KERNFORSCHUNGSZENTRUM

KARLSRUHE

Juli 1976 KFK 2303

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

MODINT — Ein FORTRAN IV-Programm zur zwei- und dreidimensionalen Interpolation in Gittern

K. Küfner

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 2303

Institut für Neutronenphysik und Reaktortechnik

Projekt Schneller Brüter

MODINT - Ein FORTRAN IV-Programm zur zwei- und dreidimensionalen Interpolation in Gittern

K. Küfner

Gesellschaft für Kernforschung mbH, Karlsruhe

Zusammenfassung

MODINT löst die Interpolationsaufgabe in 2 und 3 Raumdimensionen durch eine Verallgemeinerung des (eindimensionalen) Newton'schen Ansatzes für das Interpolationspolynom (dividierte Differenzen). Wesentliche Voraussetzung hierfür ist, daß die Stützwerte einem kartesischen Koordinatengitter zugeordnet sind (es wird insbesondere nicht das Ausgleichsproblem gelöst). Literaturangaben zur mehrdimensionalen Interpolation und modularer Aufbau des Programms sollen helfen, gegebenenfalls einen neuen Interpolationsalgorithmus zu implementieren.

MODINT - A FORTRAN-IV Code for Interpolation Grid-Functions of Two and Three Variables

Abstract

MODINT interpolates functions of two and three independent variables by using a generalized Newton-scheme (divided differences). Theory and application are restricted to cartesian grid-functions (in particular the general problem of multidimensional fitting is not solved). Literature given as well as the modular design of the programme should enable a user to move to another interpolation algorithm if necessary.

Inhaltsverzeichnis

1. Einleitung

Für Funktionen einer Veränderlichen stehen eine Vielzahl von bewährten numerischen Verfahren zur (Polynom-) Interpolation bereit. Viele dieser Algorithmen sind in die Unterprogrammbibliotheken der einzelnen Computersysteme aufgenommen und damit leicht zu benutzen.

Anders ist die Situation für Funktionen von zwei und mehr Veränderlichen. Abgesehen von einigen Standardformeln (/1/, /6/, /3/) (fast immer nur für 2 Veränderliche) findet man nur selten unmittelbar brauchbare Verfahren. Nun zwingt die Vielzahl der möglichen Kombinationen von Basispunkten im mehrdimensionalen Raum sowie die Auswahlmöglichkeiten für die Basisfunktionen auch dazu eher Rezepte zur Herstellung von Formeln anzugeben als die Formeln selbst. Deswegen wurde das Programm MODINT so aufgebaut, daß durch Austausch von wenigen Unterprogrammen eine neue Interpolationsformel verwendet werden kann.

Um dieses Konzept zu unterstützen, enthält dieser Bericht außer der Programm- und Eingabebeschreibung die Herleitung einiger elementarer Interpolationsformeln (für 2 und 3 Raumdimensionen) sowie Literaturhinweise auf weiterführende Literatur. Dabei beschränkt sich der Bericht auf die eigentliche Interpolationsaufgabe, d. h. Probleme der (mehrdimensionalen) Approximation (Ausgleichsrechnung) sind ausgeklammert.

Das beschriebene Programm MODINT löst in der vorliegenden Version das Interpolationsproblem durch die Berechnung von (verallgemeinerten) dividierten Differenzen in 2 und 3 Raumdimensionen. Durch die Einschränkung, daß die Stützstellen auf einem Gitter liegen sollen, wird die Eindeutigkeit dieser Polynominterpolation (innerhalb der Klasse $P(x,y,z) := \sum_{i \leq im} \sum_{j \leq jm} \sum_{k \leq km} a_{ijk}\, x^i y^j z^k$) gewährleistet. Der Höchstgrad der konstruierten Polynome kann, durch Eingabe der Anzahl der Stützpunkte entlang jeder Koordinatenachse, vom Benutzer festgelegt werden, während das Programm danach für jeden Interpolationsschritt die geeigneten Stützstellen aussucht.

Zum Druck im Juni 1976

2. Grundlagen der mehrdimensionalen Interpolation

a. Bezeichnungen

Im folgenden bezeichnet $\mathbb{R}$ (oder $\mathbb{R}^1$) den Raum der reellen Zahlen, $\mathbb{R}^n$ (meist n = 2 oder n = 3) den Raum der Vektoren mit n Komponenten aus $\mathbb{R}$.
Gegeben ist die Stützstellenmenge $S \subset \mathbb{R}^n$, $S = \{\vec{s}_i \in \mathbb{R}^n, i = 1, \ldots m\}$.
Die $\vec{s}_i \in S$ heißen Stützstellen (oder Interpolations-Knoten).
Es sei $F : \mathbb{R}^n \to \mathbb{R}$ eine reellwertige Funktion. Für eine Menge linear unabhängiger Funktionen

$$\{\phi_j : \mathbb{R}^n \to \mathbb{R},\ j = 1, \ldots, m\}$$

(sog. Basisfunktionen) besteht dann die Interpolationsaufgabe darin, Koeffizienten a_i, $i = 1,\ldots,m$, zu bestimmen, so daß die Funktion ψ mit

$$\psi(\vec{x}) := \sum_{j=1}^{m} a_j \, \phi_j(\vec{x}) \qquad (1)$$

die m Bedingungen

$$\psi(\vec{s}_i) = F(\vec{s}_i) \qquad i = 1,\ldots,m \qquad (2)$$

erfüllt.

Bemerkung:
1. Es genügt für (1), (2), daß anstelle von F nur die Menge M,

$$M = \{f_i \in \mathbb{R},\ f_i = F(\vec{s}_i),\ i = 1,\ldots, m\} \qquad (3)$$

bekannt ist. M heißt Stützwertmenge. Anstatt M als eine Menge von Skalaren zu betrachten ist es manchmal sinnvoll, die abgekürzte Schreibweise mit dem Stützstellenvektor $\vec{f}$ zu verwenden:

$$\vec{f} := (f_1, \ldots, f_m)^T \in \mathbb{R}^m \tag{4}$$

(Das hochgestellte T bedeutet die Transposition des Vektors).

2. Die hier gewählten Definitionen zur Formulierung der Interpolationsaufgabe lassen sich noch allgemeiner fassen (s. z. B. /4/), so daß auch die Spline-Interpolation beschrieben werden kann.

Für zwei Mengen $A = \{a_1, \ldots, a_n\}$ und $B = \{b_1, \ldots b_m\}$ ist das kartesische Produkt $A \times B$ (wie üblich) definiert als die Menge aller geordneten Paare mit der ersten Komponente aus A und der zweiten aus B, d. h.

$$A \times B = \{(a_i, b_j) \mid a_i \in A, i = 1,\ldots,n, b_j \in B, j = 1,\ldots,m\}$$

Ein kartesisches Produkt aus k Mengen: $A_1 x \ldots x A_k$ besteht analog aus allen geordneten k-tupel mit Komponenten aus den einzelnen Mengen.

Die Stützstellenmenge bildet ein Gitter, d. h. jedes $\vec{s}_i \in S$ läßt sich darstellen als

$$\vec{s}_i = (x_{i_1}^{(1)}, \ldots x_{i_n}^{(n)})^T \quad \text{und}$$
$$S \subset S_1 \times \ldots \times S_n, \; S_j \subset \mathbb{R} \tag{5}$$

Für unregelmäßig verteilte Punkte wird jedes S_j ungefähr m Elemente enthalten, für Punkte auf einem achsenparallelen Gitter reduziert sich aber diese Zahl (da nun die Punkte auf Linien parallel zu den Koordinatenachsen liegen, s. Beispiel 3 in 1.d).

Soweit es zweckmäßig ist, werden bei einer Darstellung der $\vec{s}_i$ nach (5) auch die folgenden Schreibweisen verwendet:

$$\vec{s}_i = s_{i_1 \ldots i_n}, \; F(\vec{s}_i) = f_{i_1 \ldots i_n} \tag{6}$$

Für eine Menge $M_1 = \{x_o,\ldots,x_n\}$ von reellen Zahlen bedeutet $[M_1]$ (oder auch $[x_o,\ldots,x_n]$) das kleinste, alle Zahlen von M_1 enthaltende, abgeschlossene Intervall.

Für $F(\vec{z}) = F(x^{(1)},\ldots,x^{(n)})$ ist

$$D_j^i \ F(\vec{z}) := \frac{\partial^i}{(\partial x^{(j)})^i} F(\vec{z})$$

die i-te partielle Ableitung von F nach der Variablen $x^{(j)}$,

$$D_{j,\ell}^{i,k} F(\vec{z}) : = \frac{\partial^{i+k}}{(\partial x^{(j)})^i (\partial x^{(\ell)})^k} F(\vec{z})$$

die i-te partielle Ableitung nach $x^{(j)}$ angewandt auf $D_\ell^k F(\vec{z})$. Der <u>Grad eines Polynoms</u> P, $P = \sum_{i=0}^{n} a_i x^i$, $a_n \neq 0$, ist der maximale Exponent n von x.

b. Prinzipieller Ansatz

Der im folgenden beschriebene Weg ist wohl der allgemeinste, um das Problem der Interpolation mit einer Linearkombination von Basisfunktionen zu lösen. Die Art der Darstellung entspricht im wesentlichen /4/.

