J)=Y(K)+H1*DY(K)+H3*Z
      IF(NAB) 15,15,14
   14 N1=M+J
      F(N1)=DY(K)+Z*(H1+H2)+H3*S(K)/6.
   15 CONTINUE
      GOTO 17
   16 NR=1
   17 RETURN
      END
```

4. Ein MAINPROGRAM

Das folgende Programm dient als Beispiel für den Aufruf von APPROX. Als Eingabe wird verlangt (Eine neue Ziffer bedeutet den Beginn einer neuen Karte):

i) IFF — Wenn IFF ≥ 0, dann gehe nach ii)
sonst Ende der Eingabe

ii) N, M — Bedeutung siehe 2.
Wenn M < 0, dann werden in jedem Intervall (x_i, x_{i+1}) i = 0,...,n-1) die Funktionswerte an M gleichmäßig verteilten Abszissen berechnet (siehe Sample Problem)
Es muß sei $/ M / * N + 1 \leq 1000$

NZ — Wenn NZ < 0, dann werden die Startwerte $z_i^{(o)}$ nach (10) bestimmt; andernfalls werden sie eingelesen

NH — Wenn NH < 0, dann werden die Schrittweiten h_i nach (11) bestimmt; andernfalls werden sie eingelesen

IPR, ITMAX, EPS1, EPS2 — Bedeutung siehe 2.

iii) XX
iv) Y
v) Z
vi) H
vii) T

Felder: Länge und Bedeutung siehe 2.
Das Feld Z darf nur für NZ ≥ 0,
das Feld H nur für NH ≥ 0 vorhanden sein
Das Feld T darf nur für M > 0 vorhanden sein
(Bedeutung siehe 2.)

Start bei i)

Im folgenden wird das MAINPROGRAM gelistet.

```
C     MAINPROGRAM
C
      DIMENSION XX(27),Z(26),F(26),H(26),XY(28),Y(28),S2(28),S(28),
     1          A1(28),A2(28),A3(28),A4(28),TT(1000),ST(1000)
      COMMON XX,H,XY,Y,S2,S,A1,A2,A3,A4,TT,ST,
     1       SS,EPS1,EPS2,N,M,IPR,ITMAX,KENN,
     2       KI,KO,N1,N2,N11
C
      KI=8
      KO=9
C
   99 READ(KI,1) IFF
      IF(IFF)9999,100,100
  100 WRITE(KO,1)
      READ(KI,1)N,M,NZ,NH,IPR,ITMAX,EPS1,EPS2
      WRITE(KO,2) N
      N1=N+1
      N2=N1+1
      N11=2*N1
      READ(KI,1)(XX(I),I=1,N1)
      READ(KI,1)(Y(I),I=1,N2)
      WRITE(KO,3)
      DO 101 I=1,N1
      WRITE(KO,4) Y(I)
  101 WRITE(KO,5) XX(I)
      WRITE(KO,4) Y(N2)
      IF(NZ)102,103,103
  102 WRITE(KO,6)
      GO TO 105
  103 READ(KI,1) (Z(I),I=1,N)
  105 IF(NH)106,104,104
  106 WRITE(KO,7)
      GO TO 107
  104 READ(KI,1)(H(I),I=1,N)
  107 IF(NZ)111,113,113
  111 DO 112 I=1,N
  112 Z(I)=.5*(XX(I)+XX(I+1))
  113 WRITE(KO,15)
      WRITE(KO,16) (Z(I),I=1,N)
      IF(NH)114,121,121
  114 DO 50 I=1,N
      IF(Z(I))14,13,14
   13 H(I)=5.E-3*XX(I+1)
      GOTO 50
   14 H(I)=Z(I)*5.E-3
   50 CONTINUE
  121 WRITE(KO,12)
      WRITE(KO,16) (H(I),I=1,N)
C
      IF(M) 108,110,110
  110 READ(KI,1)(TT(I),I=1,M)
      GO TO 1111
  108 MM=-M
      M=MM*N+1
      TT(M)=XX(N1)
      DO 1112 I=1,N
      JI=(I-1)*MM
      HH=(XX(I+1)-XX(I))/FLOAT(MM)
      DO 1113 J=1,MM
      JJ=JI+J
 1113 TT(JJ)=XX(I)+FLOAT(J-1)*HH
```

