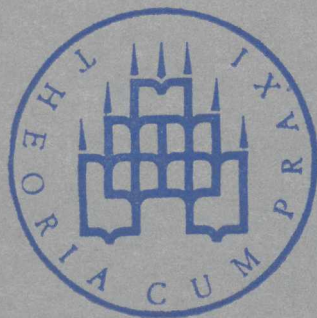


Rostocker

Mathematisches Kolloquium

Heft 6



WILHELM-PIECK-
UNIVERSITÄT ROSTOCK
1977



Rostocker Mathematisches Kolloquium

Heft 6

Tagung Digitalgraphik
in Rostock-Warnemünde
vom 30. 10. bis 5. 11. 1976

Kurzvorträge, Teil 1

1977

Wilhelm-Pieck-Universität Rostock
Sektion Mathematik

Redaktion: Abt. Wissenschaftspublizistik der Wilhelm-Pieck-Universität Rostock
DDR - 25 Rostock, Vogelsang 13/14
Fernruf 369577

Verantw. Redakteur: Dipl.-Ges.-Wiss. Bruno Schrage
Herausgegeben von der Wilhelm-Pieck-Universität Rostock unter
Genehmigungs-Nr. C 1.103/77
Druck: Ostsee-Druck Rostock, Werk II

BÄR, Gert:	Hermitesche und Spline-Interpolation empirischer Kurven und Flächen mit geometrischen Interpolationsbedingungen	5
BERNDT, Erhard:	Realisierungsvarianten der Dialogsteuerung	23
BÖHME, Wolfgang und PFLUG, Arndt:	Nutzung der graphischen Datenverarbeitung für den Informationsfonds einer Kombinatsleitung	33
BROSCHINSKI, Dagmar:	Rekonstruktion geometrischer Objekte aus ihrer Darstellung im Grund-Aufriß-Verfahren	51
COHEN, Serge:	Programmpaket zur Interpolation empirischer Kurven und Flächen im Raum mit Möglichkeiten der Steuerung	57
DOBROWOLNY, Volker:	Bildmanipulationen am Koordinatenmeßtisch	61
FEHLAUER, Klaus-Uwe:	Fehlerabschätzung bei der Interpolation von Flächenstücken	67
FISCHER, Manfred:	ORTEP- ein Zeichenprogramm zur Darstellung von Kristallstrukturen	75
FRANZE, Klaus:	Zur algorithmischen Notation von Konstruktionsbeschreibungen als Vorstufe der Digitalgraphik	81
GEISE, Gerhard:	Ein Kalkül für geometrische Aufgaben	95
JAECKEL, Ulrich:	Einsatz der Bildschirmtechnik bei der Untersuchung von Schiffsschwingungen	103
JAHN, Gerhard und MEYER, Bernd:	Anwendung der Digitalgraphik bei der Herstellung von Behälterzeichnungen	107

		Seite
KOCH, Lutz	Ein Verfahren zur automatischen Eingabe technischer Zeichnungen in DVA	113
KRAUSE, Heidrun:	Anwendung geometrischer Grundoperationen in einem Kleinrechner-Dialogsystem für KRS 4200	115
KUSSEROW, Peter:	Der Einsatz der Digitalgraphik bei der automatischen Fertigung von Leiterplatten	121
LOBER, Reinhard:	Rechnerinterne Darstellung technischer Ge- bilde (RID)	125
LUDWIG, Manfred:	Eine Fachsprache für die Konstruktion von Werkzeugmaschinengestellen	137

Hermiteische und Spline-Interpolation empirischer Kurven und Flächen mit geometrischen Interpolationsbedingungen

0. Einleitung

Die analytische Behandlung von empirischen Kurven und Flächen hat u. a. im Zusammenhang mit der Entwicklung von NC-Maschinen für Konstruktion und Fertigung von skulpturartigen Flächen in jüngster Zeit stark an Bedeutung gewonnen. Eine Aufgabe ist dabei die Bestimmung von Interpolationskurven oder -flächen durch eine gegebene Stützpunktmenge, wobei evtl. weitere geometrische Vorgaben (Tangente, Krümmung) einzuhalten sind. In diesem Problemkreis geht es um die Darstellung geometrischer Objekte. Dabei sind differentialgeometrische Aspekte bisher wenig beachtet worden. Die Interpolationsverfahren dieses Beitrages berücksichtigen vier Forderungen bei der Interpolation geometrischer Objekte, die vom Autor in den Aufsätzen /1/, /2/ motiviert wurden:

1. Das Interpolationsergebnis (Kurve, Fläche) soll unabhängig von der Wahl des Koordinatensystems sein. Dies erreicht man durch die Verwendung von Vektorfunktionen zur Darstellung der interpolierenden Kurven oder Flächen.
2. Das Interpolationsergebnis soll invariant sein gegenüber zulässigen Parametertransformationen der beschreibenden Vektorfunktion. Dies wird durch die Verwendung geometrischer Invarianten (Bogenlänge, Tangente, Krümmung, Windung) als Interpolationsbedingungen erreicht.
3. Die interpolierenden Kurven und Flächen sollen möglichst glatt verlaufen. Deshalb werden Verallgemeinerungen des Hermiteischen bzw. Spline-Interpolationsverfahrens verwendet und als Maß für die Glätte die Ordnung einer Kurve eingeführt, die der Anzahl der gemeinsamen invarianten Ableitungen in einem Verknüpfungspunkt zweier Kurvenstücke oder Parameterlinienstücke von Flächen entspricht. Für die einzelnen Verfahren wird die erreichbare Ordnung angegeben.

4. Die Interpolationskurven und -flächen sollen stückweise manipulierbar sein, um z. B. im interaktiven Dialog mit Bildschirmgerät und Rechner gewisse Kurven- oder Flächenstücke mit Hilfe geeigneter Parameter solange variieren zu können, bis alle konstruktiven und ästhetischen Gesichtspunkte erfüllt sind. Bei den Interpolationsverfahren eingeführte "freie" Parameter gestatten es, die geometrische Gestalt der Kurven oder Flächen lokal zu beeinflussen, ohne deren Ordnung zu verändern.

Dieser Aufsatz behandelt nur theoretische Grundlagen. Sie werden durch Probleme der rechentechnischen Realisierung einiger der hier dargelegten Verfahren sowie durch Anwendungsbeispiele ergänzt, die Herr S. COHEN /3/ in einem Beitrag dieses Heftes vorstellt.

1. Interpolation von Kurven im Raum

1.1. Definition und Eigenschaften der Interpolationskurven

Für die Darstellung stückweise definierter Interpolationskurven werden sogenannte Bindefunktionen (oder Mischungsfunktionen) $f_{kr}(u)$ benutzt, deren Verwendung durch die Arbeiten /2/, /4/, /7/ motiviert ist:

Definition 1.1. Es sei $F^{a,b}$ die Menge der Funktionen

$$f_{kr}(u) \in C^a \left[\underline{0}, \underline{1} \right], \quad a, b \in \mathbb{N} \ (a \geq b), \quad k = 0, 1, \quad r = 0 \ (1) \ b,$$

mit den Eigenschaften

$$f_{kr}^{(\ell)}(q) := \frac{d^\ell}{du^\ell} f_{kr}(q) = \left\{ \begin{array}{ll} 1 & k = q \text{ und } r = \ell \\ \text{für} & \\ 0 & k = q \text{ oder } r = \ell \end{array} \right\} \quad (1.1)$$

$$r, \ell = 0 \ (1) \ b, \quad k, q = 0, 1$$

An eine Bindefunktion $f_{kr} \in F^{a,b}$ werden $2 \cdot (b+1)$ Forderungen gestellt; folglich kann eine solche Funktion durch einen linearen Ansatz mit $2 \cdot (b+1)$ linear unabhängigen Grundfunktionen aus $C^a \left[\underline{0}, \underline{1} \right]$ konstruiert werden. Zum Beispiel folgen polynomiale Bindefunktionen aus dem Ansatz

$$f_{kr}(u) = \sum_{j=0}^{2b+1} a_{kr,j} u^j.$$

Eine rekursive Berechnungsvorschrift für diese Polynome gibt SCHMIEDER /6/ an. Die polynomialen Bindefunktionen $f_{kr} \in F^{a, b}$ sind die bekannten Hermite-Polynome von Grad $2b + 1$ und dem Rang b über dem Intervall $[0, 1]$.

Zur Beschreibung einer Kurve im Raum R^2 (ebene Kurve) oder im R^3 (Raumkurve) wird in Erweiterung der Methode der finiten Elemente /7/ auf Vektorfunktionen eine stückweise Darstellung verwendet.

Definition 1.2. Es sei $X_n^{a, b}$ die Menge aller Folgen

$$x(u) = (x_i(u))_{i=1}^{n-1}, \quad u \in [0, 1]$$

von $n - 1$ reellen Vektorfunktionen $x_i : I \rightarrow R^d$ ($d = 2$ bzw. 3)

mit

$$x_i(u) = \sum_{r=0}^b (\varphi_i^r c_{r,i} f_{or}(u) + \lambda_{i+1}^r c_{r,i+1} f_{1r}(u)) \quad (1.2)$$

für beliebige Koeffizienten $\varphi_i, \lambda_{i+1} > 0$ ($i = 1(1)n - 1$),

Koeffizientenvektoren $c_{r,i} \in R^d$ ($d \geq 2$) mit $c_{1,i} \neq 0$

($i = 1(1)n$) und $f_{kr} \in F^{a, b}$ ($k = 0, 1; r = 0(1)b$).

$x(u)$ wird IP-Kurve und $x_i(u)$ wird IP-Kurvenstück genannt.

Lemma 1.1. Es sei $x(u) \in X_n^{a, b}$ mit $a \leq b \leq 3$ und G die Bogenlänge von $x(u)$. Dann gilt

$$\frac{d^r}{dG^r} x_{i-1}(1) = \frac{d^r}{dG^r} x_i(0), \quad r = 0(1)a, \quad i = 2(1)n-1. \quad (1.3)$$

D. h., die IP-Kurvenstücke $x_{i-1}(u)$ und $x_i(u)$ schließen in den Punkten $x_{i-1}(1) = x_i(0)$, $i = 2(1)n - 1$ mit denselben invarianten Ableitungen bis zur Ordnung a aneinander an.

Im einzelnen besitzen die Kurvenstücke

für $a = 0$ denselben End- bzw. Anfangspunkt $x_{i-1}(1) = x_i(0)$,
 $i = 2(1)n - 1$,

und in diesen Punkten

für $a \geq 1$ dieselbe Tangentenrichtung

für $a \geq 2$ denselben Krümmungsvektor (also dieselbe Krümmung),

für $a = 3$ denselben Windungsvektor (also dieselbe Windung).

Beweis: Ohne explizite Kenntnis der Bogenlänge \mathcal{G} von $x(u) \in X_n^{a,b}$ ($a \leq b \leq 3$) können die invarianten Ableitungen $x_i^{[r]}(u) := \frac{d^r}{d\mathcal{G}^r} x_i(u(\mathcal{G}))$ nach bekannten Beziehungen der

Differentialgeometrie (vgl. z. B. /9/) berechnet werden. Wir kennzeichnen wie üblich die Ableitungen von $x(u)$ nach der Bogenlänge mit der entsprechenden Anzahl von Strichen, die Ableitungen nach einem beliebigen Kurvenparameter mit der entsprechenden Anzahl von Punkten:

$$x_i' := x_i^{[1]}(u), x_i'' := x_i^{[2]}(u), \dots, \dot{x}_i := \frac{d}{du} x_i(u), \ddot{x}_i := \frac{d^2}{du^2} x_i(u)$$

usw. Weiter sei

$$|x_i| = \sqrt{x_i^T x_i} \text{ für } x_i = (x_{i1}, \dots, x_{di})^T \in \mathbb{R}^d.$$

Es gilt

$$x_i' = | \dot{x}_i |^{-1} \dot{x}_i$$

$$x_i'' = | \dot{x}_i |^{-2} P_i \ddot{x}_i$$

$$x_i''' = | \dot{x}_i |^{-3} P_i \ddot{x}_i' - | \dot{x}_i |^{-5} (\dot{x}_i^T \dot{x}_i') (E - 4 P_i) \dot{x}_i - | \dot{x}_i |^{-5} | \dot{x}_i |^2 \ddot{x}_i$$

wobei E die $d \times d$ -Einheitsmatrix und

$$P_i = E - (x_i')^T \cdot (x_i')$$

ist.

Dabei wurden zur Vereinfachung der späteren Beweisführung die Matrizen $P_i = P_i(u)$, $i = 1(1)n-1$, eingeführt. Man bestätigt leicht, daß P_i eine symmetrische Projektionsmatrix vom Rang $d-1$ ist:

$$P_i = P_i^T, P_i^2 = P_i, \text{rg } P_i = d-1 \quad (1,5)$$

und daß

$$\left. \begin{array}{l} P_i y = 0, \quad y \in \mathbb{R}^d \\ \text{genau dann gilt, wenn} \\ y = \alpha x_i', \quad \alpha \in \mathbb{R}. \end{array} \right\} \quad (1,6)$$

Wir bilden nun die Ableitungen $x_i^{(r)}(u) = \frac{d^r}{du^r} x_i(u)$ unter Beachtung von (1.1) an den

Stellen $u = 0$ und $u = 1$:

$$\left. \begin{aligned} x_i^{(r)}(0) &= \delta_i^r c_{r,i} \\ x_i^{(r)}(1) &= \lambda_{i+1}^r c_{r,i+1} \end{aligned} \right\} i = 1(1)n-1 \quad (1.7)$$

Damit ist für $r = 0$

$x_{i-1}(1) = x_i(0) = c_{0i}$, womit (1.3) für $a = 0$ bewiesen ist.

Setzen wir (1.7) in (1.4) ein, so folgt

$$\left. \begin{aligned} x_i'(0) &= \delta_i c_{1,i}, \quad x_i'(1) = \delta_{i+1} c_{1,i+1} \text{ mit } \delta_i := |c_{1i}|^{-1} \\ x_i''(0) &= \delta_i^2 P_i(0) c_{2,i}, \quad x_i''(1) = \delta_{i+1}^2 P_i(1) c_{2,i+1} \\ x_i'''(0) &= \delta_i^3 P_i(0) c_{3,i} + \delta_i^5 (c_{1i}^T c_{2i}) (E - 4 P_i(0)) c_{2i} - \delta_i^5 |c_{2i}|^2 c_{1i} \\ x_i'''(1) &= \delta_{i+1}^3 P_i(1) c_{3,i+1} + \delta_{i+1}^5 (c_{1,i+1}^T c_{2,i+1}) (E - 4 P_i(1)) c_{2,i+1} - \\ &\quad \delta_{i+1}^5 |c_{2,i+1}|^2 c_{1,i+1} \end{aligned} \right\} \quad (1.8)$$

Für $a \geq 1$ gilt $x'_{i-1}(1) = x'_i(0)$ und damit ist (1.3) für $a = 1$ gezeigt. Für $a \geq 1$ ist aber weiter

$$P_{i-1}(1) = P_i(0)$$

und aus (1.8) folgt mit (1.9), daß (1.3) für $a = 2$ und $a = 3$ gilt.

Definition 3 Gelten die invarianten Anschlußbedingungen (1.3) für $x(u) \in X_n^{a,b}$ ($a, b \in \mathbb{N}$), so heie $x(u)$ eine IP-Kurve der Ordnung a.

Nach Lemma 1.1 ist jede Kurve $x(u) \in X_n^{a,b}$ für $a \leq b \leq 3$ eine IP-Kurve der Ordnung a.

Es können aber auch IP-Kurven der Ordnung a konstruiert werden, bei denen der Rang b der Bindefunktionen kleiner als die Ordnung a ist, indem die Bedingungen (1.3) als Forderungen an die Koeffizienten δ_i, λ_{i+1} ($i = 1(1)n-1$) und Koeffizientenvektoren $c_{r,i}$ ($i = 1(1)n$) von (1.2) aufgefat werden.

Hierzu das folgende Lemma:

Lemma 1.2 Sei $x \in X_n^{1,1}$ und $f_{kr} \in F^{2,1}$ ($k, r = 0, 1$).

Genau dann ist $x \in X_n^{2,1}$, wenn das lineare Gleichungssystem

$$\begin{aligned} & \mathcal{S}_{i-1} \mathcal{S}_i^{2,1} f_{01,1} c_{1,i-1} + \alpha_i c_{1,i} - \lambda_i \lambda_{i+1} \ddot{f}_{11,0} c_{1,i+1} = \\ & \lambda_i^2 (\ddot{f}_{00,0} c_{0,i} + \ddot{f}_{10,0} c_{0,i+1}) - \mathcal{S}_i^2 (\ddot{f}_{00,1} c_{0,i-1} + \ddot{f}_{10,1} c_{0,i}), \quad (1.10) \end{aligned}$$

$$i = 2(1)n-1, \alpha_i \in \mathbb{R},$$

erfüllt ist, wobei

$$\ddot{f}_{kr,q} := \frac{d^2}{du^2} f_{kr}(q), \quad k, r, q = 0, 1,$$

gesetzt wurde.

Beweis: $x \in X_n^{2,1}$ gilt unter Beachtung von (1.4) genau dann, wenn $x'_{i-1}(1) - x'_i(0) =$

$$|x'_{i-1}(1)|^{-2} p_{i-1}(1) \dot{x}'_{i-1}(1) - |x'_i(0)|^{-2} p_i(0) \dot{x}'_i(0) = 0$$

für $i = 2(1)n-1$ gilt. Äquivalent dazu ist wegen (1.9) das homogene Gleichungssystem

$$p_i(0) (\lambda_i^{-2} \ddot{x}'_{i-1}(1) - \mathcal{S}_i^{-2} \ddot{x}'_i(0)) = 0, \quad i = 2(1)n-1,$$

das wegen (1.6) die vollständige Lösung

$$\lambda_i^{-2} \ddot{x}'_{i-1}(1) - \mathcal{S}_i^{-2} \ddot{x}'_i(0) + \bar{\alpha}_i c_{1,i} = 0, \quad \bar{\alpha}_i \in \mathbb{R},$$

besitzt.

Durch Einsetzen von $\ddot{x}'_{i-1}(1)$ und $\ddot{x}'_i(0)$ und eine einfache Umformung folgt die Behauptung.

1.2. Interpolationsaufgaben

Eine zu interpolierende empirische Kurve des \mathbb{R}^d ($d = 2$ bzw. 3) sei durch eine Folge von Stützpunkten $p_i = (p_{1i}, p_{2i}, \dots, p_{di})^T \in \mathbb{R}^d$, $i = 1(1)n$, gegeben. Des weiteren können Tangenteneinheitsvektoren $p'_i = (p'_{1i}, \dots, p'_{di})^T \in \mathbb{R}^d$, $i = 1(1)n$, der in den Stützpunkten an die empirischen Kurven gelegten Tangenten sowie Krümmungsvektoren p''_i , $i = 1(1)n$, und auch Windungsvektoren p'''_i , $i = 1(1)n$, gegeben sein, falls man Kenntnisse über die Krümmung und Windung der empirischen Kurve in den Stützpunkten p_i besitzt. Diese Vorgaben sind nur sinnvoll, wenn sie den differentialgeometrischen Bedingungen

$$(p_i')^T p_i' = 1,$$

$$(p_i')^T p_i'' = 0, \quad (1.11)$$

$$(p_i'')^T p_i'' + (p_i')^T p_i''' = 0$$

genügen, was in folgendem vorausgesetzt wird.

Um die Länge bzw. eine Abschätzung für die Länge der empirischen Kurve berücksichtigen zu können, sei jedem Stützpunkt p_i ein Stützparameterwert $\sigma_i \in \mathbb{R}$ so zugeordnet, daß

$$\sigma_1 < \sigma_2 < \dots < \sigma_n$$

gilt und

$$\sigma_i = \sigma_{i+1} - \sigma_i, \quad i = 1(1)n-1,$$

die Länge des Kurvenstückes $p_i p_{i+1}$ oder ein Näherungswert dafür ist. Zur Festlegung von Stützparameterwerten bietet sich z. B. die Länge σ_i des Sehnepolygons

$p_1 p_2 \dots p_i$ an:

$$\sigma_i = \sigma_{i-1} + |p_i - p_{i-1}|, \quad \sigma_1 = 0, \quad i = 2(1)n.$$

Ein nach diesem Prinzip festgelegter Parameter σ , $\sigma_1 \leq \sigma \leq \sigma_n$, heiße empirische Bogenlänge der zu interpolierenden Kurve.

Ist σ die exakte Bogenlänge von $x(u) \in X_n^{a,b}$ ($a \geq 1$),

so gilt

$$\left| \frac{d}{d\sigma} x(u(\sigma)) \right| = 1$$

für $\sigma_1 \leq \sigma \leq \sigma_n$.

Wir wollen deshalb für einige IP-Aufgaben die praktische Forderung stellen, daß diese Eigenschaft in den Stützpunkten bezüglich der empirischen Bogenlänge σ erfüllt wird:

$$\left| \frac{d}{d\sigma} x_i(u(\sigma_i)) \right| = 1, \quad i = 1(1)n-1, \quad (1.12)$$

$$\left| \frac{d}{d\sigma} x_{i-1}(u(\sigma_i)) \right| = 1, \quad i = 2(1)n,$$

wobei wir die Parametertransformationen

$$u = u(\sigma) = \sigma_i^{-1}(\sigma - \sigma_i) \text{ für } \sigma_i \leq \sigma \leq \sigma_{i+1} \quad (i = 1(1)n-1)$$

annehmen.

Lemma 1.3 Für $x \in X_n^{a,b}$ mit $a \geq 1$ sind die Forderungen (1.12) genau dann erfüllt, wenn

$$\begin{aligned} \delta_i &= \delta_i \delta_i, \lambda_{i+1} = \delta_{i+1} \delta_i, u = \delta_i^{-1} (\sigma - \sigma_i), i = 1(1)n-1 \\ \text{mit } \delta_i &:= |c_{1i}|^{-1} \text{ gilt.} \end{aligned} \quad (1.13)$$

Ohne Beweis.

Zur Abkürzung der gestellten IP-Bedingungen führen wir folgende Bezeichnung ein:

$$\begin{aligned} X_n^{a,b}(\underline{p_i} \underline{\int^c \int}) &= \left\{ x(u) \in X_n^{a,b} / x_i \underline{\int^r \int}(0) = p_i \underline{\int^r \int}, i = 1(1)n-1, \right. \\ &\left. x_{n-1} \underline{\int^r \int}(1) = p_n \underline{\int^r \int} \text{ für } r = 0, 1, \dots, c \right\} \end{aligned}$$

Das heißt, für $c = 0$ ist $x(u) \in X_n^{a,b}(\underline{p_i} \underline{\int^0 \int})$ eine IP-Kurve der Ordnung a bei Verwendung von Bidefunktionen vom Rang b und verläuft durch die Stützpunkte $p_i, i = 1(1)n$. Für $c = 1, 2$ bzw. 3 stimmen zusätzlich die invarianten Ableitungen $x_i \underline{\int^c \int}$ der IP-Kurve in den Stützpunkten mit vorgegebenen Tangenteneinheitsvektoren $p_i \underline{\int^1 \int}$, Krümmungsvektoren $p_i \underline{\int^2 \int}$ bzw. Windungsvektoren $p_i \underline{\int^3 \int}$ überein.

1.3. Verallgemeinerungen des Hermiteschen und des Spline-Interpolationsverfahrens

Eine Verallgemeinerung des bekannten Hermiteschen IP-Verfahrens ist der Inhalt von Satz 1.1

- a) $x \in X_n^{a,a}(\underline{p_i} \underline{\int^0 \int})$ für $a \geq 0$ genau dann, wenn $c_{0i} = p_i, i = 1(1)n$
- b) $x \in X_n^{a,a}(\underline{p_i} \underline{\int^1 \int})$ für $a \geq 1$ genau dann, wenn (a) und $c_{1i} = \alpha_i p_i', \alpha_i \in \mathbb{R}^+, i = 1(1)n$
- c) $x \in X_n^{a,a}(\underline{p_i} \underline{\int^2 \int})$ für $a \geq 2$ genau dann, wenn (b) und

$$c_{2i} = \alpha_i^2 p_i'' + \beta_i p_i$$

$$\beta_i \in \mathbb{R}, i = 1(1)n$$

d) $x \in X_n^{3,3}(\underline{p}_i, \underline{\overset{\sim}{3}})$ genau dann, wenn (c) und

$$c_{3i} = \alpha_i \overset{3}{p_i} + 3\alpha_i \beta_i \overset{3}{p_i} + \alpha_i \overset{3}{p_i},$$

$$\alpha_i \in \mathbb{R}, i = 1(1)n.$$

Auf den Beweis soll hier verzichtet werden; er wird im wesentlichen mit den im Beweis von Lemma 1.1 zur Verfügung gestellten Aussagen geführt.

Stellen wir an die nach Satz 1.1 konstruierbaren IP-Kurven die Forderung (1.12), so finden wir mit Lemma 1.3 die

Folgerung 1.1: Die invarianten Ableitungen $\underline{p}_i, \underline{\tilde{r}} (r = 0, 1, \dots, c)$ bis zur c -ten Ordnung und die Stützparameterwerte $\underline{\sigma}_i$ einer empirischen Kurve seien gegeben.

Es gibt genau eine IP-Kurve $x(u) \in X_n^{c,c}(\underline{p}_i, \underline{\tilde{c}})$, $c = 1, 2, 3$, die den Bedingungen (1.12) genügt, nämlich

$$x_i(u) = \sum_{r=0}^c \sigma_i^r (\underline{p}_i, \underline{\tilde{r}} f_{or}(u) + \underline{p}_{i+1}, \underline{\tilde{r}} f_{1r}(u)), \sigma_i = \sigma_{i+1} - \sigma_i,$$

$$i = 1(1)n-1. \quad (1.15)$$

Die nach (1.15) bestimmte IP-Kurve ist für polynomiale Bindefunktionen identisch mit jener IP-Kurve, die man durch Anwendung des Hermiteschen IP-Verfahrens $(2c+1)$ -ten Grades in jeder Koordinate einer Vektorfunktion erhält, welche über den Stützstellen $\underline{\sigma}_i$ die Stützpunkte $\underline{p}_i, \underline{\overset{\sim}{0}}$ und Ableitungen $\underline{p}_i, \underline{\tilde{c}}$, $c = 1, 2, 3$, $i = 1(1)n$, interpoliert (vgl. '77).

Satz 1.2 (Verallgemeinerte Spline-Interpolation).

Von der empirischen Kurve seien die Stützpunkte \underline{p}_i , $i = 1(1)n$ sowie die Anfangs- und Endtangentialrichtung \underline{p}_1' und \underline{p}_n' bekannt. Ist $x \in X_n^{1,1}(\underline{p}_i, \underline{\overset{\sim}{0}})$ mit $f_{kr} \in F^{2,1}$ ($k, r = 0, 1$) und $\lambda_i, \lambda_{i+1} > 0$ ($i = 1(1)n-1$), $\alpha_i \in \mathbb{R}$ ($i = 1(1)n$), dann gibt es genau die IP-Kurve $x \in X_n^{2,1}(\underline{p}_i, \underline{\overset{\sim}{0}})$ mit $x_1, \underline{\overset{\sim}{1}}(0) = \underline{p}_1'$, $x_{n-1}, \underline{\overset{\sim}{1}}(1) = \underline{p}_n'$, falls das lineare

Gleichungssystem

$$A c = a \quad (1.16)$$

mit $c_{1i} \neq 0$, $i = 2(1)n-1$, eindeutig lösbar ist. Dabei ist

$$c = (c_{12}, c_{13}, \dots, c_{1,n-1})^T$$

$$A = (a_{ij})_{n-2, n-2}$$

$$a_{ij} = \begin{cases} \alpha_i + 1 & 1 \leq i = j \leq n-2, \\ \mathcal{S}_i \mathcal{S}_{i+1}^2 \ddot{f}_{01,1} & \text{für } 2 \leq i = j+1 \leq n-2 \\ \lambda_{i+1}^2 \lambda_{i+2} \ddot{f}_{11,0} & 1 \leq i = j-1 \leq n-3 \\ 0 & \text{sonst} \end{cases}$$

$$a = (a_1, a_2, \dots, a_{n-2})^T$$

$$a_1 = \lambda_2^2 h_1 - \mathcal{S}_2^2 k_1 - \mathcal{S}_1 \mathcal{S}_2^2 \ddot{f}_{01,1} p'_1$$

$$a_i = \lambda_{i+1}^2 h_{i+1} - \mathcal{S}_{i+1}^2 k_i \quad (i = 2(1)n-3)$$

$$a_{n-2} = \lambda_{n-1}^2 h_{n-1} - \mathcal{S}_{n-1}^2 k_{n-2} + \lambda_{n-2}^2 \lambda_n \ddot{f}_{11,0} p'_n$$

wobei abkürzend

$$h_i = \ddot{f}_{00,0} p_i + \ddot{f}_{10,0} p_{i+1}$$

$$k_i = \ddot{f}_{00,1} p_i + \ddot{f}_{10,1} p_{i+1}$$

gesetzt wurde.

$x \in X_n^{2,1}(\rho_i, \sqrt[0]{-})$ besitzt dann die Darstellung

$$x_1(u) = f_{00} p_1 + f_{10} p_2 + \mathcal{S}_1 f_{01} p'_1 + \lambda_2 f_{11} c_{12}$$

$$x_i(u) = f_{00} p_i + f_{10} p_{i+1} + \mathcal{S}_i f_{01} c_{1i} + \lambda_{i+1} f_{11} c_{1i+1},$$

$$i = 2(1)n-2,$$

$$x_{n-1}(u) = f_{00} p_{n-1} + f_{10} p_n + \mathcal{S}_{n-1} f_{01} c_{1n-1} + \lambda_n f_{11} p'_n.$$

Der Beweis wird durch Umstellen des Gleichungssystems aus Lemma 1.2 erbracht.

Hinreichend für die eindeutige Lösbarkeit von (1.16) ist die diagonale Dominanz der tri-diagonalen Koeffizientenmatrix A , die durch geeignete Wahl der Parameter $\mathcal{S}_i, \lambda_{i+1}, \lambda_i$ stets erreicht werden kann.

Stellt man die Forderungen (1.12) und verwendet kubische oder trigonometrische Binde-funktionen, wo wird die Matrix A in Satz 1.2 positiv definit; dann ist (1.16) eindeutig lösbar. Für kubische Binde-funktionen stimmt die daraus resultierende IP-Kurve $x \in X_n^{2,1}$

mit jener IP-Kurve überein, die man durch Anwendung der Spline-Interpolation 3. Grades in jeder Koordinate einer Vektorfunktion erhält, die über den Stützstellen G_i entsprechende Vorgaben interpoliert (vgl. /2/, /8/).

2. Interpolation von Flächen

2.1. Definition und Eigenschaften der Interpolationsflächen

Wir definieren die Menge der IP-Flächen als Produkt zweier Mengen von IP-Kurven nach Definition 1.2.

Definition 2.1 Es sei $X_{n,m}^{a,b}$ die Menge aller Folgen

$$x(u,v) = (x_{ij}(u,v))_{i=1, j=1}^{n-1, m-1}, \quad (u, v) \in I \times I$$

von $(n-1)(m-1)$ reellen Vektorfunktionen $x_{ij} : I \times I \rightarrow \mathbb{R}^3$

mit

$$x_{ij}(u,v) = \sum_{r=0}^b \sum_{s=0}^b (S_{ij}^r \bar{S}_{ij}^s f_{or} \bar{f}_{os} c_{rsij} + \lambda_{i+1,j}^r \bar{S}_{i+1,j}^s f_{1r} \bar{f}_{os} c_{rs,i+1,j} + S_{i,j+1}^r \bar{\lambda}_{i,j+1}^s \cdot f_{or} \bar{f}_{1s} c_{rsi,j+1} + \lambda_{i+1,j+1}^r \bar{\lambda}_{i+1,j+1}^s f_{1r} \bar{f}_{1s} c_{rs,i+1,j+1}) \quad (2.1)$$

für beliebige Koeffizienten $S_{ij}, \lambda_{i+1,j} > 0$ ($i = 1(1)n-1, j = 1(1)m$), $\bar{S}_{ij}, \bar{\lambda}_{i+1,j} > 0$ ($i = 1(1)n, j = 1(1)m-1$) und Koeffizientenvektoren $c_{rsij} \in \mathbb{R}^3$ ($r = 1(1)n, j = 1(1)m$) mit $c_{10ij} \neq 0, c_{01ij} \neq 0$ und Bindefunktion $f_{kl} = f_{kl}(u) \in F^{a,b}$
 $\bar{f}_{kl} = f_{kl}(v) \in F^{a,b}$ ($k = 0, 1; \ell = 0(1)b$).

Wir nennen $x(u,v)$ IP-Fläche und $x_{ij}(u,v)$ IP-Flächenstück.

Es sei $\bar{x}_i(v) := (x_{ij}(0,v))_{j=1}^{m-1}$ für $i = 1(1)n-1$,

$$\bar{x}_n(v) := (x_{nj}(1,v))_{j=1}^{m-1},$$

$$x_j(u) := (x_{ij}(u,0))_{i=1}^{n-1} \quad \text{für } j = 1(1)m-1,$$

$$x_m(u) := (x_{im}(u,1))_{i=1}^{n-1}.$$

Wir bezeichnen $\bar{x}_i(v)$ ($i = 1(1)n$) bzw. $x_j(u)$ ($j = 1(1)m$) als v - bzw. u -Netzkurve der IP-Fläche $x(u,v)$.

Lemma 2.1 Für $x(u,v) \in X_{n,m}^{a,b}$ gilt

$$\begin{aligned} \bar{x}_i(v) &\in X_m^{a,b} \text{ für } i = 1(1)n, \\ x_j(u) &\in X_n^{a,b} \text{ für } j = 1(1)m. \end{aligned} \quad (2.2)$$

Zum Beweis zeigt man unter Ausnutzung der Eigenschaften (1.1) der Bindefunktionen, daß die Netzkurven den Voraussetzungen von Definition 1.2 genügen.

Weiter bezeichnen wir

$$x(u^*, v) = (x_{ij}(u^*, v))_{j=1}^{m-1}, \quad u=u^* = \text{const.}, \text{ für festes } i=1(1)n-1$$

bzw.

$$x(u, v^*) = (x_{ij}(u, v^*))_{i=1}^{n-1}, \quad v=v^* = \text{const.}, \text{ für festes } j=1(1)m-1$$

als v - bzw. u -Parameterlinie von $x(u,v) \in X_{n,m}^{a,b}$.

Man bestätigt leicht, daß $x(u^*, v) \in X_m^{0,b}$ und

$$x(u, v^*) \in X_n^{0,b} \quad (b \in \mathbb{N}) \text{ gilt. Die Parameterlinien sind also}$$

stetige Kurven.

Weiter ist offensichtlich, daß die Netzkurven spezielle Parameterlinien sind. Uns interessieren die geometrischen Eigenschaften der v - bzw. u -Parameterlinie in ihren Schnittpunkten mit den u - bzw. v -Netzkurve. Damit kann der Anschluß benachbarter IP-Flächenstücke beschrieben werden. Ein hinreichendes Kriterium dafür, daß die Parameterlinien Kurven der Ordnung a sind, gibt

Lemma 2.2 Es sei $x(u,v) \in X_{n,m}^{a,b}$ mit $1 \leq a \leq b \leq 3$.

$$\text{Wenn } \mathcal{P}_{ij} : \lambda_{ij} = \mathcal{P}_{i+1,j} : \lambda_{i+1,j}$$

$$\bar{\mathcal{P}}_{ij} : \bar{\lambda}_{ij} = \bar{\mathcal{P}}_{i+1,j} : \bar{\lambda}_{i+1,j} \quad (2.3)$$

und

$$\sum_{r=0}^b (\mathcal{P}_{ij}^{(r)} \bar{\lambda}_{ij}^{(r)} \text{ or } c_{r1ij} + \bar{\mathcal{P}}_{i+1,j}^{(r)} \bar{\lambda}_{i+1,j}^{(r)} + c_{r1i+1,j}) = 0$$

$$\sum_{s=0}^b (\bar{S}_{ij}^s \lambda_{ij} \bar{f}_{os} c_{1sij} + \lambda_{i+1} \bar{S}_{i+1}^s \bar{f}_{1s} c_{1sij+1}) \neq 0 \quad (2.4)$$

für

$i = 1(1)n - 1, j = 1(1)m - 1$ gilt, dann ist

$$x(u^*, v) \in X_m^{a,b} \text{ und } x(u, v^*) \in X_n^{a,b}.$$

2.2. Definition zur Interpolation empirischer Flächen

Es seien $n \cdot m$ Stützpunkte $p_{ij} \in \mathbb{R}^3$ ($i = 1(1)n, j = 1(1)m$) gegeben, die so indiziert sind, daß für einen festen Index i die Punkte $p_{i1}, p_{i2}, \dots, p_{im}$ auf einer empirischen Kurve \bar{C}_i im Sinne von Kapitel 1 liegen. Analog sollen für einen festen Index j die Punkte $p_{1j}, p_{2j}, \dots, p_{nj}$ auf einer empirischen Kurve C_j liegen.