Es sei also

$M : = \{f_i \epsilon \mathbb{R},\ i=1,\ldots,m\}$ die Menge der Stützwerte,

$S : = \{\vec{s}_i \epsilon \mathbb{R}^n,\ i=1,\ldots,m\}$ die Menge der Stützstellen

und schließlich

$B : = \{\phi_j : \mathbb{R}^n \to \mathbb{R},\ j=1,\ldots,m\}$ die Menge der Basisfunktionen.

Für den Ansatz

$$\psi(\vec{x}) = \sum_{i=1}^{m} a_i \ \phi_i(\vec{x}) \tag{7}$$

für die Interpolationsfunktion ψ lautet dann die Interpolationsaufgabe:

Bestimme $\{a_i, i=1,\dots,m\}$ so, daß

$$\psi(\vec{s}_i) = f_i \qquad (i=1,\dots,m) \tag{8}$$

Umformuliert in Matrizenschreibweise heißen (7), (8):

$$\left.\begin{array}{l} \psi(\vec{x}) = \vec{\phi}^T(\vec{x}) \cdot \vec{a}\,, \quad D \cdot \vec{a} = \vec{f}\,, \text{ wobei} \\ \vec{a} = (a_1,\dots,a_m)^T,\ \vec{f} = (f_1,\dots,f_m)^T,\ D = (\phi_j(\vec{s}_i))_{\substack{j=1,\dots m \\ i=1,\dots m}} \\ \vec{\phi} = (\phi_1,\dots,\phi_m)^T \end{array}\right\} \tag{9}$$

(9) ist genau dann lösbar*) (und damit (7) möglich und eindeutig), falls

$$\det D = \det(\phi_j(\vec{s}_i)) \neq 0\,, \tag{10}$$

und dann gilt:

$$\psi(\vec{x}) = \vec{\phi}^T(\vec{x}) \cdot D^{-1} \cdot \vec{f}\,. \tag{11}$$

Für numerische Zwecke kann es nützlich sein, statt D eine (durch nicht-singuläre Matrizen D_1, D_2) transformierte Matrix $\tilde{D}$

$$\tilde{D} = D_1 \cdot D \cdot D_2$$

zu betrachten (Verbesserung der Kondition von D!). Nun gilt offenbar:

$$\psi(x) = \vec{\phi}^T(\vec{x}) \cdot (D_2 \cdot D_2^{-1}) \cdot D \cdot (D_1^{-1} \cdot D_1) \cdot \vec{f} = \vec{\phi}^T(\vec{x}) \cdot D_2 \cdot \tilde{D}^{-1} \cdot D_1 \cdot \vec{f}\,, \tag{12}$$

d.h. D kann durch die ähnliche Matrix $\tilde{D}$ ersetzt werden, falls nur $\vec{\phi}$ und $\vec{f}$ gemäß (12) mittransformiert werden. Der Übergang von D auf $\tilde{D}$ beschreibt z.B. Skalierungen des zugrunde liegenden Koordinatensystems, Orthogonalisierung der Basisfunktionen u.s.w.

Die zu jeder Interpolationsformel gehörende Fehlerabschätzung ergibt sich relativ einfach (aber nur formal) unter folgender Voraussetzung:

---

*) Hier wird vorausgesetzt, daß $\vec{f}$ nicht der Nullvektor ist. Für $\vec{f}=0$ ist $\psi=0$ eine Lösung von (9), aber falls (10) nicht erfüllt ist, kann es noch weitere Lösungen geben.

Sei F die zu interpolierende Funktion (mithin: $f_i = F(\vec{s}_i)$), $\varepsilon(\vec{x}) := F(\vec{x}) - \psi(\vec{x})$ hinreichend oft differenzierbar*) und sei $\vec{z} \in \mathbb{R}^n$ der Punkt, für den die Abschätzung gemacht werden soll.

Eine Anwendung des Taylor'schen Satzes (/17/, Kap.V,4) führt zu:

$$\left.\begin{aligned} \varepsilon(\vec{x}) &= \sum_{i=0}^{k} \frac{1}{i!} (\vec{h}\cdot\nabla)^{(i)} \varepsilon(\vec{z}) + R_k(\vec{x},\vec{z}) \\ \vec{h} &:= \vec{x} - \vec{z} \\ R_k(\vec{x},\vec{z}) &= \frac{1}{(k+1)!} \cdot (\vec{h}\cdot\nabla)^{(k+1)} \varepsilon(\vec{z}+\vartheta\cdot\vec{h}), \quad 0<\vartheta<1. \end{aligned}\right\} \qquad (13)$$

Dabei bedeutet $(h = (h_1,\ldots,h_n)^T)$

$$(\vec{h}\cdot\nabla)^{(i)} = \left(h_1\frac{\partial}{\partial x_1} + \ldots + h_n\frac{\partial}{\partial x_n}\right)^{(i)} = \sum_{j_1,\ldots,j_i=1}^{n} h_{j_1}\cdot\ldots\cdot h_{j_i} \frac{\partial^i}{\partial x_{j_1}\ldots\partial x_{j_i}}$$

Wegen $\varepsilon(\vec{s}_i) = 0$ $(i=1,\ldots,m)$ ergibt sich aus (13) ein lineares inhomogenes Gleichungssystem für $\varepsilon(\vec{z})$ und einige partielle Ableitungen von ε im Punkte $\vec{z}$. Über die in R_n enthaltenen Terme hängt die Lösung dieses Systems noch ab von Werten von ε an Zwischenstellen. Diese Tatsache, sowie die beachtliche Kompliziertheit des analytischen Ausdrucks der Lösung ε von (13) (s. /4/ für Beispiele) machen es schwer, den Fehler in dieser Form abzuschätzen.

c. Existenz und Eindeutigkeit der Lösung der Interpolationsaufgabe

Für m Stützstellen im $\mathbb{R}^1$ existiert genau ein Polynom P_{m-1} vom Grade m-1, für das gilt:

$$P_{m-1}(x_i) = f_i \qquad i=1,\ldots,m,$$

wobei die x_i bzw. f_i $(i=1,\ldots,m)$ die Stützstellen bzw. Stützwerte sind. (m-1) ist außerdem minimal, d.h. kein Polynom mit einem Grad kleiner als m-1 kann die Bedingungen (14) erfüllen.

*) "hinreichend oft differenzierbar" heißt hier: (k+1)-mal stetig partiell differenzierbar (nach allen Variablen) in einem konvexen Gebiet, das $\vec{z}$ und alle Stützstellen enthält.

Für die Interpolationsaufgabe in 2 und mehr Raumdimensionen dagegen kann man eine solche Aussage nur dann machen, wenn für die Wahl der Interpolationsknoten und/oder der Basisfunktionen Einschränkungen gemacht werden (siehe Gl.(10)).

Beispiel 1: sei $n=2$ und $m=4$; $S = \{(0,0), (1,0), (0,1), (1,1)\}$

a) Der Ansatz $\psi(x,y) = a_1+a_2x+a_3y+a_4x^2$

führt zu

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix} \quad \text{also } \det D = 0 \text{ (zwei gleiche Spalten).}$$

Dieser Ansatz ist nicht möglich.

b) Der Ansatz $\psi(x,y) = a_1+a_2x+a_3y+a_4xy$

dagegen liefert

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, \quad \det D = 1,$$

also ist dieser Ansatz erfolgreich.

Beispiel 2: sei $n=2$ und $m=4$; $S = \{(0,0), (-1,0), (1,0), (0,1)\}$

Der Ansatz $\psi(x,y) = a_1+a_2+a_3y+a_4y^2$

führt zu

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

Trotz $\det D = 0$ ist das homogene Problem $(\vec{f} = \vec{0})$ nicht trivial lösbar, und zwar gilt: $\psi(x,y) = c \cdot y(1-y)$ erfüllt die vier Bedingungen $\psi(x_i,y_i) = 0$ für jedes $c \in \mathbb{R}$ (keine Eindeutigkeit der Lösung).

Durch Nachprüfen von (10) kann man sich prinzipiell stets versichern, ob ein Ansatz (eindeutig) lösbar ist. Andere Kriterien für die eindeutige Lösbarkeit der (mehrdimensionalen) Interpolationsaufgabe finden sich in /3/,/8/.

d. Dividierte Differenzen, Entwicklungssatz

Ein wesentlicher Nachteil bei der Vorgehensweise nach b. besteht in der dafür nötigen Auflösung des linearen Gleichungssystems für jeden zu interpolierenden Punkt. Für spezielle Anordnungen kann man jedoch auch auf elegantere Art und Weise zur Funktion ψ aus (7) kommen. Solche Möglichkeiten werden im folgenden beschrieben.