```
 1112 CONTINUE
C
 1111 CALL APPROX(Z,F)
C
      WRITE(KO,8)ITMAX,KENN
      WRITE(KO,9)
      DO 1114 I=1,N
      IF(Y(I+1))1116,1115,1116
 1115 HH=F(I)
      GOTO 1114
 1116 HH=F(I)/(Y(I+1)*(XX(I+1)-XX(I)))*100.
 1114 WRITE(KO,10) Z(I),F(I),HH,S2(I+1)
      IF(KENN) 20,20,21
   20 WRITE(KO,17)
      DO 19 I=1,N1
   19 WRITE(KO,16) A1(I),A2(I),A3(I),A4(I)
   21 WRITE(KO,11)
      DO 115 I=1,M
  115 WRITE(KO,10)TT(I),ST(I)
      GO TO 99
C
C
    1 FORMAT(1H1,49HFLAECHENTREUE APPROXIMATION VON TREPPENFUNKTIONEN)
    2 FORMAT(1H0,I2,11H UNBEKANNTE/1H0/)
    3 FORMAT(1X,37HDIE TREPPENFUNKTION IST GEGEBEN DURCH/1H0,31H
     1   X                    Y)
    4 FORMAT(1H ,21X,E20.8)
    5 FORMAT(1H ,1X,E20.8)
    6 FORMAT(1H0,39H STARTWERTE WERDEN AUTOMATISCH BESTIMMT)
    7 FORMAT(1H0,42H SCHRITTWEITEN WERDEN AUTOMATISCH BESTIMMT)
    8 FORMAT(1H0,5H NACH,I3,28H ITERATIONEN WURDE MIT KENN=,I2,29H DIE F
     1OLGENDEN WERTE ERHALTEN)
    9 FORMAT(1H0,98H        UNBEKANNTE        ABSOLUTE FLAECHENDIFFERENZEN
     1        ABWEICHUNG IN PROZENT          S2(2..N+1)/1H0)
   10 FORMAT(1X,E20.8,4X,E20.8,16X,F13.2,7X,E20.8)
   11 FORMAT(1H1,36H TABELLIERUNG DER GESUCHTEN FUNKTION/1H0)
   12 FORMAT(1H0,14H SCHRITTWEITEN)
   15 FORMAT(1H0,11H STARTWERTE)
   16 FORMAT(1X,7E17.7)
   17 FORMAT(1H1,54H TABELLIERUNG DER KOEFFIZIENTEN DER POLYNOME 3. GRAD
     1ES/1H0)
 9999 STOP
      END
```

5. Sample Problem

Input:

```
0
18 -4 -1 -1 0 20 1.-10 1.-4
0. 2. 3.5 4.5 6. 7. 9. 10. 12. 13. 14. 15. 16. 17. 19. 20. 21. 22. 23.
0. 1. 2.5 6.5 4. 2. 5.5 12. 13.5 8.5 7.5 6.5 7.5 8.5 5. 4. 3. 2. 1. .5
-1
```

Output:

```
FLAECHENTREUE APPROXIMATION VON TREPPENFUNKTIONEN

18 UNBEKANNTE


DIE TREPPENFUNKTION IST GEGEBEN DURCH

           X                    Y
                               0.
          0.
                               1.0000000E 00
          2.0000000E 00
                               2.5000000E 00
          3.5000000E 00
                               6.5000000E 00
          4.5000000E 00
                               4.0000000E 00
          6.0000000E 00
                               2.0000000E 00
          7.0000000E 00
                               5.5000000E 00
          9.0000000E 00
                               1.2000000E 01
          1.0000000E 01
                               1.3500000E 01
          1.2000000E 01
                               8.5000000E 00
          1.3000000E 01
                               7.5000000E 00
          1.4000000E 01
                               6.5000000E 00
          1.5000000E 01
                               7.5000000E 00
          1.6000000E 01
                               8.5000000E 00
          1.7000000E 01
                               5.0000000E 00
          1.9000000E 01
                               4.0000000E 00
          2.0000000E 01
                               3.0000000E 00
          2.1000000E 01
                               2.0000000E 00
          2.2000000E 01
                               1.0000000E 00
          2.3000000E 01
                               5.0000000E-01
```

STARTWERTE WERDEN AUTOMATISCH BESTIMMT

SCHRITTWEITEN WERDEN AUTOMATISCH BESTIMMT

STARTWERTE

1.000000E 00	2.750000E 00	4.000000E 00	5.250000E 00	6.500000E 00	8.000000E 00	9.500000E 00
1.100000E 01	1.250000E 01	1.350000E 01	1.450000E 01	1.550000E 01	1.650000E 01	1.800000E 01
1.950000E 01	2.050000E 01	2.150000E 01	2.250000E 01			

SCHRITTWEITEN

5.000000E-03	1.375000E-02	2.000000E-02	2.625000E-02	3.250000E-02	4.000000E-02	4.750000E-02
5.500000E-02	6.250000E-02	6.750000E-02	7.250000E-02	7.750000E-02	8.250000E-02	9.000000E-02
9.750000E-02	1.025000E-01	1.075000E-01	1.125000E-01			

NACH 6 ITERATIONEN WURDE MIT KENN= 0 DIE FOLGENDEN WERTE ERHALTEN

UNBEKANNTE	ABSOLUTE FLAECHENDIFFERENZEN	ABWEICHUNG IN PROZENT	S2(2..N+1)
4.6494700E-01	-1.2000000E-07	-0.00	-4.2451309E 00
2.8736693E 00	-3.0000000E-07	-0.00	6.3228257E 00
3.7755496E 00	-2.0000000E-08	-0.00	-9.7182947E 00
5.2089942E 00	2.1200000E-07	0.00	1.8228457E 00
6.1425297E 00	-3.0000000E-08	-0.00	3.1185765E 00
8.0782994E 00	-4.3000000E-07	-0.00	2.1204835E 00
9.4769803E 00	2.5000000E-08	0.00	-2.2472063E 00
1.1312309E 01	4.6000000E-07	0.00	-6.2171789E 00
1.2405522E 01	-1.5700000E-06	-0.00	7.4945160E 00
1.3636762E 01	-8.0800000E-07	-0.00	-4.4472996E 00
1.4270559E 01	9.2000000E-06	0.00	4.3666857E 00
1.5509777E 01	1.5000000E-07	0.00	6.2092760E-01
1.6702722E 01	-2.4000000E-07	-0.00	-6.9105140E 00
1.7751440E 01	1.2900000E-06	0.00	4.9463614E 00
1.9672942E 01	-4.5000000E-08	-0.00	-2.7241674E 00
2.0473256E 01	9.4620000E-07	0.00	1.1875628E 00
2.1513670E 01	6.1390000E-07	0.00	-4.4371701E-01
2.2494185E 01	4.9390000E-07	0.00	2.0967153E-01

TABELLIERUNG DER KOEFFIZIENTEN DER POLYNOME 3. GRADES

-1.521726E 00	0.	2.479743E 00	0.
7.312283E-01	-3.142513E 00	3.940844E 00	-2.264447E-01
-2.964384E 00	2.871739E 01	-8.761398E 01	8.747300E 01
1.341889E 00	-2.005825E 01	9.654088E 01	-1.442890E 02
2.313304E-01	-2.703573E 00	6.140461E 00	1.267612E 01