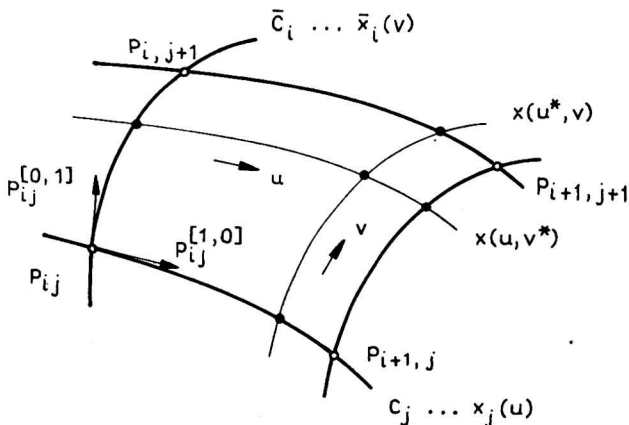


Abb. 1

Bei einigen Aufgabenstellungen werden Tangenteneinheitsvektoren $\vec{p}_{ij}^{[1,0]}$ ($i = 1(1)n$), der in den Stützpunkten p_{ij} an die empirische Kurve C_j ($j = 1(1)m$) gelegten Tangenten und entsprechend Tangenteneinheitsvektoren $\vec{p}_{ij}^{[0,1]}$, $j = 1(1)m$, der in den Stützpunkten p_{ij} an die empirische Kurve \bar{C}_i ($i = 1(1)n$) gelegten Tangenten vorausgesetzt

(s. Abb. 1). Weiter können Krümmungsvektoren

$p_{ij}^{[2,0]}$, $i = 1(1)n$, von C_i in p_{ij} ($j = 1(1)m$) bzw.

$p_{ij}^{[0,2]}$, $j = 1(1)m$, von \bar{C}_i in p_{ij} ($i = 1(1)n$) und entsprechend auch Windungsvektoren gegeben sein.

Um die Länge bzw. eine Abschätzung für die Länge der Kurven des empirischen Kurvennetzes berücksichtigen zu können, seien für jeden (festen) Index i den Stützpunkten p_{ij} Stützparameterwerte $\bar{\sigma}_{ij} \in \mathbb{R}$ so zugeordnet, daß $\bar{\sigma}_{i1} < \bar{\sigma}_{i2} < \dots < \bar{\sigma}_{im}$ gilt und

$$\bar{\sigma}_{ij} = \bar{\sigma}_{ij+1} - \bar{\sigma}_{ij}, \quad i = 1(1)m-1,$$

die Länge des Kurvenstückes $\widehat{p_{ij} p_{ij+1}}$ von \bar{C}_i oder ein Näherungswert dafür ist. Entsprechend werden für jeden (festen) Index j den Stützpunkten p_{ij} Stützparameterwerte $\sigma_{ij} \in \mathbb{R}$ mit $\sigma_{1j} < \sigma_{2j} < \dots < \sigma_{nj}$ zugeordnet, so daß

$$\sigma_{ij} = \sigma_{i+1j} - \sigma_{ij}, \quad i = 1(1)n-1$$

die Länge des Kurvenstückes $\widehat{p_{ij} p_{i+1j}}$ von C_i oder ein Näherungswert dafür ist.

Zur Abkürzung der gestellten IP-Bedingungen führen wir folgende Bezeichnung ein:

$$X_{n,m}^{a,b}(p_{ij}^{[c]}) := \left\{ x(u,v) \in X_{n,m}^{a,b} \mid x_i(v) \in X_m^{a,b}(p_{ij}^{[c]}) \right.$$

$$\left. \text{für jedes } i = 1(1)n, x_i(u) \in X_n^{a,b}(p_{ij}^{[c]}) \text{ für jedes } i = 1(1)m \right\}$$

Für $c = 0$ ist $x(u,v) \in X_{n,m}^{a,b}$ eine IP-Fläche, deren v - bzw. u -Netzkurven Kurven der Ordnung a (bei Verwendung von Bindefunktionen vom Rang b) sind und die die empirischen Flächenkurven \bar{C}_i bzw. C_i interpolieren. Für $c = 1, 2$ bzw. 3 stimmen zusätzlich die invarianten partiellen Ableitungen $x_{ij}^{[0,c]}$ und $x_{ij}^{[c,0]}$ der IP-Fläche in den Stützpunkten mit vorgegebenen Tangenteneinheitsvektoren $p_{ij}^{[0,1]}$ und $p_{ij}^{[1,0]}$, Krümmungsvektoren $p_{ij}^{[0,2]}$ und $p_{ij}^{[2,0]}$ bzw. Windungsvektoren $p_{ij}^{[0,3]}$ und $p_{ij}^{[3,0]}$ der empirischen Flächenkurven \bar{C}_i und C_i überein. Weitere Bedingungen werden an der jeweils benötigten Stelle aufgeführt.

2.3. Verallgemeinerungen des zweidimensionalen Hermiteschen und des Spline-Interpolationsverfahrens

Satz 2.1 (Verallgemeinerte zweidimensionale Hermitesche Interpolation).

a) $x \in X_{n,m}^{a,a} (p_{ij}^{[0]})$ für $a \geq 0$ genau dann, wenn

$$c_{00ij} = p_{ij}, \quad i = 1(1)n, \quad j = 1(1)m.$$

b) $x \in X_{n,m}^{a,a} (p_{ij}^{[1]})$ für $a \geq 1$ genau dann, wenn (a) und

$$c_{10ij} = \alpha_{ij} p_{ij}^{[1,0]}, \quad c_{01ij} = \bar{\alpha}_{ij} p_{ij}^{[0,1]}, \quad \alpha_{ij}, \bar{\alpha}_{ij} \in \mathbb{R}^+, \\ i = 1(1)n, \quad j = 1(1)m.$$

c) $x \in X_{n,m}^{a,a} (p_{ij}^{[2]})$ für $a \geq 2$ genau dann, wenn (b) und

$$c_{20ij} = \alpha_{ij}^2 p_{ij}^{[2,0]} + \beta_{ij} p_{ij}^{[1,0]}, \quad c_{02ij} = \bar{\alpha}_{ij}^2 p_{ij}^{[0,2]} + \bar{\beta}_{ij} p_{ij}^{[0,1]} \\ \beta_{ij}, \bar{\beta}_{ij} \in \mathbb{R}, \quad i = 1(1)n, \quad j = 1(1)m$$

d) $x \in X_{n,m}^{3,3} (p_{ij}^{[3]})$ genau dann, wenn (c) und

$$c_{30ij} = \alpha_{ij}^3 p_{ij}^{[3,0]} + 3\alpha_{ij} \beta_{ij} p_{ij}^{[2,0]} + \alpha_{ij} p_{ij}^{[1,0]} \\ c_{03ij} = \bar{\alpha}_{ij}^3 p_{ij}^{[0,3]} + 3\bar{\alpha}_{ij} \bar{\beta}_{ij} p_{ij}^{[0,2]} + \bar{\alpha}_{ij} p_{ij}^{[0,1]} \\ \alpha_{ij}, \bar{\alpha}_{ij} \in \mathbb{R}, \quad i = 1(1)n, \quad j = 1(1)m.$$

Der Beweis wird durch Betrachtung der u- und v-Netzkurven von $x \in X_{n,m}^{a,a} (p_{ij}^{[c]})$ mit Hilfe von Satz 1.1 geführt. Die mit Satz 2.1 konstruierbaren Hermiteschen IP-Flächen interpolieren die jeweils vorgegebenen geometrischen Invarianten der empirischen Flächenkurven \bar{C}_i und C_j für eine beliebige Wahl der Parameter

$$S_{ij}, \lambda_{i+1, j} \quad (i = 1(1)n-1, \quad j = 1(1)m), \quad \bar{S}_{ij}, \bar{\lambda}_{i, j+1}$$

($i = 1(1)n, \quad j = 1(1)m-1$) und der "freien" Koeffizientenvektoren c_{rsij} mit $r \neq 0$ oder $s \neq 0$, sowie beliebigen Parametern $\alpha_{ij}, \bar{\alpha}_{ij}, \beta_{ij}, \bar{\beta}_{ij}, \alpha_{ij}, \bar{\alpha}_{ij}$ ($i = 1(1)n, \quad j = 1(1)m$).

Sollen auch die Parameterlinien einer Hermiteschen IP-Fläche $x \in X_{n,m}^{a,a}$ Kurven der Ordnung a sein, so sind die genannten Größen so zu wählen, daß sie (2.3) und (2.4) erfüllen (vgl. Lemma 2.2).

Folgerung 2.1 Es seien $\bar{\sigma}_{ij}$ bzw. $\bar{\sigma}_{ij}$ die Stützparameter der empirischen Flächenkurven \bar{C}_i bzw. \bar{C}_i . Für jede Wahl der Koeffizientenvektoren $c_{rs,ij}$ mit $r \neq 0$ oder $s \neq 0$ existiert genau eine IP-Fläche $x(u,v) \in X_{n,m}^{c,c}(p_{ij} \begin{smallmatrix} \nearrow c \\ \searrow \end{smallmatrix})$, $c = 1, 2, 3$, deren Netzkurven den Bedingungen (1.12) genügen. Diese ergibt sich aus (2.1), indem man dort

$$\left. \begin{aligned} c_{roi} &= p_{ij} \begin{smallmatrix} \nearrow r, o \\ \searrow \end{smallmatrix} \\ c_{osi} &= p_{ij} \begin{smallmatrix} \nearrow o, s \\ \searrow \end{smallmatrix} \end{aligned} \right\} \quad r, s = 0(1)c$$

$$\sigma_{ij} = \lambda_{i+1j} = \sigma_{ij}, \quad i = 1(1)n-1, \quad j = 1(1)m$$

$$\bar{\sigma}_{ij} = \bar{\lambda}_{i+1j} = \bar{\sigma}_{ij}, \quad i = 1(1)n, \quad j = 1(1)m-1 \quad (2.9)$$

setzt.

Der Beweis ergibt sich durch Anwendung der Folgerung 1.1 auf die u - und v -Netzkurven von $x(u,v) \in X_{n,m}^{c,c}$ aus Satz 2.1.

Entsprechend läßt sich Satz 1.2 für ein verallgemeinertes zweidimensionales Spline-IP-Verfahren ausnutzen, indem die empirischen Flächenkurven \bar{C}_i bzw. C_i durch Netzkurven $\bar{x}_i(v) \in X_m^{2,1}(p_{ij} \begin{smallmatrix} \nearrow 0 \\ \searrow \end{smallmatrix})$ bzw. $x_i(u) \in X_n^{2,1}(p_{ij} \begin{smallmatrix} \nearrow 0 \\ \searrow \end{smallmatrix})$ interpoliert werden. Man hat hierzu $n \cdot m$ lineare Gleichungssysteme der Gestalt (1.16) für die Koeffizientenvektoren c_{oiij} ($j=2(1)m-1, i=1(1)n$) und c_{ioij} ($i=2(1)n-1, j=1(1)m$) zu lösen und die Anfangs- und Endtangenteurichtungen von \bar{C}_i bzw. C_i vorzugeben. Werden die so bestimmten Größen in (2.1) (unter Beachtung von $b=1$ und $c_{ooij} = p_{ij}$) eingesetzt, so erhält man für jede Wahl der freien Parameter und Koeffizientenvektoren genau eine IP-Fläche $x(u,v) \in X_{n,m}^{2,1}(p_{ij} \begin{smallmatrix} \nearrow 0 \\ \searrow \end{smallmatrix})$.

2.4. Anwendungsbeispiel

Von einer empirischen Fläche seien die Stützpunkte p_{ij} und zugehörige Stützparameterwerte σ_{ij} und $\bar{\sigma}_{ij}$ bekannt, die man z. B. aus den entsprechenden Abständen der Stützpunkte ermittelt. Wir nehmen an, daß eine Interpolation der Fläche mit kubischen Bindefunktionen, die den Rang 1 besitzen, ausgeführt werden soll. Wir betrachten zwei Fälle:

1. Sind die Tangentenrichtungen der empirischen Flächenkurven in allen Stützpunkten vorgeschrieben, so ermittelt man nach Folgerung 2.1. zunächst die Hermiteische IP-Fläche $x(u,v) \in X_{n,m}^{1,1}(p_{ij}, \frac{f}{\lambda})$, die nach Wahl von $c_{11,ij} = 0$, $i = 1(1)n$, $j = 1(1)m$, eindeutig bestimmt ist. Befriedigt das Interpolationsergebnis nicht, so kann durch Variation der Parameter $S_{ij}, \lambda_{i+1j}, S_{ij+1}, \lambda_{i+1,j+1}, \bar{S}_{ij}, \bar{\lambda}_{ij+1}, \bar{S}_{i+1j}, \bar{\lambda}_{i+1,j+1}$ für die zunächst (2.9) angenommen wurde, die geometrische Form des Flächenstücks $x_{ij}(u,v)$ verändert werden. Eine Verkleinerung dieser Parameter bewirkt eine Straffung der Flächenhaut im Bereich der Netzmasche $p_{ij}, p_{i+1j}, p_{i+1,j+1}, p_{ij+1}$ - eine Vergrößerung ein stärkeres Ausschwingen, wobei die vorgeschriebenen Tangentenrichtungen in den Stützpunkten eingehalten werden. Erfüllt man bei dieser Variation die Bedingungen (2.3) und (2.4) des Lemmas 2.2 (evtl. durch neue Wahl der $c_{11,ij}$), so sind auch die Parameterlinien der IP-Fläche Kurven der Ordnung 1.

2. Sind nur die Tangentenrichtungen in den Anfangs- und Endpunkten der empirischen Flächenkurven bekannt, so ermittelt man zunächst die Tangentenrichtungen in den übrigen Stützpunkten nach dem Verfahren der Spline-Interpolation (Lösung von $n \cdot m$ Gleichungssystemen (1.16), wobei die Parameter zuerst nach (2.9) gewählt werden). Das Ergebnis ist eine Spline-IP-Fläche $x(u,v) \in X_{n,m}^{2,1}(p_{ij}, \frac{f}{\lambda})$. Genügen die ermittelten Tangentenrichtungen den konstruktiven Forderungen, so kann es dennoch notwendig sein, daß die IP-Fläche lokal abgeändert werden muß. Dies kann wie im 1. Fall durchgeführt werden, und man erhält dann eine Hermiteische IP-Fläche. Anderenfalls muß man durch Wahl anderer Parameter $S_{ij}, \lambda_{i+1j}, \dots, \lambda_{i+1,j+1}$, deren geometrische Bedeutung oben beschrieben wurde, neue Tangentenrichtungen ermitteln.

Literatur:

- /1/ Bär, G.: Parametrische Interpolation empirischer Kurven und Flächen.
Rechentechnik/Datenverarbeitung 1/76.
- /2/ Bär, G.: Parametrische Interpolation empirischer Raumkurven.
ZAMM 56 (in Vorbereitung).
- /3/ Cohen, S.: Interpolation glatter Kurven und Flächen mit Steuerungsmöglichkeiten
für die rechnergestützte Konstruktion.
Dipl.-Arb. Sektion Mathematik, TU Dresden 1975.
- /4/ Coons, S. A.: Surfaces for computer aided design of space forms.
MIT, MAC-TR-41 (1967).
- /5/ Manning, J. R.: Continuity conditions for spline curves.
Comp. J., 17, 2, 181-186 (1974).
- /6/ Schmieder, L.: Rekursive Erzeugung, Differentiation und Integration Hermite'scher
Interpolationspolynome nebst einem Anwendungsbeispiel. ZAMM 53, 433-440 (1973).
- /7/ Schultz, M. H.: Spline Analysis. Prentice-Hall 1973
- /8/ Späth, H.: Spline-Algorithmen zur Konstruktion glatter Kurven und Flächen.
R. Oldenbourg Verlag 1973.
- /9/ Strubecker, K.: Differentialgeometrie, Bd. I. W. d. Gruyter 1955.

Anschrift des Verfassers:

Dr. Gert Bär
Technische Hochschule Karl-Marx-Stadt
Sektion FPM
DDR - 90 Karl-Marx-Stadt
Postfach 964

Realisierungsvarianten der Dialogsteuerung

Beim Erstellen von Dialogprogrammen bzw. digitalgraphischen Dialogsystemen stehen vor allem zwei Fragen im Mittelpunkt des Interesses:

- Wie kann der Nutzer dem Automaten seine Befehle übermitteln?
- Wie wird das Wechselspiel "Aktion des Nutzers- Reaktion des Automaten" durch Programme im Automaten gesteuert?

Mit anderen Worten geht es um die konkrete Festlegung der Kommandosprache und die Realisierung der Dialogsteuerung. Als Kommandosprache oder auch Dialogsprache bzw. interaktive Sprache wollen wir die Sprache bezeichnen, die es dem Nutzer erlaubt, Befehle in den Automaten einzugeben, die dann sofort ausgeführt werden (s. /2/).

Im wesentlichen haben die Befehle der Kommandosprache zwei Aufgaben zu erfüllen:

- Sie ermöglichen zu steuern, welche Prozesse (Programme) aktiviert werden sollen.
- Sie ermöglichen die Eingabe von Daten (Parametern) für diese Prozesse.

Es ist notwendig, die Kommandosprachen von den Sprachen zu unterscheiden, die für die programmtechnische Realisierung eines Dialogsystems benutzt werden (z. B. FORTRAN, PL-1, Assemblersprachen).

Kommandosprachen weisen zahlreiche Besonderheiten auf:

- Bedeutend einfacherer Aufbau, d. h. im allgemeinen keine komplexen Befehle für iterative oder bedingte Operationen.
- Ihre Befehle bestehen aus einer Folge von Elementen, von denen jedes durch eine spezielle Nutzeraktivität eingegeben wird.

- Die Befehle beginnen vielfach mit einem speziellen Element und berücksichtigen eine Mnemonik.
- Häufig existiert die Möglichkeit, Abkürzungen (Kurzformen) der Befehle zu benutzen.
- Die Implementation der Kommandosprachen erfolgt interpretativ.

Die Befehle einer Kommandosprache enthalten im wesentlichen entsprechend ihrer Aufgabenstellung drei Elementtypen:

- das Verb (zur Definition des zu aktivierenden Prozesses),
- die Datenelemente (zur Eingabe der Parameter) und
- Begrenzer für das Verb und /oder die Datenelemente.

Die Realisierung der Kommandosprachen erfolgt häufig in Form der sogenannten Menütechnik. Auch im DIGRA 73-System (§. /1/) hat sich diese Realisierungsart bewährt, weil sie am besten den Anforderungen an Kommandosprachen (Einfachheit, Erweiterbarkeit, vorbeugende Fehlervermeidung, Nutzerfreundlichkeit) entspricht.

Was verstehen wir unter der Menütechnik?

Die Eingabe der Befehle bzw. Befehlselemente erfolgt in diesem Fall durch die Auswahl in einem angebotenen Menü möglicher Befehle bzw. Befehlselemente. Die rechnerische Realisierung eines Menüs ist z. B. als Menge von Lichttasten mit entsprechender Beschriftung möglich. Das bedeutet, daß jedem Befehl bzw. Befehlselement eine auf dem Bildschirm dargestellte Textinformation (Menüzeile) zugeordnet wird, die auf die Semantik hinweist. Die Eingabe könnte dann durch das Identifizieren eines Zeichens einer Menüzeile mit dem Lichtstift erfolgen.

Eine zusammenfassende Darstellung der inneren Struktur aller Befehle und der Kontextbedingungen einer Kommandosprache als Graph bezeichnen wir als Menügraph. Realisierungen der Kommandosprache durch die Menütechnik sind z. B. in den Systemen PRADIS, (§. /3/), SOPSY (§. /5/) und DIGRA 73 (§. /1/) vorgenommen worden.

Einen gewissen Eindruck vom Aufbau eines Menügraphen vermittelt Abb. 1, die einen Untergraphen des Standardmenübaumes des DIGRA 73-Systems darstellt. Alle Menüs der Hauptmenüfunktion DARSTELL, die zur Beeinflussung der Projektionsparameter durch den Nutzer am Bildschirm dient, werden so dargestellt, wie sie dem Nutzer bei der Dialogarbeit begegnen.

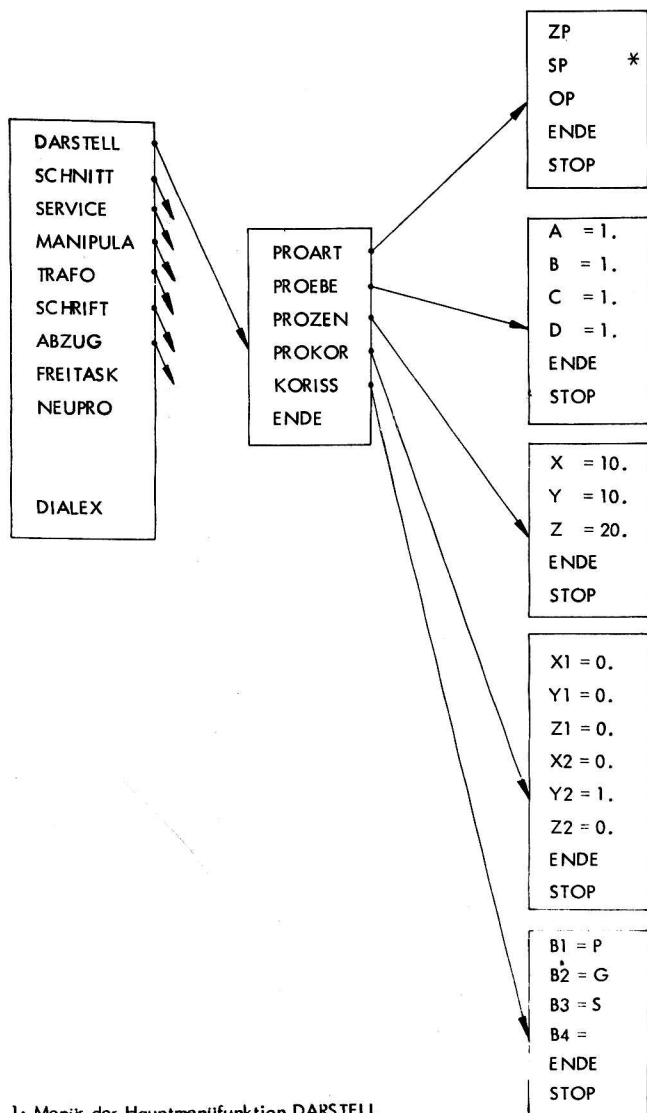


Abb. 1: Menüs der Hauptmenüfunktion DARSTELL

Nun zur zweiten eingangs aufgeworfenen Frage, der Realisierung der Dialogsteuerung. Die Programme, die die wechselseitige Kommunikation Nutzer-Automat steuern, haben drei Grundaufgaben zu erfüllen:

- Warten auf Eingabeinformationen des Nutzers (sogenannte Nutzerkontaktinformationen)
- Steuerung der Reaktion des Automaten, d. h. der Verarbeitung der Nutzerkontaktinformationen
- Vorbereitung weiterer Aktionen des Nutzers, d. h. Übergang in einen erneuten Wartezustand auf Nutzerkontaktinformationen.

Dabei besteht die Reaktion im allgemeinen aus verschiedenartigen, umfangreichen Informationsverarbeitungsprozessen, wie z. B. der Transformation der Nutzerkontaktinformationen in eine für bestimmte Programme akzeptable Form, der Aktivierung ausgewählter Prozesse, dem Auslösen notwendiger Fehlermaßnahmen, der Erzeugung von Ausgabeinformationen an den Nutzer (sogenannten Automatenkontaktinformationen) oder der Datenvermittlung an bestimmte Programme.

Unter dem Aspekt der Implementation von Kommandosprachen heißt das, daß die Dialogsteuerung die Programme umfaßt, die die Eingabe der Befehle bzw. Befehlselemente ermöglichen, eingegebene Befehle bzw. Befehlselemente interpretieren und die Abarbeitung veranlassen.

Bezüglich der Realisierung der Dialogsteuerung sind mehrere verschiedene Methoden bekannt geworden.

Worin liegen die Ursachen dafür?

Es sei auf die folgenden Ursachen hingewiesen:

- Existenz verschiedenartiger Konzeptionen der Kommandosprachen,
- unterschiedliche hard- und grundsoftwareseitige Voraussetzungen und nicht zuletzt
- Unterschiede in der Problemstellung (Einzelproblem oder Problemklasse, Dialogprogramm oder Dialogsystem).

Aus der Literatur sind vor allem drei Realisierungsvarianten bekannt (§./4/); im DIGRA 73-System wurde eine vierte Variante realisiert, die mehrere Erweiterungen beinhaltet (§./2/). Im folgenden sollen die verschiedenen Varianten grob charakterisiert und einige Vor- bzw. Nachteile vorgestellt werden.

- Benutzung herkömmlicher Programmiersprachen

Für die Implementation von Kommandosprachen mit nur wenigen einfachen Befehlen zur Lösung eines Einzelproblems wird häufig diese einfachste, problemabhängige Variante gewählt. Dabei besteht die Aufgabe im allgemeinen darin, vorliegende Programme dialogfähig zu machen, indem an vorbestimmter Stelle ermöglicht wird, daß gewisse aktuelle Parameter eingegeben werden können. Es muß also die Eingabe von Daten, z. B. durch Unterprogrammaufrufe einer Dialoggrundsoftware, realisiert werden, während der zu aktivierende Prozeß fest fixiert ist. Problematisch wird die Anwendung dieser Methode bei starkem Anwachsen der Befehle, bei einer nicht elementaren Struktur der Befehle, bei einer komplizierten Programmstruktur, beim Versuch, mehrere periphere Geräte für die Parametereingaben zuzulassen und bei notwendigen Erweiterungen. Sehr schnell wird die große Speicherplatzbelastung zu einem Problem, wenn die Parametereingabe und die Parameterverarbeitung innerhalb eines Programms erfolgen. Andererseits ergeben sich daraus gewisse Vorteile beim Aufbau eines Baukastensystems, da ein einfachstes Zusammenspiel nahezu abgeschlossener Programmbausteine zu realisieren ist.

- Tabellenbenutzende Sprachprozessoren

Eine zweite Realisierungsvariante der Dialogsteuerung ist die Erstellung eines Sprachprozessors der eine Tabelle benutzt, die neben einer Beschreibung aller Befehle auch den Namen des jeweils für die Gestaltung der Reaktion zu aktivierenden Prozesses enthält.

In diesem Falle sieht der Arbeitsablauf also wie folgt aus: Der Sprachprozessor ermöglicht die Eingabe eines Befehls. Nach Abschluß der Eingabe erfolgt ein Durchmustern der Tabelle, bei dem zunächst die Zulässigkeit des Befehls festgestellt, dann seine Abarbeitung veranlaßt wird.

Im Anschluß an die Abarbeitung des aktivierten Programms geht der Sprachprozessor erneut in einen Wartezustand auf die Eingabe eines Befehls über. Das Neue dieser Variante ist die Problemunabhängigkeit, d. h., ein Austauschen der Tabelle der Befehle und der Menge aktivierbarer Programme entspricht der Implementation einer neuen Kommandosprache. Schwierigkeiten treten bei der Verwendung komplexer Befehle, der Zulassung verschiedener Eingabegeräte und der Bereitstellung von Parametern für die zu aktivierenden Programme auf.

- Übersetzung von Zustandsdiagrammen

Eine Berücksichtigung der Teilschritte bei der Eingabe eines Befehls, d. h. des mehrfachen Wechsels Aktion-Reaktion, und der Charakterisierung der Reaktion als Übergang in einen Wartezustand auf die nächste Aktion führt zum Begriff des Zustandsdiagramms oder Zustandsgraphen. Wählt man als Knotenmenge die Menge aller Wartezustände, die bei der Realisierung einer Kommandosprache auftreten, und verbindet entsprechend den Zustandsübergängen die Knoten durch Bögen, dann erhält man das Zustandsdiagramm einer konkreten Kommandosprache. Den Bögen dieser Graphen sind dabei praktisch zwei unterschiedliche Bewertungen zugeordnet, zum einen die Aktion, die den jeweiligen Zustandsübergang veranlaßt, und zum anderen die Reaktion, die beim Zustandsübergang abläuft. Auf der Tatsache basierend, daß eine Kommandosprache vollständig durch ein Zustandsdiagramm beschreibbar ist, wurde die Methode der Übersetzung von Zustandsdiagrammen realisiert.

Nach einer Eingabe des Zustandsdiagramms erfolgt dessen Übersetzung, indem jedem Zustand eine Gruppe von Anweisungen zugeordnet wird.

Als wesentliche Vorteile ergeben sich die Einfachheit der Definition von Kommandosprachen und die Problemunabhängigkeit. Probleme treten vor allem dann auf, wenn durch eine Erhöhung der Anzahl der Befehle und Befehlselemente und dem damit verbundenen schnellen Anwachsen der Anzahl der Zustände der Umfang und die Kompliziertheit der übersetzten Programme sehr stark zunimmt.

Als mögliche Auswege werden eine Standardisierung der Eingabemöglichkeiten und die Einführung einer Hierarchie im Zustandsdiagramm, einschließlich der Zusammenfassung von Zuständen zu sogenannten "Superzuständen", vorgeschlagen (§. 4/).

- Interpretation von Zustandsdiagrammen

Die im DIGRA 73-System realisierte Dialogsteuerung stellt, basierend auf der Verwendung eines speziellen Zustandsdiagramms, eine Weiterführung der beschriebenen dritten Variante dar. Sie berücksichtigt dabei die Besonderheiten, die sich bei der Realisierung einer Kommandosprache auf der Basis der Menütechnik ergeben. Das Zustandsdiagramm wird in Form der sogenannten Menüdaten eingegeben, die inhaltlich sowohl die Informationen des Menügraphen als auch des Zustandsgraphen enthalten.

Eine wichtige Eigenschaft der realisierten Dialogsteuerung ist die konsequente Trennung der Eingabe der Befehlselemente von der Abarbeitung der Befehle. Dabei wird ausgenutzt, daß es eine begrenzte, im allgemeinen problemunabhängige Anzahl von Datentypen gibt, wie z. B.

- . reellen und ganzen Zahlen,
- . alphanumerischen Zeichen,
- . graphischen Parametern,
- . konjunktiven oder disjunktiven Auswahlparametern (also logischen Größen) und
- . dem Koordinatenpaar (X,Y) bei der Positionierung am Bildschirm.

Es wurden Teilprogramme für die Ermöglichung der Menüauswahl (Eingabe von Verb, Hilfsverben und Begrenzern) und die Eingabe der Datentypen erstellt, d. h. eine Typisierung der Befehlselemente erfolgte.

Dadurch war es möglich, den Dialogsteuerteil problemunabhängig zu gestalten, und somit den Umfang der problemunabhängigen Software eines digitalgraphischen Dialogsystems zu vergrößern. Dieser problemunabhängige, fixe Dialogsteuerteil arbeitet das Zustandsdiagramm interpretativ ab, ruft Teilprogramme zur Ermöglichung der Eingabe der Befehlselemente und Programme zur Befehlsabarbeitung auf.

In Abb. 2 wird ein Schema der im DIGRA 73-System realisierten Dialogsteuerungsvariante dargestellt, aus dem die drei Hauptaufgaben des Dialogsteuerteils ersichtlich sind:

- . Interpretation der Menüdaten und Ermöglichung von Aktionen
- . Interpretation der den Befehlselementen entsprechenden Nutzerkontaktinformationen in Verbindung mit den Menüdaten und
- . Aktivierung von Programmen, die eingegebene und übermittelte Nutzerkontaktinformationen verarbeiten und die Probleminformationen und den Problemlösungsalgorithmus zielgerichtet verändern.

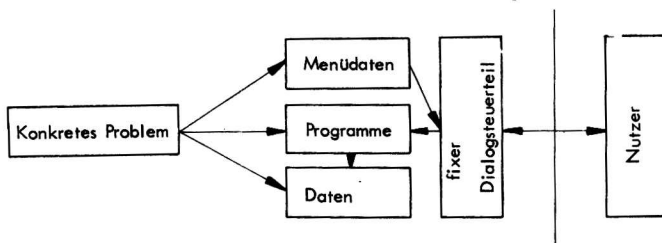


Abb. 2: Schema der 4. Dialogsteuerungsvariante

Im folgenden soll kurz auf einige Charakteristika dieser Dialogsteuerungsvariante hingewiesen werden. Den Ausgangspunkt für die Erstellung der Menüdaten bildet der Menügraph, der rechenstechnisch in Form des zugehörigen Bindgraphen abgespeichert wird. Jedem Knoten wird ein Tripel von Informationen (t, s, l) zugeordnet, wobei

- t den sogenannten Internreaktionstyp beinhaltet (hängt mit der Typisierung der Befehlselemente zusammen),
- s der Verweis auf den Nachfolger und
- l die Textinformation der einem Knoten zugeordneten Lichttaste (Bezeichnung) ist.

Im Dialogsteuerteil wird zu einem diskreten Zeitpunkt immer genau ein Knoten des Menügraphen abgearbeitet, oder mit anderen Worten, die Reaktion nach der Auswahl eines bestimmten Knotens gesteuert (zu Details s. /2/). Sollte eine Kommandosprache mit diesem Dialogsteuerteil implementiert werden, dann bedeutet das, neue Menüdaten und eine Menge von Programmen für die Befehlsabarbeitung zu erstellen und den fixen Dialogsteuerteil für die Befehlseingabe und Interpretation zu nutzen. Aus den bisherigen praktischen Erfahrungen bei der Arbeit mit dem problemunabhängigen Dialogsteuerteil in Programmpaketen für die Erzeugung von Maschinenaufstellungsplänen, von Rohrleitungs- und von Schiffsliniendarstellungen ergibt sich folgendes Resümee:

- . Die Nutzung als sogenannte Arbeit mit dem freien Menübaum erfordert keine Kenntnisse über die Grundsoftware für die Dialoggestaltung.
- . Durch die Verwendung relativ kleiner Verarbeitungsprogramme (wegen der vorgenommenen Trennung Befehlseingabe- Befehlsabarbeitung) ergibt sich eine günstige Speicherplatzbelastung.

- . Am besten haben sich Menügraphen bewährt, die eine Unterteilung in semantisch abgrenzbare Untergraphen beinhalten und dreistufig aufgebaut sind (Eingabe Verb, Eingabe Hilfsverb, Eingabe Datenelemente).
- . Unsere Realisierungsvariante ist auf einfachste Art mit der 1. Realisierungsvariante koppelbar (in Form des sogenannten FREITASK-Konzepts).
- . Die Teilprogramme für die Dateneingabe sind auch separat in eigenen Dialogprogrammen verwendbar.
- . Bisher sind nur Textmenüs, d. h. zum Beispiel keine Menüs mit Symbolen für die Bemaßung, zugelassen.
- . Bei zu einfacher Struktur des Menügraphen ist unsere Dialogsteuerungsvariante nicht effektiv einsetzbar, d. h., bei einem einfachen Einzelproblem könnte ein vorhandenes Programm besser durch die 1. Realisierungsvariante dialogfähig gestaltet werden.