Wir betrachten dabei nur noch die Interpolationsaufgabe für Gitter, d.h.:

$$S = S_1 x \ldots x S_n, \quad S_i = \{x_o^{(i)}, \ldots, x_{m_i}^{(i)}\} \ i=1,\ldots,n, \quad \prod_{i=1}^{n} (m_i+1) = m.$$

Die Elemente $\vec{s}_i \in S$ und $F(\vec{s}_i)$ werden abgekürzt geschrieben:

$$s_i = (x_{i_1}^{(1)}, \ldots, x_{i_n}^{(n)})^T = s_{i_1 \ldots i_n}, \qquad F(\vec{s}_i) = f_{i_1 \ldots i_n}$$

<u>Beispiel 3:</u> Als Beispiel für die obigen Begriffsbildungen sei hier n=2 ($x^{(1)}$=x, $x^{(2)}$=y) gewählt. Dann ist

$$S_1 = \{x_o, \ldots, x_{m_1}\}, \ S_2 = \{y_o, \ldots, y_{m_2}\}$$

$$S = \{(x_o,y_o),(x_o,y_1),\ldots,(x_o,y_{m_2}),(x_1,y_o),\ldots,(x_{m_1},y_{m_2})\}$$

Stützwerte und Stützstellen lassen sich in tabellarischer Form angeben:

x \ y	y_o	y_i	...	y_{m_2}
x_o	f_{oo}	f_{o1}	...	f_{om_2}
x_1	f_{1o}	f_{11}	...	f_{1m_2}
⋮	⋮	⋮	⋮	⋮
x_{m_1}	f_{m_1o}	f_{m_11}	...	$f_{m_1m_2}$

Dividierte Differenzen für Funktionen einer Veränderlichen

Gegeben sei die Funktionswerttafel*)

x_o	x_1	...	x_m
f_o	f_1	...	f_m

(15)

und der Interpolationspunkt x (d.h. gesucht ist F(x)).

Wir definieren:

$$\left.\begin{aligned} f[x_i] &:= f_i \qquad i=0,\dots,m \\ f[x_i,\dots,x_{i+k}] &:= \frac{f[x_{i+1},\dots,x_{i+k}]-f[x_i,\dots,x_{i+k-1}]}{x_{i+k}-x_i} \qquad \begin{array}{l} i=0,\dots,m \\ 1\le i+k\le m \\ 1\le k \end{array} \\ c_o &:= 1,\ c_1 := (x-x_o) \\ c_i &:= c_{i-1}\cdot(x-x_{i-1}) \qquad i=2,\dots,m \end{aligned}\right\} \quad (16)$$

Für $f[x_o,\dots,x_k]$ verwenden wir auch die kürzere Schreibweise $f[k]$.

$f[x_i,\dots,x_{i+k}]$ heißt k-te dividierte Differenz. Elementare Eigenschaften der dividierten Differenzen lassen sich allen einführenden Numerik-Lehrbüchern (z.B. /9/) entnehmen.

Die Newtonsche Interpolationsformel (s. /9/, /10/) läßt sich mit den Definitionen in (16) schreiben als:

$$\left.\begin{aligned} F(x) &= \sum_{i=0}^{k} c_i\, f[x_o,\dots,x_i] + c_{k-1} f[x,x_o,\dots,x_n] \\ &= \sum_{i=0}^{k} c_i\, f[i] + R_k\ , \qquad 0\le k\le m \\ R_k &= c_{k-1}\, f[x,x_o,\dots,x_k] = \frac{F^{(k+1)}(\bar{x})}{(k+1)!}\ , \quad \bar{x}\varepsilon[x_o,x_k] \end{aligned}\right\} \quad (17)^{**)}$$

*) Im Gegensatz zu den vorhergehenden Abschnitten werden hier (wegen der Indizierung ab 0) (m+1) Stützstellen anstatt m benutzt.

**) falls eine Funktion F mit f_i=F(x_i) existiert und (k+1)-mal stetig differenzierbar ist (Anwendung des Mittelwertsatzes).

Beispiel 4: Analytische Ausdrücke für die dividierten Differenzen 0. bis 2. Ordnung.

$$f[x_i] = f_i = F(x_i) \qquad \text{0. Ordnung}$$

$$f[x_i, x_{i+1}] = \frac{F(x_{i+1}) - F(x_i)}{x_{i+1} - x_i} \qquad \text{1. Ordnung}$$

$$f[x_i, x_{i+1}, x_{i+2}] = \frac{f[x_{i+1}, x_{i+2}] - f[x_i, x_{i+1}]}{x_{i+2} - x_i} \qquad \text{2. Ordnung}$$

Dividierte Differenzen für Funktionen mehrerer Veränderlicher

Es genügt, die Erweiterung von (16) für Funktionen von zwei Veränderlichen zu zeigen; für höhere Dimensionen geht alles analog (s. auch /10/).

Es seien also gegeben (vgl. Beispiel 3):

$$S_1 = \{x_o, \ldots, x_{m_1}\}, \; S_2 = \{y_o, \ldots, y_{m_2}\}, \; S = S_1 \times S_2,$$

$$M = \{f_{oo}, f_{o1}, \ldots, f_{10}, f_{11}, \ldots, f_{m_1 m_2}\}.$$

S_1, S_2 und M lassen sich tabellarisch darstellen:

	y_o	y_1	...	y_{m_2}
x_o	f_{oo}	f_{o1}	...	f_{om_2}
x_1	f_{10}	f_{11}	...	f_{1m_2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{m_1}	$f_{m_1 o}$	$f_{m_1 1}$	...	$f_{m_1 m_2}$

Wir definieren:

$$\left.\begin{aligned}
&f[x_i;y_j] := f_{ij}\\
&f[x_i\ldots x_{i+k};y_j] := \frac{f[x_{i+1}\ldots x_{i+k};y_j]-f[x_i\ldots x_{i+k-1};y_j]}{x_{i+k}-x_i}\\
&f[x_i\ldots x_{i+k};y_j\ldots y_{j+l}] := \frac{f[x_i\ldots x_{i+k};y_{j+1}\ldots y_{j+l}]-f[x_i\ldots x_{i+k};y_j\ldots y_{j+l-1}]}{y_{j+l}-y_j}\\
&\qquad (i=0,\ldots,m_1,\ j=0,\ldots,m_2,\ 1\leq i+k\leq m_1,\ 1\leq j+l\leq m_2)\\
&c_o^{(1)} := 1,\ c_1^{(1)} := (x-x_o),\ c_i^{(1)} := c_{i-1}^{(1)}\cdot(x-x_{i-1})\ ,\ i=2,\ldots,m_1\\
&c_o^{(2)} := 1,\ c_1^{(2)} := (y-y_o),\ c_j^{(2)} := c_{j-1}^{(2)}\cdot(y-y_{j-1})\ ,\ j=2,\ldots,m_2
\end{aligned}\right\}\quad(18)$$

Für $f[x_o\ldots x_i;y_o\ldots y_j]$ verwenden wir wieder die kürzere Schreibweise $f[i;j]$ und entsprechend auch $f[i;y_j\ldots y_{j+l}]$ bzw. $f[x_o\ldots x_n;j]$.

Beispiel 5: Auch hier sollen einige dividierte Differenzen explizit angegeben werden:

$$f[x_i;y_k] = F(x_i,y_k)$$

$$f[x_i,x_{i+1};y_k] = \frac{F(x_{i+1},y_k)-F(x_i,y_k)}{x_{i+1}-x_i}$$

$$f[x_i;y_k,y_{k+1}] = \frac{F(x_i,y_{k+1})-F(x_i,y_k)}{y_{k+1}-y_k}$$

$$f[x_i,x_{i+1};y_k,y_{k+1}] = \frac{f[x_i,x_{i+1};y_{k+1}]-f[x_i,x_{i+1};y_k]}{y_{k+1}-y_k}$$

Die Formel (17) läßt sich auf den vorliegenden Fall verallgemeinern, wenn man (17) zweimal hintereinander bei festgehaltener erster bzw. zweiter Koordinate anwendet:

Für alle y gilt:

$$F(x,y) = \sum_{i=0}^{m_1} c_i^{(1)} \, f[i;y] + R_{m_1} .$$

Da $f[i;y] = \sum_{j=0}^{m_2} c_j^{(2)} \underbrace{f[i;y_o \ldots y_j]}_{= f[i;j]} + R_{m_2}$

gilt

$$\left.\begin{aligned} F(x,y) &= \sum_{i=0}^{m_1} c_i^{(1)} \sum_{j=0}^{m_2} c_j^{(2)} \, f[i;j] + R \\ R &= R_{m_1} + \sum_{i=0}^{m_1} c_i^{(1)} R_{m_2} \end{aligned}\right\} \qquad (19)$$

Dabei hängt R_{m_2} natürlich von i ab.