-8.593421E-02	3.142849E 00	-2.977135E 01	8.620589E 01
-5.204534E-01	1.367338E 01	-1.148401E 02	3.152762E 02
-3.605142E-01	9.126155E 00	-7.174617E 01	1.791427E 02
2.090427E 00	-7.405127E 01	8.691825E 02	-3.368883E 03
-1.616502E 00	6.390793E 01	-8.422732E 02	3.708284E 03
2.317773E 00	-9.704440E 01	1.352595E 03	-6.268679E 03
-5.037798E-01	2.375100E 01	-3.712227E 02	1.931268E 03
-1.052220E 00	4.926954E 01	-7.670094E 02	3.977451E 03
1.884344E 00	-9.787629E 01	1.690726E 03	-9.706165E 03
-6.653240E-01	3.790456E 01	-7.195792E 02	4.555962E 03
8.146240E-01	-4.944024E 01	9.987504E 02	-6.712240E 03
-2.613190E-01	1.664393E 01	-3.542081E 02	2.520910E 03
1.110621E-01	-7.389921E 00	1.628482E 02	-1.187010E 03
-6.908703E-02	4.767005E 00	-1.106120E 02	8.634125E 02

TABELLIERUNG DER GESUCHTEN FUNKTION

0.	0.
5.0000000E-01	1.0497526E 00
1.0000000E 00	1.3031152E 00
1.5000000E 00	1.0820639E 00
2.0000000E 00	9.3502017E-01
2.3750000E 00	1.2032141E 00
2.7500000E 00	2.0528896E 00
3.1250000E 00	3.6567408E 00
3.5000000E 00	5.5141215E 00
3.7500000E 00	6.4339230E 00
4.0000000E 00	6.8234257E 00
4.2500000E 00	6.7185520E 00
4.5000000E 00	6.2450325E 00
4.8750000E 00	5.1187337E 00
5.2500000E 00	3.8705414E 00
5.6250000E 00	2.8453138E 00
6.0000000E 00	2.1576222E 00
6.2500000E 00	1.9224323E 00
6.5000000E 00	1.8777718E 00
6.7500000E 00	2.0165027E 00
7.0000000E 00	2.3305687E 00
7.5000000E 00	3.4524807E 00
8.0000000E 00	5.1790569E 00

8.5000000E 00	7.4132618E 00
9.0000000E 00	9.8481657E 00
9.2500000E 00	1.1018976E 01
9.5000000E 00	1.2093638E 01
9.7500000E 00	1.3026608E 01
1.0000000E 01	1.3782217E 01
1.0500000E 01	1.4626162E 01
1.1000000E 01	1.4355085E 01
1.1500000E 01	1.2714805E 01
1.2000000E 01	1.0183423E 01
1.2250000E 01	9.0455150E 00
1.2500000E 01	8.2509730E 00
1.2750000E 01	7.8536222E 00
1.3000000E 01	7.7158601E 00
1.3250000E 01	7.6861394E 00
1.3500000E 01	7.6129133E 00
1.3750000E 01	7.3503471E 00
1.4000000E 01	6.9183111E 00
1.4250000E 01	6.5240324E 00
1.4500000E 01	6.3507224E 00
1.4750000E 01	6.4069602E 00
1.5000000E 01	6.6455412E 00
1.5250000E 01	7.0192360E 00
1.5500000E 01	7.4808150E 00
1.5750000E 01	7.9754464E 00
1.6000000E 01	8.4140974E 00
1.6250000E 01	8.6981230E 00
1.6500000E 01	8.7288777E 00
1.6750000E 01	8.4080261E 00
1.7000000E 01	7.7131401E 00
1.7500000E 01	5.8345001E 00
1.8000000E 01	4.4425967E 00
1.8500000E 01	3.9986936E 00
1.9000000E 01	4.0443284E 00
1.9250000E 01	4.0947872E 00
1.9500000E 01	4.0805080E 00
1.9750000E 01	3.9397940E 00
2.0000000E 01	3.6600146E 00
2.0250000E 01	3.3098856E 00
2.0500000E 01	2.9657576E 00
2.0750000E 01	2.6812586E 00
2.1000000E 01	2.4438630E 00
2.1250000E 01	2.2290718E 00
2.1500000E 01	2.0123867E 00
2.1750000E 01	1.7742241E 00
2.2000000E 01	1.5181727E 00
2.2250000E 01	1.2546440E 00
2.2500000E 01	9.9404981E-01
2.2750000E 01	7.4378645E-01
2.3000000E 01	5.0000001E-01

III. Literatur

[1] Greville, T.N.E.
Spline functions, interpolation, and numerical quadrature
in: Ralston/Wilf
Mathematical Methods for Digital Computers, Vol. II, p. 156-168, J. Wiley and Sons, 1967

[2] Braess, D. und Späth, H.
Maßnahmen zur globalen Konvergenzerzwingung beim Newtonschen Verfahren für spezielle nichtlineare Gleichungssysteme.
ZAMM 47, p. 409-410 (1967)

[3] Späth, H.
The Damped Taylor's Series Method for Minimizing a Sum of Squares and for Solving Systems of Nonlinear Equations (Algorithm 315).
Communications of the Association of Computing Machinery 10, p. 726-728 (1967)