Literatur

- /1/ Autorenkollektiv aus der Sektion Mathematik der Wilhelm-Pieck-Universität Rostock
Programmierhandbuch DIGRA 73, Rostock, 1975.
- /2/ Berndt, E.: Der Dialog zwischen Nutzer und Automat und seine Realisierung im
DIGRA 73-System, Forschungsbericht SM-DIGRA 92 (Dissertation), Rostock, 1976.
- /3/ Encarnacao, J. L.: PRADIS- Ein Programmsystem für räumliche Darstellungen auf
Displays, Elektronische Datenverarbeitung, 7, 315-325 (1970).
- /4/ Newman, W. M., Sproull, R. F.: Principles of interactive computer graphics,
Mc Graw-Hill B., 1973.
- /5/ Rothhardt, G.: Programmausdruck SOPSY.
VEB Carl-Zeiss-Jena (betriebsinterne Dokumentation), 1973.

Anschrift des Verfassers:

Dr. Erhard Berndt
 Wilhelm-Pieck-Universität Rostock
 Sektion Mathematik
 DDR - 25 Rostock, Universitätsplatz 1

Nutzung der graphischen Datenverarbeitung für den Informationsfonds einer Kombinarsleitung

1. Anforderungen an graphische Darstellungen im Rahmen der Nutzung eines Informationsfonds einer Kombinarsleitung

Die Anwendung der elektronischen Datenverarbeitung und ökonomischer Methoden hat mit dem Ziel zu erfolgen, "daß die Leitungs- und Planungsarbeiten rationeller gestaltet werden, der Verwaltungsaufwand vermindert wird, die erforderlichen Entscheidungen sachkundiger getroffen werden können und insbesondere die operative Leitung und Planung in der Volkswirtschaft sowie die Produktionsdurchführung verbessert werden." (1/, S. 173).

Bei der Erfüllung dieser vom IX. Parteitag gestellten Aufgaben auf der Ebene der Leitung von Kombinatn gilt es, die Informationsbereitstellung und -aufbereitung entsprechend einer wissenschaftlichen Leitungstechnologie zu gewährleisten.

Untersuchungen lassen zwei grundsätzliche Richtungen der Informationsversorgung erkennen. Sie enthalten:

1. die Bereitstellung aktueller Informationen über die Entwicklung der wichtigsten ökonomischen und technisch-ökonomischen Kenngrößen des Reproduktionsprozesses in der Gegenüberstellung von Plan-, Ist- und Vorausschauwerten für festgelegte Zeiträume und
2. komplexe, aktuelle und ballastfreie, möglichst bewertete Führungs- und Leitungsinformationen über
 - wissenschaftlich-technische und ökonomische Kenngrößen neuer Erzeugnisse und Technologien sowie Tendenzen der Entwicklung des Industriezweiges,

- Forschungs- und Entwicklungsarbeiten, Neuerertätigkeit,
- internationale wirtschafts- und handelspolitische Tendenzen, Preisentwicklungen u.ä. (§. /8/).

Die Lösung dieser Aufgaben könnte mit Hilfe eines Informationsfonds für die Kombinatleitung möglich werden. Dieser Fonds sollte deshalb die in Abb. 1 dargestellte Struktur besitzen, deren Hauptbestandteile

- die Datenbasis,
- spezielle Methoden und Methodengruppen zur Aufbereitung von Informationen sowie
- Modelle, z. B. zum Ableiten von Handlungsalternativen, sind (§. /9/).

Im Rahmen dieses Vortrages soll nicht auf die Entwicklung eines solchen Informationsfonds eingegangen, sondern nur die daraus resultierenden Anforderungen an graphische Darstellungen gezeigt werden .

Diese Anforderungen lassen sich in folgende zwei Klassen unterteilen:

1. inhaltlich-informatorische Anforderungen,
2. technisch-organisatorische Voraussetzungen.

Die inhaltlich-informatorischen Anforderungen an Grafiken gehen davon aus, daß die Darstellung von Informationen es ermöglichen muß, das Wesentliche schnell, rationell und vollständig zu erfassen.

Grafiken sollen:

- ein der Zielstellung des Nutzers entsprechendes Abbild des Sachverhaltes sein,
- die gewünschten Informationen vollständig und mit großer Anschaulichkeit und Bildhaftigkeit darstellen,
- komplizierte, komplexe informationelle Zusammenhänge von Prozessen übersichtlich vermitteln und
- wesentliche Informationen anschaulich hervorheben (§. /5/).

Eine hohe Informationsdichte in Verbindung mit den Möglichkeiten der übersichtlichen, aussagekräftigen und ansprechenden Darstellungsformen bei graphischen Darstellungen lassen diese zu einem wirkungsvollen Hilfsmittel in der Leitungstätigkeit werden. Sie können jederzeit durch weitere Informationen aus den Informationsfonds ergänzt werden.

Die Grundlage für die Erfüllung der inhaltlich-informatischen Anforderungen ist die Schaffung der technisch-organisatorischen Voraussetzungen. Dazu gehören

- a) programmtechnische Voraussetzungen in Form von Algorithmen für die Aufbereitung der Daten zur Ausgabe als Grafik,
- b) die technische Ausrüstung mit elektronischen Rechanlagen und entsprechenden peripheren Geräten einschließlich der Organisation im ORZ sowie
- c) die Gewährleistung einer zweckmäßigen Kommunikation zwischen Nutzer und Gerätetechnik.

Die programmtechnischen Voraussetzungen sind durch das Bereitstellen von Algorithmen, Programmodulen und Programmpaketen für die grafische Ausgabe zu schaffen.

Zur Gewährleistung der Kommunikation bei der Nutzung eines Informationsfonds kann die alphanumerische Bildschirmtechnik eingesetzt werden, die eine aktive Arbeit des Nutzers mit den Daten und Algorithmen unterstützt. Die graphische Ausgabe mittels dieser Technik ist jedoch auf stilisierte Grafiken begrenzt. Sie entspricht damit nur bedingt den Anforderungen der Anwender für die eingangs genannten Aufgaben. Andererseits wird die aktive graphische Bildschirmtechnik in absehbarer Zeit aus den verschiedensten hier nicht zu diskutierenden Gründen nicht in dem an sich möglichen Umfang verfügbar sein. Es gilt daher einen Weg zu finden, um die graphische Datenverarbeitung und die damit verbundenen Vorteile auch für Leitungsaufgaben sinnvoll einzusetzen.

Das vorzustellende Programmpaket stellt einen möglichen Weg dar und zeigt, daß eine Einbeziehung der graphischen Datenverarbeitung für die Nutzung eines Informationsfonds möglich ist.

Dabei wird davon ausgegangen, daß im Rahmen der gegenwärtig in der DDR verfügbaren Gerätetechnik, die geeignetste graphische Ausgabemöglichkeit der Einsatz von automatischen Zeichenmaschinen ist.

2. Das Programmpaket DASYGRAF für die Verbindung eines Dateiverwaltungssystems mit der graphischen Ausgabe von Informationen

Die Arbeit mit einem Informationsfonds stellt erhöhte Anforderungen sowohl an die wissenschaftliche Arbeitsweise des Nutzers als auch an die Nutzerfreundlichkeit des EDV-Systems.

Bei der Gewährleistung einer nutzerfreundlichen graphischen Ausgabe sind folgende Gesichtspunkte zu berücksichtigen:

- Die einfachen und vielseitigen Möglichkeiten der graphischen Ausgabe müssen vom Nutzer zu jeder Zeit ohne großen Aufwand veranlaßt werden können.
- Die wahlfreie Ausgabe jedes Datenbestandes eines Dateiverwaltungssystems in Form einer geeigneten Grafik muß möglich sein.
- Die graphische Ausgabe eines beliebigen Datenbestandes, von Ergebnissen eines Programmes oder manuell am Bildschirmgerät eingegebener Daten, ist zuzulassen.
- Die Bereitstellung eines Sortiments von Zeichenprogrammen für die Erzeugung von Grafiken muß sicher sein.
- Mit einem relativ geringen Aufwand an menschlicher Arbeit ist die Erzeugung einer Grafik möglich.

Zur Lösung dieser Aufgaben wurde das zweiteilige Programmpaket DASYGRAF entwickelt. Es arbeitet mit folgenden Systemunterlagen:

1. DASY - Zentrales Dateiverwaltungssystem des SOPS KOMPASS (/15/).
2. BS-SS/R21 - Bildschirmsteuersystem (/13/, /14/).
3. DIGIGRAF - Software für ESER-Rechner

In DASY werden Daten in Form von Matrizen verwaltet, wobei jede Zeile und Spalte eine Bezeichnung trägt. Durch sie ist es dem Nutzer möglich, jede Information genau zu adressieren, was eine relativ gute Nutzerfreundlichkeit gewährleistet. Es werden aktuelle und Archivdaten unterschieden, so daß Trenduntersuchungen, Plan/Ist-Vergleiche und ähnliche Operationen ausführbar sind (s. auch Abb. 1).

Das BS-SS/R21 wurde für den Rechner R21 entwickelt, ist aber auf größeren Rechnern einsetzbar und wird im Beispiel auf der EDVA tS 1040 genutzt. Es unterstützt den Teilerbetrieb und verwaltet die Arbeit von 8 Sektoren, wobei pro Sektor ein Problem abgearbeitet werden kann. Die Aktivierung der Sektoren erfolgt nach dem Warteschlangenprinzip. Die Ein- und Ausgabe-Operationen, durch Makros veranlaßt, teilen das jeweilige Problem in Zeitscheiben. Das unterbrochene Problem wird wieder an die Warteschlange angereiht. Bei Beendigung der Problemabarbeitung wird der Sektor für das nächste Problem freigegeben.

Die DIGIGRAF-Software auf ESER entspricht dem Leistungsangebot der entsprechenden Software auf dem Rechner BESM-6.

Die Abb. 2 zeigt die Bestandteile und den Datenfluß im Programmpaket DASYGRAF, wobei die Schnittstellen mit den Systemunterlagen zu erkennen sind.

Das Leistungsangebot enthält die Aufbereitung von Daten und ihre Ausgabe als Kurven-, Säulen- und Sektordiagrammen sowie in Maschinenbelegungsplänen. Dabei sind die Darstellungen in den Formaten A1-A4 möglich, und es sind pro Blatt bis zu 4 Grafiken zugelassen. Die Anlage 1 zeigt eine Auswahl der möglichen Darstellungen mit den verfügbaren Varianten der Beschriftung.

Für die Steuerung der Varianten enthält der Teil 1 den Bildschirmdialog zur Gestaltung der gewünschten Grafik und die Aufbereitung der darzustellenden Daten in die Datenstruktur des Zeichenprogrammes, während Teil 2 das eigentliche Zeichenprogramm für die Erzeugung des Steuerdatenträgers der Zeichenmaschine darstellt (s. /7/).

Diese Teilung mit den entsprechenden Schnittebenen ist die Voraussetzung für die Weiterentwicklung bzw. Anpassung an andere Dateiverwaltungssysteme, Zeichenprogramme sowie eventuell andere Dialogorganisationen. Auf die Probleme der Weiterentwicklung wird im Abschnitt 3 noch eingegangen. Zunächst sollen die Funktionen der einzelnen Programmteile erläutert werden.

Abb. 3 zeigt die Varianten des Dialoges. Bei den Standardausführungen müssen nur die Angaben zur Darstellungsart und den darzustellenden Daten eingegeben werden. Alle anderen Angaben sind konstant.

Im Zweig 2 ist eine weitgehende Beeinflussung der Gestalt der Grafik möglich. Dazu werden im echten Dialog die einzelnen Steuerparameter zur Veränderung der Standardform der Grafik erarbeitet und in einem "Common"-Bereich abgespeichert. Nach Eingabe der darzustellenden Informationen erfolgt eine Aufbereitung auf Grund dieser Parameter, die den Nutzer nicht weiter belastet.

Der Dialog und die Datenaufbereitung werden im Programmteil DIADAT realisiert, dessen Struktur in Abb. 4 zu sehen ist.

In den Ebenen des Teil I werden die Daten für die Variation der Diagramme zusammengestellt, die als Sektor-, Kurven- und Säulendiagramme bzw. Maschinenbelegungspläne ausgegeben werden. Das Balkendiagramm stellt den Sonderfall des Maschinenbelegungsplanes dar. Die Ebene II enthält die Programme, die die Datenstruktur für das Zeichenprogramm generieren. Die Grundlage bilden die Programme des Dateiverwaltungssystems DASY.

Die Datenstruktur enthält 6 verschiedene Datenblöcke, von denen

Block I Format und Blatteilung,

Block II Darstellungsart und Beschriftung steuern, wählen

Block III - VI die darzustellenden Daten enthalten.

Der Aufbau der Blöcke I und II ist in Abb. 5 gezeigt.

Block II muß zu jedem der Blöcke III - VI angegeben werden, da er die jeweilige Darstellung beschreibt, d. h. bei einer vorgegebenen Verteilung des Blattes muß für jedes Viertel ein Block II erzeugt werden (s./6/).

In Abb. 6 ist die Struktur des Zeichenprogrammes PROSYT (s./6/) abgebildet. Ebene 2 ist für die Blatteinteilung verantwortlich, die es ermöglicht, daß ein bis vier Grafiken pro Blatt erzeugt werden können. Das Unterprogramm ZEICH steuert die Generierung der Steuerbefehle für den DIGIGRAF und ruft die dazu notwendigen Unterprogramme auf. Der Algorithmus der Steuerung ist den Entscheidungstabellen in Abb. 7 zu entnehmen. Dieses Zeichenprogramm kann sowohl Daten vom Magnetband als auch von Lochkarten verarbeiten. Die Abarbeitung ist in einer anderen Partition oder zu anderer Zeit vorgesehen, um den Dialogbetrieb nicht durch die Arbeit des Programmes PROSYT zu behindern.

Der Teil DIADAT des Programmpaketes ist in Überlagerungsstruktur erstellt und nimmt im Hauptspeicher jeweils nicht mehr als 12 kbyte ein. Der Teil PROSYT benötigt etwa 20kbyte Hauptspeicherkapazität. Bei der Testung des Teiles PROSYT wurden für die Grafik in Anlage 1 (Format A3 mit 4 Diagrammen - Stäulen-, Kurven- und Sektordiagramm mit Beschriftung) etwa 15 min Rechenzeit und 30 min Zeichenzeit auf dem DIGIGRAF 1008 benötigt.

3. Gesichtspunkte für die Weiterentwicklung

Das vorliegende Programmpaket DASYGRAF ist eine Basislösung für die Verknüpfung der Arbeitsmittel alphanumerisches Bildschirmgerät, der graphischen Ausgabe mittels Zeichmaschine und ein Dateiverwaltungssystem. Die Ausführungen zeigten, wie die Verbindung dieser Arbeitsmittel programmtechnisch gestaltet werden kann.

Das durchgängig eingehaltene Modularprinzip ist die Voraussetzung für eine vielseitige Erweiterung.

Möglich sind u. a. folgende Veränderungen:

- Die Erweiterung des Angebotes an graphischen Darstellungen z. B. durch Netzwerke, die nur eine Ergänzung der Steuerung im Programm ZEICH erfordert. Im Dialog- und Aufbereitungsteil DIADAT müssen die entsprechenden Modulen eingeführt werden.
- Die Nutzung eines anderen Dateiverwaltungssystems. Hier muß die Ebene II in Abb. 4 den neuen Anforderungen angepaßt werden, um die Datenstruktur für das Zeichenprogramm zu erzeugen.
- Die Nutzung einer völlig anderen Kombination von Dateiverwaltungssystemen und Zeichenprogrammpaketen. Diese Umstellung erfordert allerdings einen großen Aufwand, da alle speziellen Modulen ersetzt werden müssen.

Eine noch weitergehende Ergänzung ist die Realisierung des Zweiges 3 in Abb. 3, da durch die aktive Arbeit mit Bausteinen und Bausteingruppen die Programmierung neuer Zeichenprogramme effektiver gestaltet wird.

DIPROVA, eine Systemunterlage zur Dialog-Programm-Variation (s./11/), könnte eine solche Grundlage der Erweiterung sein. Geeignet erscheint auch die Menütechnik von DIGRA 73 (s./2/). Der Aufwand wird allerdings nicht unwesentlich sein.

Trotzdem sollte dieser Zweig in der graphischen Datenverarbeitung bei verarbeitenden Anwendungsmöglichkeiten genutzt werden. Das vorgestellte Programmpaket stellt einen Versuch dar, in die programmtechnische Gewährleistung eines Informationsfonds die Möglichkeiten der graphischen Ausgabe von Informationen einzubeziehen.

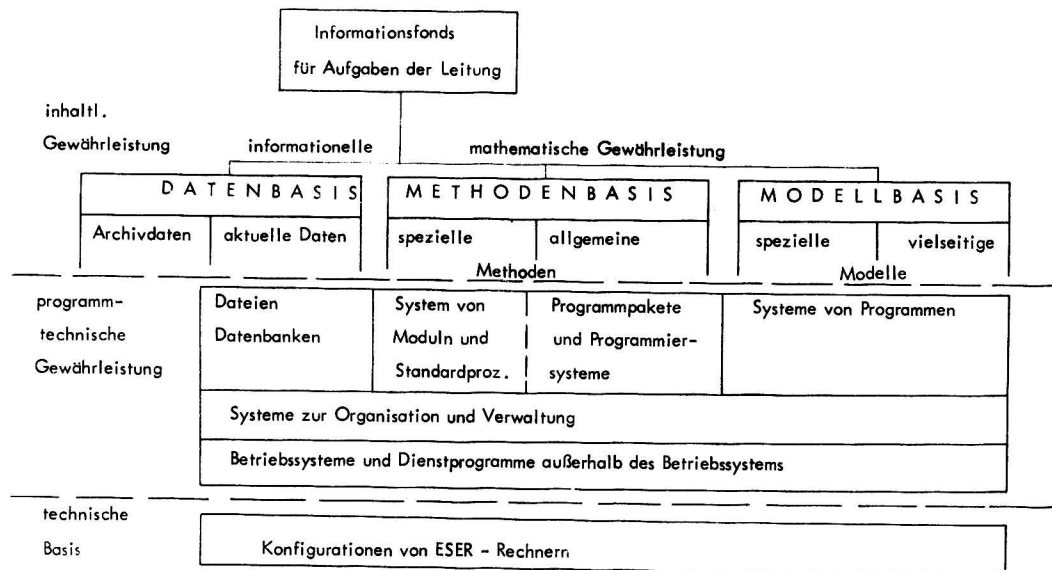


Abb. 1: Prinzipieller Aufbau eines Informationsfonds für Aufgaben der Leitung (s./9/)

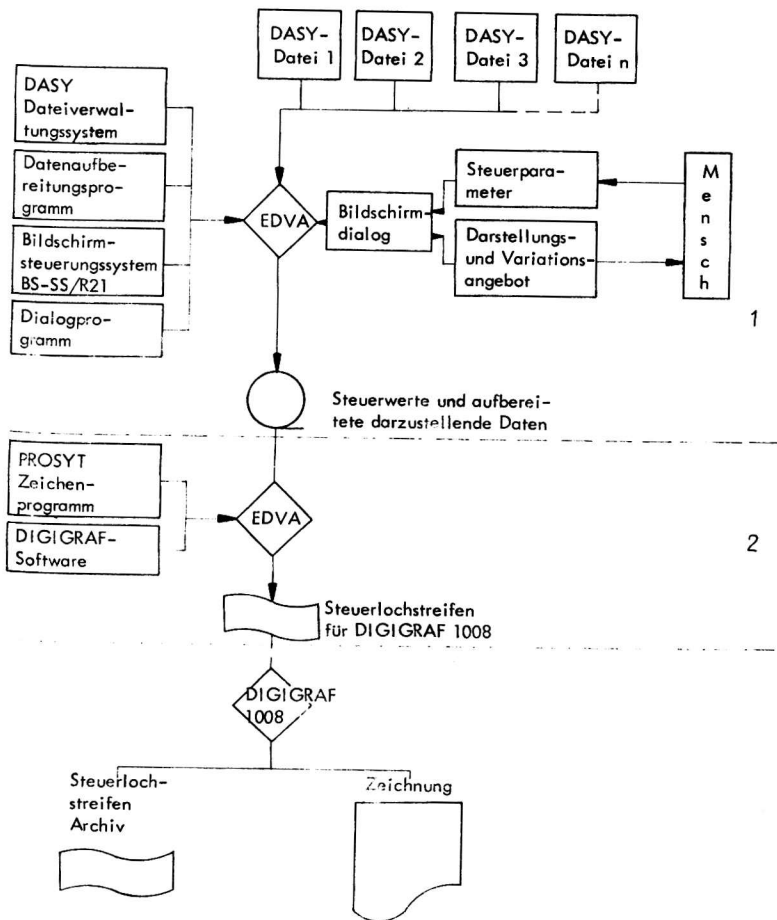


Abb. 2: Struktur und Datenflußplan des Programmpaketes DASYGRAF

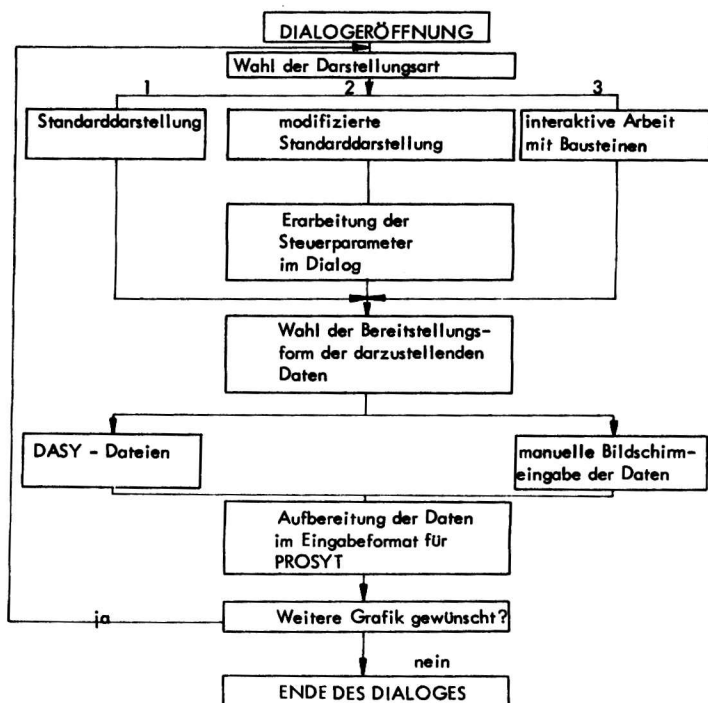


Abb. 3: Varianten des Dialoges zur Aktivierung der graphischen Ausgabe

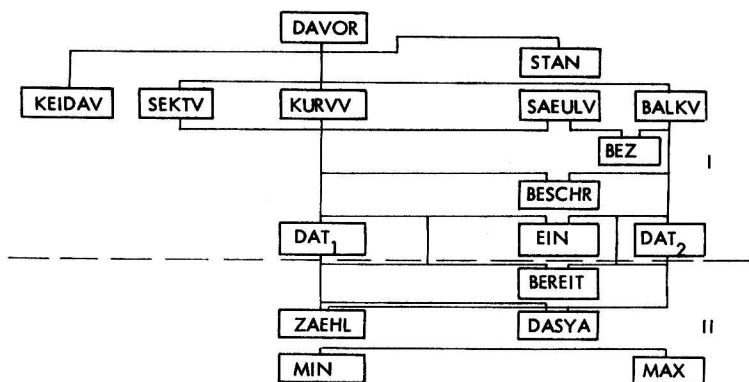


Abb. 4: Struktur des Programmpaketes DIADAT

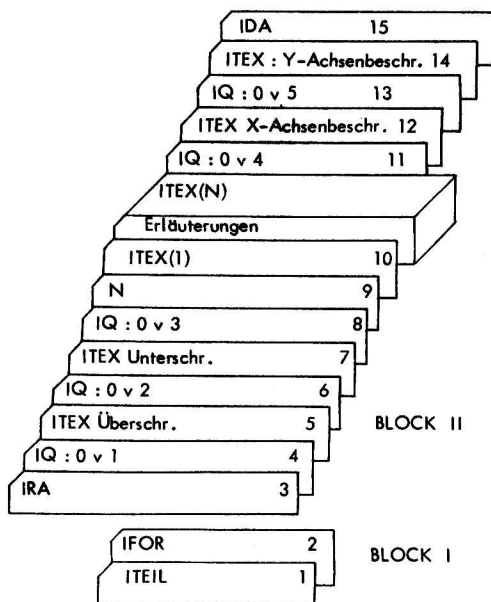


Abb. 5: Datenstruktur

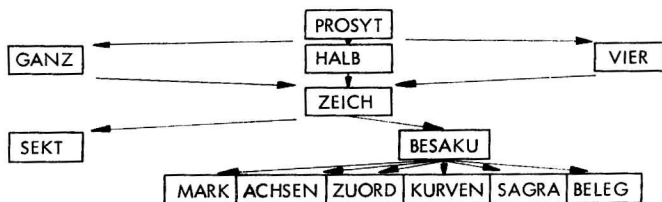


Abb. 6: Struktur des Teilpaketes PROSYT (Zeichenprogramm)

Nr.	Parameter Name	Wert	AKTIONEN	goto
1	ITELL	1	Einlesen von IFOR	5
2		2	Einlesen von IFOR	6
3		3	Einlesen von IFOR	7
4				STOPP
5			$FAK = \sqrt{2} * (6 - IFOR) * 100$	8
6			$FAK = \sqrt{2} * (5 - IFOR) * 100$	8
7			$FAK = \sqrt{2} * (4 - IFOR) * 100$	
8				ZEICH

Tab. Hierarchieebene 1 und 2

1	IRA	0		3
2		1	Rahmen zeichnen	
3	IQ	0	keine Beschriftung	
4		1	Überschrift zeichnen	
5		2	Unterschrift zeichnen	
6		3	Erläuterungen zeichnen	
7		4	X-Achsen-Bezeichnung zeichnen	
8		5	Y-Achsen-Bezeichnung zeichnen	
9	IDA	0		12
10		1	Zeichnen des Sektordiagrammes	12
11		2v3v4		BESAKU
12				STOPP

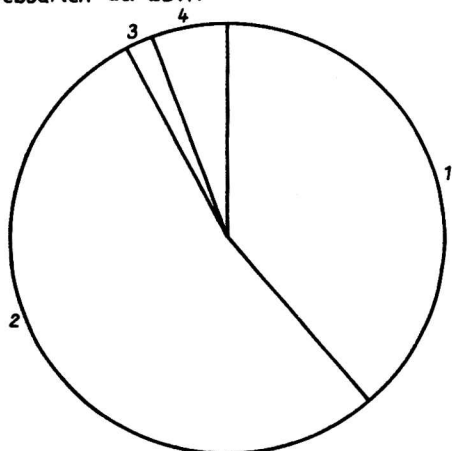
Tab. ZEICH

1			Achsen zeichnen	
2			Achsenmarkierung zeichnen	
3			Zuordnung d. Eingebewerte in formatger. MILLIMETER	
4	IDA	2	Kurvendiagramm	11 *
5		3		7
6		4		9
7	IQ	1	X-Achsenbeschr. mit Anwenderdaten	
8			Säulendiagramm	11
9	IQ	1	Y-Achsenbeschr. mit Anwenderdaten	
10			Maschinenbelegungsplan	
11				STOPP

Abb.7: Zusammenstellung der Entscheidungstabellen der Steuerung

Anlage 1

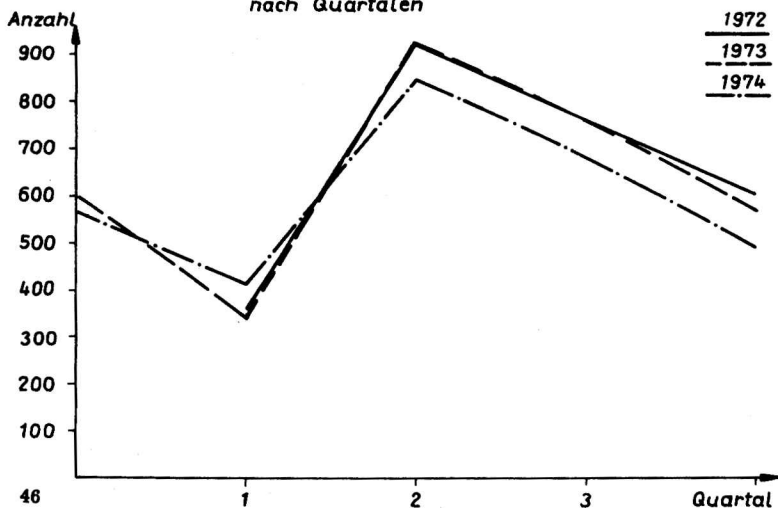
Betriebsarten der EDVA



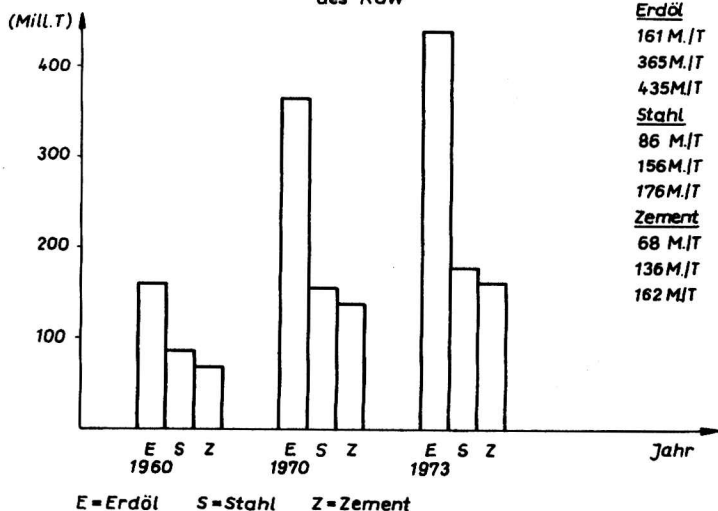
- 1 36.59 % Stapelver
- 2 53.50 % St. im Mult
- 3 2.07 % Teilnehmer
- 4 5.80 % Echtzeitve

Prozent der Computerarbeit

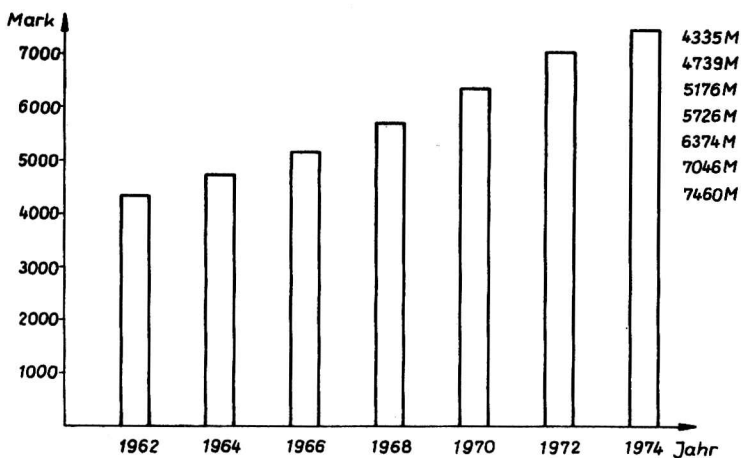
Statistik über die von Kindern verursachten Verkehrsunfällen nach Quartalen



Entwicklung der Produktion ausgewählter Industrieerzeugnisse des RGW



Produziertes Nationaleinkommen pro Kopf der Bevölkerung



Literatur:

- /1/ Direktive des IX. Parteitages der SED zum Fünfjahrplan für die Entwicklung der Volkswirtschaft der DDR in den Jahren 1976 - 1980, Dietz-Verlag, Berlin 1976.
- /2/ Autorenkoll.: Programmierhandbuch DIGRA 73. Universität Rostock, Sektion Mathematik, Rostock 1974.
- /3/ Böhme, W.: Möglichkeiten der graphischen Datenverarbeitung bei Leitungsaufgaben. Vortrag auf dem Problemseminar ASUP der TU Dresden, Sektion Informationsverarbeitung, Geising 1975.
- /4/ Böhme, W.: Integration der Bildschirmtechnik in Schwerpunktprojekte der EDV-Anwendung im VEB Büromaschinenwerk Sömmerda. TU Dresden, Sektion Informationsverarbeitung, Studie 1975 (unveröffentlicht).
- /5/ Böhme, W.: Bildschirmdialog zur Nutzung der Digitalgraphik für Aufgaben der Leitung und Planung. Wissenschaftliche Zeitschrift der TU Dresden 26, 2, in Vorbereitung (1977).
- /6/ Krömer, L., Pahlitzsch, I.: Entwicklung eines Programmpaketes für die Ausgabe von ausgewählten graphischen Darstellungen in Leitung und Planung über eine Zeichenmaschine TU Dresden, Sektion Informationsverarbeitung, Diplomarbeit 1975.
- /7/ Lehmann, G.: Bildschirmdialog mit graphischer Informationsausgabe über Zeichenmaschinen. TU Dresden, Sektion Informationsverarbeitung, Diplomarbeit 1976.
- /8/ Pflug, A.: Informationsfonds für die Leitung und Planung von Kombinat. Vortrag auf der internationalen wissenschaftlichen Tagung "Mathematik und Kybernetik in der Ökonomie (MKÖ)IV" , Dresden 1975 (veröffentlicht als Mikrofiche).
- /9/ Pflug, Ch. Informationsfonds in rechnergestützten Informationssystemen für die Leitung und Planung. Rechentechnik/Datenverarbeitung 13, 4, 35 - 39 (1976).
- /10/ Rohde, H. Einsatz des automatischen Zeichnens für graphische Darstellungen in Leitung, Planung und Organisation (in "Automatisches Zeichnen durch EDV") Institut für Energetik (Hrsg.) Leipzig 1974, S. 17/1 - 17/9

- /11/ Rothhardt, G. DIPROVA - Software zur Dialog-Programm-Variation, Anwenderinformation. VEB Carl Zeiss Jena, Forschungsleitzentrum AUTEVO, Jena 1975.
- /12/ Wilhelm, K. H. Grafische Darstellung in Leitung und Organisation, Verlag Die Wirtschaft, Berlin 1970
- /13/ Programmierhandbuch zum Bildschirmsteuersystem BS-SS/R21. VEB Kombinat Robotron Dresden, Stammbetrieb ORZ.
- /14/ Teilhaberhandbuch zum Bildschirmsteuersystem BS-SS/R21. VEB Kombinat Robotron Dresden, Stammbetrieb ORZ.
- /15/ Programm- und Anwendungsbeschreibung SOPS KOMPASS Zentrales Dateiverwaltungssystem DASY. VEB Kombinat Robotron, Dresden 1975.

Anschrift der Verfasser:

HS-Ing. Wolfgang Böhme,

Dr. Arndt Pflug

Technische Universität Dresden

Sektion Informationsverarbeitung

DDR - 8027 Dresden, Mommsenstr. 13

Rekonstruktion geometrischer Objekte aus ihrer Darstellung im Grund-Aufriß-Verfahren

1. Anliegen

Die Realisierung geometrischer Denk- und Arbeitsweisen mit Hilfe der Rechentechnik erfordert eine Modellierung der - i. a. räumlichen - geometrischen Objekte durch geeignete Datenstrukturen. Für die Entwicklung solcher Strukturen sind die Anforderungen zu berücksichtigen, die sich sowohl von Seiten der Rechentechnik als auch aus der Sicht der Geometrie ergeben, wie z. B.

- minimaler Bedarf an Speicherplatz,
 - schneller Zugriff zu den einzelnen Daten,
 - Möglichkeit der Segmentierung der Struktur
- und
- durch welche Parameter sind die einzelnen geometrischen Objekte beschreibbar,
 - welches sind die günstigsten Parameter in Bezug auf die geometrischen Operationen.