Der Ausdruck für R läßt sich noch vereinfachen,

da $\qquad R_{m_1} = c_{m_1+1}^{(1)} \; f\left[x,x_o,\ldots,x_{m_1};y\right]$

und $\qquad R_{m_2}(i) = c_{m_2+1}^{(2)} \; f\left[i;y,y_o,\ldots,y_{m_2}\right]$

gilt: $\qquad R = c_{m_1+1}^{(1)} \; f\left[x,x_o,\ldots,x_{m_1};y\right] + c_{m_2+1}^{(2)} \sum_{i=0}^{m_1} c_i^{(1)} \; f\left[i;y,y_o,\ldots,y_{m_2}\right]$

Wegen $\qquad f\left[x;y,y_o,\ldots,y_{m_2}\right] = \sum_{i=0}^{m_1} c_i^{(1)} f\left[i;y,y_o,\ldots,y_{m_2}\right] + c_{m_1+1}^{(1)} \; f\left[x,x_o,\ldots,x_{m_1}; y,y_o,\ldots,y_{m_2}\right]$

folgt daraus:

$$\left.\begin{aligned} R = c_{m_1+1}^{(1)} \; f\left[x,x_o,\ldots,x_{m_1};y\right] + c_{m_2+1}^{(2)} \; f\left[x;y,y_o,\ldots,y_{m_2}\right] - \\ c_{m_1+1}^{(1)} \; c_{m_2+2}^{(2)} \; f\left[x,x_o,\ldots,x_{m_1};y,y_o,\ldots,y_{m_2}\right] \end{aligned}\right\} \quad (20)$$

Analog zu (17) gilt:

$$\left.\begin{aligned} f\left[x,x_o,\ldots,x_{m_1};y,y_o,\ldots,y_{m_2}\right] &= \frac{1}{(m_2+1)!} \, D_2^{m_2+1} \; f\left[x,x_o,\ldots,x_{m_1};\bar{y}\right] \\ &= \frac{1}{(m_2+1)!} \cdot \frac{1}{(m_1+1)!} \, D_{2,1}^{m_2+1,m_1+1} \; F(\bar{x},\bar{y}) \\ \text{mit } (\bar{x},\bar{y}) \in I^2 &= \left[x_o,\ldots,x_{m_1}\right] \times \left[y_o,\ldots,y_{m_2}\right] \end{aligned}\right\} \quad (21)$$

Damit kann (20) geschrieben werden als:

$$\left.\begin{aligned} R &= \frac{1}{(m_1+1)!}\, c^{(1)}_{m_1+1}\, D_1^{m_1+1}\, F(\bar{x}_1,\bar{y}_1) + \frac{1}{(m_2+1)!}\, c^{(2)}_{m_2+1}\, D_2^{m_2+1}\, F(\bar{x}_2,\bar{y}_2) \\ &- \frac{1}{(m_1+1)!}\,\frac{1}{(m_2+1)!}\, c^{(1)}_{m_1+1}\, c^{(2)}_{m_2+1}\, D_{1,2}^{m_1+1,m_2+1}\, F(\bar{x}_3,\bar{y}_3) \\ &(\bar{x}_i,\bar{y}_i)\ \varepsilon I^2 \quad (i=1,3) \end{aligned}\right\} \quad (22)$$

Im Falle $m_1=m_2=1$ lautet (19)[*)]:

$$\left.\begin{aligned} F(x,y) &= f[0;0]+(x-x_o)\cdot f[1;0]+(y-y_o)\cdot f[0;1]+(x-x_o)(y-y_o)\cdot f[1;1]+R \\ R &= \frac{1}{2}(x-x_o)(x-x_1)\cdot D_1^2\ F(\bar{x}_1,\bar{y}_1)+\frac{1}{2}(y-y_o)(y-y_1)D_2^2\ F(\bar{x}_2,\bar{y}_2) \\ &-\frac{1}{4}(x-x_o)(x-x_1)(y-y_o)(y-y_1)D_1^2\ D_2^2\ F(\bar{x}_3,\bar{y}_3) \end{aligned}\right\} \quad (23)$$

Für (23) werden nur 4 Stützwerte benötigt. Für $m_1=m_2=2$ läßt sich (19) anschreiben als:

$$\left.\begin{aligned} F(x,y) &= f[0;0]+c_1^{(1)}\cdot f[1;0]+c_2^{(1)}\cdot f[2;0] \\ &+ c_1^{(2)}\cdot(f[0;1]+c_1^{(1)}\cdot f[1;1]+c_2^{(1)}\cdot f[2;1]) \\ &+ c_2^{(2)}\cdot(f[0;2]+c_1^{(1)}\cdot f[1;2]+c_2^{(1)}\cdot f[2;2]) + R \\ R &= \frac{1}{6}\,c_3^{(1)}\cdot D_1^3F(\bar{x}_1,\bar{y}_1)+\frac{1}{6}\,c_3^{(2)}\cdot D_2^3F(\bar{x}_2,\bar{y}_2)-\frac{1}{36}\,c_3^{(1)}\ c_3^{(2)}\cdot D_{1,2}^{3,3}F(\bar{x}_3,\bar{y}_3) \end{aligned}\right\} \quad (24)$$

Auch für Funktionen von mehr als zwei Veränderlichen läßt sich nach der vorgeführten Methode die Entwicklung einer Funktion F nach (entsprechend verallgemeinerten) dividierten Differenzen analog zu (19) angeben. Der Konzeption des Textes folgend sei hier noch der Fall n=3 gezeigt:

$$\left.\begin{aligned} F(x^{(1)},x^{(2)},x^{(3)}) &= \sum_{i=0}^{m_1}\ \sum_{j=0}^{m_2}\ \sum_{k=0}^{m_3} c_i^{(1)}\ c_j^{(2)}\ c_k^{(3)}\ f[i;j;k] + R \\ R &= \bar{c}_1\ \bar{D}_1(\vec{z}_1)+\bar{c}_2\ \bar{D}_2(\vec{z}_2)+\bar{c}_3\ \bar{D}_3(\vec{z}_3) + \bar{c}_1\cdot\bar{c}_2\cdot\bar{c}_3\ \bar{D}_{1,2,3}(\vec{z}_4) \\ &- (\bar{c}_1\bar{c}_2\ \bar{D}_{1,2}(\vec{z}_5) + \bar{c}_2\bar{c}_3\ \bar{D}_{2,3}(\vec{z}_6) + \bar{c}_3\bar{c}_1\ \bar{D}_{3,1}(\vec{z}_7)) \\ \bar{c}_i &= \frac{1}{(m_i+1)!}\ c^{(i)}_{m_i+1},\quad \bar{D}_i(\vec{z}) = D_i^{m_i+1}\ F(\vec{z}),\quad \overline{D_{i,j}}(\vec{z}) = D_{i,j}^{m_i+1}\ F(\vec{z}) \\ &\qquad (i=1,3) \end{aligned}\right\} \quad (25)$$

[*)] Hier wie in der vorstehenden Herleitung wurde natürlich vorausgesetzt, daß 1) die Stützwerte von einer Funktion F herstammen und 2) F genügend oft stetig differenzierbar ist.

wobei z_1 bis z_7 Punkte im Innern des Stützstellengitters sind.

Der Spezialfall $m_1=m_2=m_3=1$ von (25) kann demnach dargestellt werden als:

$$\left.\begin{aligned} F(x,y,z)=c_o^{(1)}&(c_o^{(2)}(c_o^{(3)}f[\overline{0};0;\overline{0}]+c_1^{(3)}f[\overline{0};0;\overline{1}])+c_1^{(2)}(c_o^{(3)}f[\overline{0};1;\overline{0}]+c_1^{(3)}f[\overline{0};1;\overline{1}])) \\ +c_1^{(1)}&(c_o^{(2)}(c_o^{(3)}f[\overline{1};0;\overline{0}]+c_1^{(3)}f[\overline{1};0;\overline{1}])+c_1^{(2)}(c_o^{(3)}f[\overline{1};1;\overline{0}]+c_1^{(3)}f[\overline{1};1;\overline{1}])) \\ +R& \end{aligned}\right\} \quad (26)$$

Auf die Darstellung des Restgliedes sei hier verzichtet.

Die Verwendung von dividierten Differenzen ist vor allem dann zweckmäßig, wenn das Interpolationspolynom an mehreren Stellen ausgewertet werden soll. Die Definition (16) bzw. (18) gibt einen Algorithmus zur Berechnung der $f[\overline{i};j;\ldots]$ und anschließend kann in einem Hornerschema das Polynom $P_k:=F(x) - R_k$ aus (17) bzw. $F(x,y)-R$ aus (19) (und entsprechende höherdimensionale Verallgemeinerungen) für die gewünschten Punkte ausgewertet werden.

e. Lagrange-Interpolation

Es gelten wieder die Einschränkungen bezüglich der Stützstellen (Anordnung im Gitter), wie im vorhergehenden Abschnitt. Wir betrachten die Funktionen

$$\left.\begin{aligned} L_j^{(i)}(x) &:= \prod_{\substack{k=1 \\ k\neq j}}^{m_i} (x-x_k^{(i)}) \qquad \left(\begin{matrix} i=1,\ldots,n \\ j=0,\ldots,m_i \end{matrix}\right) \\ \ell_{i,j}(x) &:= L_j^{(i)}(x)/L_j^{(i)}(x_j^{(i)}) \end{aligned}\right\} \quad (27)$$

Die Funktionen $l_{i,j}$ sind Verallgemeinerungen der (in der eindimensionalen Interpolation) wohlbekannten "Lagrangefaktoren". Insbesondere gilt die (unmittelbar zu überprüfende) Relation

$$\ell_{i,j}(x_k^{(i)}) = \delta_{j,k} \qquad i=1\ldots,n;\ j,k=0,\ldots,m_i \qquad (28)$$

Betrachtet man nun das Polynom

$$P(x^{(1)},\dots,x^{(n)}) := \sum_{k=0}^{m_1}\dots\sum_{l=0}^{m_n} \ell_{1,k}(x^{(1)})\cdot\ldots\cdot\ell_{n,l}(x^{(n)})\cdot f_{k,\dots,l} \tag{29}$$

so gilt wegen (28)

$$P(x_i^{(1)},\dots,x_j^{(n)}) = f_{i,\dots,j} = F(x_i^{(1)},\dots,x_j^{(n)})$$

also erfüllt P aus (29) die Interpolationsaufgabe (verallgemeinerte Lagrange-Interpolation).