Ein weiterer, hierbei zu beachtender Gesichtspunkt ist der gesamte Komplex des Dialoges zwischen Nutzer und Rechenanlage über einen Bildschirm. Zu diesem Zweck muß das räumliche Objekt zunächst in eine ebene Darstellung übergeführt werden. Dies kann nach den in der darstellenden Geometrie üblichen Verfahren geschehen. Weiterhin sollen an dem Bild bestimmte Manipulationen vorgenommen werden können, z. B. Verändern einzelner Teile des Objektes oder Hinzufügen neuer Elemente, d. h., in gewissem Umfang muß ein Arbeiten in dem entsprechenden Darstellungsverfahren möglich sein. Es ist also angebracht, sich mit der Frage zu befassen, welche Schlußfolgerungen auf die Datenstrukturen sich aus den Darstellungsverfahren ableiten lassen.

Eine der wesentlichsten Forderungen an die ebene Darstellung eines räumlichen Objekts ist ihre Eindeutigkeit: Aus dem Bild muß das räumliche Objekt rekonstruiert werden können. Das bedeutet, daß Untersuchungen nötig sind, durch welche Parameter das dem be-

treffenden Darstellungsverfahren entsprechende Bild beschrieben werden kann, und ob es möglich ist, aus ihnen in eindeutiger Weise die Parameter des räumlichen Objekts zu gewinnen.

Solche Untersuchungen sind begonnen worden für das Grund-Aufriß-Verfahren, und zwar zunächst für die geometrischen Grundelemente Punkt, Gerade, Ebene sowie für Kreise und konvexe Polyeder.

2. Grund-Aufriß-Punkt- und Grund-Aufriß-Mengenabbildung

Grundlage der Untersuchung war die Unterscheidung zwischen folgenden drei Abbildungen (s./1/):

1. Grund-Aufriß-Punktabbildung

$$\varphi: X \rightarrow (X', X'')$$

Jedem Punkt X des euklidischen geometrischen Raumes wird ein Paar (X', X'') von Punkten der Zeichenebene zugeordnet.

2. Die von φ induzierte Abbildung von Punktmengen

$$\Phi: M \rightarrow \{(X', X'') : X \in M\}$$

Einer Menge M von Punkten des Raumes wird eine Menge von Punktpaaren der Ebene zugeordnet.

3. Grund-Aufriß-Mengenabbildung

$$\psi: M \rightarrow (\{X' : X \in M\}, \{X'' : X \in M\})$$

Einer Punktmenge M des Raumes wird ein Paar von Punktmengen der Ebene zugeordnet.

Mit Hilfe von ψ werden im Grund-Aufriß-Verfahren die Figuren des Raumes (=Punktmengen) abgebildet. ψ ist jedoch im Gegensatz zu φ und Φ nicht eineindeutig. Bedingung für die eindeutige Existenz des Urbildes zweier Mengen ist:

Jede Ordnungslinie schneidet beide Mengen und wenigstens eine der Mengen in nur einem Punkt.

Die Tatsache, daß die Überführung aus dem Raum in die Ebene nicht in jedem Fall umkehrbar ist, muß bei der Rekonstruktion räumlicher Objekte aus ihrer Darstellung im Grund- und Aufriß stets beachtet werden.

3. Rekonstruktion aus Grund- und Aufriß

Um das Bild eines räumlichen Objekts im Grund-/Aufriß zu beschreiben, stehen meist verschiedene Möglichkeiten zur Verfügung. Legt man bestimmte Parameter fest und stellt sich das Ziel, aus diesen eindeutig die Parameter der räumlichen Beschreibung zu ermitteln, so ist dazu prinzipiell folgende Vorgehensweise nötig:

1. alle Lagen des Objektes bzgl. der beiden Bildebenen herausfinden, bei denen eine Rekonstruktion mit Hilfe der gewählten Parameter nicht eindeutig möglich wäre,
2. Überlegungen anstellen, wie diese Sonderfälle - möglichst geschlossen - behandelt werden können, z. B. Verwenden des Kreuzrisses oder Ergänzen der Beschreibung des Grund- und Aufrisses durch zusätzliche Parameter bzw. Benutzen anderer Parameter,
3. einen entsprechenden Algorithmus zur Rekonstruktion aufstellen.

3.1. Grundelemente

Die ausführliche Behandlung der Grundelemente Punkt, Gerade, Ebene verschafft einen Überblick über die Probleme bei den ausgezeichneten Lagen bzgl. der Bildebenen, der dann auch für solche geometrischen Objekte von Nutzen ist, die sich mit Hilfe dieser Grundelemente beschreiben lassen. Weiterhin können bei der Rekonstruktion solcher Objekte die für die Grundelemente aufgestellten Algorithmen verwendet werden. Für Punkt, Gerade, Ebene werden folgende Grund-Aufriß-Abbildungen festgelegt und die entsprechenden Abbildungen der homogenen Koordinatenvektoren, die zur Beschreibung der Grundelemente gewählt wurden, formuliert:

Punkt $X \in \overline{E}^3$ \overline{E}^3 : projektiv abgeschlossener euklidischer Raum

Punktabbildung auf das Punktepaar der Ebene

(Grundrißpunkt, Aufrißpunkt):

$$X \mapsto (X', X'')$$

Beschreibung durch homogene Punktkoordinaten:

$$(x_0, x_1, x_2, x_3)^T \mapsto (x_0, x_2, -x_1)^T, \quad (x_0, x_2, x_3)^T$$

Gerade $g \subset \overline{E}^3$

Mengenabbildung auf das Paar von Geraden der Ebene

(Grundrißgerade, Aufrißgerade):

$$g \mapsto (g', g'')$$

Beschreibung durch Plücker'sche Linienkoordinaten bzw. Geradenkoordinaten der Ebene:

$$(g_1, g_2, g_3, g_4, g_5, g_6)^T \rightarrow (g_6, g_1, g_2)^T, (g_4, -g_3, g_2)^T$$

Ebene $\mathcal{E} \subset \overline{E^3}$

Mengenabbildung auf das Paar von Geraden der Ebene

(Grundriß der 1. Spurgeraden, Aufriß der 2. Spurgeraden):

$$\mathcal{E}(h', v'')$$

Beschreibung durch Ebenenkoordinaten bzw. Geradenkoordinaten der Ebene:

$$(e_0, e_1, e_2, e_3)^T \rightarrow (e_0, e_2, -e_1)^T, (e_0, e_2, e_3)^T$$

Die Bedingungen für die Existenz und Eindeutigkeit des Urbildes bei diesen drei Abbildungen können durch Forderungen an die Koordinatenvektoren ausgedrückt und somit im Algorithmus überprüft werden. In den Fällen der Nicht-Eindeutigkeit wird der Kreuzriß hinzugezogen.

Geht man bei Gerade und Ebene von einer Beschreibung durch andere Bestimmungsstücke aus

(z. B. für eine Gerade: 2 Punkte,

für eine Ebene: 3 Punkte oder

2 nicht windschiefe Geraden oder die Achsenabschnitte),

so lassen sich auf dieser Grundlage weitere Algorithmen für die eindeutige Rekonstruktion angeben, die ebenfalls alle möglichen ausgezeichneten Lagen berücksichtigen.

3.2. Kreis

Ein geometrisches Objekt, das sich in seiner Beschreibung auf die Grundelemente stützt, ist der Kreis.

Ein Kreis K ist im Raum bestimmt durch die Parameter:

Kreisebene \mathcal{E}

Mittelpunkt M

Radius r .

Eine Möglichkeit zur Beschreibung der Grund- und Aufrißellipsen K' und K'' (die bei bestimmten Lagen der Kreisebene in Strecke bzw. Kreis ausarten können) besteht darin, jeweils die Koordinaten dreier Punkte anzugeben, und zwar
 des Mittelpunktes,
 eines der Hauptscheitel,
 eines der Nebenscheitel.

(Nötigenfalls werden Punkte entsprechend definiert.)

Die eindeutige Bestimmung eines Kreises aus zwei gegebenen Ellipsen E_1 (Grundriß) und E_2 (Aufriß) ist, obwohl offensichtlich in den meisten Fällen die Bedingung für die Eindeutigkeit des Urbildes verletzt ist, möglich, weil dabei die für Bildellipsen eines Kreises charakteristischen Eigenschaften ausgenutzt werden können.

Ist - unter der Voraussetzung, daß ein Urbild existiert (Bedingung für die Existenz des Urbildes: $E_1'' = E_2'$) - neben der für die Existenz eines Kreises als Urbild notwendigen Bedingung

(1) Die Hauptachsen von E_1 und E_2 sind gleich lang

noch eine der folgenden Forderungen an Lage und Gestalt von E_1 und E_2 erfüllt, so kann der Kreis $K \dots (\varepsilon, M, r)$ bis auf Fall (6) eindeutig aus den Mittelpunkten und jeweils einem Hauptscheitel von E_1 und E_2 bestimmt werden. Im Fall (6) sind zur Rekonstruktion die Mittelpunkte und Nebenscheitel nötig.

Forderungen an

Gestalt		Lage der Hauptsache	
von E_1	von E_2	von E_1	von E_2
(2)	Ellipsen	$\neq x_{12}$ und $\perp x_{12}$	
(3)	Strecken	identisch und $\perp x_{12}$	
(4) Strecke	Ellipse	$\neq x_{12}$	$\perp x_{12}$
	Kreis	$\parallel x_{12}$	$(\perp x_{12})$
Ellipse	Strecke	$\perp x_{12}$	$\neq x_{12}$
(5) Kreis		$(\perp x_{12})$	$\parallel x_{12}$
(6)	Ellipsen	$\parallel x_{12}$	$\parallel x_{12}$

(x_{12} : Rißachse)

4. Abschließende Bemerkung

Theoretische Untersuchungen dieser Art - für ausgewählte geometrische Objekte innerhalb der gebräuchlichsten Darstellungsverfahren vorgenommen - sollen die Grundlage sein zur Beantwortung der eingangs gestellten Frage, welchen Einfluß auf die Gestaltung der Datenstrukturen die Überführung der räumlichen Objekte in eine ebene Darstellung und die Arbeit an dem so gewonnenen Bild haben. Um Strukturen aufstellen zu können, die den Anforderungen des Dialoges gerecht werden, müssen aufbauend darauf natürlich noch alle die Probleme berücksichtigt werden, die mit der Realisierung einer Datenstruktur auf einer Rechenanlage mit Bildschirmgerät im Zusammenhang stehen.

Literatur:

/1/Kirsch, A.: Ist die Grund-Aufriß-Abbildung injektiv? in: Mathematisch-physikalische Semesterberichte Band XIX, Heft 2 (1972).

Anschrift des Verfassers:

Dipl.-Math. Dagmar Broschinski

Techn. Universität Dresden

Sektion Mathematik

DDR - 8027 Dresden

MommSENstr. 13

Programmpaket zur Interpolation empirischer Kurven und Flächen im Raum mit Möglichkeiten der Steuerung

Das Programmpaket wurde auf der theoretischen Grundlage der von G. Bär entwickelten Interpolationsverfahren (s./1/,/2/) aufgebaut. Bei der Realisierung an der Rechenanlage BESM-6 wurde als Programmiersprache ALGOL-60 benutzt. Die Art der Ausgabe wird durch einen Eingangsparameter bestimmt. Es bestehen prinzipiell zwei Möglichkeiten:

- nur Druck der Ergebnisse im Rechnerausdruck,
- zusätzliches Stanzen eines Lochstreifens zum automatischen Zeichnen am DIGIGRAF 1008.

Die benötigten Stanzprogramme (aus der DIGIGRAF-Bibliothek), die Interpolationsprozeduren, die Standardanwenderprogramme sowie alle Hilfsprozeduren stehen auf einem Magnetband (Nutzer: Bereich MKR, Sektion Mathematik, TU Dresden). Alle diese Programme sind extern vereinbart, wobei die Stanzprogramme als Standardmassive und alle anderen als Textinformationen auf dem Band stehen. Die Steuerung des Kurvenverlaufes bzw. des Flächenbildes erfolgt durch die Belegung von Eingabeparametern vor dem Aufruf der entsprechenden Interpolationsprozedur. Wünscht man aber keine Steuerung, so werden die o. g. Parameter innerhalb der Prozedur mit Werten belegt, die der HERMITESchen Interpolation entsprechen. Die algorithmische Errechnung dieser Werte bleibt somit dem Nutzer erspart.

1. Interpolationsprozeduren

1.1. Kurven

Es handelt sich hier um Kurvenscharen aus den Mengen $X_n^{1,1}$ und $X_n^{2,1}$, die im Beitrag von G. Bär (s./2/) definiert worden sind. Es seien

p_i : Koordinatenvektor des i-ten Stützpunktes P_i

c_i : Koordinatenvektor der Tangente an die Kurve in P_i

r_i : Steuerparameter rechts von P_i
 l_{i+1} : Steuerparameter links von P_{i+1}

Wir legen fest

$p_{0,k} : p_k$ und $p_{1,k} : c_k$ für $k = i, i+1$.

Die einheitliche Darstellung

$$x_i(u) = \sum_{s=0}^1 (r_i^s p_{s,i} f_{0,s} + l_{i+1}^s p_{s,i+1} f_{1,s})$$

bei der stückweisen Interpolation von Kurven gestattet es, mit Hilfe eines einzigen Algorithmus jedes Kurvenstück zu beschreiben. Dies wird in der Prozedur KUZEL - Baustein des Programmpaketes - geleistet. Durch eine entsprechende Aktualisierung der formalen Parameter wird beim Aufruf von KUZEL die Interpolation und eventuell eine bildliche Darstellung der Kurve zwischen beliebigen P_i und P_{i+1} rechnerisch realisiert.

Aufruf: KUZEL(XA,EA,XE,EE,R,L,EPS,ZEI,NORM,PROJ);

Aktualisierung:

XA = p_i , EA = c_i , XE = p_{i+1} , EE = c_{i+1} ,

R = r_i , L = l_{i+1}

EPS : Genauigkeitsgrenze

ZEI = true: Stanzen des Lochstreifens für den DIGIGRAF

= false: kein Stanzen

NORM = true: Normierung von EA und EE

= false: keine Normierung

PROJ : Projektionsmatrix (wenn ZEI = true, wird die Projektionsanweisung übersprungen)

Als Ausgang erhält man ein nutzerfreundliches Druckbild der Eingangsparameter XA, EA, XE, EE, R, L, EPS sowie der Ergebnisse der Interpolation und zusätzlich - im Fall ZEI = true - den Lochstreifen zum automatischen Zeichnen.

Die Prozeduren KUZUSA und KHERM ermöglichen die Interpolation der gesamten Kurve. Beide sind nach demselben Prinzip aufgebaut:

In einem ersten Teil erfolgt die Aufbereitung der in jedem Stützpunkt für die Kurvenzusammenstellung benötigten Parameter (Punkt-, Tangentenkoordinaten und Steuerparameter). Im zweiten Teil wird die Kurve aus den einzelnen Stücken durch den wiederholten Aufruf von KUZEL zusammengestellt.

KUZUSA wird eingesetzt bei der verallgemeinerten HERMITESchen bzw. Spline-Interpolation, KHERM beim Spezialfall der normalen eben erwähnten Interpolation.

1.2 Flächen

Es werden nur Flächen aus der Menge $X_{n,m}^{1,1}$ interpoliert. Dabei sind die u- und v-Parameterlinien Kurven aus den Mengen $X_n^{1,1}$ und $X_m^{1,1}$. Speziell für die Netzklinien ist es möglich, Interpolierende aus $X_n^{2,1}$ und $X_m^{2,1}$ zu konstruieren. Dies hat aber keinen geometrischen Einfluß auf den Anschluß benachbarter Pflaster, sondern bezweckt lediglich die Variante einzuräumen, wo der Nutzer Tangenten nur an die Randkurven der Fläche vorgibt.

Die Prozedur KNETZ übernimmt die Interpolation der Netzkurven. Sie ruft u. a. KHERM und KUZUSA als Hilfsprozeduren auf. KPATCH füllt ein Pflaster mit Parameterlinien in u- und v-Richtungen, wobei die Anzahl der Parameterlinien pro Pflaster und für jede Richtung vom Nutzer bestimmt werden kann. Sie benutzt als Hilfsprozedur KUZEL. KPATCH eignet sich besonders, wenn nur einzelne Pflaster der Fläche interpoliert bzw. gesteuert werden sollen. Eine Pflasterung der ganzen Fläche durch wiederholten Aufruf von KPATCH erweist sich rechentechnisch als uneffektiv. Deshalb wurde KFLAE so aufgebaut, daß die Parameterlinien über die ganze Fläche gezogen werden. Dabei werden die Netzklinien ausgesondert und durch Aufruf von KNETZ am Anfang von KFLAE erledigt. Das Ergebnis als bildliche Darstellung bleibt eine pflasterweise interpolierende Fläche, da die Pflaster durch das Gitter der Netzkurven abgegrenzt sind.

Die Algorithmen, die diesen Programmen zugrunde liegen, sichern, daß sich die u- und v-Parameterlinien immer schneiden (s./3/). Bei der Aktualisierung der Parameter vor den Prozeduraufrufen muß auf die Reihenfolge (bei der Belegung der verschiedenen Felder) besonders geachtet werden. Diese delikate Programmieraufgabe kann man durch die sogenannten Standardanwenderprogramme umgehen.

2. Standardanwenderprogramme (SAP)

Diese Programme sollen es dem Nutzer ermöglichen, lediglich durch die Eingabe von Daten eine Kurve oder eine Fläche zu interpolieren und eventuell zeichnen zu lassen.

In einem SAP werden alle Größen vereinbart, die bei den verschiedensten Varianten benötigt werden. Die vom Nutzer eingegebenen Daten enthalten logische Werte, die über die ausgewählte Art der Interpolation bzw. des Zeichnens informieren.

Das SAP liest diese Daten ein, testet sie und liest anschließend die benötigten numerischen Daten ein, mit denen es dann die entsprechenden Interpolations- bzw. Stanzprozeduren beim Aufruf aktiviert.

Mit der Interpolation von geodätischen Höhenlinien und Bruchflächen aus der Werkstoffkunde hat dieses Programmpaket in der Praxis bereits Anwendung gefunden.

Literatur

- /1/ G. Bär: Vorlesung über Differentialgeometrie. TU Dresden, Sektion Mathematik, 1974
- /2/ G. Bär: s. Vortrag in diesem Heft S. 5 - 22.
- /3/ S. Cohen: Interpolation glatter Kurven und Flächen mit Steuerungsmöglichkeiten für die rechnergestützte Konstruktion. Diplomarbeit, TU Dresden, Sektion Mathematik, 1975.

Anschrift des Verfassers:

Dipl.-Math. Serge Cohen
Techn. Universität Dresden
Sektion Mathematik
DDR - 8027 Dresden
MommSENstr. 13

Bildmanipulationen am Koordinatenmeßtisch

Koordinatenmeßtische bieten eine einfache Möglichkeit, spezielle Punkte aus Zeichnungen aufzumessen und auf einem Datenträger zu fixieren. Diese Fähigkeit kann bei Einblendung geeigneter Zusatzinformationen so ausgebaut werden, daß sich der Datenbestand kompletter Zeichnungen erfassen und abändern läßt. Darüber hinaus können Meßdaten Sonderbedeutungen zugeordnet werden, auf deren Grundlage eine einfache Menütechnik möglich wird. Als Voraussetzung dafür ist ein geeignetes Auswertungsprogramm erforderlich, mit dessen Hilfe die bereitgestellten Informationen interpretiert, in ein rechnerinternes Bildmodell eingearbeitet und gegebenenfalls über ein graphisches Ausgabegerät sichtbar gemacht werden können. Im folgenden soll kurz ein derartiges Arbeitssystem vorgestellt werden, das für die Gerätekette DIGITRON - Großrechner-DIGIGRAF 100B unter Verwendung des Programmpakets AUTOZEI entwickelt wurde.

Wir gehen aus von der Hardware, wie sie mit dem DIGITRON zur Verfügung steht. Der Meßtisch besitzt eine Arbeitsfläche der Größe DIN A0 und gestattet mittels eines Laufwagen-Meßkopfs eine Koordinatenbestimmung in Zehntel Millimetern. Der Bezugspunkt ist dabei beliebig vorgebar und kann mit einem Anfangswert ungleich Null belegt werden. Die Koordinaten werden optisch angezeigt und können einzeln oder paarweise auf Lochstreifen ausgegeben werden. Über die Tastatur eines angeschlossenen Org.-Automaten, gegebenenfalls auch über eine Funktionstastatur am Meßtisch, können zusätzliche Textzeichen in den Lochstreifen eingefügt werden.

Wir legen nun ein spezielles Eingabeschema fest. Es ermöglicht die Bereitstellung verschiedener Arten von Datenrecords, deren Unterscheidung mittels eines zweistelligen Typkennzeichens am Recordanfang erfolgt. Grundsätzlich sind 3 Haupttypen von Datensätzen zugelassen: Werterecords, Namenrecords und Steuerrecords.

Die Werterecords enthalten alle Angaben, die für eine Generierung und Modifizierung der elementaren Bildbestandteile, sogenannter Bildatome, erforderlich sind. Als Bildatome zugelassen sind Polygone, Kurven, Symbole, Texte und Kreisbögen. Zusätzliche Modifikationsatome ermöglichen eine Beeinflussung der Ausführungsform, z. B. durch Einstellen einer variablen Linienart. Hinzu kommen sogenannte Routineatome, mit deren Hilfe Folgen von Bildatomen auf einen Subroutinen-Aufruf zurückgeführt werden können. Werterecords können des weiteren Daten aufnehmen, die für eine Identifizierung und Transformierung von Bildteilen mittels Meßkopf notwendig sind.

Die Namenrecords enthalten Identifikationsnamen, mit deren Hilfe sich separat verarbeitbare Teilbilder, sogenannte Bildelemente, abgrenzen lassen. Als Typmerkmal dient eine Stufennummer, durch deren Verwendung zugleich die Erzeugung einer Elementehierarchie möglich wird. Ihre unterste Ebene bilden die Grundelemente, die unmittelbar disjunkten Teilmengen von Bildatomen zugeordnet sind.

Die Steuerrecords enthalten Angaben darüber, wie die nachfolgenden Werte- und Namenrecords im Rechner interpretiert und verarbeitet werden sollen. Mit ihrer Hilfe können unterschiedliche Manipulationen eingestellt werden, die von der Erzeugung über eine Zerstörung, Ersetzung, Transformation bis hin zu einer Auflistung bzw. Auszeichnung von Bildelementen reichen. Die Manipulationen beziehen sich dabei auf ein flexibel strukturiertes Gesamtbild, das als Einheit konserviert und bei Bedarf reaktiviert werden kann. Daneben können über Steuerrecords auch Bildelementespeicher in die Arbeit einbezogen werden, die als Zwischen- und Basisspeicher sequentiell und damit relativ starr organisiert sind.

Die Erzeugung der Datenrecords vollzieht sich nun im einzelnen so, daß eine variable Arbeitszeichnung und konstante "Menüblätter" gemeinsam auf der DIGITRON-Arbeitsfläche untergebracht werden. Anstelle einer Tastatureingabe ergeben sich dann nach einmaliger Mitteilung entsprechender Bezugspunkte folgende Vereinfachungsmöglichkeiten:

1. Werterecords zur Erzeugung von Bildatomen können durch Digitalisierung atomspezi-fischer Punkte erzeugt werden. Polygone, Kurven und Kreisbögen lassen sich auf diese Weise vollständig, Symbole und Texte bei Einführung geeigneter Hilfspunkte bis auf die Symbolnummer bzw. Textkette festlegen.

2. Werterecords zur Identifikation von Bildatomen oder Grundelementen können durch Aufmessung eines Punktes bestimmt werden, der im Falle von Polygonen, Kurven oder Kreisbögen dem Linienzug, im Falle von Symbolen und Texten der Bezugspunktumgebung angehört.

3. Werterecords zur Transformation von Bildelementen können durch Aufmessen entsprechender Meßmarken bzw. Skalenpunkte in einem Transformationsmenü erzeugt werden. Auf diese Weise kann eine Drehung, Vergrößerung oder Verkleinerung sowie eine achsenparallele Spiegelung, Stauchung oder Streckung vorgenommen werden. Die Elemente können über ihren Namen oder bei Grundelementen mittels Punkteidentifikation festgelegt werden. Insbesondere können häufig benötigte Grundelemente aus Basiselementespeichern über zugeordnete Basiselementemenüs leicht entnommen werden.

4. Namenrecords können bei Verwendung eines Namenmenüs einfach in einer Standardform $X_1, X_2, X_3 \dots$ durch Messung festgelegt werden.

5. Steuerrecords können über Meßtasten eines Steuermenüs schnell generiert werden. Die von AUTOZEI benötigte Steuerkartenstruktur wird aus den Tastenkoordinaten intern erzeugt.

Unter Nutzung dieser Möglichkeiten kann am DIGITRON ein Lochstreifen erzeugt werden, der alle Manipulationsanforderungen verschlüsselt enthält. In einem zweiten Arbeitsgang werden nun diese Informationen vom Rechner direkt mittels AUTOZEI verarbeitet. Dafür sind nur einige Steuer- und Datenkarten bereitzustellen, die insbesondere Hinweise über auszublenkende Lochstreifenpassagen oder die gewünschte Meßwertrundung mit aufnehmen können. Das Ergebnis ist i. a. ein manipuliertes Datenmodell des Bildes, das ausgezeichnet oder in anderer Weise weiterverarbeitet werden kann.

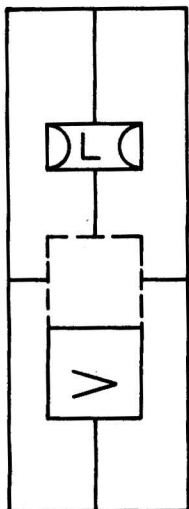


Abb. 1: Handzeichnung

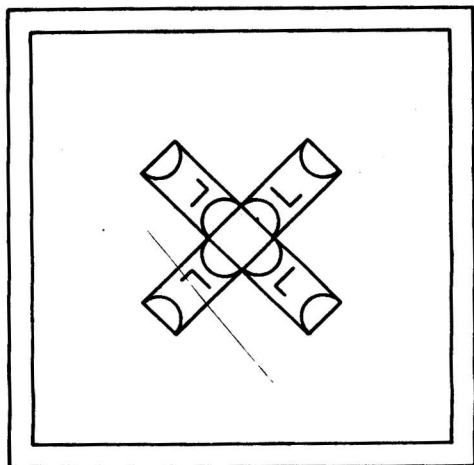
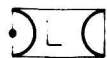


Abb. 3: Ergebnisbild



P1=

P2=

P3=

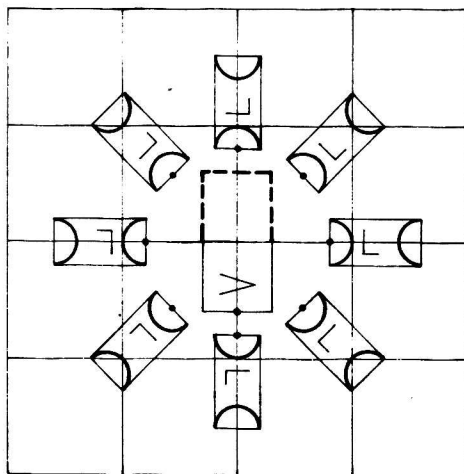


Abb. 2: Menü und manipulationsfähiges Bild

Ein Anschauungsbeispiel soll im folgenden kurz eine mögliche Einsatzvariante des Systems illustrieren. Dabei soll folgende Serie von Manipulationen durchlaufen werden:

1. Aus der Handzeichnung (Abb. 1) sind die Elemente L und V herauszulösen und in einer Normallage in einen Basiselementespeicher zu bringen. Das Element V ist dabei um 90 Grad zu drehen und auf die Hälfte zu verkleinern. Zusätzlich soll ein Routineatom abgespeichert werden, mit dessen Hilfe durch Angabe dreier Punkte P1, P2, P3 ein spezielles regelmäßiges Raster erzeugt werden kann. Das zugehörige Basiselementemenu Z2 ist auszuzeichnen, wobei die Bezugspunkte der Elemente durch Kreise zu kennzeichnen sind.

2. Mit Hilfe des Menüs (s. Abb. 2) und des Transformationsmenüs soll ein manipulationsfähiges Bild (Abb. 2) erzeugt, über Zeichentisch ausgegeben und zwecks Wiederverwendung abgespeichert werden.

3. Aus dem Bild (Abb. 2) sollen nach einer Abänderung Elemente in ein gleichfalls schon abgespeichert vorliegendes Bild übernommen werden, das der Einfachheit halber nur aus zwei Begrenzungsrahmen besteht. Das Ergebnisbild (Abb. 3) ist auszuzeichnen.

Das vorgestellte System wurde mit Einschluß des FORTRAN-Pakets AUTOZEI im Rechenzentrum des VEB Kombinat Kraftwerksanlagenbau Berlin entwickelt. Wichtige Anregungen boten dabei die DIGRA-Systeme, die an der Wilhelm-Pieck-Universität Rostock unter Leitung von Prof. Kieseewetter entstanden. Das System enthält offenbar gewisse Elemente einer Bildschirmarbeit, wobei die Ein- und Ausgabefunktion eines aktiven Displays von zwei getrennten Geräten, nämlich Zeichentisch und Meßtisch, wahrgenommen wird. Die von AUTOZEI verwendeten Zeichenroutinen sind mit den Grundroutinen des neuen Zeichenpakets für ESER-Rechner PAD-ES kompatibel. Nähere Angaben zum Programmpaket AUTOZEI sind im 4. Beiheft der Zeitschrift "Rechentechnik Datenverarbeitung" des Jahrgangs 1975 enthalten.

Anschrift des Verfassers:

Dr. Volker Dobrowolny
VEB Kombinat Kraftwerksanlagenbau
Rechenzentrum
DDR - 1017 Berlin, Beimlerstr. 91-94

Fehlerabschätzung bei der Interpolation von Flächenstücken

Der Darstellung von Flächen auf Bildschirmgeräten und Zeichenmaschinen kommt durch deren umfangreichen Informationsgehalt eine immer größer werdende Bedeutung zu.

Es gibt bekanntlich viele Möglichkeiten, um Flächen zu beschreiben. Vorzugsweise arbeitet man mit expliziten Definitionen der Form

$$\begin{aligned} x &= x(u, v), \\ (1) \quad y &= y(u, v), \\ z &= z(u, v), \end{aligned}$$

wobei die Parameter in bestimmten Bereichen einer u, v -Ebene variieren. Die Parameterdarstellung mit zweimal stetig differenzierbaren Funktionen x, y, z bildet die Grundlage für die differentialgeometrische Beschreibung von Flächen. In diesem Zusammenhang spricht man auch von Gaußschen Flächen. Für spezielle Flächentypen (Regelflächen, Minimalflächen, algebraische Flächen usw.) arbeitet man mit Beschreibungen, die dem betreffenden Typ angepaßt sind und die neben kartesischen Koordinaten noch andere Koordinatensysteme benutzen.

Für die Digitalgraphik benötigt man Flächenbeschreibungen, die ausgehend von einer kleinen Anzahl von Grundtypen möglichst viele Flächentypen erfassen. Dabei ist es notwendig, allgemeine Flächen durch einfache Elemente zu interpolieren bzw. zu approximieren. Die einfachste Variante wird darin bestehen, daß man Flächen durch Parzellierung mit Dreiecken interpoliert bzw. approximiert. Dabei erhält man aber keine glatten Übergänge von einer Zelle zu den benachbarten Zellen. Aus diesem Grunde untersuchte Steven A. Coons (s./2/) die Möglichkeit, allgemeine Flächen durch Flächenstücke glatt zu interpolieren. Steven A. Coons bezeichnete die Flächenstücke als Flecken (engl. patch). Neben den bekanntesten Flächendarstellungen von Coons 2

und Bezier (s. /1/) wurden von mehreren Autoren für spezielle Probleme Verfahren und Methoden, die den entsprechenden Problemen in Definition, Genauigkeit und Verwendbarkeit angepaßt sind, entwickelt.

In /3/ finden wir eine umfangreiche Literaturzusammenstellung dazu.

Die Interpolation eines Flächenstückes durch Spline-Funktionen bezeichnen wir als Hermite-Entwicklung.

Wir definieren die bekannten Mischfunktionen n -ter Ordnung $f_{ik}(t)$ für $i=0, 1, \dots, n$ und $k=0, 1$ für feste Werte von $n=0, 1, \dots$ über dem Intervall $[t_0, t_1]$ durch folgende Eigenschaften:

$$(2) \quad f_{ik}(t) = 0 \text{ für } t \in [t_0, t_1]$$

$$(2') \quad f_{ik}(t) \text{ sei } n\text{-mal stetig differenzierbar}$$

$$(2'') \quad \left. \frac{d^j}{dt^j} f_{ik}(t) \right|_{t=t_1} = \begin{cases} 1 & \text{für } j=i \text{ und } k=1 \\ 0 & \text{sonst.} \end{cases}$$

Die Forderungen (2) - (2'') lassen sich durch Polynomansätze $(2n+1)$ -ten Grades realisieren (s. 3, 4, 5). Intervallfunktionen lassen sich mit bekannten Ansätzen in Verbindung bringen, dazu seien nur die Beziehungen zu den Spline-Funktionen erwähnt (s. /5/).

Wir bezeichnen das zu interpolierende Flächenstück mit

$$(3) \quad F(u, v) = \{X(u, v), Y(u, v), Z(u, v)\},$$

das über dem Bereich $u_0 \leq u \leq u_1$ und $v_0 \leq v \leq v_1$ definiert sei. Um $F(u, v)$ nach Hermite entwickeln zu können, muß $F(u, v)$ ausreichend oft stetig differenzierbar sein. Für die partiellen Ableitungen von $F(u, v)$ führen wir folgende abkürzende Schreibweise ein:

$$(4) \quad \frac{\partial^{i+j}}{\partial u^i \partial v^j} F(u, v) = ! F_{ij}^{\cdot}(u, v) \quad 1$$

Somit können wir $F(u, v)$ nach Hermite entwickeln.

1 $A! = B$ bzw. $B = !A$ bedeutet, daß A durch B definiert wird

$$(5) \quad F(u, v) = \sum_{k, \ell=0,1} \sum_{i,j=0}^n F_{ij}(u_k, v_\ell) f_{ik}(u) f_{jl}(v) + R^n(u', v')$$

mit $u_0 \leq u' \leq u_1$ und $v_0 \leq v' \leq v_1$.

$$(6) \quad P(u, v) = \sum_{k, \ell=0,1} \sum_{i,j=0}^n F_{ij}(u_k, v_\ell) f_{ik}(u) f_{jl}(v)$$

Aus der Definition (6) folgt, daß $P(u, v)$ bis auf das Restglied $R^n(u, v)$ die Fläche $F(u, v)$ darstellt.

Mit der Hermite-Entwicklung⁽⁵⁾ ist uns ein einfaches Mittel gegeben, um komplizierte Flächen darstellen zu können. Wir können eine gegebene Fläche in Flächenstücke unterteilen und diese Flächenstücke nach⁽⁵⁾ entwickeln. Die Flächenstücke lassen sich stetig (d. h. die Randkurven der angrenzenden Flächenstücke sind gleich) und glatt (d. h. gemeinsame Ableitung beliebig hoher Ordnung quer zur Randkurve) zusammensetzen. Den Beweis finden wir hierzu in /3/. Für $n=1$ ergeben sich aus⁽⁶⁾ die Flächen-darstellungen von Coons, s. /2/, bei denen ein Flächenstück durch bikubische Funktionen interpoliert wird.