Man kann zeigen, daß für die gewählte Anordnung der Stützstellen das Interpolationspolynom[+)] eindeutig bestimmt ist (/3/, /10/), also hat (29) das gleiche Restglied, wie die n-dimensionale Verallgemeinerung von (19).

Nimmt man zu den bereits vorhandenen Stützstellen weitere hinzu, so muß bei der Lagrange-Interpolation das ganze Polynom (29) neu berechnet werden - ein Nachteil gegenüber der Interpolation mit dividierten Differenzen.

Lösbarkeit und Eindeutigkeit der Interpolationsaufgabe im vorliegenden Falle ergeben sich aus der Ableitung dieser Formeln. Im wesentlichen wird sukzessive parallel zu jeder Koordinatenrichtung eindimensional interpoliert. Interessant dabei ist, daß

1. die Reihenfolge, in der die einzelnen Koordinatenachsen abgearbeitet werden, auf das Resultat keinen Einfluß hat und
2. der Grad der n-dimensionalen Interpolation größer sein kann, als der Grad der komponentenweise Interpolation. Z.B. entspricht (23) (bzw. (26)) einer linearen Interpolation parallel zu den 2 (bzw. 3) Koordinatenachsen. Dabei ist aber (23) von der Form

$$f(x,y) = a_0 + a_1x + a_2y + a_3xy$$

(bzw. (26): $\quad f(x,y,z)=a_0+a_1x+a_2y+a_3z+a_4xy+a_5xz+a_6yz+a_7xyz$

mithin vom Grade 2 (bzw. 3).

+) Dabei ist die Form des Interpolationspolynoms entscheidend:

$$P(x_1,x_n) = \sum_{k_1=0}^{m_1} \dots \sum_{k_n=0}^{m_n} a_{k_1\dots k_n}\cdot x_1^{k_1}\dots x_n^{k_n}$$

f. Literatur zur mehrdimensionalen Interpolation

Zum Abschluß dieses Kapitels soll noch die Literaturliste (Kap. 5) erläutert werden. Dort finden sich im wesentlichen alle einschlägigen Veröffentlichungen, die dem Autor bei der Programmerstellung bekannt waren. Neuere Forschungen befassen sich vorwiegend mit Anwendungen und Entwicklung der Spline-Interpolation.

Lehrbücher, Monografien

Es wird meistens die zweidimensionale Interpolation ausführlich behandelt und auf höherdimensinale Verallgemeinerungen hingewiesen. Die hier am meisten verwendeten Werke waren /1/, /3/ und /10/. /11/ enthält eine umfangreiche Literaturliste (bis 1960). Anregungen (für zweidimensionale Formeln) können auch in /6/ gefunden werden. Einzelheiten über mehrdimensionale Spline-Interpolation finden sich in Büchern über die "Finite Elementen"-Methode und über Spline-Theorie (z.B. /12/, /13/).

Zeitschriftenartikel

H.C. Thacher /8/ beschäftigt sich mit einer strengeren Formulierung von Kriterien für Existenz und Eindeutigkeit von (mehrdimensionalen) Interpolationspolynomen. /4/, /5/ geben im wesentlichen die in Abschnitt b) vorgeführte Interpolationsmethode wieder. In /14/ wird versucht, die in Abschnitt d) gegebenen Definitionen der dividierten Differenzen allgemeiner zu fassen.
Eine (vom methodischen Standpunkt aus mehr oder weniger willkürliche) Auswahl an neuerer Literatur über mehrdimensionale Spline-Interpolation wird durch die Referenzen /15/ bis /20/ gegeben. Zusammen mit den ausführlichen Literaturangaben in /21/ und /22/ kann damit der Anschluß an die moderne Theorie der Spline-Interpolation gefunden werden. Während der Dokumentation erschien /24/.

3. Programmbeschreibung

a. Programminformationen

Sowohl das Stützwertfeld, als auch das erzeugte Feld mit den interpolierten Werten haben eine ganz bestimmte Struktur (die sich ableitet aus dem üblichen Aufbau von Multigruppen-Neutronenflußfeldern).Diese Struktur soll hier am Beispiel eines Feldes A dargestellt werden. Da es vorgesehen ist, zu einer gegebenen Menge von Stützstellen (d.h. zu einem gegebenen Gitter) in einem Programmaufruf mehrere Stützwertmengen (verschiedene Gruppen) zu interpolieren, kann man A auffassen als vierfach indiziertes Feld:[++]

$$A(i,j,k,l)$$

und bezeichnet die Indizes der Reihe nach als Zeilen-, Spalten-, Ebenen- und Gruppenindex. Unter der Ebene k_o der Gruppe l_o versteht man dann das Zahlenfeld

$$E(k_o,l_o) = \{A(i,j,k_o,l_o),\quad i=1,\, i_{max},\ j=1,\, j_{max}\}$$

die Gruppe l_o besteht aus den Ebenen 1 bis k_{max} der Gruppe l_o. (Dabei bedeuten i_{max}, k_{max}, l_{max} jeweils die maximalen Indizes für die entsprechende Dimension.) Daraus ergibt sich der in Abb.1 gezeigte Aufbau von A.

Gruppe 1				Gruppe 2				Gruppe l_{max}		
E(1,1)	E(2,1)	...	$E(k_{max},1)$	E(1,2)	...	$E(k_{max},2)$	...	$E(1,l_{max})$	...	$E(k_{max},l_{max})$

Abb.1 Struktur der bearbeiteten Felder

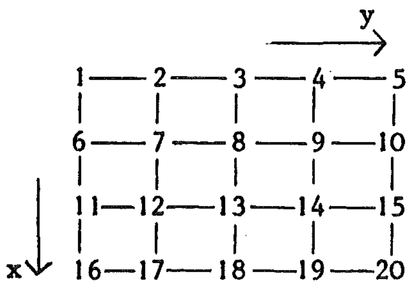
Jede Ebene hat $i_{max} \cdot j_{max}$, jede Gruppe $i_{max} \cdot j_{max} \cdot k_{max}$ Werte[+]. Gespeichert wird das Feld auf einer externen Einheit in Sätzen, die jeweils eine Ebene umfassen.

[+] Sowohl das Feld der Stützwerte, als auch das Feld mit den interpolierten Werten ist aufgebaut aus der gleichen Anzahl von Gruppen, während die Anzahl von Ebenen und die Anzahl der Punkte pro Ebene verschieden sein kann.

[++] Der vierte Index dient also nur zur Kennzeichnung des für die Interpolation zu benutzenden Stützwertfeldes (s. auch nächste Seite).

Jede Gruppe entspricht einem separaten Satz von Stützwerten, aber die Stützpunkte sind für alle Gruppen gleich. In der Terminologie des 2. Kapitels heißt das, daß M aus (3) jeweils den Werten in einer Gruppe entspricht, während die Stützstellenmenge S für alle Gruppen gleich ist. Im Prinzip wird hier also l_{max}-mal die Interpolationsaufgabe gelöst.

Den Indizes i,j,k sind die Koordinatenachsen X,Y,Z zugeordnet. Innerhalb der Ebenen sind die Stützwerte zeilenweise abgespeichert (MODINT behandelt jede Ebene als lineares Feld; vgl. Abb.2).



Die Zahlen geben die Nummer des Speicherplatzes des jeweiligen Wertes an. Element (i,j) wird gespeichert in Platz Nr. $(i-1)\cdot j_{max}+j$

Abb.2 Speicheranordnung innerhalb einer Ebene ($i_{max}=4$, $j_{max}=5$)

Grundsätzlich wird vom Programm versucht, die jeweils benötigte Gruppe im Kernspeicher unterzubringen. Gelingt dies nicht (Ausdruck: OPTIMIERUNGSSTUFE II), so wird die Gruppe umkopiert auf eine "direct-access"-Datei (Dateinummer 3), und im Kernspeicher befinden sich jeweils nur die NZ (s. Kap. 4) im aktuellen Schritt benötigten Ebenen. Dieser Fall kann nur eintreten bei der dreidimensionalen Interpolation (MEA>1, s. Kap. 4), und da er dort nicht unbedingt vorhersehbar ist, sollte jeder Lauf eine Betriebssystemkarte (für IBM/370 OS-MVT:DD-Karte)für die Dateinummer 3 enthalten.

Mit Hilfe des (Assembler-) Unterprogramms XTAREA (von W. Höbel)+) wird der benötigte Kernspeicherbereich in Form eines Arbeitsfeldes ARB zur Verfügung gestellt. Die Aufteilung dieses Bereiches in die verschiedenen Felder ist in Tabelle A gezeigt. Das Unterprogramm CALSTE besetzt die Felder mit den Koordinaten der Stütz- und Interpolationspunkte. Für die zweidimensionale Interplation verzweigt dann das Programm nach IN2DIM (Interpolation 2-dimensional), während im dreidimensionalen Fall noch unterschieden wird, ob

+) siehe auch Abschnitt b) zur genaueren Erklärung der Wirkungsweise der Unterprogramme.

Optimierungsstufe I oder II vorliegt; im zweiten Falle erfolgt vor dem Sprung nach IN3DIM zusätzlich noch ein dynamisches "define-file"+) mit Hilfe der (Assembler-) Unterprogramme DDTEST, DINF und DEFI++) (von G.Arnecke, H. Bachmann). Dieser Programmablauf ist in Abb.3 dargestellt.

+) d.h. die Beschreibung der Attribute der definierten "direct access"-Datei

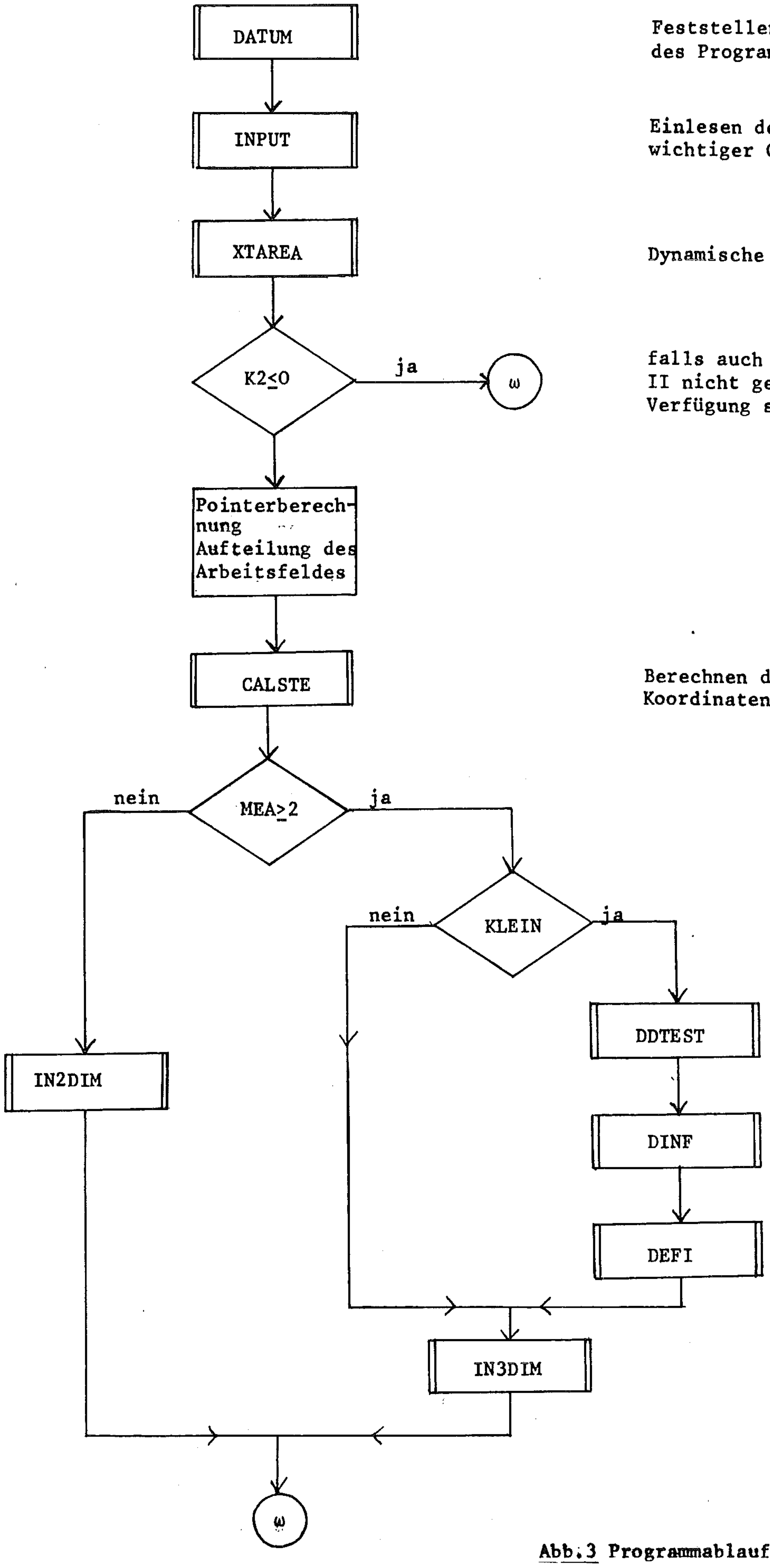
++) siehe auch Abschnitt b) zur genaueren Erklärung der Wirkungsweise der Unterprogramme.

Tabelle 1 Aufteilung des Arbeitsfeldes ARB

Anfangsadresse[1]	Länge in Wörtern[2]	Feldname in Unterprogrammen	Inhalt des Feldes
IFALT	NEB	FALT	Stützwerte (MZA•MSA•NZ bzw. MZA•MSA•MEA Werte je nach Optimierung
IFNEU	NEU	FNEU	Interpolierte Werte (jeweils nur 1 Ebene)
IXA	MZA	XA	X-Koordinaten der Stützpunkte
IYA	MSA	YA	Y-Koordinaten der Stützpunkte
IZA	MEA	ZA	Z-Koordinaten der Stützpunkte
IXN	MZN	XN	X-Koordinaten der Interpolationspunkte
IYN	MSN	YN	Y-Koordinaten der Interpolationspunkte
IZN	MEN	ZN	Z-Koordinaten der Interpolationspunkte
IXS	INPG	X	X-Koordinaten für alle Werte in F
IYS	INPG	Y	Y-Koordinaten für alle Werte in F
IZS	INPG	Z	Z-Koordinaten für alle Werte in F
IFS	INPG	F	Stützwerte für 1 Interpolationsschritt (NX•NY•NZ Werte)

[1] raltiv zu ARB(1), d.h. FALT(1) hat die gleiche Adresse wie ARB(IFALT)

[2] 1 Wort = 4 Bytes; Bedeutung der Variablen: siehe Tab. 2



Feststellen von Datum und Zeitpunkt des Programmaufrufs

Einlesen der Eingabe, Besetzen wichtiger COMMON-Größen

Dynamische Feldausdehnung

falls auch bei Optimierungsstufe II nicht genug Kernspeicher zur Verfügung steht: Abbruch

Berechnen der Felder für die Koordinaten der Stützpunkte

KLEIN=.TRUE. nur bei Optimierungsstufe II

Dynamisches "define-file"-Statement

Interpolation 3-dimensional

STOP

Abb.3 Programmablaufplan von MODINT

Der Aufbau der COMMON-Blöcke in MODINT ist in allen Einzelheiten der Tabelle 2 zu entnehmen.

Das Programm benötigt in seiner jetzigen Version (übersetzt mit dem IBM FORTRAN H-Extended Compiler) ca. 56 K Bytes an Speicherplatz. Dazu kommen noch:

Speicherplatz für die dynamische Feldausdehnung:

$MZA+MSA+MEA+MZN+MSN+MEN+4\cdot NX\cdot NY\cdot NZ+MZN\cdot MSN+MEX\cdot MZA\cdot MSA$
(MEX=MEA für Optimierungsstufe I, =NZ für Stufe II; Bedeutung der Variablen siehe Eingabebeschreibung Kap. 4)

sowie Puffer für die Dateien (siehe Eingabeparameter IBUF)

1 : Stützwerte
2 : Interpolierte Werte
3 : direct-access Zwischenspeicher (sofern benötigt)
und für die Standard-Eingabe- und -Ausgabe-Einheit.

Durch die in Abb.4 wiedergegebene Overlay-Struktur können ca. 10 K Bytes an Kernspeicher gespart werden.

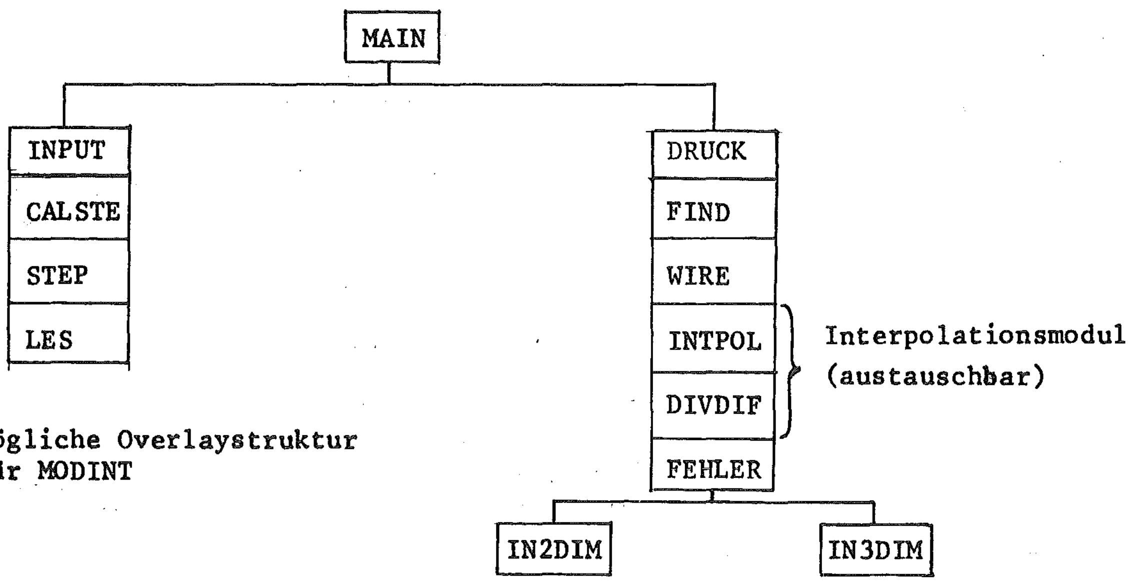


Abb.4 Mögliche Overlaystruktur für MODINT

Ein Lademodul (ohne Overlay-Struktur) von MODINT befindet sich in der Programmbibliothek LOAD.NUSYS. des Rechenzentrums der GfK.