Von großem Interesse bei der stückweisen Interpolation oder Approximation von Flächen ist die notwendige Anzahl von Stützstellen, die bei der Interpolation verwendet werden müssen, um eine gewisse Approximations- oder Interpolationsgenauigkeit zu erhalten. Fehlerformel für die eindimensionale Hermite-Entwicklung kann man nach bekannten Techniken aus dem Satz von Rolle herleiten (s. /5/). Das ist für den mehrdimensionalen Fall nach dieser Technik nicht möglich. Um nach folgender Methode eine Fehlerabschätzung geben zu können, müssen wir fordern, daß die partiellen Ableitungen von $F(u, v)$ bis zu der partiellen Ableitung $F_{2n+2, 2n+2}(u, v)$ existieren und stetig sind. Es gilt nach (5) und (6):

$$(7) \quad F(u, v) = P(u, v) + R^n(u', v')$$

mit $u_0 \leq u' \leq u_1$ und $v_0 \leq v' \leq v_1$. Wir betrachten den Fehler

$$(8) \quad \|R^n(u, v)\| \leq \|F(u, v) - P(u, v)\|$$

mit

$$(9) \quad \|F(u, v)\| = \max_{(u, v)} |F(u, v)|$$

Zur Abschätzung von $\|R^n(u,v)\|$ führen wir Hilfsfunktionen $(PUV(u,v), PVU(u,v))$ durch mehrere Eigenschaften ein, die eine Fehlerabschätzung ermöglichen.

$$(10) \quad RUV^n(u,v) = F(u,v) - PUV(u,v) + PUV(u,v) - P(u,v)$$

$$(11) \quad RVU^n(u,v) = F(u,v) - PVU(u,v) + PVU(u,v) - P(u,v)$$

$$(12) \quad 1. \quad PUV_{i0}(u_k, v) = F_{i0}(u_k, v)$$

$$2. \quad PUV_{0l}(u, v_l) = P_{0l}(u, v_l)$$

$$(13) \quad 1. \quad PVU_{0l}(u, v_l) = F_{0l}(u, v_l)$$

$$2. \quad PVU_{i0}(u_k, v) = P_{i0}(u_k, v)$$

für $i, l = 0, 1, 2, \dots, n$ und $k, l = 0, 1$.

Es gilt:

$$(14) \quad PUV_{ij}(u, v) = \frac{\partial^{i+j}}{\partial u^i \partial v^j} PUV(u, v)$$

Folgende Hilfsfunktionen genügen (12) bzw. (13)

$$(12') \quad PUV(u, v) = \sum_{i=0}^n \sum_{k=0}^1 f_{ik}(u) F_{i0}(u_k, v)$$

$$(13') \quad PVU(u, v) = \sum_{j=0}^n \sum_{l=0}^1 f_{jl}(v) F_{0l}(u, v_l)$$

Die Funktionen $F_{i0}(u_k, v)$ bzw. $F_{0l}(u, v_l)$ mit $i, l = 0, 1, \dots, n$ und $k, l = 0, 1$ sind die Funktionen der Ableitungen der Randkurven quer zum Rand. Es lassen sich (10) und (11) betragsmäßig abschätzen

$$(15) \quad \|RUV^n\| \leq \|F - PUV\| + \|PUV - P\|$$

$$(16) \quad \|RVU^n\| \leq \|F - PVU\| + \|PVU - P\|$$

Das Minimum von (15) und (16) ist eine Abschätzung für den Fehler (8).

$$(17) \quad \|R^n(u, v)\| \leq \min(\|RUV^n\|, \|RVU^n\|)$$

Betrachten wir (15) und führen folgende Abkürzungen ein

$$(18) \quad R1 = \|F(u, v) - PUV(u, v)\|$$

$$(19) \quad R2 = \|PUV(u, v) - P(u, v)\|$$

Zur Abschätzung von R_1 führen wir folgende Hilfsfunktion für $v=v^x$ fest, aber beliebig und u variabel ein

$$(20) \quad H_1(u, v^x) = F(u, v^x) - \text{PUV}(u, v^x) - c (u - u_0)^{n+1} (u - u_1)^{n+1}$$

$H_1(u, v^x)$ hat $2(n+1)$ Nullstellen bzgl. u (durch Wahl von PUV). Eine weitere Nullstelle läßt sich durch Wahl von $c = c(u', v^x)$ finden:

$$(21) \quad c = c(u', v^x) = \frac{F(u', v^x) - \text{PUV}(u', v^x)}{(u' - u_0)^{n+1} (u' - u_1)^{n+1}}$$

Damit hat $H_1(u, v^x)$ $(2n+3)$ Nullstellen im Intervall $[u_0, u_1]$. Aus dem Satz von Rolle läßt sich herleiten, daß die $(2n+2)$ -te Ableitung von $H_1(u, v^x)$ eine Nullstelle im Intervall $[u_0, u_1]$ hat. Es sei vermerkt, daß $H_1(u, v^x)$ eine Funktion nur von u ist, da v^x fest gewählt ist.

$$(22) \quad H_{1, 2n+2, 0}(u^0, v^x) = 0 \text{ mit } u_0 \leq u \leq u_1.$$

Die Funktion $\text{PUV}(u, v^x)$ ist eine Funktion $(2n+1)$ -ten Grades in u , d. h.

$$\text{PUV}_{2n+2, 0}(u, v^x) = 0.$$

Somit folgt aus (20)

$$H_{1, 2n+2, 0}(u^0, v^x) = 0 = F_{2n+2, 0}(u^0, v^x) - 0 - c (2n+2)!$$

Aus (21) und (22) folgt

$$(23) \quad F(u', v^x) - \text{PUV}(u', v^x) = \frac{F_{2n+2, 0}(u^0, v^x)}{(2n+2)!} (u' - u_0)^{n+1} (u' - u_1)^{n+1}$$

Da die Wahl der Stelle u' beliebig ist, können wir u' durch u ersetzen. Wir führen folgende Abkürzungen für die Intervalllängen ein

$$(24) \quad h_1 = u_1 - u_0, \quad k_1 = v_1 - v_0.$$

Nach (24) gilt $u = u_0 + s h_1$ für $0 \leq s \leq 1$, es läßt sich abschätzen $|s(s-1)| \leq 1/4$, damit läßt sich (23) betragsmäßig abschätzen:

$$(25) \quad |F(u', v^x) - \text{PUV}(u', v^x)| \leq \frac{|F_{2n+2, 0}(u^0, v^x)|}{(2n+2)! 4^{2n+2}} h_1^{2n+2}$$

Wir bestimmen das Maximum über $F_{2n+2, 0}(u, v)$ für alle u und für jedes feste $v=v^x$ aus den Intervallen $[u_0, u_1]$ und $[v_0, v_1]$. Falls das Maximum von $F_{2n+2, 0}(u, v)$ existiert, gilt

$$(26) \quad R1 = \|F - PUV\| \leq \|F_{2n+2,0}(u,v)\| \cdot \frac{h^{2n+2}}{(2n+2)! 4^{2n+2}}$$

In analoger Vorgehensweise läßt sich R2 abschätzen und eine Abschätzung für $\|RVU^n\|$ (s.16) finden.

Wir erhalten

$$(27) \quad \|RUV^n\| \leq \frac{\|F_{2n+2,0}\| h_1^{2n+2} + \sum_{i=0}^n \sum_{k=0}^1 \|f_{ik}\| \|F_{i,2n+2}(u_k,v)\| k_1^{2n+2}}{(2n+2)! 4^{2n+2}}$$

$$(28) \quad \|RUV^n\| \leq \frac{\|F_{0,2n+2}\| k_1^{2n+2} + \sum_{j=0}^n \sum_{l=0}^1 \|f_{jl}\| \|F_{2n+2,j}(u,v_l)\| h_1^{2n+2}}{(2n+2)! 4^{2n+2}}$$

Für $n=1$ erhalten wir aus (27) bzw. (28) die Formeln, die die Güte der Interpolation einer Fläche durch Coon'sche Flächenstücke bewertet.

Die Fehlerformeln lassen sich auf unterschiedliche numerische Fragestellungen anwenden, so z. B. auf Quadraturformeln (s./3/).

Literatur

- /1/ Bezier, P.: Definition numérique des courbes et surfaces. Automatisation', No.12, S. 625 ff (1966).
- /2/ Coons, S. A.: Surfaces for Computer-Aided Design of Space Forms. MAC-TR-41, MIT, 1967.
- /3/ Fehlauer, K -U.: Interpolation und Darstellung von Flächenstücken im Zusammenhang mit dem DIGRA 73-System. Dissertation, Rostock, 1976.
- /4/ Kiesewetter, H.: Interpolation mit Intervallfunktionen und einige Anwendungen. Wiss. Zeitschrift der Univ. Rostock, Heft 5/6 (1971).

/5/ Kiesewetter, H.; Maeß, G.: Elementare Methoden der numerischen Mathematik,
Akademie Verlag Berlin, 1974.

Anschrift des Verfassers:

Dr. Klaus-Uwe Fehlaue
Wilhelm-Pieck-Universität Rostock
Rechenzentrum
DDR - 25 Rostock
Albert-Einstein-Str.

ORTEP – ein Zeichenprogramm zur Darstellung von Kristallstrukturen –

1. Allgemeines

ORTEP ist ein FORTRAN-Programmpaket zur graphischen Darstellung von Kristallstrukturen. Es kann als ein Hilfsmittel angesehen werden, das die in zahlreichen Versuchen und Rechnungen erlangten Informationen im Forschungsprozeß der Röntgenkristallstrukturanalyse in stark komprimierter Form darstellt. Für die Strukturforschung wird damit eine höhere Qualität der Ergebnisbereitstellung erreicht, da mit den konventionellen Methoden derartige komplexe Darstellungen einen praktisch nicht realisierbaren Arbeitsaufwand zur Folge hätten. Darüber hinaus werden durch die Nutzung von ORTEP manuelle Arbeiten eingespart.

Der Übergang von der 3-dimensionalen zur 2-dimensionalen Information erfolgt in ORTEP vollautomatisch. Jedoch sind für den Aufbau des 3-dimensionalen Modells Eingriffe des Bearbeiters notwendig. Der Nachteil dieser Vorgehensweise besteht darin, daß Kenntnisse auf dem Fachgebiet der Strukturforschung erforderlich sind. Demgegenüber ist der Bildaufbau flexibler und dem Kristallografen stehen mehr Varianten zur Verfügung als bei einer vollautomatischen Verarbeitung.

Auch bei diesem Anwendungsfall der graphischen Ausgabe stellen Überlagerungen ein wichtiges Problem dar. Verdeckte Teile werden vollständig gezeichnet und müssen anschließend manuell entfernt werden. Beschriftungen können automatisch positioniert werden, jedoch erscheint es bei nichtstereoskopischen Darstellungen mitunter günstiger, dies von Hand vorzunehmen, da man auf diese Weise ästhetische Ansprüche besser berücksichtigen kann.

Es besteht für alle Kristallstruktur-Darstellungen das Problem, eine geeignete Auswahl für die graphische Ausgabe zu treffen. Eine Struktur umfaßt eine nahezu unbegrenzte Anzahl von Atomen. Deshalb werden oft mehrere Einheitszellen ausgeblendet, so daß neben den Atomlagen (Molekülanordnungen) auch die periodischen Eigenschaften klar ersichtlich werden.

2. Beschreibung einer Figur

Für die Beschreibung einer Struktur sind Angaben zu den Zellen, den Symmetrien und den Atomen notwendig. Daraus ergeben sich Punktkoordinaten, die durch einen Graphen miteinander verbunden sind. Neben der eigentlichen graphischen Ausgabe können zusätzliche Informationen wie z. B. Angaben zu Atomlagen, inneratomaren Abständen, Bindungswinkeln oder Hauptachsen der thermischen Bewegung erhalten werden.

Prinzipiell werden die Kristallstrukturen nach dem "ball-and-stick"-Typ aufgebaut. Die Atome (balls) können durch Kreise, Kugeln oder anisotrope Schwingungsellipsoide beschrieben werden, die Bindungen (sticks) durch Striche oder "Röhren". Beschriftungen (z. B. für Atombezeichnungen oder Bindungsabstände) vervollständigen die Aussage. Im Programm erfolgt kein Stiftwechsel, sondern ausgewählte Linien werden durch mehrmaliges Nebeneinanderzeichnen hervorgehoben. So können Ungenauigkeiten beim Justieren verschiedener Zeichenstifte umgangen werden.

Als Projektionsmethoden bietet ORTEP die Parallelprojektion, die Zentralprojektion oder ein stereoskopisches Bildpaar an. Eine solche Stereoabbildung setzt sich aus zwei Zentralprojektionen zusammen, deren Projektionszentren jeweils um einen bestimmten Winkel gegenüber einer normalen Projektion gedreht werden. Es soll im Folgenden zunächst die prinzipielle Zusammenstellung einer nichtstereoskopischen Darstellung mit ORTEP beschrieben werden.

Nach der Eingabe der erforderlichen Daten zur Beschreibung der zu zeichnenden Struktur – das sind für das vorliegende Beispiel 47 Lochkarten – wird das Zeichenprogrammpaket initialisiert sowie Zeichengrenzen und ein Betrachtungsabstand definiert. Für unterschiedliche Darstellungsweisen werden die gewünschten Atome ausgewählt und die Figur durch Rotation um die Achsen eines Bezugssystems geeignet orientiert. Eine Skalierung gestattet das Positionieren relativ zur Zeichenfläche. Die eigentliche Darstellung erfolgt

im Anschluß an diese Vorarbeiten durch das Zeichnen der Kreise, Kugeln oder Ellipsoide, der Striche oder "Stäbe" und der Markierungen. Diese angeführte Reihenfolge ist allerdings nicht zwingend, da die Abbildung durch eine beliebige Auswahl der beschriebenen Elemente modifiziert werden kann.

Es soll jedoch an dieser Stelle auf eine Besonderheit dieses Anwendungsfalles der Digitalgraphik aufmerksam gemacht werden. In der überwiegenden Mehrzahl der Fälle beziehen sich die Eingabedaten bzw. allgemein die Probleme auf ein kartesisches Koordinatensystem. In der Röntgenkristallstrukturanalyse dagegen wird mit einem schiefwinkligen Koordinatensystem, das sich auf die Achsen der Kristalle bezieht, gearbeitet.

Viele Rechnungen und die Operationen für den Bildaufbau in ORTEP beziehen sich auf Grund des vorliegenden triclinen Kristallsystems auf schiefwinklige Koordinatensysteme. Jedoch erfordern andere Schritte die Einführung orthonormaler Basisvektoren (kartesisches Koordinatensystem). Deshalb kann ein geeignetes Koordinatensystem definiert werden, auf das die oben beschriebene Darstellungsmethode Bezug nehmen kann. Von ORTEP werden jedoch zwei kartesische Koordinatensysteme definiert - nämlich ein Bezugssystem, das für alle Operationen bezüglich der Struktur außer dem Zeichnen benutzt wird, und ein Arbeitssystem, das ausschließlich für die eigentliche graphische Ausgabe zuständig ist.

Im Falle einfacher Parallel- oder Zentralprojektionen stimmen beide Systeme überein. Für ein Stereo-Bildpaar wird für die Abbildungen - für das rechte und das linke Auge - das Arbeitssystem gegenüber dem Bezugssystem um eine Achse des letzteren gedreht. Um ein exaktes stereoskopisches Bildpaar zu erhalten, müssen während der "Programmierung der Zeichnung" Entscheidungen getroffen werden, die sich dann auf das Bezugssystem beziehen, z. B. die Wahl einer geeigneten Blickrichtung. Da das Arbeitssystem relativ zum Bezugssystem erklärt wird, ist es damit automatisch auch davon betroffen. Der Nutzer kann beide Systeme definieren und orientieren.

Damit ist bereits ein Hilfsmittel für die Erzeugung stereoskopischer Bildpaare genannt. Ein weiteres wird dadurch gegeben, daß eine beliebige sequentielle Anweisungsfolge gespeichert und zu einem späteren Zeitpunkt beliebig oft abgearbeitet werden kann.

3. Zur Implementierung

Das Programmpaket ORTEP bezieht sich mit seinem Konzept auf eine Zeichenmaschine. Dabei wird eine modifizierte Form der CALCOMP-Basis-Zeichensoftware verwendet. Von dieser Software werden jedoch nur die folgenden Programme benutzt:

- | | | |
|--------|---|--|
| PLOTS | - | zum Initialisieren des Zeichenpaketes |
| PLOT | - | zum Verfahren mit gehobenem/gesenktem Stift zu einem Zielpunkt und zur Nullpunktdefinition |
| SYMBOL | - | zur Darstellung alphanumerischer Symbole |
| NUMBER | - | zur Darstellung von Zahlenwerten. |

ORTEP greift also nur auf Geradenelemente zurück und nutzt die gegebenen Hardware-Funktionen des DIGIGRAF nicht aus. Daß das kein erheblicher Nachteil ist, zeigen die Zeichnungen.

Derartige Strukturdarstellungen sind international üblich. Im Rahmen des Programmaustausches der Kristallografen wurde ORTEP übernommen und an die BESM6 angepaßt. (s./1/) Es beschreibt ein ESER-Magnetband mit den Steuerdaten für den DIGIGRAF 1612 mit der Steuerelektronik DAPOS D-3G. Ausgezeichnet werden die Strukturen im on-line-Betrieb durch einen ES 1020, der als Satellitenrechner für die BESM6 arbeitet (s./2/).

Dabei einer stereoskopischen Darstellung mittlerer Größe ca. 8000-10000 Koordinatenpunkte anfallen, ist eine Steuerung des Zeichenautomaten im off-line-Betrieb über Lochstreifen sehr unökonomisch, zumal das Verfahren in seiner Anlage iterativ ist, d. h. ein optimaler Betrachtungswinkel und eine günstige Beschriftung erst ermittelt werden müssen.

Im Ergebnis entstehen Zeichnungen, die (i.a. mit wenig manueller Nacharbeit) eine für Veröffentlichungen ausreichende Qualität aufweisen (s. Abb. 1). Im Normalfall erfolgt dann noch eine Verkleinerung auf phototechnischem Wege.

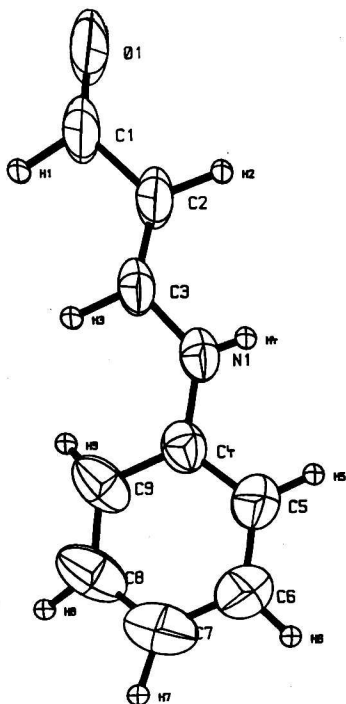
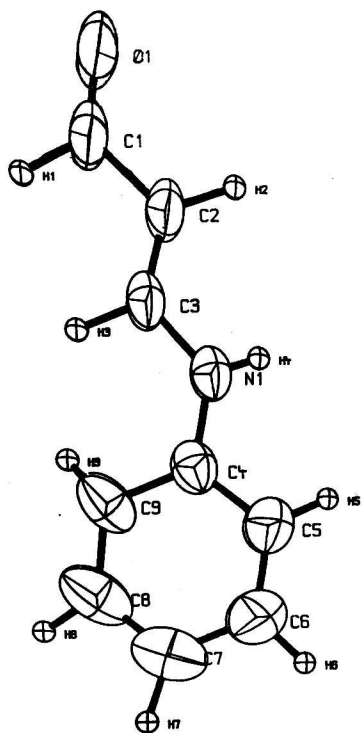


Abb. 1

4. Entwicklungstendenzen

International existieren bereits vervollkommneter Varianten, die einerseits neue Methoden berücksichtigen - Bearbeitung des Problems der Überdeckungen und das vollautomatische Erstellen der Abbildungen - und andererseits eine weiterentwickelte Gerätetechnik - aktiver graphischer Bildschirm - nutzen.

Darüber hinaus muß festgestellt werden, daß nicht mehr die zeichnerische Darstellung eines Modells im Vordergrund steht, sondern das Einwirken auf methodische Prozesse in der Arbeit der Röntgenkristallstrukturanalyse. Ein Modell wird mit experimentellen Daten - meist in anderer Darstellungsform wie Isolinien gegeben - überlagert und in einem aktiven Prozeß so abgewandelt, daß es sich der Realität nähert. Diese Methodik erfordert einerseits eine schnelle und leistungsfähige Rechentechnik und andererseits eine interaktive Arbeitsweise, da das Ziel mit automatischen Zeichenanlagen nicht zu erreichen ist.

Schnelle und leistungsfähige Rechner sind im Rahmen ESER in Sicht. Ebenso sind graphische Bildschirmgeräte angekündigt. Es bleibt die Frage nach der Software auf dieser gerätetechnischen Basis. Da eine solche Entwicklung eine gewisse Zeit benötigt, sollte man bereits heute diese Problemstellung berücksichtigen, zumal sich Anwendungsfälle in unterschiedlichen Fachgebieten bereits deutlich abzeichnen.

Literatur

- /1/ Johnson, C. K.: ORTEP A FORTRAN Thermal-Ellipsoid Plot Program for Crystall Structure Illustrations. Oak Ridge, Tennessee 1965.
- /2/ Fischer, M., Schulz, R., Teegler, U., Weiß, H.-G., Wiedemann, R.: DIGIGRAF 1612 DAPOS D-3G - ein Zeichengerät des ESER.
ZfR-Informationen ZfR-C-76. 12, Berlin 1976.

Anschrift des Verfassers:

Dipl.-Math. Manfred Fischer
Akademie der Wissenschaften der DDR
Zentrum für Rechentechnik
1199 Berlin
Rudower Chaussee 5

Zur algorithmischen Notation von Konstruktionsbeschreibungen als Vorstufe der Digitalgraphik

Der schnellen und breiten Einführung der Digitalgraphik stehen die begrenzte Zahl von Geräten und erhebliche Kosten entgegen. Beide Faktoren werden in Zukunft an Bedeutung verlieren, so daß bereits heute dem algorithmischen Denken bei geometrischen Konstruktionen als Basis vieler Anwendungen Aufmerksamkeit gewidmet werden sollte.

Im Beitrag wird eine "Programmiersprache" besprochen, die - als Vorstufe der Digitalgraphik - einerseits von konventionellen Konstruktionsbeschreibungen und andererseits von syntaktischen Elementen universeller Programmiersprachen ausgeht. Sie ist vorzugsweise auf eine manuelle Interpretation ausgerichtet, kann aber auch als Vorstufe der Codierung in einer implementierten graphischen Sprache betrachtet werden.

Wesentliches Ziel dieser Sprache ist es (neben der Vorbereitung der Digitalgraphik), die Ausführung geometrischer Konstruktionsaufgaben zu rationalisieren, indem aus der in Abhandlungen und in Lehrbüchern üblichen Verflechtung von Erläuterungen, Durchführung an einem Beispiel und Beweisen der eigentliche Konstruktionsweg herausgelöst wird.

Sie ist vor allem für den zeichnenden Ingenieur und Wissenschaftler gedacht, der graphische Verfahren routinemäßig benutzt. Auch im pädagogischen Bereich läßt sie sich vorteilhaft sowohl zur Beschreibung von Konstruktionen als auch zur Schulung des systematischen logischen Denkens verwenden.

1. Einführungsbeispiel

An der Konstruktion der Mittelsenkrechten (Abb. 1) und ihrer Verwendung für die Bestimmung eines Kreises durch 3 Punkte (Abb. 2) wollen wir die wichtigsten Eigenschaften der Konstruktionsbeschreibungssprache kennenlernen.

Die konventionelle Beschreibung würde lauten:

Es ist die Mittelsenkrechte m zwischen den Punkten A und B zu konstruieren. Wir nehmen eine beliebige Strecke r in den Zirkel und schlagen je einen Kreis um A und B . Die Verbindungsgerade der Schnittpunkte C_1 und C_2 dieser Kreise a und b ist die gesuchte Mittelsenkrechte.

Zum Zeichnen eines Kreises k durch die Punkte P_1 , P_2 und P_3 errichten wir zwei Mittelsenkrechten zwischen den Punktepaaren P_1P_2 , P_1P_3 oder P_2P_3 . Ihr Schnittpunkt ist der Mittelpunkt M des gesuchten Kreises.

In der Konstruktionsbeschreibungssprache, die man KONZIL nennen könnte (Konstruktion mit Zirkel und Lineal), würde die Beschreibung lauten:

1. Konstruktion einer Mittelsenkrechten

MITTELS: GEG (A, B) / A, B = gegebene Punkte/

GES (m); / m = Mittelsenkrechte/

r = STR BEL; / $r \approx$ STR A, B /

a = KRS UM A RAD r ;

b = KRS UM B RAD r ;

$C_{1,2}$ = PKT a, b ;

m = GER C_1, C_2 ;

2. Konstruktion eines Kreises durch 3 Punkte P_i

KREIS 3: GEG (P_1, P_2, P_3)

GES (k, M); / k = Kreis, M = Mittelpunkt/

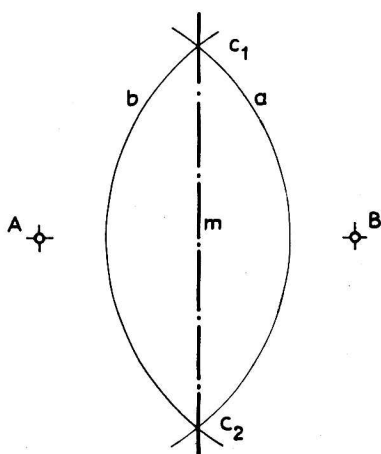


Abb. 1: Mittelsenkrechte

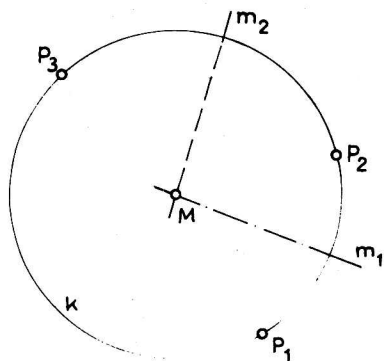


Abb. 2: Kreis durch 3 Punkte

2.1. Bestimmung von 2 Mittelsenkrechten

$$m_1 = \text{MITTELS } (P_1, P_2);$$

$$m_2 = \text{MITTELS } (P_2, P_3);$$

oder

$$m_3 = \text{MITTELS } (P_1, P_3);$$

2.2. Bestimmung des Kreises

$$M = \text{PKT } m_1, m_2, m_3;$$

$$k = \text{KRS } P_1 \text{ oder } P_2 \text{ oder } P_3 \text{ UM } M;$$

Aus dem Einführungsbeispiel ersehen wir folgende Eigenschaften der Konstruktionsbeschreibungssprache:

- Orientierung auf geometrische Konstruktionen mit Zirkel und Lineal,
- Verbindung weniger mnemonisch gebildeter Schlüsselworte und Wahrung normalsprachlicher Formulierungen,
- Verwendung gebräuchlicher syntaktischer Mittel höherer Programmiersprachen für die einzelnen Anweisungen und die Darstellung der Struktur des geometrischen Algorithmus,
- Verwendung von streng formalisierten und relativ freien Formulierungen nebeneinander,
- Prozedurkonzept für einen hierarchischen Aufbau höherer Operationen,
- Einbeziehung von Kommentaren (Zwischenüberschriften) zur Strukturierung.

Die syntaktischen Elemente werden den Programmiersprachen FORTRAN, ALGOL, PL/I sowie EXAPT unter starker Beachtung der Prinzipien der strukturierten Programmierung entlehnt.

Die konventionelle Form ist zwar etwas kürzer, betrachtet man sie aber unter dem Aspekt, daß der Konstruktionsgang normalerweise unbekannt ist und daß er anhand der Beschreibung nachvollzogen werden soll, so werden einige Nachteile offenkundig, z. B. erhält der Kreis k zu einem Zeitpunkt den Namen k , an dem er noch nicht vorhanden ist, dafür aber wird er nie explizit gezeichnet, denn die Beschreibung endet mit dem Auffinden des Mittelpunktes U . Das Herausfinden des Konstruktionsganges aus der

Beschreibung ist im Beispiel relativ leicht, kann aber schwierig werden, wenn er mit Ableitungen, Beweisen und Erläuterungen verflochten ist und/oder wenn technische Kräfte mit Routineaufgaben betraut werden sollen. Weiter sind die nicht seltenen Verweise der Art "... nun verfahren wir weiter wie im Abschnitt xxx dargelegt ..." oder "... nach der allgemein bekannten Methode ..." mitunter sehr unangenehm für das effektive Nachvollziehen einer Konstruktion.

2. Beschreibung der Sprache

Das Lesen von in KONZIL geschriebenen Konstruktionsbeschreibungen bedarf kaum umfangreicher Erläuterungen zur Sprache. Für das Schreiben sind folgende Regeln und Empfehlungen zu beachten:

2.1. Die Konstruktionsbeschreibung (Prozedur) wird in Einzelschritte zerlegt, die in einzelnen Anweisungen formuliert und durch gliedernde Kommentare erläutert werden.

Logisch-dezimale Gliederung der Programmabschnitte, zeilenweises Schreiben der Anweisungen und Einrückungen erleichtern die Übersicht.

Die Konstruktionsbeschreibung sollte durch ein Bild mit den in der Prozedur verwendeten Variablen und einem Hinweis auf verwendete Sätze, Ableitungen, usw. ergänzt werden.

2.2. Die formale Konstruktionsbeschreibung (Prozedur) beginnt mit einer Eröffnungsanweisung im Format

```
prozedurname:  GEG (argumentliste)
               GES (funktionsliste);
```

bestehend aus

- dem Namen der Konstruktionsbeschreibung 'prozedurname',
- einem Doppelpunkt ':' als Trennzeichen,
- dem Schlüsselwort 'GEG' für 'gegeben' und der Liste der gegebenen Größen (Argumente),
- dem Schlüsselwort 'GES' für 'gesucht' und der Liste der gesuchten Größen (Funktionen),
- dem abschließenden Semikolon ';'.

Die Schlüsselworte 'GEG' und 'GES' können durch 'ARG' und 'FKT' (für 'Argumente' und 'Funktionen') ersetzt werden.

2.3. Jeder Konstruktionsschritt wird in einer Ergibtanweisung im Format

variablenname(n) = typenbezeichnungvariablenname (n) relationsabgabe (n);

beschrieben, sie besteht aus

- dem Namen der sich ergebenden Größe(n),
- dem Ergibtzeichen '=',
- einem Schlüsselwort für den Typ der sich ergebenden Größe (n) ('PKT' für Punkt, 'GER' für Gerade, usw. nach Regel 2.9),
- den Namen der bereits bekannten Größen, die in diesem Schritt verwendet werden, und den Angaben der geltenden Relationen ('PAR' für parallel, 'RAD' für Radius, usw. nach Regel 2.10),
- dem abschließenden Semikolon ';'.

2.4. Eine Konstruktionsbeschreibung oder eine Anweisungsfolge (Block, Folge von Ergibtanweisungen) endet mit der letzten Ergibtanweisung oder bei Bedarf mit einer Endeanweisung im Format

END prozedurname;

bestehend aus:

- dem Schlüsselwort 'END' für 'Ende',
- bedarfsweise dem Namen der Prozedur oder der Anweisungsfolge,
- dem abschließenden Semikolon ';'.

2.5. Bereits beschriebene Konstruktionen können als Ganzes durch einen Prozeduraufruf im Format

(funktionsliste) = prozedurname (argumentliste);

in größere Konstruktionen eingefügt werden. Er besteht aus

- der eingeklammerten Funktionsliste '(funktionsliste)',
- dem Ergibtzeichen '=',
- dem Namen der verwendeten Prozedur 'prozedurname',
- der eingeklammerten Argumentliste '(argumentliste)',
- dem abschließenden Semikolon ';'.

in genauer Korrespondenz mit der Bedeutung und Anordnung der Variablennamen in der Eröffnungsanweisung der definierenden Funktionen nach Regel 2.2.

2.6. Jeder Anweisung kann ein Anweisungsname im Format `anweisungsname : anweisung` vorgesetzt werden, bestehend aus

- dem Anweisungsnamen 'anweisungsname',
- einem Doppelpunkt ':',

2.7. Eine Prozedur wird durch gliedernde Kommentare im Format
 dezimalnummer verbaler Ausdruck

logisch gegliedert und erläutert. Die Dezimalnummer beginnt mit der höchsten Hierarchiestufe und wird bedarfsweise rechts durch Untergliederungen ergänzt. Der verbale Ausdruck sollte eine kurze Charakteristik des gesamten zugehörigen Programmabschnittes geben. Bedarfsweise sind gliedernde Kommentare durch einen Punkt abzuschließen. Wenn gleichberechtigte Verfahrensvarianten(s. hierzu auch 2.12.3.) angegeben werden, sollte dies in der Dezimalgliederung durch eingeklammerte Zahlen ausgedrückt werden.

Beispiele:

Gliederung ohne Varianten

1. Hauptprozedur
 - 1.1 Teil 1
 - 1.2 Teil 2
 - 1.2.1 Unterteil
 - 1.2.2 Unterteil
 - 1.2.3 Unterteil
 - 1.3 Teil 3
2. Unterprogramm
3. Unterprogramm.

Gliederung mit Varianten

- :
- 2.2 Teilprogramm
 - 2.2(1). Variante 1
 - 2.2(2). Variante 2
 - 2.2(3). Variante 3
 - 2.3

2.8. Freie Kommentare (Erläuterungen, Verweise auf benutzte Sätze, usw.) können an jeder Stelle eingefügt werden. Bedarfsweise sind sie in Schrägstriche einzuschließen.

'freier Kommentar'

Siehe hierzu auch Regel 2.12.6.

2.9. Jede geometrische Variable bekommt einen Namen (siehe auch Regel 2.12.2.), der nach Bedarf indiziert wird, und gehört einem bestimmten Typ an, der in der definierenden Ergibtanweisung nach Regel 2.3 festgelegt wird. Die Namen der Variablen sollen im allgemeinen bestimmten Buchstabenklassen angehören.

Typ	Schlüsselwort	Buchstabenklasse	Beispiele
Gebilde			
- Punkt	PKT	lat. Großbuchst.	A, B, C, P, Q
- Linie	LIN	lat. Kleinbuchst.	a, b, x, y
- - Gerade	GER		
- - Kreis	KRS		
- Fläche	FLA	lat.od.grch. Großb.	A, B, π , $\sqrt{\quad}$
- Körper	KOR	lat.od.grch. Großb.	A, B
Maße			
- Strecke	STR	lat. Kleinbuchstabe	a, d, r
- Winkel	WIN	grch. Kleinbuchstabe	α, β, γ

Indizes werden tief nachgestellt, z. B. A_1 , P_i , oder in Klammer nachgestellt, z. B. $A(1)$, $P(i)$;

Mehrfachindizes sind zulässig, z. B. A_{x_1} bzw. $AX(1)$ oder $A(X,1)$.

Jede Variable darf nur einmal auf der linken Seite einer Ergibtanweisung erscheinen. Alle in einer Konstruktion verwendeten Größen sind mit einem Variablennamen zu belegen.

2.10. Den Relationen zwischen den Variablen sind folgende Schlüsselwörter zugeordnet:

Relation	Schlüsselwort	Beispiele (Abb. 3)
Allgem. Relationen		
- beliebig	BEL	$A = \text{PKT BEL}$; beliebiger Punkt (B 2.1)
- Listenbildung	(,,)	$p = \text{LIN}(P_1, P_2, P_3, P_4)$; Linie durch die Punkte (B 2.2)
Lagerrelationen		
- Schnitt, gemeinsam -		$A = \text{PKT } a, b$; Schnittpunkt (B 2.3) zweier Linien

Relation	Schlüsselwort	Beispiele
		$a = \text{GER } A, B$; Gerade durch zwei Punkte (B 2.4)
- normal, senkrecht	NOR	$n = \text{GER } A \text{ NOR } a$; Ger. durch A normal zu a (B 2.5)
- parallel	PAR	$b = \text{GER } B \text{ PAR } C, D$; (B 2.6)
- im Winkel	WIN	$c = \text{GER } F \text{ WIN } a F c = \alpha$; (B 2.7)
- tangential	TAN	$t_{1,2,3,4} = \text{TAN KRS } k1 \text{ KRS } k2$; (B 2.8)
- mittig	MIT	$c = \text{GER MIT NOR } A, B$; Mittelsenkrechte zu A, B (B 2.9)
- Kreismittelpunkt	UM	$k = \text{KRS } A \text{ UM } M$; (B 2.10)
- Thaleskreis	THA	$k = \text{KRS THA } A, B$ Thaleskreis über Strecke A, B (B 2.11)
- rechts, links	RE, LI	$C = \text{PKT krs1, krs2 RE}$; (B 2.12)
Maßrelationen		
- Gleichheit	=	$\alpha = \text{WIN ASC}$ (B 2.13)
		$\text{STR } A, C = f$ (B 2.14)
- Radius	RAD	$k = \text{KRS UM } M \text{ RAD } r$; (B 2.15)
- Koordinaten		$A = \text{PKT } x = 17 \text{ cm},$ $y = 3 \text{ cm};$ (2. 16).