Tabelle 2 Aufbau der COMMON-Blöcke in MODINT

Der COMMON-Bereich des Programms besteht aus den vier Blöcken:

/FORMAT/	(variable Formate und Fallkennzeichnung; der Reihe nach der Inhalt der Sätze K3, K5 und K0 aus der Eingabe)
/KONSTA/	(Informationen für Stützwerte und Stützstellen)
MZA MSA MEA KPOSA NFORMA MODEA	gleiche Bedeutung wie die gleichnamigen Variablen aus dem Satz K2 der Eingabe
NALT	Anzahl der Stützwerte pro Ebene (=MZA·MSA)
NFA	Dateinummer für die Eingabe der Stützwerte (=1)
/KONSTN/	(Informationen für die interpolierten Werte)
MZN MSN MEN KPOSN NFORMN MODEN	Gleiche Bedeutung wie die gleichnamigen Variablen aus dem Satz K4 der Eingabe
NEU	Anzahl der zu interpolierenden Werte pro Ebene (=MZN·MSN)
NFN	Dateinummer für die Ausgabe der interpolierten Werte (=2)
/KONST/	(Allgemeine Konstanten)
MG IBUF NWRIT INDA NX NY NZ	Gleiche Bedeutung wie die gleichnamigen Variablen aus dem Satz K1 der Eingabe
NFO NFI NDA	Dateinummer für Standardausgabe (=6) Dateinummer für Standardeingabe (=5) Dateinummer für "direct-access"-Datei (=3)
NEB	Anzahl der Stützwerte, die gleichzeitig im Kernspeicher sind
KA	Dimension der Interpolation (=2 oder =3)
NXL bis NZR	Verschiebungsindizes für Subroutine FIND
INPG	Anzahl der Stützwerte für jeden Interpolationsschritt (=NX·NY·NZ)
NASS	Assoziierte Variable zur direct-access-Datei

b. Unterprogramme

1. Assembler-Unterprogramme, Installationsabhängigkeit

DATUM (DDAT, DZEIT) : (Assembler; Autor: C.H.Hinze)
Liefert in den 8-Bytes-Variablen DDAT bzw. DZEIT das Datum bzw. den Zeitpunkt des Aufrufs.

DDTEST (N, DDFELD, ICON, NST) : (Assembler; Autor: G. Arnecke)
prüft die "Job Control Language" auf das Vorhandensein von N, im (doppelt genauen) Feld DDFELD gespeicherten, "data definition" (DD) Karten-Namen. ICON = 0 veranlaßt einen Stop des Programms, falls eine DD-Karte fehlt. NST = 0, falls alle Karten vorhanden sind, = 1 sonst. Zu DDTEST gehören das Entry DDFIND, sowie ein FORTRAN Unterprogramm DDDRU mit den Entries DDFELD und DDSTOP.

DEFI (IUNIT, NPRIQ, FORMAT, NBLK, NASS) : (Assembler; Autoren: G.Arnecke und H. Bachmann) dient der dynamischen Ausführung der "define-file" Anweisung in FORTRAN. Der Aufruf mit der o.a. Argumentenliste entspricht im wesentlichen der FORTRAN Anweisung

```
DEFINE FILE IUNIT (NPRIQ,NBLK,FORMAT,NASS) ;
```

In Standard IBM-FORTRAN können aber die vier ersten Argumente nur Konstanten sein.

DINF (DDNAME, NBLK, NPRIQ) : (Assembler; Autoren: G.Arnecke und H. Bachmann) ist ein Vorschaltprogramm zu DEFI. Es beschafft von der DD-Karte die Anzahl der Sätze (NPRIQ) sowie deren Länge (NBLK).

XTAREA (K1, K2, K3, A(1)) : (Assembler, Autor: W. Höbel)
stellt einen Kernspeicherbereich von K2 Bytes zur Verfügung, dessen Anfangspunkt durch A((K1-K3)/4+1) adressiert werden kann. Eine Freigabe des so angeforderten Platzes kann über einen Aufruf des Entrys REXTAR erreicht werden.

2. Andere Unterprogramme

CALSTE steuert die Berechnung und Bereitstellung der Koordinaten der (alten und neuen) Gitterpunkte.

DIVDIF berechnet dividierte Differenzen (s.a. Abschn. c.)

DRUCK druckt jeweils eine Ebene mit interpolierten Wërten aus.

FEHLER reagiert auf eine Fehlerbedingung in INTPOL (s.a. c.)

FIND sucht zu gegebenem Feld X und zu gegebenen Zahlen Y und NX einen Index NXL, so daß Y möglichst zentral im Intervall [X(NXL), X(NXL+NX-1)] liegt.

INPUT liest die Sätze KO bis K5 der Eingabe (s. Kap. 4)

INTPOL ist die eigentliche Interplationsroutine und wird näher in Abschn. c. beschrieben.

IN2DIM steuert die zweidimensionale Interpolation.

IN3DIM steuert die dreidimensionale Interpolation.

LES liest bei jedem Aufruf einen der Sätze K9 bis K11 bzw. K15 bis K17 (siehe Kap. 4).

STEP berechnet aus einer Eingabe analog zu K6 (siehe Kap. 4) die einzelnen Koordinatenwerte.

WIRE mit den Eingängen OUTPUT und OUTP dient dem vereinfachten Lesen und Schreiben von Feldern.

c. Austausch der Interplationsroutine

Wie schon erwähnt, ist MODINT so aufgebaut, daß mit wenigen, örtlich begrenzten Eingriffen ein neues Interpolationsverfahren verwirklicht werden kann. Diese Eingriffe erfolgen bei den Unterprogrammen INTPOL und DIVDIF sowie möglicherweise in der Fehlerbehandlung FEHLER.

Die Informationen über FEHLER sind in Tab.3 zusammengefaßt. Tabelle 4 erläutert die Argumentenliste von INTPOL. Anzumerken ist hier noch, daß die Stützstellen so ausgesucht werden, daß der Interpolationspunkt möglichst zentral liegt.

Nach Neu- oder Umprogrammierung von FEHLER und/oder INTPOL kann z.B. mit Hilfe des Linkage-Editors der zur Verfügung gestellte "Loadmodule" leicht verändert werden (siehe Anhang C).

Die gegenwärtige Version von INTPOL (u. DIVDIF) implementiert die sukzessive Berechnung der dividierten Differenzen parallel zu den einzelnen Koordinatenachsen gemäß Formel (18), (19) bzw. (25). DIVDIF berechnet dabei jeweils dividierte Differenzen für eine Masche des Gitters. Der (Zweit-) Eingang HORNER von DIVDIF wiederum bestimmt aus den dividierten Differenzen den interpolierten Wert.

Für NX = NY = 2, NZ = 1 bzw. NZ = 2 ist in Anhang C eine schnellere Version von INTPOL angegeben (die außerdem auf DIVDIF und HORNER verzichten kann).

Tabelle 3: Argumentenliste von FEHLER

SUBROUTINE FEHLER (KS, I, J, K, L, X, Y, Z. F, ARG, FN, I1)

KS = Fehlercode aus INTPOL

I = Gruppen
J = Ebenen
K = Zeilen
L = Spalten
—Index des Punktes, für den die Fehlerbedingung auftrat.

X, Y, Z, F, ARG = gleiche Bedeutung wie bei INTPOL (s. Tab. 4)
FN, I1 = gleiche Bedeutung wie FKT, IF1 bei INTPOL

Bem.: 1. In der Standardversion von INTPOL werden die Werte von F überspeichert durch die berechneten dividierten Differenzen.

2. In der implementierten Fassung von FEHLER wird nur eine Fehlernachricht ausgedruckt und KS wieder auf 0 gesetzt.

3. Nach dem Rücksprung aus dem Unterprogramm FEHLER setzt MODINT die Interpolation fort.

d. Testbeispiele

In Anhang B sind die Ergebnisse der Interpolation für die nachfolgenden Testbeispiele angegeben+). Die Stützwerte (und die exakten Werte) wurden von einem Vorschaltprogramm berechnet; in Testbeispiel 2 ist INDA>0 (s.Kap.4, Eingabe K1).

+) Mit den Bezeichnungen aus Tab. 2, /KONST/ gilt in (22) bzw. (25):
$NX=m_1+1$, $NY=m_2+1$, $NZ=m_3+1$, da die Indizierung im Programm erst bei 1 beginnt.

Tabelle 4 Argumentenliste von INTPOL

SUBROUTINE INTPOL (X, Y, Z, F, ARG, KS, FKT, IF1)

X(INPG) : X
Y(INPG) : Y } -Koordinaten der Stützpunkte in u.a. Anordnung
Z(INPG)*) : Z

F(INPG) : Stützwerte (in u.a. Anordnung)

Bem.: INPG ist die Anzahl der Stützstellen und wird im COMMON-Block /KONST/ übergeben (siehe Tab. 2)

ARG(3) : (X-, Y-, Z-)Koordinaten des Interpolationspunktes

KS : Fehlercode (wird nach dem Rücksprung abgefragt)
= 0 : weiterrechnen
≠ 0 : Verzweigen in das Unterprogramm FEHLER zur Fehlerbehandlung
FKT : Anfangsadresse des Feldes für die interpolierten Werte
IF1 : Zeiger (s.u.)