Bei Richtungen wird, soweit notwendig, auf die Reihenfolge der Variablen Bezug genommen:

Bsp 1. $\text{GER } A, B$ = Gerade von A nach B

Bsp 2. $t_2 = \text{GER TAN } k1 \text{ RE TAN } k2 \text{ LI}$ = Tangente an die Kreise 1 und 2, die Kreis 1 rechts, Kreis 2 links, in Richtung der Verbindungsgerade der Kreismittelpunkte gesehen, berührt. (B 2.3)

Richtungen sind grundsätzlich auf Variable, nie auf die Zeichenfläche (oben, unten) zu beziehen, da deren Orientierung nicht beständig ist.

Weitere Relationen können in verbalen Ausdrücken, vergleichenden ($\neq, >, <, \leq, \geq$) oder arithmetischen Operationen formuliert werden.

z. B.: $x = \text{GER auf A4 Blatt quer}$

PAR unterer Blattrand, Abstand 2 cm;

$x = \text{Koordinatenachse};$

$p = \text{LIN Polygon } (P_1, P_2, \dots, P_8);$

2.11. Die lineare Folge der Bearbeitung von Anweisungen kann durch Steueranweisungen modifiziert werden.

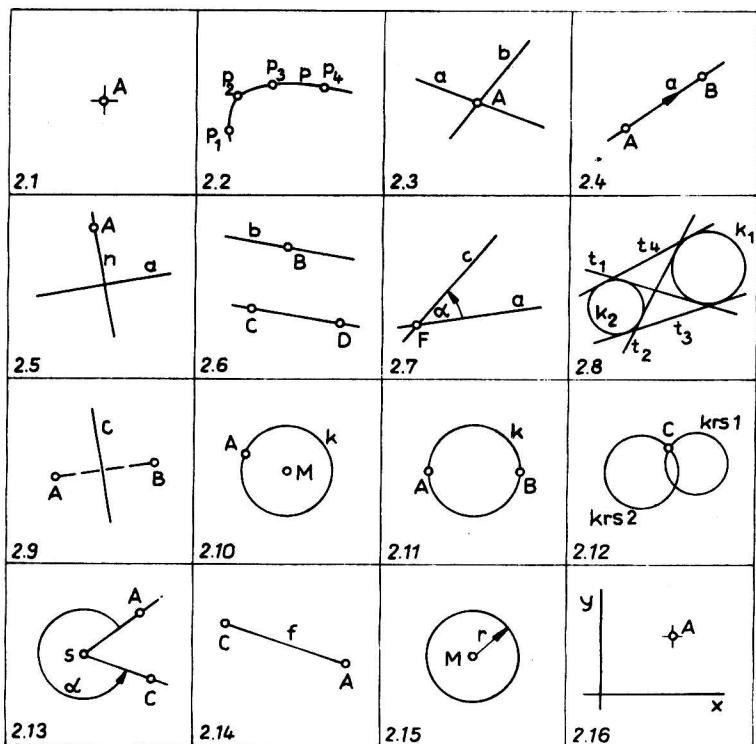


Abb. 3: Beispiele zu den Relationen und Ergibtanweisungen

2.11.1 Sprunganweisung im Format

GOTO anweisungsname;

Der Gebrauch der Sprunganweisung sollte nach Möglichkeit völlig vermieden werden.

Nach der Sprunganweisung wird ab der Anweisung mit dem Namen 'anweisungsname' weitergearbeitet.

2.11.2 Bedingungsanweisungen in den Formaten

IF bedingung THEN anweisungsfolge END;

IF bedingung THEN anweisungsfolge END;

ELSE anweisungsfolge END;

Wenn die Bedingung erfüllt ist, dann wird die Anweisungsfolge nach THEN abgearbeitet, sonst wird sie übersprungen bzw. die Anweisungsfolge nach ELSE abgearbeitet.

2.11.3 Alternativanweisung im Format

CASE ausdrück

OF wert 1: anweisungsfolge END;

OF wert 2: anweisungsfolge END;

.

.

ELSE anweisungsfolge END;

Wenn der Ausdruck nach CASE (im Fall) einen der aufgeführten Werte annimmt, wird die zugeordnete Anweisungsfolge abgearbeitet, sonst wird, wenn vorhanden, eine nach ELSE angegebene Anweisungsfolge abgearbeitet bzw. es werden alle Anweisungsfolgen übersprungen.

2.11.4 Wiederholungsanweisungen in den Formaten

FOR (werteliste) DO anweisungsfolge END;

Die Anweisungsfolge wird für alle in der Liste aufgeführten Werte abgearbeitet.

WHILE bedingung DO anweisungsfolge END;

Die Anweisungsfolge wird wiederholt bearbeitet, solange die Bedingung, die vor der Anweisungsfolge steht, erfüllt ist.

Die Anweisungsfolge wird wiederholt bearbeitet, bis die Bedingung, die nach der Anweisungsfolge steht, erfüllt ist.

2.12.1 Die Bezeichnung aller auftretenden Elemente mit Variablenamen ist zur Verständigung notwendig, bei der Anwendung der Konstruktionsbeschreibungen sollten nur die wichtigsten Elemente mit Buchstaben usw. bezeichnet werden. Bei Prozeduraufrufen (Regel 2.5) sollten die zugeordneten Bezeichnungen paarweise notiert werden.

- mnemonische Verbindungen unterstützt werden,
z. B. M = Mittelpunkt, h = Höhe, r = Radius, α = Winkel bei A;
- Zahlen als untere Indizes verschiedene gleichartige Objekte unterscheidbar machen,
z. B. P₁, P₂ ... P₁₂ = zwölf Punkte auf einer Kurve, die sich aus der punktwweisen Konstruktion ergeben,
P₀ = ausgezeichnete Punkt auf dieser Kurve;
- Buchstaben als untere Indizes die Zuordnung zu anderen Objekten ausdrücken,
z. B. h_A, h_B, h_C = die den Ecken A, B, C eines Dreiecks zugeordneten Höhen;
- obere Indizes zur Unterscheidung verschiedener Darstellungen desselben Objekts unterscheidbar machen,
z. B. P
P', P'', P'''

P^o, P^a

Punkt im Raum
Darstellung von P im Grund-,
Seiten- und Aufriß
Diesem Punkt zugeordnete Punkte
auf Hilfslinien oder -flächen.

92

- 2.12.4 Grundkonstruktionen, insbesondere seltener verwendete, sollten in einer Programm-bibliothek gesammelt werden.
- 2.12.5 Anweisungen, die sich mit den Mitteln der Konstruktionsbeschreibungssprache nur schwer verständlich ausdrücken lassen, sind als verbale Anweisungen frei zu formulieren.
- 2.12.6 Sätze oder Ableitungen werden nicht als Kommentare in Konstruktionsbeschreibungen eingeführt, sondern besser vor oder hinter zusammengehörenden Konstruktionsbeschreibungen angeordnet und numeriert. In den Kommentaren genügen dann kurze Verweise, z. B. /nach Satz 17/ als Begründung oder /beachte 9.8/ als Hinweis.

Anschrift des Verfassers:

Dr. sc. techn. Klaus Franze

VEB Robotron, ZFT

DDR - 801 Dresden

PSF 330

Ein Kalkül für geometrische Aufgaben

Es gibt viele geometrische Aufgaben, die sich auf die durch "Verbinden", "Schneiden", "durch ... parallel zu ..." und "durch ... senkrecht zu ..." umschriebenen Grundaufgaben der Orthogonalgeometrie des euklidischen Raumes zurückführen lassen.

Sie lassen sich einheitlich im projektiv-abgeschlossenen euklidischen Raum unter Beachtung der Polarität an der imaginären Einheitskugel (absolute Polarität) behandeln. Es sollen hier die diesbezüglichen wohl bekannten und völlig klassischen Sachverhalte dargestellt werden, da sie nicht zum Allgemeingut jüngerer Mathematikergenerationen gehören und auch den Anwendern im allgemeinen nicht geläufig sind. Zugleich ist dies eine Einführung in den Aufsatz

17 Ortleb, R.: Rechentechnisch-orientierte Darstellung einfacher geometrischer Grundaufgaben.

Rechentechnik/Datenverarbeitung 13, 1, 46 - 47 (1976),

der zwar äußerst klar, aber vielleicht doch etwas knapp gehalten ist. Über den Stand der inzwischen erfolgten rechentechnischen Realisierung der darzulegenden Vorgehensweise, um die es uns ja eigentlich geht, wird Fr. Krause vortragen. Daß Herr Dr. Ortleb, obwohl anwesend, nicht über das eine oder andere oder beides vorträgt, bedeutet übrigens nichts Schlimmes.

1. Der projektiv-abgeschlossene euklidische Raum

Der euklidische Raum E^3 sei wie üblich mit einem kartesischen Koordinatensystem $(0; x, y, z)$ versehen. Er wird durch Hinzunahme der uneigentlichen Punkte und Geraden und der uneigentlichen Ebene projektiv abgeschlossen, wobei die besondere Rolle dieser uneigentlichen Elemente erhalten bleibt. So entsteht der projektiv-abgeschlossene eukli-

dische Raum; er wird als Raum seiner Punkte mit \bar{E}^3 und (wenn für nötig erachtet) als Raum seiner Ebenen mit \hat{E}^3 bezeichnet. Das gegebene Koordinatensystem ist das Koordinatentetraeder $(0, X_u, Y_u, Z_u; E) = (0, X_{1u}, X_{2u}, X_{3u}; E)$ abzuändern, wo $X_u = X_{1u}$, $Y_u = X_{2u}$, $Z_u = X_{3u}$ die uneigentlichen Punkte von x-, y- bzw. z-Achse sind und E den Einheitspunkt bedeutet. Auf dieses projektive Koordinatensystem bezieht sich die Darstellung des \bar{E}^3 als die Menge aller 1-dimensionalen Unterräume des R^4 und des \hat{E}^3 als die Menge aller 1-dimensionalen Unterräume eines weiteren Vektorraumes R^4 , der zweckmäßigerweise mit \hat{R}^4 bezeichnet werde. Irgendein erzeugender Vektor eines 1-dimensionalen Unterraumes von R^4 bzw. \hat{R}^4 heißt ein Koordinatenvektor des durch diesen Unterraum angegebenen Punktes von \bar{E}^3 bzw. der entsprechend dargestellten Ebene von \bar{E}^3 . Die Koordinatenvektoren von Punkten werden mit

$$x = (x_0 \ x_1 \ x_2 \ x_3)^T,$$

die von Ebenen mit

$$\hat{x} = (\hat{x}_0 \ \hat{x}_1 \ \hat{x}_2 \ \hat{x}_3)^T$$

bezeichnet. Die beiden Vektorräume R^4 und \hat{R}^4 sind ein duales Raumpaar (\hat{R}^4, R^4) ; das die Dualität festlegende skalare Produkt ist

$$(\hat{x}, y) := \hat{x}_0 y_0 + \hat{x}_1 y_1 + \hat{x}_2 y_2 + \hat{x}_3 y_3 = \hat{x}^T y = y^T \hat{x}.$$

Genau im Falle $(\hat{y}, x) = 0$ liegt Inzidenz von Punkt y und Ebene \hat{x} vor. Der Vektorraum \hat{R}^4 , der \hat{E}^3 im angegebenen Sinne beschreibt, ist auch als zu R^4 dualer Vektorraum (Vektorraum der auf R^4 operierenden Linearformen) aufzufassen, und umgekehrt. Es ist dann

$x_0 = 0$ eine Gleichung der uneigentlichen Ebene und

$\hat{x}_0 = 0$ eine Gleichung des Koordinatenursprungs.

2. Verbinden, Schneiden; Inzidenz

Bei der Einführung des projektiv-abgeschlossenen euklidischen Raumes geht es weniger um die Erfassung der uneigentlichen Elemente dieses geometrischen Raumes (für die in vorliegendem Zusammenhang ihre historische Benennung "Fernelemente" als kennzeichnend ungesehen werden kann), sondern vielmehr um das Erschließen einer sachgemäßen Möglichkeit, sehr weit vom Koordinatenursprung liegende Punkte, Geraden und Ebenen auch rechnerisch in den Griff zu bekommen. Hierbei die Eigenarten der in einem Re-

chenautomaten vorliegenden Zahlensysteme und ihre Arithmetik zu bedenken, ist ein anderes Problem, auf das hier nicht eingegangen werden kann. Die homogenen Koordinaten besitzen auch den Vorteil, die bei einfachen geometrischen Aufgaben auftretenden linearen Gleichungssysteme divisionsfrei lösen zu können. Natürlich muß man nicht bei jeder anfallenden geometrischen Aufgabe solche Rücksichten üben (so auch bei einigen der folgenden Betrachtungen), doch verdienen diese Dinge grundsätzliche Beachtung.

2.1. Zur Rolle von Normen

Um schon das "sehr weit Entfernte" rechentechnisch bzw. rechnerisch in den Griff zu bekommen, um ferner gewisse Invarianten geometrischer Gebilde (beispielsweise Winkel) ermitteln, um Tests und numerische (Iterations-)Verfahren durchführen zu können, sind für \mathbb{R}^4 und $\hat{\mathbb{R}}^4$ (und weitere Vektorräume) gewisse Normen einzuführen, und dies trotz erhöhten Programmier- und Speicheraufwandes und verwandter Probleme.

Mit Hilfe der Norm

$$\|x\|_1^2 := x^T x \quad \text{bzw.} \quad \|\hat{x}\|_1^2 := \hat{x}^T \hat{x}$$

kann auf "Unelement" $(0 \ 0 \ 0 \ 0)^T$ getestet und die gnomonische Projektion des \hat{E}^3 auf die Einheitssphäre des E^4 bewerkstelligt werden; durch diese Projektion gelangen alle Elemente des \hat{E}^3 "ganz ins Endliche" (allerdings eines 4-dimensionalen Raumes). Mit Hilfe der Pseudonormen

$$\|x\|_0^2 := x_0^2 \quad \text{und} \quad \|\hat{x}\|_3^2 := \hat{x}_1^2 + \hat{x}_2^2 + \hat{x}_3^2$$

können "sehr weit entfernte" bzw. "unendlichferne" Elemente als solche erkannt und beispielsweise der Abstand eines Punktes von einer Ebene ermittelt werden:

$$\text{dist}(\hat{x}, y) := \frac{\hat{x}^T y}{\|\hat{x}\|_3 \cdot \|y\|_0}$$

Auf die Normierungsnotwendigkeit und -problematik soll hier nur hingewiesen werden.

2.2. Verbinden, Schneiden; Inzidenz

Um die projektiven Aufgaben des Verbindens und Schneidens zu lösen, sind für Geraden die Plückerschen Geradenkoordinaten 1. und 2. Art (Linien- und Achsenkoordinaten) einzuführen. Sie sind die wesentlichen Elemente der folgenden schiefsymmetrischen

4 x 4-Matrizen vom Range 2 (hinter $\text{rg} = 2$ verbirgt sich die PLÜCKER-Bedingung): Die Punkte x und y definieren die Gerade g mit der Koordinatenmatrix 1. Art

$$G = xy^T - yx^T = \begin{bmatrix} 0 & g_1 & g_2 & g_3 \\ -g_1 & 0 & g_6 & -g_5 \\ -g_2 & -g_6 & 0 & g_4 \\ -g_3 & g_5 & -g_4 & 0 \end{bmatrix};$$

$$g = (g_1 \ g_2 \ g_3 \ g_4 \ g_5 \ g_6)^T \text{ mit } g_1g_4 + g_2g_5 + g_3g_6 = 0$$

ist der zugehörige Koordinatenvektor aus den Geradenkoordinaten 1. Art. Wird dieselbe Gerade g durch die Ebenen \hat{x} und \hat{y} definiert, so erhält man die Koordinatenmatrix 2. Art

$$\hat{G} = \hat{x}\hat{y}^T - \hat{y}\hat{x}^T = \begin{bmatrix} 0 & \hat{g}_1 & \hat{g}_2 & \hat{g}_3 \\ -\hat{g}_1 & 0 & \hat{g}_6 & -\hat{g}_5 \\ -\hat{g}_2 & -\hat{g}_6 & 0 & \hat{g}_4 \\ -\hat{g}_3 & \hat{g}_5 & -\hat{g}_4 & 0 \end{bmatrix};$$

$$\hat{g} = (\hat{g}_1 \ \hat{g}_2 \ \hat{g}_3 \ \hat{g}_4 \ \hat{g}_5 \ \hat{g}_6)^T \text{ mit } \hat{g}_1\hat{g}_4 + \hat{g}_2\hat{g}_5 + \hat{g}_3\hat{g}_6 = 0$$

ist der zugehörige Koordinatenvektor aus den Geradenkoordinaten 2. Art, und bis auf einen gemeinsamen Faktor ist

$$g_1 = \hat{g}_4, \ g_2 = \hat{g}_5, \ g_3 = \hat{g}_6, \ g_4 = \hat{g}_1, \ g_5 = \hat{g}_2, \ g_6 = \hat{g}_3.$$

Als genaues Kriterium für die Gleichheit zweier Punkte x und y bzw. zweier Ebenen \hat{x} und \hat{y} hat man neben der linearen Abhängigkeit der Koordinatenvektoren auch

$$G = 0 \quad \text{bzw.} \quad g = 0 \quad \text{und} \quad \hat{G} = 0 \quad \text{bzw.} \quad \hat{g} = 0$$

zur Auswahl.

Die Gerade g und der Punkt x inzidieren oder definieren eine (Verbindungs-)Ebene \hat{v} , je nachdem

$$\hat{G}x = 0 \quad \text{oder} \quad \hat{G}x \cdot \hat{v} \neq 0$$

gilt. Die Gerade g liegt in der Ebene \hat{x} oder hat genau einen Schnittpunkt mit ihr gemein-

sam, je nachdem

$$G\hat{x} = 0 \quad \text{oder} \quad G\hat{x} = s \neq 0$$

ist. Offensichtlich hat man damit divisionsfrei jene linearen Gleichungssysteme gelöst, die vorliegen, wenn die Verbindungsebene dreier Punkte bzw. der Schnittpunkt dreier Ebenen bestimmt werden soll.

Man erkennt, daß zum Prüfen von Inzidenzrelationen oder der Gleichheit zweier geometrischer Elemente oder des Vorliegens von Geradenkoordinaten (Plückerbedingung) notwendig Normen verwendet werden müssen. So auch bei dem folgenden Problem.

Zwei Geraden können identisch sein, in genau einer Ebene liegen oder nicht in einer Ebene liegen (windschief sein). Dies trifft beziehungsweise genau dann zu, wenn gilt:

$$\text{rg } \hat{GH} = \begin{cases} 0 \\ 1 \\ 2 \end{cases} \quad \text{bzw., gleichwertig,} \quad \text{rg } \hat{GH} = \begin{cases} 0 \\ 1 \\ 2 \end{cases};$$

dabei können die Aussagen der 2. und 3. Zeile auch ersetzt werden durch die gleichwertigen Kriterien

$$g_1 h_4 + g_2 h_5 + g_3 h_6 + g_4 h_1 + g_5 h_2 + g_6 h_3 =: d \begin{cases} = 0 \\ \neq 0 \end{cases}$$

Bestimmen die Geraden g und h eindeutig eine Ebene, so auch eindeutig einen Punkt (und umgekehrt), und es ist irgendeine von 0 verschiedene Spalte von \hat{GH} ein Koordinatenvektor dieser Verbindungsebene bzw. eine solche Spalte von \hat{GH} ein Koordinatenvektor jenes Schnittpunktes.

Insgesamt sind damit für zwei Elemente (Punkte, Geraden, Ebenen) als Operanden alle Aufgaben des Verbindens und Schneidens und der Inzidenz beschrieben worden.

2.3. Beispiel

Als Beispiel für eine Aufgabe des Verbindens/Schneidens und der Inzidenz mit drei Operanden diene das Folgende: Durch einen Punkt x , eine Gerade g und eine Gerade h sind alle jene Geraden t zu legen, die alle drei Elemente treffen ("Treffgerade"). Im Falle der eindeutigen Existenz (die nicht gewährleistet ist) hat man, wenn das Zeichen x für die gleichberechtigten Operationen Schneiden und Verbinden genommen wird:

$$(1) \quad t = (x * g) * (x * h)$$

bzw. in Koordinatenvektoren und -matrizen:

$$(2) \quad \hat{T} = (\hat{G}x) (\hat{H}x)^T - (\hat{H}x) (\hat{G}x)^T = \hat{H}x x^T \hat{G} - \hat{G}x x^T \hat{H}.$$

Es ist klar, daß solche fertigen Formeln gar nicht gefragt, sondern durch Verfahren zu ersetzen sind, deren Bestandteile eine einfache Bedeutung und bequeme Programmierung besitzen, etwa wie in (1). Wegen der Zweideutigkeit von $*$ bzw. der Abhängigkeit der Operation $*$ von den eingehenden Operanden ist der Typ des Ergebnisses noch jeweils 'zu berechnen'. Dies geschieht mit Hilfe von Kennzahlen. In (1) können g und h nur Geraden bedeuten, ebenso kann t nur eine Gerade sein, aber es ist möglich, in x eine Ebene \hat{x} zu sehen. Dann aber ist (falls definiert)

$$(1) \quad t = (\hat{x} * g) * (\hat{x} * h)$$

die Verbindungsgerade der Schnittpunkte von g und h mit der Ebene \hat{x} ; die Rechnung in Koordinaten sieht genau so aus wie (2).

Da Existenz und Eindeutigkeit des Ergebnisses nicht vorliegen muß, ist es wünschenswert, daß ein Abbruch des Verfahrens unter Bekanntgabe eines wohl definierten, möglichst geometrischen Grundes erfolgt. Dies läuft auf ein Testen verschiedener Inzidenzen hinaus. Sie erhöhen den Programmieraufwand erheblich, sind aber gegebenenfalls unerlässlich. So beispielsweise bei Sichtbarkeitsentscheidungen, wenn Polyeder dargestellt werden.

Mit den bisher beschriebenen Mitteln beherrscht man etwa das Projizieren, das Schneiden von Pyramiden, Prismen und Polyedern sowie das Schneiden von Quadriken sowohl in der Ebene wie im Raum.

3. Orthogonalität

Für Paare der Art (Gerade, Gerade), (Ebene, Gerade), (Ebene, Ebene), an denen kein uneigentliches Element beteiligt ist (Test!), kann Orthogonalität vorliegen. Die Orthogonalität in E^3 wird vollständig von der Polarität an der imaginären Einheitskugel

$$K: \quad x_0^2 + x_1^2 + x_2^2 + x_3^2 - x^T x = 0$$

beherrscht. Diese Polarität kann im projektiv-abgeschlossenen euklidischen Raum auch auf uneigentliche Elemente ausgeübt werden und induziert in der uneigentlichen Ebene die Polarität bezüglich des absoluten Kugelkreises

$$k: \quad x_0 = 0, \quad x_1^2 + x_2^2 + x_3^2 = 0.$$

Die Orthogonalität von Paaren (oben genannter) eigentlicher Elemente spiegelt sich genau wider in dem Konjugiertsein ihrer uneigentlichen Elemente bezüglich k :

- Zwei Geraden stehen (genau dann) senkrecht aufeinander, wenn jeder ihrer uneigentlichen Punkte auf der Polaren des anderen liegt, wenn sie also bzgl. k konjugiert liegen.
- Eine Gerade steht zu einer Ebene senkrecht, wenn ihr uneigentlicher Punkt der Pol der uneigentlichen Geraden der Ebene ist.
- Zwei Ebenen stehen senkrecht aufeinander, wenn jede ihrer uneigentlichen Geraden durch den Pol der anderen geht, wenn sie also bzgl. k konjugiert liegen.

Damit lassen sich alle Aufgaben "durch ... senkrecht zu ...", auch Lotaufgaben genannt, lösen. Zwei Beispiele mögen zur Erläuterung dienen.

Es sei $n = p \perp \hat{x}$ die Gerade n , die durch p geht und senkrecht zu \hat{x} steht. Man bestimmt von \hat{x} die uneigentliche Gerade, ermittelt die konjugierte Gerade bzgl. K , sucht deren uneigentlichen Punkt auf (d. i. der Pol der uneigentlichen Geraden von \hat{x} bzgl. k) und verbindet diesen mit p . Bezeichnet π die Polarität an K und \hat{w} die uneigentliche Ebene, dann hat man (von rechts nach links)

$$n = p * (\hat{w} * \pi(\hat{w} * \hat{x}))$$

und wegen der Linearität von π auch

$$n = p * (\hat{w} * (w + \pi(\hat{x}))),$$

wobei $\pi(\hat{w}) = w$ einen Koordinatenvektor des Koordinatenursprungs liefert.

Die Gemeinlotgerade n zweier (eigentlicher) Geraden g und h wird so bestimmt: Die (verschiedenen (Test!))uneigentlichen Punkte von g und h werden verbunden, deren konjugierte Gerade bzgl. K ermittelt, davon den uneigentlichen Punkt, also den Pol jener Verbindungsgerade bzgl. k aufsuchen, und durch diesen und g und h die Treffgerade legen. In zwei Etappen zusammengefaßt, erhält man

$$x = \hat{w} * \pi \left[\frac{(h * \hat{w}) * (g * \hat{w})}{(x * g) * (x * h)} \right]$$

$$n = (x * g) * (x * h).$$

Führt man die zugehörigen Rechnungen in Koordinaten aus, so stellt man fest, daß sich gewisse Vereinfachungen gegenüber diesen geometrischen Grundoperationen ergeben und insbesondere etliche arithmetische Grundoperationen wiederholt auftreten, die

eigenständig zu erfassen, äußerst sinnvoll ist. Außerdem wird sich zum bequemen Arbeiten ein Zusammenfassen von Operationen aus \ast und Π zu 'zusammengesetzten' Grundoperationen anbieten. Wie die den Anwender primär interessierenden geometrischen Operationen mit Hilfe von im Hintergrund ablaufenden arithmetischen Grundoperationen beherrscht werden können, ist Inhalt des Vortrags von Frl. Krause.

Anschrift des Verfassers:

Prof. Dr. habil. Gerhard Geise

Techn. Universität Dresden
Sektion Mathematik

DDR - 8027 Dresden

Mommstr. 13

Einsatz der Bildschirmtechnik bei der Untersuchung von Schiffsschwingungen

In dem Artikel soll kurz auf den Einsatz der Bildschirmtechnik bei der Untersuchung von Schiffsschwingungen, speziell der Antriebsanlage Motor-Welle-Propeller, eingegangen werden. Die später angeführten Beispiele zählten zu den ersten Anwendungen und dienten der Sammlung von Erfahrungen für den Einsatz der Bildschirmtechnik, hier konkret zur Steuerung von Berechnungsabläufen. Bei diesen Anwendungen wurden daher bewußt nicht völlig neue Programme konzipiert, sondern es wurden bereits erprobte erweitert, um möglichst schnell zu brauchbaren Aussagen über Aufwand und Nutzen zu kommen.

Am Rechenzentrum des Institutes für Schiffbau Rostock wurden bereits vor mehreren Jahren umfangreiche Programmsysteme zur Berechnung von Biege-, Längs- und Drehschwingungen von Schiffsantriebsanlagen entwickelt. Hierbei handelte es sich um reine Stapelverarbeitungsprogramme, die für bestimmte Eingabewerte Berechnungen ausführten und anschließend die erzielten Resultate ausgaben. Die mit der Bildschirmtechnik gegebenen Möglichkeiten zu einem Dialog Nutzer-Rechenmaschine sollten zu einer wesentlichen Steigerung der Effektivität dieser Programme genutzt werden. Mit Hilfe des am Institut für Schiffbau entwickelten Programmsystems GIPS wurden neue Programmabschnitte als Ergänzung zu den vorhandenen entwickelt, die das Bildschirmgerät an geeigneten Schnittstellen als Ein- und Ausgabegerät sowie zur Steuerung des Rechenablaufes vorsehen. Diese Schnittstellen waren auf Grund des Stapelverarbeitungscharakters der bestehenden Programme recht schnell und sehr günstig festzulegen, nach der Beendigung der Berechnungen und dem Listen der Ergebnisse konnten die "Bildschirmprogramme" zur Darstellung der Resultate und zum Neustart von Berechnungen nach Eingabeparametervariationen im Dialog am Bildschirm eingefügt werden. Dieses Vorgehen ergab sich aus

der Problematik der Untersuchungen von Schwingungsvorgängen und soll etwas näher am Beispiel von Biegeschwingungen erläutert werden.

Zur Konstruktion einer Schiffsantriebsanlage sind umfangreiche theoretische Variantenuntersuchungen mit Hilfe eines mathematisch-physikalischen Modells notwendig. Mit der Erzeugung eines solchen Modells beginnend, werden anschließend Modelle gesucht, die gute konstruktive Lösungen an der realen Anlage ermöglichen, d. h. solche, die zur Vermeidung unerwünschter Schwingungsvorgänge beitragen.

Zur Berechnung der Biegeschwingungen wird die gesamte Antriebsanlage im Modell als aus mehreren Teilstücken zusammengesetzt dargestellt. Als Teilstücke sind dabei "Wellenelemente" mit den Parametern Masse, Länge und Biegesteifigkeit, "Konzentrierte Massen", "Elastische Federn" und "Starre Lager" vorgesehen. Der Aufbau der Wellenanlage aus den aufgeführten Teilstücken wird in einem Schlüssel für die Anordnung der durchnummerierten Teilstücke angegeben. Nach der Eingabe der Parameterwerte der einzelnen Teilstücke erfolgt die Eigenschwingungsberechnung in einem Iterationsverfahren mit Hilfe von Übertragungsmatrizen, auf das hier jedoch nicht näher eingegangen werden soll.

Nach Abarbeitung des Berechnungsteils erfolgt die Freigabe der Funktionstastatur des Bildschirmgerätes für Dialogaktivitäten. Der Anwender hat jetzt die Möglichkeit, sich als Ergebnisse die Schwingungsgrade, die Eigenkreisfrequenzen und die dazugehörigen Wellendrehzahlen alphanumerisch und die Durchbiegungen der einzelnen Teilstücke in Form von Kurven graphisch darstellen zu lassen.

Wenn die erzielten Ergebnisse noch nicht seinen Anforderungen entsprechen, so kann der Anwender mit Hilfe der alphanumerischen und der Funktionstastatur beliebige Modellveränderungen vornehmen. Es können Parameterwerte von Teilstücken verändert werden, Teilstücke aus dem Modell gestrichen oder neu eingefügt werden. Daran anschließend können die Berechnungen für das veränderte Modell der Schiffsantriebsanlage von Neuem initialisiert werden. Durch diese effektive Arbeit mit der Bildschirmtechnik konnte die Suche des optimalen Modells gegenüber der früheren Arbeitsweise (mehrere Variantenuntersuchungen in aufeinanderfolgenden Rechnungen an verschiedenen Tagen) auf ein oder zwei Arbeitssitzungen am Bildschirm verkürzt werden.

Zusammenfassend kann gesagt werden, daß die Untersuchungen und Erweiterungen an den angeführten Programmsystemen erwiesen haben, daß auch Problemlösungen, die ursprünglich nicht die Datenverarbeitung mit der Bildschirmtechnik beinhalteten, mit verhältnismäßig geringem Aufwand wesentlich effektiver gestaltet werden konnten. Günstiger wird es aber auf jeden Fall sein, wenn die jetzt zur Verfügung stehenden umfangreichen Mittel und Wege der Digitalgraphik gleich in die Konzeption zur Lösung eines Problems mit eingearbeitet werden.

Anschrift des Verfassers:

Dipl.-Math. Ulrich Jaeckel

Institut für Schiffbau
Rechenzentrum

DDR - 25 Rostock

Doberaner Straße

Anwendung der Digitalgraphik bei der Herstellung von Behälterzeichnungen

Ausgehend von einer Analyse des Projektierungsprozesses werden im Chemieanlagenbau- und Montagekombinat Leipzig Arbeiten zur Rationalisierung der Projektierung von Chemieanlagen durchgeführt. Sie erfolgen in Übereinstimmung mit den Richtlinien und Festlegungen für Projektierungsleistungen in der chemischen Industrie: Investitionsvorentcheidung (IVE), Grundsatzentscheidung (GE) und Durchführung (Projekt).

In den Vorbereitungsphasen IVE und GE wird der Hauptteil der wissenschaftlich-technischen Arbeit geleistet. Hier wird die technische Lösung des Vorhabens fixiert, werden Preise und Realisierungszeiten festgelegt. In diesen Phasen werden auch sogenannte Anfragespezifikationen erarbeitet.

Anfragespezifikationen dienen dem Informationsaustausch zwischen Projektierung und Ausrüstungsherstellern. Sie erleichtern auch die im Projektierungsprozeß notwendigen Koordinierungsarbeiten. Die Anfragespezifikationen bestehen aus Zeichnungen, Texten und Listen.

Die Zeichnung vermittelt einen Überblick über die technische Konzeption und Funktion einer Ausrüstung. Sie beinhaltet die Darstellung von Form, Größe und Abmessung sowie sämtlicher Einbauten und Anbauteile. Ihre Ausführlichkeit ist abhängig von der Komplexität der darzustellenden Ausrüstung. Grundsätzlich ist die Zeichnung so zu gestalten, daß der Ausrüstungshersteller in der Lage ist, auf deren Grundlage ein verbindliches Angebot abzugeben und die erforderlichen Fertigungszeichnungen herzustellen.

Eine Untersuchung von 17 Vorhaben des Chemieanlagenbaues ergab, daß bei diesen Vorhaben 726 Behälter mit einem Nenninhalt bis 100 m^3 eingesetzt wurden. Der größte Teil davon waren liegende Behälter. U. o. waren diese Gründe der Anlaß, ein Programmsystem-Behälter zu entwickeln.

Das Programmsystem-Behälter besteht aus Rechenprogrammen und einem Programm zum automatischen Zeichnen der Behälter. Die Rechenprogramme dienen u. a. zur Berechnung des Behältervolumens sowie zur Auswahl eines Behälters aus dem Lieferkatalog, zur wärmetechnischen Berechnung, zur Werkstoffauswahl und zur Masseberechnung.

Auf der Zeichnung werden der Behälter in 2 Ansichten mit Hauptabmessungen sowie Sattellager, Schriftfeld, Stützentabelle und Bemerkungen dargestellt. Einbauten und Anbauteile sind manuell nachzutragen. Es zeigte sich nämlich, daß das Zeichenprogramm bei Darstellung der verschiedenartigen An- und Einbauten des Behälters und deren Bemaßung sehr aufwendig wird. Letzteres vor allem war der Grund, Möglichkeiten des Einsatzes der Bildschirmtechnik zu untersuchen. Durch diese ergeben sich Vorteile insbesondere dadurch, daß die Behälterdarstellung auf dem Bildschirm ergänzt und bemaßt werden kann, bevor sie als Zeichnung ausgegeben wird.

Die Arbeiten, die in dieser Hinsicht durchgeführt wurden, hatten nur den Charakter von Forschungsarbeiten. Sie dienten anfangs auch nur dazu, Programmierungskenntnisse zur Bildschirmtechnik zu erlangen und die Arbeit am Bildschirmgerät kennenzulernen. Da im CMK Leipzig kein Bildschirmgerät zur Verfügung steht, mußten die erforderlichen Testarbeiten am Institut für Schiffbau in Rostock durchgeführt werden. Die große Entfernung von Leipzig nach Rostock bereitete dabei zusätzliche Schwierigkeiten.

Für die Erzeugung einer Behälterdarstellung auf dem Bildschirm, deren Vervollständigung und Bemaßung im Dialog und die Ausgabe des erzeugten Bildes für die Zeichenmaschine wurde ein Programm erarbeitet, das diese Aufgaben in folgenden Schritten löst:

1. Erzeugung eines Zeichnungsrahmens und Zeichnungsschriftfeldes sowie eines Behälterkenngrößenschriftfeldes auf dem Bildschirm mit Hilfe von Bilderzeugungsroutinen.
2. Eintasten der Werte von folgenden Größen mit der alphanumerischen Tastatur:
 - Länge des zylindrischen Teils des liegenden Behälters,
 - Behälterdurchmesser,
 - Wanddicke des Behälters,
 - Abstand der Sattellager,
 - Höhe der Behälterachse über der Aufstandsfläche.

Die in Millimeter eingetasteten Werte werden auf dem Bildschirm dargestellt. Sie können gegebenenfalls korrigiert werden.

3. Aus den eingegebenen Werten der Behälterkenngrößen wird der Maßstab für die Darstellung des Behälters auf einer A3-Zeichnung berechnet; danach wird der Maßstab automatisch in das Zeichnungsschriftfeld eingetragen.
4. Mit Hilfe der alphanumerischen Tastatur werden die Werte von zwei Größen a und b vorgegeben. Durch diese Größen ist es möglich, die Behälterdarstellung an der entsprechenden Stelle der Zeichnungsfläche zu positionieren. Die Werte von a und b werden ebenfalls auf dem Bildschirm dargestellt und können gegebenenfalls auch korrigiert werden.
5. Nach Drücken einer Funktionstaste wird der Behältergrundkörper mit Sattellagem in Vorder- und Seitenansicht auf dem Bildschirm dargestellt.
6. Die Vervollständigung der Behälterdarstellung erfolgt mit Hilfe der Menütechnik. Dazu werden zunächst - nach Auslösung mittels Funktionstaste - die anzutragenden Anbauteile (Arbeitsöffnung, Stutzen für Rohrleitungsanschlüsse) im Maßstab der Behälterdarstellung sowie Lichttasten am oberen Bildrand erzeugt. Danach werden die mit dem Lichtstift identifizierten Anbauteile z. B. kopiert. Das geschieht durch Aktivieren der entsprechenden Lichttaste mit dem Lichtstift. Die jeweilige Kopie erscheint an der Stelle des Bildschirms, wo momentan das Lichtkreuz positioniert ist.

Verschieben und Löschen der Kopie ist gegebenenfalls möglich.

7. Nachdem die Behältervorderansicht durch Anbauteile vervollständigt worden ist, wird nach Auslösen mittels Funktionstaste die bisherige Seitenansicht entsprechend ergänzt.
8. Erzeugung der Bemaßung im Dialog.

Dazu wird mit Hilfe des Lichtkreuzes der Anfangs- und Endpunkt der betreffenden Maßlinie markiert sowie der Abstand zu den anderen Maßlinien bzw. zur Behälterdarstellung festgelegt. Nach Drücken einer Funktionstaste erscheint die Maßlinie mit Maßpfeilen und Maßhilfslinien auf dem Bildschirm. Es können sowohl waagerechte als auch senkrechte Maßlinien dargestellt werden.

Nach der Erzeugung der Maßlinien werden die Maßzahlen an der durch das Lichtkreuz festgelegten Stelle angetragen. Die Darstellung der Maßzahlen erfolgt mit Hilfe der alphanumerischen Tastatur.

9. Berechnung des Behältervolumens.

Durch Druck einer Funktionstaste wird das Behältervolumen berechnet und das Ergebnis auf dem Bildschirm angezeigt.

10. Nach Betätigen einer weiteren Funktionstaste wird das erzeugte Bild über einen Lochstreifen für die Zeichenmaschine ausgegeben. Anschließend wird die Darstellung auf dem Bildschirm gelöscht und es können weitere Behälterdarstellungen erzeugt, vervollständigt und bemaßt sowie über Lochstreifen für die Zeichenmaschine ausgegeben werden.

11. Beendigung der Arbeit am Bildschirmgerät, wenn keine weiteren Behälterdarstellungen erzeugt werden sollen.

Zur Unterstützung der Arbeit am Bildschirmgerät liegt ein Regieplan vor, der die einzelnen Schritte der Programmabarbeitung bzw. die zu fällenden Entscheidungen aufzeigt. Es ist Vorsorge getroffen, daß bei Fehlhandlungen, z. B. bei Nichteinhaltung der Reihenfolge der Abarbeitung oder bei Drücken einer falschen Funktionstaste, kein vorzeitiger Abbruch der Rechnung erfolgt.

Bei der Projektierung von Chemieanlagen ist ein Einsatz der Bildschirmtechnik u. a. auch bei der Ausrüstungstypenauswahl bzw. bei der Ausrüstungsauswahl vorgesehen. Z. Z. werden dazu folgende Untersuchungen durchgeführt:

1. Untersuchungen zur Methodik der Ausrüstungstypenauswahl vor allem unter dem Gesichtspunkt der Kopplung von Ausrüstungen. Auswahlgrößen sind dabei solche Größen, die den Typ der Ausrüstung qualitativ und quantitativ beschreiben.

2. Untersuchungen zur Methodik der Ausrüstungsauswahl, wobei die Dimensionen der Ausrüstung so zu bestimmen sind, daß ihre Montage und ihr Betreiben bestimmten Zielen entspricht.

Die Kompliziertheit und Komplexität des Problems der Ausrüstungstypenauswahl bzw. Ausrüstungswahl läßt eine geschlossene Programmierung der Problemlösung nur mit sehr großem Aufwand zu. Effektiver ist deshalb die Realisierung eines Dialogsystems, bei dem die nicht programmierbaren Erfahrungen des Menschen mit den programmierbaren Routineprozessen gekoppelt werden können.

Ausrüstungsauswahl und Erarbeitung der Anfragespezifikationen sind voneinander abhängige Tätigkeiten, die beide auf verfahrenstechnischen Vorgaben basieren. Das Ziel besteht darin, diese Tätigkeiten zweckmäßig miteinander zu verbinden.

Anschrift der Verfasser:

Dipl.-Math. Gerhard Jahn,
Ing. Bernd Meyer

VEB Chemieanlagenbau- und
Montagekombinat Leipzig

DDR - 701 Leipzig

Georgiring 1 - 3

Ein Verfahren zur automatischen Eingabe technischer Zeichnungen in DVA

Die gegenwärtig bekannten halbautomatischen Methoden zur Eingabe technischer Zeichnungen in DVA erfordern einen hohen Arbeitszeitaufwand, weil der Mensch zur Auswahl der zu erfassenden geometrischen Informationen erforderlich ist und die Zuordnung von Bedeutungen durch den Menschen vorgenommen werden muß. Das Ziel der automatischen Eingabe ist die Eingabe von Zeichnungen mit geringem personellen Arbeitszeitaufwand.

Das Verfahren verläuft in folgenden Schritten:

- Digitalisierung:

Die kontinuierliche analoge technische Zeichnung wird in ein diskontinuierliches digitales Abbild gewandelt.

- Ermittlung der elementaren Beschreibung:

Aus dem digitalen Abbild der technischen Zeichnung werden einfache geometrische Gebilde (Geradenstücke, Kreisbögen) ermittelt.

- Strukturanalyse:

Die Struktur der technischen Zeichnung wird durch die Bestimmung der in ihr enthaltenen Substrukturen und deren Zusammenhänge ermittelt. Bestimmten Substrukturen werden Bedeutungen zugeordnet.

Das Verfahren wird an einem Beispiel (Eingabe eines Schaltplanes) erläutert.

Eine ausführliche Fassung des Vortrages ist in /1/ enthalten.

Literatur:

/1/Koch, L.: Ein Verfahren zur automatischen Eingabe von technischen Zeichnungen in
DVA. Feingerätetechnik, in Vorbereitung.

Anschrift des Verfassers:

Dipl.-Phys. Lutz Koch

Institut für Nachrichtentechnik

DDR - 116 Berlin

Edisonstr. 63

Anwendung geometrischer Grundoperationen in einem Kleinrechner-Dialogsystem für KRS 4200

Wird bei der Lösung geometrischer Aufgaben an die Benutzung einer Rechenanlage gedacht, dann kann man sich die erforderliche Programmierung des Problems so einrichten, daß bestimmte rechentechnische Bausteine genau den geometrischen Bausteinen, also etwa den Teilaufgaben, in die man derartige Probleme gewöhnlich zerlegt, entsprechen.

So wurden Teilaufgaben, die in umfassenderen Aufgabenstellungen immer wieder auftreten, zusammengestellt und für den KRS 4200 in der Autocodesprache SYPS programmiert. Die Auswahl der Teilaufgaben erfolgte wie in /1/ beschrieben.

Im Ergebnis entstand ein kleines autonomes Dialogsystem, mit dem vielfältige Konstruktionsaufgaben bearbeitet werden können unter der Voraussetzung, daß die Beschreibung mit den Elementen Punkt, Gerade und Ebene auskommt.

1. Analytische Grundoperationen (AGOP)

Die den geometrischen Teilaufgaben entsprechenden rechentechnischen Bausteine wurden aus analytischen Grundoperationen aufgebaut. Ein Aufbau dieser Art ist hinsichtlich der Einsparung von Speicherplatz und bei der Programmierung weiterer Bausteine sinnvoll, da immer wieder gleiche analytische Operationen auftreten.

Durch Betrachtung des Beispiels Schnittpunktbestimmung von Gerade und Ebene wird offensichtlich, daß Bildung des inneren sowie äußeren Produktes unumgängliche Operationen sind (s. /1/). Dieselben Operationen sind auch bei der hierzu dualen Aufgabe, Bestimmung von Verbindungsebene aus Punkt und Gerade, notwendig.

Zur Berechnung der Verbindungsgerade zweier Punkte bzw. Schnittgerade zweier Ebenen benötigt man das äußere Produkt und eine neu eingeführte Operation, die mit "Differenzprodukt" bezeichnet wird (s. /1/, /2/).

Die drei Rechenoperationen inneres Produkt, äußeres Produkt und Differenzprodukt werden auf eine einheitliche Form $R^4 \times R^4 \rightarrow R^4$ gebracht, um

- a) die Organisation mit den Operationen in der Rechenanlage zu vereinfachen; es wird jetzt nur noch mit 4- und 6-Tupeln anstelle von Zahlen, Tripeln, 4- und 6-Tupeln gearbeitet, und 4×4 -Matrizen werden vermieden;
- b) die Freiheiten der zusätzlichen Koordinate bzw. der zusätzlichen Koordinaten zur Anpassung der Operationen an möglichst viele Aufgabenstellungen ausnutzen zu können.

Die oben genannten analytischen Grundoperationen sind im einzelnen die folgenden:

Rechenoperationen

$(x, y) = (x_1y_1 + x_2y_2 + x_3y_3, 0, 0, 0)^T$	Skalarprodukt
$\vec{x} \wedge \vec{y} = (x_0y_3 - y_2x_3, x_3y_1 - y_3x_1, x_1y_2 - y_1x_2)^T$	Kreuzprodukt
$\langle x, y \rangle = (x_0y_0, x_0y_1 - y_0x_1, x_0y_2 - y_0x_2, x_0y_3 - y_0x_3)^T$	Differenzprodukt

Organisationsoperationen

symbolisch	Bedeutung, falls	
	z Punkt oder Ebene	z Gerade ¹
$z = \begin{bmatrix} x \\ y \end{bmatrix}$	$z = (x_0, y_1, y_2, y_3)^T$	$z = (x_1, x_2, x_3, y_1, y_2, y_3)^T$
\bar{z}	$(-z_0, 0, 0, 0)^T$	$(1, z_{-2}, z_{-1}, z_0)^T$
\underline{z}	$(-1, z_1, z_2, z_3)^T$	$(1, z_1, z_2, z_3)^T$
z'	z_0	1
$z \hat{x}$	$(x, 0, 0, 0)^T$ mit x reell	nicht definiert

¹ Um die Rechenoperationen auch für Geraden anwendbar zu machen, werden Geradenkoordinaten folgendermaßen indiziert:

$$g = (g_{-2}, g_{-1}, g_0, g_1, g_2, g_3)^T$$

Bei rein projektiver Betrachtungsweise macht es sich notwendig, Linearkombinationen aus 4-Tupeln zu bilden. Dies leisten die analytischen Grundoperationen nicht unmittelbar, da sie auf Rechnungen im euklidischen, wenn auch projektiv-abgeschlossenen Raum abgestimmt sind.

Mittels der eingeführten Operationen lassen sich die Aufgaben Verbinden und Schneiden folgendermaßen lösen:

- a) Verbindungsgerade z zweier Punkte x und y bzw. Schnittgerade z zweier Ebenen x und y :

$$z = \begin{bmatrix} \langle x, y \rangle \\ \tilde{\langle x, y \rangle} \end{bmatrix}$$

- b) Verbindungsebene von Punkt und Gerade, falls die Gerade in Achsenkoordinaten gegeben ist bzw.

Schnittpunkt von Ebene und Gerade, falls die Gerade in Linienkoordinaten gegeben ist:

$$z = \begin{bmatrix} (x, y) \\ \langle \tilde{y}, \tilde{x}, \tilde{y} \rangle \end{bmatrix} \quad \begin{array}{l} x \text{ Koordinatenvektor von Punkt bzw. Ebene} \\ y \text{ Koordinatenvektor der Geraden} \\ z \text{ beschreibender Vektor des Ergebniselements} \end{array}$$

- c) Verbindungsebene von Punkt und Gerade, falls die Gerade in Linienkoordinaten gegeben ist bzw.

Schnittpunkt von Punkt und Gerade, falls die Gerade in Achsenkoordinaten gegeben ist:

$$z = \begin{bmatrix} (x, y) \\ \langle \tilde{y}, \tilde{x}, \tilde{y} \rangle \end{bmatrix} \quad x, y, z \text{ wie in b)}$$

Um intern Punkte von Ebenen sowie Geraden der einen Koordinatenart von der anderen unterscheiden zu können, wurden folgende Kennzahlen eingeführt:

Gerade in Linienkoordinaten	2
Punkt	1
Element ohne geometrische Interpretation	0
Ebene	- 1
Gerade in Achsenkoordinaten	- 2

In Abhängigkeit von diesen Kennzahlen wird der Algorithmus Verbinden und Schneiden gesteuert sowie der Ergebnistyp festgelegt:

$$K(z) = K(x) + K(y) \bmod 4 \quad \text{mit } K(a) \text{ Kennzahl von } a.$$

2. Geometrische Grundoperationen (GGOP)

Mit Verbinden und Schneiden hat man eine wichtige geometrische Grundoperation erhalten. Eine zweite häufig benötigte ist, wie in /1/ gezeigt ist, das Ausüben der absoluten Polarität, die rechentechnisch durch Ändern der Kennzahl zu realisieren ist und aus diesem Grund (insbesondere für den der projektiven Geometrie unkundigen Nutzer) treffend Umdeuten genannt wurde. Umdeuten wird durch Multiplikation der Kennzahl mit -1 bewerkstelligt.

Hinsichtlich der Programmierung an der Rechenanlage wird bei der Lösung von Aufgaben mittels Verbinden und Schneiden (Operationssymbol ' \ast ') und Umdeuten (Operationssymbol ' Π ') in einer höheren Ebene gearbeitet.

Es ist dabei z. B. die Programmierung mit den Operationen $+$, $-$, die 0. Ebene, und bei Verwendung analytischer Grundoperationen wird in der 1. Ebene gearbeitet; bei Lösung geometrischer Aufgaben mittels geometrischer Grundoperationen befindet man sich in einer 2. Ebene.

Für die Lösung komplexerer geometrischer Aufgaben erweist es sich als sinnvoll, gewisse aus geometrischen Grundoperationen zusammengesetzte Aufgaben wie Grundaufgaben zu behandeln, um zu einer echt geometrischen Denkweise zu gelangen sowie bei der Arbeit an der Rechenanlage und in der Programmierung Zeit zu sparen. Das bedeutet, Aufgaben, die sich durch Hintereinanderausführung von ' \ast ' und ' Π ' lösen lassen, werden programmiert, also die Sprünge zu den sie realisierenden Makros in SYPS aufgeschrieben. Damit werden neue geometrische Bausteine erhalten, wobei sich die Arbeitsweise mit diesen in nichts von der mit geometrischen Grundoperationen unterscheidet.

3. Arbeitsweise mit dem System

Das erarbeitete System kann man als Dialogsystem bezeichnen, weil in Abhängigkeit von Zwischenergebnissen der weitere Rechengang bestimmt werden kann. Dieser Dialog erfolgt über die Bedienschreibmaschine am KRS. Hier werden bestimmte Kommandos ausge-

führt, damit ist gemeint, daß durch einen Namen, das können vier beliebige, aber vorher im System festgelegte Zeichen sein, der Ausführungsblock, der durch dieses Kommando repräsentiert wird, abgearbeitet wird und ein Ergebnisdruckbild auf der Schreibmaschine erscheint. Die Abarbeitung eines Ausführungsblockes bedeutet Ausführung einer geometrischen Operation.

Bei Verwendung geometrischer Operationsebenen braucht der Nutzer den analytischen Hintergrund des Systems nicht zu kennen. Das Wesentliche der Arbeitsweise besteht darin, daß der Dialogführende ähnlich wie bei der Konstruktion einer technischen Zeichnung die Situation im Raum ständig vor Augen haben muß. Eventuell arbeitet er mit einer Skizze am Kleinrechner. Bei mehrmaliger Verwendung ein und derselben von ihm zusammengesetzten Aufgabe kann er leicht einen neuen Baustein programmieren, um ein wiederholtes und damit ermüdendes Erarbeiten dieser Aufgabe zu umgehen.

Zur Lösung unübersichtlicher Aufgaben macht es sich notwendig, dem Nutzer fertige Bausteine in die Hand zu geben. Je komplizierter diese Bausteine sind, desto größer ist auch die zu verwaltende Datenmenge und die dazugehörige Datenstruktur wird komplizierter, so daß an einer bestimmten Stelle ein Kleinrechner nicht mehr genügt.

Im gegenwärtigen Entwicklungsstand des Systems sind noch nicht alle aufbereiteten Teilaufgaben programmiert; da das Prinzip klar ist, soll das auch nicht Ziel sein, sondern im Hinblick auf die baldigste Verwendung eines Displaygerätes ist der nächste Schritt das Sichtbarmachen der Zwischenergebnisse am Bildschirm, um hier den Konstruktionsgang direkt verfolgen zu können.

Literatur:

- /1/ Geise, G.: Ein Kalkül für geometrische Aufgaben. Vortrag auf der Tagung Digitalgraphik in Rostock - Warnemünde, Nov. 1976; im gleichen Heft.
- /2/ Krause, H.: Rechentechnische Aufbereitung geometrischer Grundaufgaben für Punkte, Geraden und Ebenen. Diplomarbeit, TU Dresden, 1976.

Anschrift des Verfassers:

Dipl.-Math. Heidrun Krause

Technische Universität Dresden
Sektion Mathematik

DDR - 8027 Dresden; Mommsenstr. 13

Der Einsatz der Digitalgraphik bei der automatischen Fertigung von Leiterplatten

Der Einsatz eines aktiven Bildschirmgerätes bei der Konstruktion von Leiterplatten erweist sich insbesondere für dieses Problem als äußerst vorteilhaft, weil die Algorithmisierung der Leiterplattenkonstruktion noch nicht so weit entwickelt ist, daß die Konstruktion ohne Eingriff des Menschen durchgeführt werden kann. Sehr oft treten Besonderheiten auf, die teils elektrischer, teils topologischer Natur sind. Durch die Arbeit an einem aktiven Bildschirmgerät wird der Konstrukteur in die Lage versetzt, diese Besonderheiten unmittelbar während der Programmabarbeitung zu berücksichtigen.

Außer der Neukonstruktion einer Leiterplatte existiert das Problem der Änderung einer bereits konstruierten Leiterplatte. Im Programm ist es möglich, die Konstruktionsergebnisse von Leiterplatten auf einem Magnetband zu archivieren. Diese Informationen können zu einem beliebigen Zeitpunkt in den Rechner übertragen werden. Am Bildschirm werden dann die entsprechenden Änderungen vorgenommen.

Das am Institut für Schiffbau, Rostock, entwickelte graphische interaktive Programmsystem zur Konstruktion von Leiterplatten, abgekürzt GISKOL, geht aus von einem üblichen Stromlaufplan, einer Bauelementeliste und den Abmessungen der Leiterplatte. Die Bauelemente werden vom Programm auf der Leiterplatte angeordnet und die entsprechenden Leitungen verlegt. Die erste Phase wird als Platzierung und die zweite als Trassierung bezeichnet.

Der Algorithmus für die Platzierung der Bauelemente wurde vom Institut für Regelungstechnik, Berlin, übernommen. Der Dialogteil zur Arbeit mit dem Bildschirm wurde zusätzlich programmiert. Der Algorithmus beruht auf der Festlegung, daß sich Bauelemente anziehen, wenn sie miteinander elektrisch verbunden sind. Sie stoßen sich ab, wenn sie

einen bestimmten Abstand unterschreiten. Außerdem wirkt eine Randabstoßung auf alle Bauelemente. Die Bauelemente werden als Massepunkte betrachtet. Nach den Gesetzen der Kinetik ergibt sich für N Bauelemente ein System von N linearen, voneinander unabhängigen homogenen Differentialgleichungen. Die Lösung des Differentialgleichungssystems erfolgt nach dem Seidelschen Iterationsverfahren. Das theoretische Modell wurde der Literaturquelle: Matchjuchin, N., I.: "Die Anwendung von Rechnern beim Entwurf digitaler Geräte", Moskau 1968, entnommen.

Der Bildschirmteil bietet dem Konstrukteur auf dem Schirm folgende Informationen:

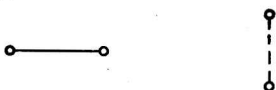
- Die Bauelemente sind auf dem Schirm mit ihrer Rechteckfläche, den Anschlußpunkten und dem Kurzzeichen nach jedem Iterationsschritt abgebildet.
- Der Zykluszähler der Iteration und die Größe ZC , die die Summe der Koordinatenänderungen aller Bauelemente darstellt, werden auf dem Schirm dargestellt.
- Die elektrischen Beziehungen der Bauelemente untereinander sind durch Linien mittels Tastendruck darstellbar.

Der Konstrukteur kann folgende Aktionen auslösen:

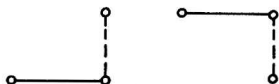
- Unterbrechen der Iteration,
- Verschieben von Bauelementen mittels Lichtstift und Rollkugel,
- "Festsetzen" oder "Losbinden" von gewünschten Bauelementen,
- Änderung der Konstanten des Differentialgleichungssystems.

Mit diesen Dialogmöglichkeiten kann der Konstrukteur die Platzierung der Bauelemente verfolgen und korrigierend einwirken. Ungünstiges Schwingverhalten der Bauelemente wird erkannt und kann durch Änderung der Konstanten des Differentialgleichungssystems beeinflußt werden. Vorhandene Überschneidungen am Ende der Platzierung können durch Verschieben der Bauelemente beseitigt werden. Muß die letztgenannte Aufgabe ohne Bildschirmgerät erledigt werden, ist ein umfangreiches Programm erforderlich mit der erheblichen Laufzeit von 20 - 30 Min.

Geraden



Winkelverbindungen mit einer Durchkontaktierung



Winkelverbindungen mit zwei Durchkontaktierungen

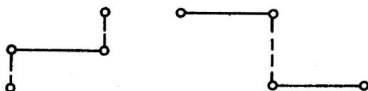


Abb. 1: Leiterzugmuster

Der Algorithmus für die automatische Trassierung wurde vom Institut für Nachrichtentechnik, Berlin, entwickelt. Er benutzt eine Potentialliste als Ausgangspunkt. Die Potentialliste enthält die Koordinaten aller Anschlußpunkte, die auf gleichem Potential liegen. Hergestellt wird sie aus der eingegebenen Potentialliste unter Berücksichtigung der Platzierungsergebnisse, die die konkreten Koordinaten liefern. Die Potentialliste wird in eine Verbindungsliste umgeformt, die einen Baum mit möglichst kurzen Kanten je Potential darstellt. In ihr sind je zwei zu verbindende Anschlußpunkte aufgeführt. Zur Verbindung dieser Anschlußpunkte werden unterschiedliche Leiterzugmuster vom Programm durchprobiert (s. Abb. 1).

Die Erlaubnis zur Verlegung eines Leiterzugmusters wird nach Prüfung der bereits vorhandenen Leiterzugstruktur in der rechnerinternen Leiterplattendarstellung erteilt. Nach dem Durchprobieren aller Möglichkeiten bleiben in der Regel nichtverlegbare Verbindungen übrig. Es sind ca. 3 % aller Verbindungen. Die nicht verlegten Verbindungen werden durch Blinken der Anschlußpunkte angezeigt. Der Konstrukteur hat dann die Möglichkeit, diese mit dem Bildschirm manuell zu verlegen.

Das Programm ist bei der automatischen/manuellen Arbeitsweise für die Konstruktion von gemischt bestückten Zweiebenenleiterplatten einsetzbar. Werden nur die manuellen Programmteile zur Platzierung und Trassierung benutzt, können auch Einebenenleiterplatten konstruiert werden.

Als Ergebnis der Berechnung erhält der Konstrukteur die Fertigungsunterlagen der Leiterplatte in Form von Zeichnungen, die auf einer Präzisionszeichenmaschine hergestellt werden.

Folgende Zeichnungen werden ausgegeben:

- Druckstockzeichnung,
- Bearbeitungszeichnung,
- Beschriftungsaufdruckzeichnung,
- Kontrollzeichnung.

Infolge des hohen manuellen Aufwandes bei der Konstruktion von Leiterplatten und der Genauigkeitsforderungen der Leiterplattenhersteller an die Fertigungsunterlagen lassen sich durch die automatische Konstruktion von Leiterplatten mit Hilfe der EDV folgende Vorteile gegenüber manueller Konstruktion angeben:

- Höhere Genauigkeit der Fertigungsunterlagen,
- Reduzierung des Bearbeitungsaufwandes um 2 bis 3 Wochen,
- Verringerung der Konstruktionskosten um ca. 200,- M pro Leiterplatte,
- Reduzierung des Bearbeitungsaufwandes bei nachträglichen Veränderungen,
- Fertigung von Leiterplatten mit den Schwierigkeitsgraden 4 und 5.

Diese Vorteile wirken sich besonders bei Leiterplatten aus, die in geringen Stückzahlen gefertigt werden.

Anschrift des Verfassers:

Dipl.-Ing. Peter Kusserow

Institut für Schiffbau
Rechenzentrum

DDR - 25 Rostock

Doberaner Straße 110/111

Rechnerinterne Darstellung technischer Gebilde (RID)

1. Einleitung

Die Forderung nach ständiger Steigerung der Produktivität verlangt die konsequente Anwendung moderner, wissenschaftlich durchdrungener Verfahren auf allen Gebieten, was im Bereich der unmittelbaren Produktion vor allem durch ständige Erhöhung der Mechanisierung und Automatisierung erreicht wird. Im Maschinenbau z. B. heißt das, daß der Einsatz von NC-Maschinen und Prozeßrechnern immer mehr an Bedeutung gewinnt. Das erfordert jedoch wiederum eine prinzipiell neue Art der Arbeitsvorbereitung, indem mit der Konstruktion beginnend alle Unterlagen in eine Form gebracht werden, die später vom Rechner bzw. NC-Teil akzeptiert wird. Außerdem kann schon auf dieser Stufe, der Konstruktion, die Anwendung von Rechenanlagen eine erhebliche Verringerung des Aufwandes bringen, indem der Konstrukteur von der EDVA weitgehend von routinemäßigen Arbeiten, wie der Standardermittlung, dem Ausarbeiten von Stücklisten, der Zeichnung und Bemaßung entlastet wird. Ziel der Automatisierung der technischen Produktionsvorbereitung ist ein mehr oder weniger komplexer Rechnereinsatz in Konstruktion und Technologie. Dazu ist eine rechnerinterne Darstellung der geometrischen, technisch-organisatorischen und bemaßungstechnischen Sachverhalte technischer Gebilde erforderlich.

Die Untersuchungen und praktischen Arbeiten an der Sektion Rechentechnik und Datenverarbeitung der Technischen Hochschule Karl-Marx-Stadt, beschränken sich zunächst auf einfache technische Gebilde, also auf Einzelteile und Baugruppen. Das zu erarbeitende System für die Verarbeitung dieser technischen Gebilde (externes Beschreiben und Eingabe, internes Verarbeiten bis zur Ausgabe von Steuerinformationen für graphische Geräte) ist durch starke Orientierung auf die Spezifik des Werkzeugmaschinenbaues und durch die Behandlung aller Informationen vom Einzelteil, also Geometrie, Bemaßung

und technisch-organisatorische Parameter, gekennzeichnet. Das Ziel der Arbeiten ist zunächst die technische Zeichnung, da dies die vorherrschende graphische Darstellungsart im Werkzeugmaschinenbau ist.

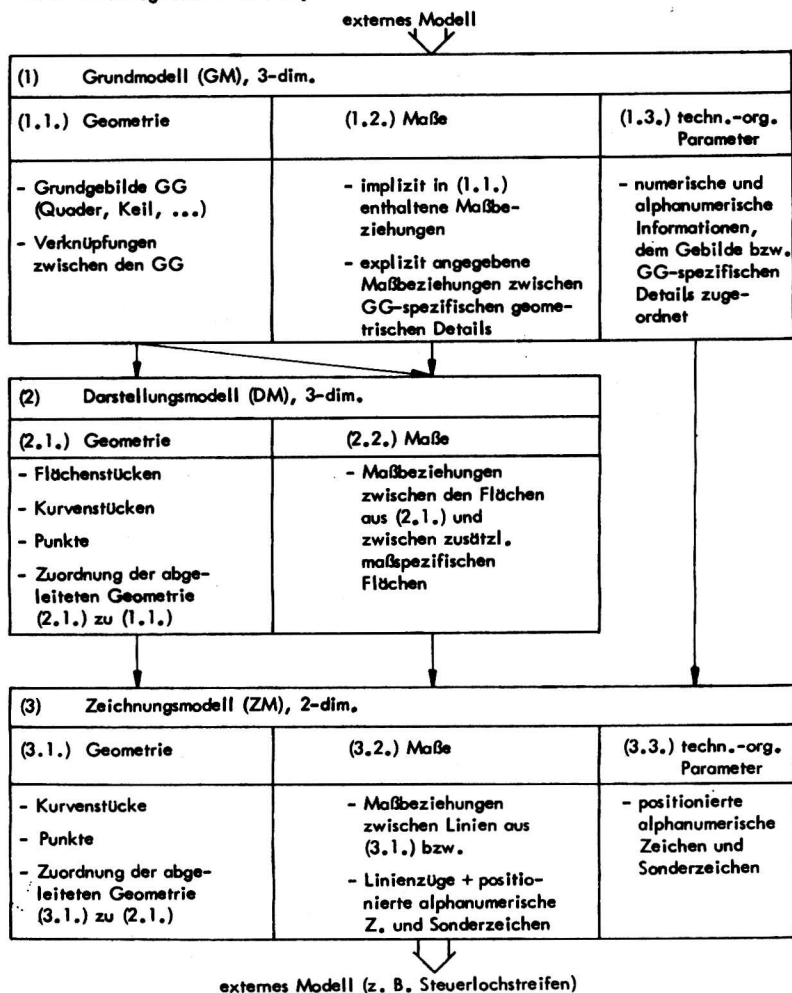


Abb. 1: Übersicht über die Modelle und ihre Beziehungen

2. Bemerkungen zum Gesamtkonzept RID

Beim Lösen der Darstellungsprobleme (perspektivische Darstellungen, technische Zeichnungen) wird mit drei relativ eigenständigen Modellformen gearbeitet:

- (1) Grundmodell (GM)
- (2) Darstellungsmodell (DM)
- (3) Zeichnungsmodell (ZM).

Das Grundmodell ist Ausgangspunkt jeglicher internen Verarbeitung und Ausgabe. Beschreibungen bzw. Änderungen der Gebilde sind nur hier möglich. Das Darstellungsmodell und Zeichnungsmodell sind für die graphische Ausgabe aufbereitete und explizit gespeicherte Formen. Ihre Erzeugung geschieht automatisch und ist nur bezüglich Vollständigkeit (z. B. abgerüsteter Informationsgehalt durch Beschränkung auf Geometrie) und der geforderten Darstellungsvarianten von "außen" beeinflussbar. Alle drei Modellformen sind dreigeteilt, wobei nur der Geometrieteil eigenständig existieren kann. Maße und technisch-organisatorische Parameter können bei Bedarf einzeln oder beide an den Geometrieteil angelagert werden, was auch erst nachträglich erfolgen kann. Abb. 1 gibt eine Übersicht über die Spezifika der Modelle und Beziehungen zwischen ihnen.

2.1. Das Grundmodell

Das Grundmodell ((1) in Abb. 1) ist eine kompakte, der externen Beschreibung sehr naheliegende Modellform. Es enthält den vollständigen Informationsgehalt im wesentlichen unaufbereitet bzgl. irgendeiner nachfolgenden Verarbeitungsaufgabe.

Die Geometrie der Einzelteile ((1.1) in Abb. 1) wird mit Volumenbausteinen und der Mengentheorie ähnlichen Operationen beschrieben. Ausgangspunkt der Beschreibung ist eine Anzahl elementarer geometrischer Körper mit einer homogenen Massenverteilung, die die Menge der Grundgebilde darstellen.

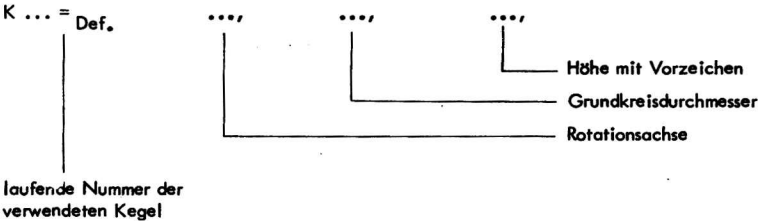
Als Grundgebilde (GG) werden zunächst verwendet:

Quader (Q), Keil (L), Zylinder (Z), Kegel (K), Halbraum (H). Jedes Grundgebilde hat ein grundgebildeeigenes Koordinatensystem (KSGG), in dem es durch Parameterangaben eindeutig beschrieben wird. Dabei geben Formparameter die Abmessungen der Grundgebilde und Lageparameter die Lage des Grundgebildes im KSGG an. Durch Festlegung

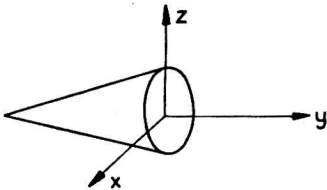
einer Reihenfolge in den Parameterangaben lässt sich die Beschreibung formal durch Definitionsgleichungen erreichen.

Beispiel Kegel: Der Ursprung liegt im Mittelpunkt des Grundkreises. Die Formparameter sind der Grundkreisdurchmesser d und der Betrag der Kegelhöhe h . Die Lageparameter sind die Angabe einer Achse x , y oder z , mit der die Kegelachse zusammenfällt und das Vorzeichen der Kegelhöhe für die Richtung der Ausdehnung des Kegels.

Definitionsgleichung



z. B.



$K2 = \text{Def. } y, 2, -4$

Das geometrische Leistungsvermögen hängt von dem zugelassenen Grundgebildespektrum ab. Zunächst wird mit diesem minimalen Spektrum gearbeitet, welches ohne prinzipielle Schwierigkeiten erweitert werden kann. Die Grundgebilde haben i. a. noch keinen direkten Bezug zu technischen Belangen.

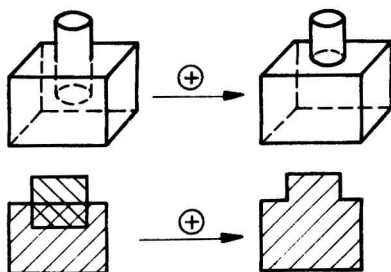
Ein Teilgebilde (TG) ist eine Kombination von GG mit ständig gleichen Lagebeziehungen zueinander. Das technische Äquivalent sind Formelemente.

Ein Gebilde (G) ist eine Kombination von verschiedenen GG und TG bzw., auch nur GG oder nur TG. Gebilde sind vollständigen Einzelteilen zuzuordnen. Grundgebilde, Teilgebilde und Gebilde werden im weiteren als Volumenkörper bezeichnet.

Als absoluter Bezug dient das gebildeeigene Koordinatensystem (KSG), in dem die Lage der Grundgebilde und Teilgebilde zueinander durch Translation und Rotation beschrieben werden. Verknüpfungen: Für den Aufbau von Teilgebilden und Gebilden werden die folgenden Verknüpfungsoperationen festgelegt:

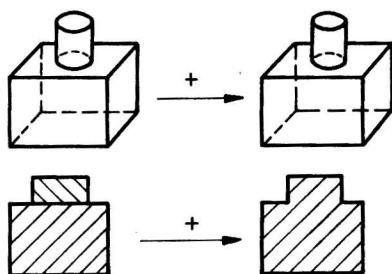
Begriff	Symbol	Bedeutung
Addition 1	\oplus	Addition zweier Volumenkörper im Sinne einer mengentheoretischen Vereinigung, wobei sich die Volumenkörper auf Grund ihrer Lagebeschreibung überschneiden können.

Bsp.:



Begriff	Symbol	Bedeutung
Addition 2	+	Spezialfall der Addition \oplus , bei dem sich die Volumenkörper nur in den Flächenstücken berühren dürfen (keine Überschneidungen).

Bsp.:

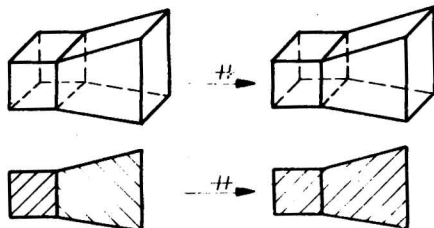


Addition 3

$\#$

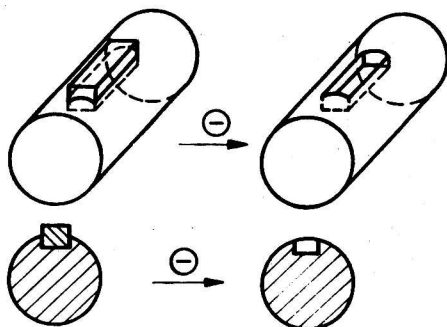
Zwei Volumenkörper, die sich auf Grund der Lagebeschreibung "berühren", werden durch diese Operation als "fest verbunden" betrachtet, obwohl keine Flächenverschmelzung im Sinne der Mengentheorie erfolgt. Diese Verknüpfung dient zur Erzeugung von Gebilden im Sinne von inhomogenen Einzelteilen (z. B. Verbindung durch Schweißen).

Bsp.: Geschweißte Hebel



Begriff	Symbol	Bedeutung
Subtraktion 1	\ominus	Subtraktion im Sinne einer mengentheoretischen Differenz zwischen zwei Volumenkörpern, wobei der zu subtrahierende Volumenkörper nicht vollständig enthalten sein muß.

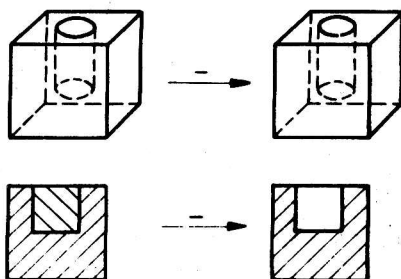
Bsp.:



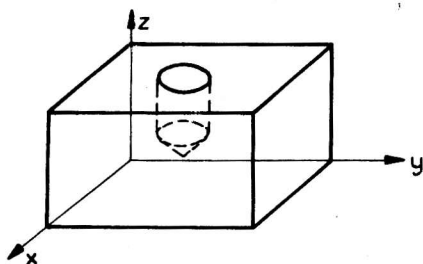
Subtraktion 2

Spezialfall der Subtraktion \ominus , bei dem sich der zu subtrahierende Volumenkörper vollständig im anderen Körper befindet und ihn in Flächenstücken berührt.

Bsp.:



Beispiel: Es ist folgendes Gebilde zu erzeugen:



mit GG1: Quader
GG2: Zylinder
GG3: Kegel

Die Geometriebeschreibung ist durch

$$G = GG1 - TG$$

$$TG = GG2 + GG3$$

gegeben.

Eine Maßstruktur ((1.2.) in Abb. 1) ist als Standardmaßstruktur in der Geometriebeschreibung der Einzelteile im Grundmodell implizit durch Größen und Strukturangaben enthalten. Sie bedarf aber noch einer Interpretation bzw. Bearbeitung zur Vereindeutigung, da sie gewöhnlich nicht den Erfordernissen genügen wird. So können z. B. durch Zusammenfallen von Koordinatenebenen von Volumenkörpern bei formaler Zuordnung Parameter - Maß zu jenen Ebenen mehrere Maße hinführen. Es sind dann Strategien zum Streichen von Maßen anzuwenden.

Durch Hinzunahme weiterer Informationen kann die aus der Beschreibung der Geometrie abgeleitete Standardmaßstruktur zu einer Maßstruktur verändert werden, welche enthält:

- alle Maße,
- für jedes Maß: Art des Maßes, Größe und evtl. Toleranzen, Flächenbezüge (Bezugsfläche, bezogene Fläche), Bezugselementbezüge,
- Angabe der Flächen (Ebenen-, Zylinder-, Kegelflächen),
- Angabe der Bezugselemente.

Die Maßstruktur spiegelt die notwendigen Zusammenhänge der Maße am Einzelteil wider, wobei Hauptbezugsflächen bzw. Hauptbezugsachsen festgelegt werden. Die hauptsächlichen Bemaßungsrichtungen sind die Koordinatenachsenrichtungen. Die logische Abfolge der Bemaßung beginnt bei den Hauptbezugsebenen und schreitet entsprechend der maßlichen Beziehungen voran. Aus Darstellungsgründen werden Maße zu Kanten, Achsen und speziellen Punkten, als Maße zu Ebenen aufgefaßt, die diese Elemente enthalten.

Um im Beschreibungssystem den Informationsgehalt einer technischen Zeichnung zu erreichen, müssen neben der Beschreibung der Geometrie und der Bemaßung, Informationen über technische und organisatorische Parameter ((1.3.) in Abb. 1) enthalten sein. Für die rechnerinterne Beschreibung ist es notwendig, die Parameterangaben entsprechend den Bezugsgrößen wie folgt vorzunehmen:

Parameter	Bezug
Organisatorische Parameter (global), z. B. Benennung, Zeichnungsnummer, Bearbeiter	Gebildegebunden
technische Parameter (global), z. B. Werkstoff bzw. Halbzeug, Oberflächenangaben	Gebilde- bzw. Teilgebilde- gebunden
technische Parameter (lokal), z. B. Form- und Lageabweichung, Angaben für standardisierte Modifikatoren wie Gewindeangaben, Zentrier- bohrungen.	Flächengebunden

2.2. Das Darstellungsmodell

Da es sehr schwer ist, aus dem Grundmodell direkt die Steuerinformationen für ein graphisches Ausgabegerät zu ermitteln, wird mit Zwischenmodellen gearbeitet. Das erste davon ist das Darstellungsmodell ((2) in Abb. 1). Es enthält die Gebildegeometrie ((2.1.) in Abb. 1) so aufbereitet, daß Darstellungsverfahren (zur Projektion der 3-dimensionalen Gebilde in eine Bildebene) günstig ansetzen können. Dazu wird die Geometrie

eines Einzelteiles durch begrenzende Flächen-, Linien- und Punktelemente beschrieben, und es sind Bezüge zu einzelnen Flächen und Kanten zur Angabe von Maßen und technischen Parametern möglich (Zuordnungsteil). Das Darstellungsmodell wird, wie schon das Grundmodell, in die drei relativ getrennten Informationsgruppen, geometrische Informationen, Maßinformationen und technisch-organisatorische Informationen unterteilt, wobei die technisch-organisatorischen Informationen ohne Anwendung von Überführungsalgorithmen aus dem Grundmodell übernommen werden können, da sich eine Zwischenform in diesem Modell nicht als nötig erweist.

Für die vier Arten der Beschreibungstücke werden die vier Listen, Flächenliste, Linienliste, Punktliste und Zuordnungsliste, eingerichtet, die bei der Überführung eines Volumenkörpers aus dem Grundmodell, d. h. bei einem schrittweisen Aufbau des Gebildes im Darstellungsmodell, ausgefüllt und ergänzt werden, wobei für jedes Element ein Datensatz bereitgestellt wird. Strukturelle Beziehungen werden mittels Verweisen realisiert.

Die im Grundmodell enthaltene Maßstruktur (2.2.) in Abb. 1) bedarf einer weiteren Aufbereitung, um eine aus einer Menge von Maßen bestehenden, strukturierten und einer weiteren Verarbeitung zur Darstellung in technischen Zeichnungen zugänglichen maßlichen Beschreibung zu erhalten. Diese maßliche Beschreibung entsteht im Darstellungsmodell. Voraussetzung dazu ist neben dem vorliegenden Grundmodell die vollständig abgeleitete Geometrie im Darstellungsmodell. Die vorliegenden Listen der Flächenstücke, Linienstücke, Punkte und Zuordnungen werden durch maßspezifische Elemente erweitert.

2.3. Das Zeichnungsmodell

Das Zeichnungsmodell (3) in Abb. 1) nimmt die aus dem Darstellungsmodell durch spezielle Abbildungsverfahren erzeugten 2-dimensionalen Bilder auf und ist analog zum Grundmodell dreigeteilt. Es enthält somit die Informationen für das Anfertigen einer vollständigen technischen Zeichnung. Die für eine technische Zeichnung geltenden Standards werden mit dem Zeichnungsmodell realisiert, insbesondere gilt das für die verschiedenen Darstellungsarten (z. B. Rißdarstellung, perspektivische Darstellung, Schnittdarstellung).

Ausgehend vom Zeichnungsmodell erzeugt ein Ausgabeprogramm direkt Steuerlochstreifen für eine Zeichenmaschine. Bei entsprechend vorhandenen Ausgabeprogrammen kann aber auch mit anderen graphischen Ausgabegeräten gearbeitet werden.

Abschließend noch einige Bemerkungen zur Eingabe: Für die z. Z. anstehende Testphase wurde ein Eingabeprogramm geschrieben, für welches die Einzelteilbeschreibung extern in einem Tableau (festformatiert) anzugeben ist. Für die spätere praktische Anwendung des Systems wird eine nutzerfreundlichere Form geschaffen, für deren Konzipierung Erkenntnisse aus der internen Verarbeitung erwartet werden.

3. Bemerkungen zur programmtechnischen Realisierung

Alle Modellformen werden mit Hilfe der allgemeinen Datenstruktur - Ringlistenstruktur RLS - realisiert. Deshalb erfolgt die Arbeit mit und in den Modellen auf der Grundlage ein- und derselben Menge von RLS-Programmen normiert. Dieses Konzept ermöglicht eine gute Übersicht über die Modellrealisierungen und garantiert gute Erweiterungs- und Änderungsmöglichkeiten. Das Programmpaket zur automatischen Erstellung technischer Zeichnungen ist streng modular aufgebaut und setzt sich aus folgenden Programmkomplexen zusammen:

Name des Komplexes	Leistung
G - GD	Ableitung des Geometrieteiles des DM aus dem GM
G - DZ	" " " " ZM aus dem DM
M - GD	" " Bemaßungsteiles " DM aus dem GM
M - DZ	" " " " ZM aus dem DM
P - GZ	" der techn.-org. Parameter im ZM a. d. GM
A - ZZ	Gestaltung der tech. Zeichnung, also Zuordnen der Maße zu den Rissen, Zuordnen der techn.-org. Parameter zu den Rissen, Anordnen der Risse auf dem Zeichenblatt
A - US	Ausgabe des ZM als Steuerinformation einer Zeichenmaschine.

Als besonderer Schwerpunkt beim Anlegen des Programmpaketes ist die Erweiterungsfähigkeit angesehen worden, insbesondere wurden die Modelle dialogfähig konzipiert. Deshalb ist es möglich, daß zu gegebener Zeit mit diesen Modellen und Verarbeitungsmodul ein Dialogsystem erstellt werden kann.

4. Schlußbemerkungen

Die Überföhrungsalgorithmen sind fast vollständig erarbeitet und zum Teil programmiert. Ein Teil der Programme befindet sich in der Testphase. Es ist vorgesehen, bis Ende 1978 die Implementation einer 1. Ausbaustufe auf dem Rechner ES 1020 abzuschließen.

Neben der automatischen Zeichnungserstellung zeichnen sich bereits weitere konkrete Aufgabenstellungen ab, z. B. Steuerlochstreifenkontrolle für NCM mit Hilfe des konzipierten Systems.

Bei der Erarbeitung des Systems der rechnerinternen Darstellung technischer Gebilde wird sehr eng mit Konstrukteuren aus der Sektion Maschinenbauelemente der THK zusammenge- arbeitet, um einerseits die Ergebnisse der digitalen Modellierung sofort der kompetenten Kritik des Konstrukteurs zu unterziehen und andererseits Anforderungen seitens der Kon- struktion aus theoretischen Untersuchungen und der Lösung praxisnaher Aufgaben zu er- halten.

Literatur:

- / 1/ Braid/Lang: Rechnergestützte Entwicklung mechanischer Komponenten mit Volumenbau- steinen. Beiträge zu PROLAMAT 73, Budapest 1973
- / 2/ Kurth: Compac - Ein System zur rechnerorientierten Werkstückbeschreibung. ZWF 68, Heft 2 u. 3 (1973).
- / 3/ Debler: Beitrag zur rechnergestützten Verarbeitung von Werkstückinformationen in pro- duktionsbezogenen Planungsprozessen. Diss., TU Berlin, 1973.
- / 4/ Krause/Vassilacopoulos: Systeme zur digitalen, rechnerinternen Darstellung technischer Gebilde. Konstruktion 23, Heft 12 (1971).
- / 5/ Forschungsberichte der Forschungsgruppe "Rechnerinterne Darstellung technischer Ge- bilde" (RID) der TH Karl-Marx-Stadt.

Anschrift des Verfassers:

Dr. Reinhard Lober
TH Karl-Marx-Stadt
Sektion Rechentechnik und Datenverarbeitung
DDR 901 Karl-Marx-Stadt; Postfach 964

Eine Fachsprache für die Konstruktion von Werkzeugmaschinengestellen

Die Konstruktion von Baugruppen im Werkzeugmaschinenbau erfordert recht unterschiedliche Arbeitsetappen. Im wesentlichen geht es dabei um das (geometrische) Gestalten und Dimensionieren unter Berücksichtigung der geforderten technischen (und technologischen) Parameter. Dieser Prozeß verläuft heute noch vielfach äußerst diskontinuierlich. So wird das Gestalten in der Regel manuell mit konventionellen Hilfsmitteln durchgeführt. Das Dimensionieren bei Beachtung der technischen Parameter erfolgt zumindest für Teilaufgaben in zunehmendem Maße mit Hilfe der elektronischen Datenverarbeitung. Die erhaltenen Resultate werden wieder meist manuell ausgewertet und finden z. B. bei dem (geometrischen) Gestalten Berücksichtigung.

Ein wesentlicher Beitrag für die Rationalisierung des Konstruktionsprozesses wäre es, sowohl das (geometrische) Gestalten als auch das Dimensionieren unter Zuhilfenahme geeigneter rechentechnischer Hilfsmittel in einem durchgängigen Prozeß zu vereinigen. Dabei sollte die Inanspruchnahme derartiger Konstruktionshilfen den Gedankengang des Konstrukteurs nicht stören. Im Gegenteil, sie müßten so beschaffen sein, daß er in seiner eigenen Terminologie und Denkweise vorgehen kann und damit durch den Rechner echt unterstützt wird.

Dieser Forderung käme eine problemangepaßte Eingangssprache für eine entsprechende EDVA entgegen, die es dem Konstrukteur erspart, graphische Informationen zur Beschreibung meist dreidimensionaler Objekte in digitale Worte umsetzen.

Im folgenden soll der prinzipielle Aufbau einer derartigen problemorientierten Programmiersprache speziell für die Konstruktion von Gestellen des Werkzeugmaschinenbaus beschrieben werden.

Diese Arbeiten sind erste Ergebnisse einer langfristigen Zusammenarbeit zwischen den Sektionen Fertigungstechnik und Werkzeugmaschinen (Bereich Fertigungsmittel!) und Mathematik (Bereich Math. Kybernetik und Rechentechnik) der TU Dresden (s./3/, /4/, /5/).

1. Anforderungen an die Fachsprache

Neben den allgemeinen Forderungen, die der Nutzer an die problemorientierte Programmiersprache (Fachsprache) stellt, wie flexibler Anwendungsbereich, leichte Erlernbarkeit, geringes Fehlerrisiko, gute Korrekturmöglichkeiten und möglichst geringer Aufwand bei der Objektbeschreibung, muß sie selbstverständlich durch Wahl geeigneter Operanden und Operatoren den fachspezifischen Aufgabenstellungen gerecht werden. Die bei der Konstruktion von Gestellen zu lösenden Teilaufgaben kann man wie folgt unterteilen (s. Abb. 1):

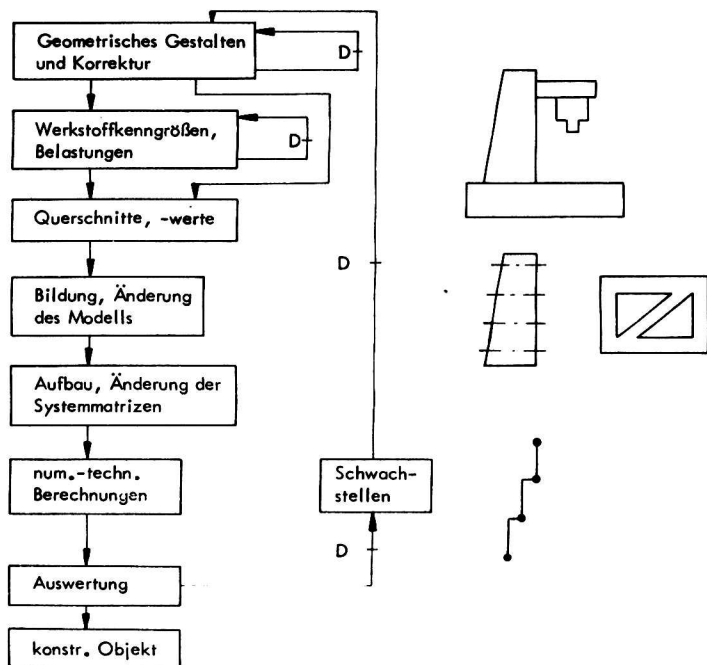


Abb. 1: Teilaufgaben bei der Konstruktion von Werkzeugmaschinengestellen

- Konstruktive (geometrische) Beschreibung des Objektes,
- Berechnung von Kennwerten, die aus der geometrischen Gestalt und entsprechenden Werkstoffkenngrößen resultieren,
- Bildung bzw. Änderung des Modells,
- Numerisch-technische Berechnungen unter Beachtung der technischen und Modell-Parameter,
- Auswertung der Ergebnisse und eventuelle geometrische Änderung des Objektes.

Die zu entwerfende Fachsprache muß demnach die geometrische Beschreibung der Objekte und die Ausführung der numerisch-technischen Berechnungen gewährleisten. Außerdem muß sie in der Lage sein, die Verbindung zwischen den (geometrischen) Objekten und den numerisch-technischen Algorithmen herzustellen. Weiterhin sollten bereits konstruierte Teilobjekte wieder verwendet werden können, um dem Nutzer einen möglichst hohen Komfort bereitzustellen.

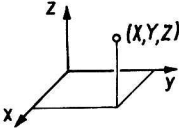
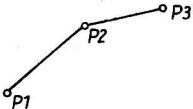
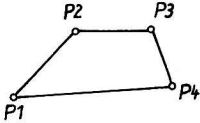
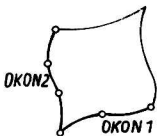
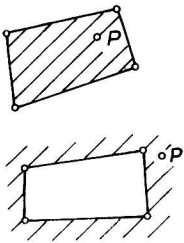
2. Aufbau der Fachsprache

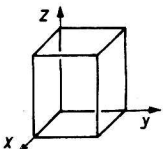
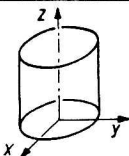
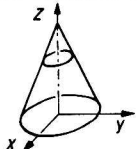
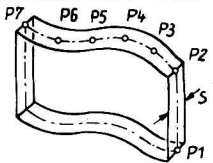
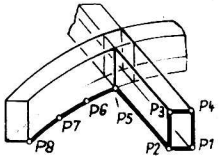
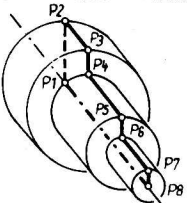
Der syntaktische Aufbau der Fachsprache erfolgte analog der Systemsprache des "Digitalgeometrie-orientierten Systems DIGOS" (s. /1/). In ihrer ersten Ausbaustufe beschränkt sie sich im wesentlichen auf die (geometrisch) konstruktive Beschreibung der Objekte und auf die Wechselbeziehung zwischen den geometrischen Parametern und den numerisch-technischen Algorithmen.

Als Operanden der Fachsprache dienen Konstruktionselemente (Formelemente), die ähnlich wie in ALGOL 60 vereinbart werden müssen, durch Operatoren in Ausdrücken verknüpft werden können und in Anweisungen Verwendung finden. Ein Unterprogrammkonzept erleichtert die Behandlung oft vorkommender Teilobjekte. Die Auswahl der Konstruktionselemente und der Operatoren erfolgte entsprechend den in der Gestellbautechnik bestehenden Anforderungen. Dabei wurde stets darauf geachtet, daß zur Charakterisierung der Operanden und Operatoren möglichst wenige Parameter ausreichen.

Im folgenden sollen die wesentlichsten Konstruktionselemente und Operatoren kurz beschrieben werden (s. Tab. 1).

Tab. 1: Konstruktionselemente

Typ	Parameter	Beispiel
Punkt	X, Y, Z	
Kontur offene Kontur	P_1, P_2, P_3 $P_i \ (i=1, 2, \dots, n) \dots$...Punkte OKON (vgl. Bild 2)	
geschlossene Kontur	P_1, P_2, P_3, P_4 $P_i \ (i=1, 2, \dots, n) \dots$...Punkte GKON (vgl. Bild 2)	
Fläche	OKON 1, OKON 2 OKON 1, OKON 2.... ... offene Konturen	
	GKON, P GKON ... geschlossene Kontur P ... Punkt	

Körper Quader	L,B,H L ... Ausdehnung in x-Richtung B ... Ausdehnung in y-Richtung H ... Ausdehnung in z-Richtung	
Kreis- zylinder	STGZ,H,D STGZ... Standardgerade Z D ... Durchmesser H ... Höhe	
Kreiskegel (-stumpf)	STGZ,H,D1,D2 STGZ... Standardgerade Z H ... Höhe D1 ... Durchmesser der Grundfläche D2 ... D.d.Deckfläche	
Wandkörper	F,S F ... Fläche S ... Wandstärke	
Schiebe- profil- körper	KON,GKON KON ... offene oder ge- schlossene Kontur (Leitlinie) GKON... geschlossene Kontur (Fläche)	
Drehprofil- körper	GERA,GKON GERA ... Rotationsachse GKON ... geschlossene Kontur	

Konstruktionselemente

Punkt

Ein Punkt wird durch drei Parameter (x, y, z) zur Angabe seiner kartesischen Koordinaten beschrieben.

Kontur

Ein Kontur besteht aus einer Folge von in einer Ebene liegenden Punkten P_i ($i = 1, 2, \dots, n; n \geq 2$), die miteinander durch Geradenstücken, Kreisbögen oder Interpolationskurven verbunden sind. Zur Kennzeichnung der Verbindungsart wurden Zusatzsymbole eingeführt:

OKON (P_1, P_2, \dots, P_n) ... Verbindung durch Geradenstücke

OKON ($P_1^*, P_2^*, P_3^*, \dots, P_{n-1}^*, P_n$) ... Verbindung durch Interpolationskurve

OKON (P_1^{**}, P_2, P_3) ... Verbindung durch Kreisbogen

Durch geeignete Wahl dieser Zusatzsymbole können innerhalb einer Kontur alle drei Verbindungsarten auftreten (s. Abb. 2).

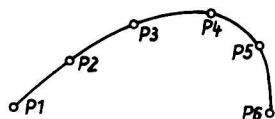
Für geschlossene Konturen wird der Punkt P_n mit dem Punkt P_1 nach der vorgeschriebenen Verbindungsart verbunden. Demnach kann man zwischen offenen Konturen OKON und geschlossenen Konturen GKON unterscheiden (s. Abb. 2)

Fläche

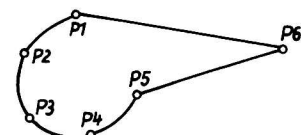
Eine in der Fachsprache definierte Fläche entsteht entweder durch Verschieben einer offenen Kontur OKON 1 entlang einer zweiten offenen Kontur OKON 2 (Voraussetzung: Beide Konturen schneiden sich in genau einem Punkt), oder sie wird durch die Angabe einer geschlossenen Kontur GKON und einen im Inneren der Fläche liegenden Punkt PU beschrieben.

Grundkörper

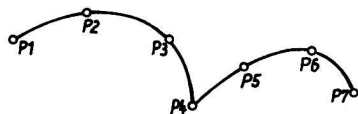
Neben den bekannten geometrischen Körpern Quader, Kreiszylinder und Kreiskegel (-stumpf) werden den fachspezifischen Anforderungen entsprechend Wandkörper, Schiebeprofilkörper und Drehprofilkörper eingeführt.



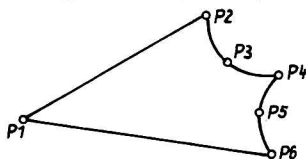
OKON (P1,*P2,*P3,*P4,*P5,P6)



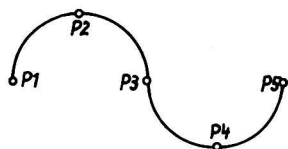
GKON (P1,*P2,*P3,*P4,P5,P6)



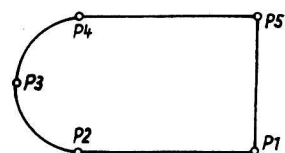
OKON (P1,*P2,*P3,P4,*P5,*P6,P7)



GKON (P1,P2,*P3,P4,*P5,P6)



OKON (P1,**P2,P3,**P4,P5)



GKON (P1,P2,**P3,P4,P5)

Abb. 2: Konturen

Um möglichst einfache (parameterarme) Beschreibungsformen zu gewährleisten, wurde die jeweilige Bezugslage der Grundkörper, Quader, Kreiszylinder und Kreiskegel so gewählt, daß bei dem Quader ein Eckpunkt mit dem Ursprung des Koordinatensystems zusammenfällt und alle Flächen parallel zu den Koordinatenebenen liegen, während bei Kreiszylinder und Kreiskegel der Mittelpunkt der Grund- (oder Deckfläche) im Ursprung des Koordinatensystems liegt und die Rotationsachse mit einer der drei Koordinatenachsen übereinstimmt.

Wandkörper (Parameter: Fläche F und Wandstärke s) entstehen durch Verlagerung der Fläche F um jeweils $s/2$ derart, daß die Strecke s in jedem Punkt der Fläche senkrecht zu ihr steht. Die so entstandenen Begrenzungsflächen eines Wandkörpers sind Äquidistante zu der angegebenen Fläche F mit dem Abstand $s/2$. Schiebeprofilkörper (Parameter: Leitlinienkontur KON , Fläche $GKON$) entstehen, indem eine durch eine geschlossene Kontur begrenzte Fläche $GKON$ entlang einer offenen oder geschlossenen Kontur KON so bewegt wird, daß die erzeugende Fläche in jedem Punkt der Leitlinienkontur senkrecht zu ihr steht.

Drehprofilkörper (Parameter: Rotationsachse $GERA$, geschlossene Kontur $GKON$) entstehen durch Rotation einer geschlossenen Kontur $GKON$ um eine vorgegebene Achse $GERA$. Dabei wird gefordert, daß Kontur und Achse in einer Ebene liegen.

Diese Konstruktionselemente werden durch Standardkonstruktionselemente ergänzt, wie Einheitskugel, achsenparalleler Einheitsquader, Koordinatenachsen, Koordinatenebenen u. a., die den Umgang mit der Fachsprache erleichtern helfen.

Operatoren

Neben einigen in DIGOS speziell geometrie-orientierten Operatoren (Tangente, Normale), den Transformationen (Translation, Rotation speziell um eine Koordinatenachse, Skalierung, Spiegelung an einer Ebene) sowie den mengentheoretischen Operationen (Durchschnitt, Vereinigung, Differenz) wurden eine Reihe problemorientierter Operatoren eingeführt. Sie können unterteilt werden in

- konstruktionsspezifische Operatoren
- Operatoren zur Modellbildung
- Zuweisungen numerischer Werte zu geometrischen Objekten.

Die konstruktionsspezifischen Operatoren sind in Tab. 2 zusammengefaßt.

Operation	Parameter
Translation des Koordinatensystems <u>KOTRA</u>	X, Y, Z ... Koordinaten des Translationsvektors
Rotation einer Koordinatenebene <u>KOROT</u>	KOEB, phi KOEB ... Koordinatenebene phi ... Drehwinkel in Grad
Rücktransformation auf Standardkoordinatensystem <u>KOANF</u>	-
Bohrung <u>BO</u>	X, Y, Z, P, D, T X, Y, Z ... Richtungsvektor f. Bohrrichtung P ... Ansatzpunkt des Bohrers D ... Durchmesser d. Bohrung T ... Bohrlochtiefe
Anfügen (eines Körpers an einen anderen) <u>AFU</u>	P oder P1, P2 P, P1, P2 ... Berührungspunkte
Wandkörper einpassen in einen bereits definierten Körper <u>WE</u>	-
Abschneiden eines Teils eines bereits definierten Körpers <u>AS</u>	P ... innerer Punkt des abzuschneidenden Körpers

Tab. 2: Konstruktionsspezifische Operatoren

Die Operatoren zur Modellbildung dienen im wesentlichen der Berechnung der Querschnittswerte (u. a. Querschnittsfläche, statische Momente der Schnittfläche, Trägheitsmomente, Schubkoeffizienten) und dem Aufbau des dem Gestell entsprechenden Modells (elastischer Balken, Starrelement, Biege-, Torsions-, Zugbettung, konzentrierte Steifigkeit, konzentrierte Trägheit).

Den Zusammenhang zwischen den geometrischen (konstruktionsspezifischen) Operanden und den sie charakterisierenden numerisch-technischen Parametern liefern die Operatoren (eigentlich spezielle Anweisungen) der dritten Gruppe, deren wesentlichste im folgenden angeführt werden.

Zuweisung einfacher Variabler

NW - Name eines numerischen Wortes

KOV - Name einer Konstruktionsvariablen

c - ALGOL 60 - Variable

NW(KOV) : c , c : NW(KOV)

NW(KOV1) : NW(KOV2)

Zuweisung von Feldern

FNW - Name eines Feldes numerischer Werte

MAT - ALGOL 60 - Feldvariable

FNW(KOV) := MAT

MAT := FNW(KOV)

Zuweisung indizierter Variabler

P, Q, R, S - Indizes

$FNW[\underline{P}, \underline{Q}] (KOV) := c$, $c := FNW[\underline{P}, \underline{Q}] (KOV)$,

$FNW[\underline{P}, \underline{Q}] (KOV) := FNW[\underline{R}, \underline{S}] (KOV)$,

$FNW[\underline{P}, \underline{Q}] (KOV) := NW(KOV)$ Spezialfall

Zuweisung von Teilfeldern

$FNW[\underline{P}: Q, R: \underline{S}] (KOV) := MAT$,

$MAT := FNW[\underline{P}: Q, R: \underline{S}] (KOV)$,

$FNW(KOV) := FNW1[\underline{P}: Q, R: \underline{S}] (KOV)$

Dabei müssen die Dimensionen der Felder auf beiden Seiten der Anweisung übereinstimmen. Die Zuweisung sowohl geometrischer als auch numerisch-technischer Werte zu Variablen bleibt innerhalb einer Datenstruktur solange erhalten bis der Nutzer durch STREICHE sie aus der Datenstruktur löscht.

Operanden/Operatoren
der Fachsprache



DIGOS

Kontur
Fläche
Quader

Punktfolge
Schiebfläche, ebene Fläche
Polyederkörper

Kreiszylinder, -kegel
Drehprofilkörper

Drehkörper

Wandkörper
Schiebprofilkörper

Schiebfläche, innerer Punkt
spezielles Körperelement

Transformation des
Koordinatensys:

Transformation der Elemente

BO
AFU
WE
AS

Körper 1 \ Körper 2
Zusammengesetzte Elemente
 $Körper\ 1 \left\{ \begin{matrix} n \\ 2 \end{matrix} \right\} Körper\ 2$
Körper 1 \ Körper 2

Tab. 3: Fachsprache - DIGOS

Mit Hilfe des Programmtransformationssystems DEPOT (s. /2/) kann eine Transformation der Quellprogramme der Fachsprache in Zielsprachenprogramme einer höheren Programmiersprache (z. B. BESM 6 - ALGOL 60 oder ESER PL 1) erfolgen. Die Übersetzung der so erhaltenen Programme in Objektprogramme wird mit den vorhandenen Compilern der entsprechenden Anlagen realisiert (s. Abb. 3).

In der vollständigen Ausbaustufe wird ein Programmsystem angestrebt, das unter Einbeziehung eines interaktiven graphischen Terminals den unmittelbaren Eingriff des Nutzers in die verschiedenen Konstruktionsphasen gestattet. Die Fachsprache erhält die Funktion einer interaktiven Eingangssprache, wobei Teile von ihr durch vorhandene Dialogmöglichkeiten (Lichtstift, Funktionstastatur) aktiviert werden können.

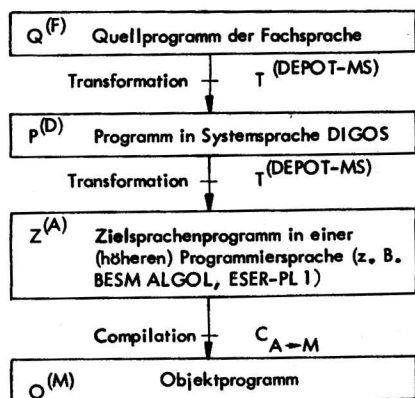


Abb. 3: Übersetzung in Objektprogramme

Literatur:

- / 1 / Ludwig, M., Chr. Richter u. R. Weber: Sprache und Datenstrukturen des digital-geometrie-orientierten Systems DIGOS.
Wiss. Z. TU Dresden 25, 3, S. 405 - 415 (1976).
- / 2 / Bormann, J., u. J. Löttsch: Definition und Realisierung von Fachsprachen mit DEPOT.
Dissertation, TU Dresden 1974.
- / 3 / Wiedwald, H.-J.: Konzeption einer Fachsprache zum rechnergestützten Konstruieren von Gestellen im Werkzeugmaschinenbau.
Diplomarbeit TU Dresden, Sect. Mathematik, 1975.
- / 4 / Picht, M.: Eingabe geometrischer Daten zur Berechnung von Werkzeugmaschinenstellen mittels elektronischer Datenverarbeitungsanlagen.
Diplomarbeit, TU Dresden, Sect. Fertigungstechnik und Werkzeugmaschinenbau, 1975.
- / 5 / Paul, M.: Erweiterung geometrieorientierter Datenstrukturen auf numerische Anschlußbedingungen.
Diplomarbeit, TU Dresden, Sect. Mathematik, 1975.

Anschrift des Verfassers:

Doz. Dr. Manfred Ludwig
Technische Universität Dresden
Sektion Mathematik

DDR - 8027 Dresden

Mommensenstr. 13