Bem. 1: IF1 wird im Programm als Indikator zur (ordnungsgemäßen) Beendigung der Interpolation in der betrachteten Ebene benutzt. Zu Beginn jeder neuen Ebene wird IF1 auf 1 gesetzt; nach dem Aufruf von INTPOL führt die Bedingung IF1>NEU (s. Tab.2) zu einem Sprung ans Ende der Ebenenschleife. In der standardmäßig implementierten Interpolationsroutine wird der interpolierte Wert abgespeichert im FKT(IF1) und anschließend IF1 um 1 erhöht.

Bem. 2: X, Y, Z und F werden so angeliefert, daß (X(i), Y(i), Z(i)) und F(i) Stützpunktkoordinaten und zugehöriger Stützwert bilden. Die innere Ordnung geht aus folgender Übersicht hervor:

I	1	2	...	NY	NY+1	...	NX•NY	NX•NY+1	2•NX•NY	2NX•NY+1	NX•NY•NZ
Z	Z_1	Z_1	...	Z_1	Z_1	...	Z_1	Z_2 ...	Z_2	Z_3 ...	Z_{NZ}
X	X_1	X_1	...	X_1	X_2	...	X_{NX}	X_1	X_{NX}	X_1	X_{NX}
Y	Y_1	Y_2	...	Y_{NY}	Y_1	...	Y_{NY}	Y_1	Y_{NY}	Y_1	Y_{NY}

*) Z wird nur für die echte dreidimensionale Interpolation besetzt.

Testbeispiel 1: Dreidimensionale Interpolation

Zu interpolierende Funktion:

$$F(x,y,z) = 1.0 + 2.0x + 3.0y + 4.0z + 1.5xy + 1.5xz + 1.5yz + 1.7x^2 + 1.9y^2 + 2.1z^2 + 9.0\cdot xyz$$

Anzahl der Gruppen: 1

Art der Interpolation entlang der Koordinatenachsen:

a) quadratisch, d.h. NX = NY = NZ = 3
(nach (25) ist dann das Restglied R = 0, also muß diese Funktion exakt interpoliert werden)

b) linear, d.h. NX = NY = NZ = 2
(nach (25) ist dann $R = 1.7(x-x_1)(x-x_2) + 1.9(y-y_1)(y-y_2) + 2.1(z-z_1)(z-z_2)$)

c) linear, aber mit dem Unterprogramm aus Anhang C

Das Gitter der Stützpunkte ist gegeben durch die Koordinaten

x : 0.0, 1.0,...,18.0, 19.0
y : 0.0, 1.0,...,18.0, 19.0
z : 0.0, 1.0,...,18.0, 19.0,

die Interpolationspunkte liegen auf dem Gitter

x : 0.5, 2.5,...,16.5, 18.5
y : 0.5, 2.5,...,16.5, 18.5
z : 0.5, 2.5,...,16.5, 18.5.

Im Falle b) ergibt sich damit die (a-priori) Abschätzung

$$R \leq 1.425$$

In Tabelle 5 sind die Ergebnisse zusammengestellt.

Fall	Region	Zeit	ε_1	ε_2	R
a.	108	1.95	$0.195 \cdot 10^{-5}$	0.012	0
b.	108	0.91	0.155	1.426	1.425
c.	106	0.50	0.155	1.426	1.425

Tab. 5: Ergebnisse für Testbeispiel 1

Region = benötigter Kernspeicherbereich in K Bytes (1K = 1024 Bytes)
Zeit = CPU-Zeit für IBM/370 - 168 in Sekunden
ε_1 = max{|exakter-interpolierter)/exakter Wert|}
ε_2 = max{|exakter-interpolierter Wert|}
R = Restglied (zu vergleichen mit ε_2)

Testbeispiel 2: Zweidimensionale Interpolation

Zu interpolierende Funktion:

$$F(x,y) = 1.0 + 2.0x + 3.0y + 1.5xy + x^2 + y^2$$

Anzahl der Gruppen: 2

Die Gruppenabhängigkeit wird erreicht durch Addition von 100.0 zu den Funktionswerten in der zweiten Gruppe.

Art der Interpolation entlang der Koordinatenachsen:

a) quadratisch (NX = NY = 3; R aus (22) ist dann 0)

b) linear (NX = NY = 2)

c) linear, aber mit dem Unterprogramm aus Anhang C.

Das Gitter der Stützwerte ist gegeben durch

x : 0.0 1.0 ... 18.0 19.0

y : 0.0 1.0 ... 18.0 19.0,

die Interpolationspunkte haben die Koordinaten

x: 0.5, 2.5, ..., 16.5, 18.5

y: 0.5, 2.5, ..., 16.5, 18.5

Für b) und c) ist $R = (x-x_1)(x-x_2) + (y-y_1)(y-y_2) = 0.5$

Fall	Region	Zeit	ε_1	ε_2	R
a.	74	0.40	0.0	0.0	0.0
b.	74	0.39	0.114	0.500	0.5
c.	72	0.36	0.114	0.500	0.5

Tab. 6: Ergebnisse für Testbeispiel 2

4. Eingabebeschreibung

Die Eingabe für alphanumerischen Text erfolgt im Format 18A4, während die numerischen Werte formatfrei nach den Regeln des "List-directed" (s. /23/) Lesens eingegeben werden (Eingabe numerischer Werte in der üblichen FORTRAN Schreibweise; Wiederholungsfaktoren können in der Form n* angegeben werden - 4 * 1 entspricht 1 1 1 1; Trennzeichen ist das Komma oder Blank; Schrägstrich -/- beendet die Eingabe für den aktuellen Lesebefehl).

Reihenfolge und Bedeutung der Eingabegrößen

Jedes Ki (i=0, 18) repräsentiert einen Satz der Eingabe, der durch einen Schrägstrich (/) beendet wird. Die S_j (j=1,7) stellen Entscheidungspunkte dar.

K0 : TEXT (18)
Alphanumerischer Text zur Fallkennzeichnung

K1 : (Dimensionierungs- und Steuergrößen)

MG — Anzahl der Gruppen, ≥ 1

IBUF — Anzahl der für Puffer zu reservierenden K-Bytes Blöcke (1 KByte = 1024 Bytes), ≥ 1

NWRIT — > 0 : Ausdruck der interpolierten Werte
≤ 0 : kein Ausdrucken der Ergebnisse

INDA — > 0 : Koordinaten der Stützwerte liegen als 3 unformatierte Sätze achsenweise (Reihenfolge: x-, y-, z-Achse) auf der Eingabeeinheit 1 vor.
≤ 0 : ohne Bedeutung

$$\left.\begin{array}{ll} \mathrm{NX} & \geq 2, \\ \mathrm{NY} & \geq 2, \\ \mathrm{NZ} & \geq 1, \end{array}\right\} \text{Grad der Interpolation entlang der} \left\{\begin{array}{c} X \\ Y \\ Z \end{array}\right\}\text{-Achse ist} \left\{\begin{array}{l} \mathrm{NX}-1 \\ \mathrm{NY}-1 \\ \mathrm{NZ}-1 \end{array}\right. \quad {}^{+)}$$

K2 : (Eingabegrößen für die Ausgangssituation)

MZA — ≥ 2, Anzahl der Gitterzeilen einer Ebene (X-Richtung)

MSA — ≥ 2, Anzahl der Gitterspalten einer Ebene (Y-Richtung)

MEA — ≥ 1, Anzahl der Gitterebenen (Z-Richtung)

+) d.h. zur Interpolation im Punkt P=(x,y,z) wird ein Gitter $G = \{x_1,\ldots,x_{NX}\} \times \{y_1,\ldots,y_{NY}\} \times \{z_1,\ldots,z_{NZ}\}$ verwendet

KPOSA > 0 : auf der Datei der Stützwerte (Dateinummer 1) werden KPOSA Sätze zu Beginn der Verarbeitung überlesen

$\leq$ 0 : keine Wirkung

NFORMA > 0 : die Stützwerte werden mit dem Format FORMA (s. K3) von der Datei 1 gelesen

$\leq$ 0 : die Stützwerte wurden formatfrei auf die Datei 1 geschrieben

MODEA $\leq$ 0 : Eingabe der Koordinaten der Stützpunkte gemäß K9 - K11

> 0 : Eingabe gemäß K6 - K8

$\boxed{\text{S1}}$ Für NFORMA$\leq$0:Sprung nach K4

K3 : (Formatangabe für die Stützwerte; alphanumerischer Text)

FORMA(18) Formatangabe gemäß den Regeln für "run-time"-Format (d.h. ohne vorgestellte Qualifizierung "FORMAT", z.B. FORMAT(10G 13.4) wird eingegeben als (10 G13.4)).

K4 : (Eingabegrößen für die Interpolationssituation)

MZN, MSN, MEN: Die Größen haben eine analoge Bedeutung wie in K2 (der Endbuchstabe A steht für alt, N für neu), nur daß sich die Variablen auf das Gitter der zu interpolierenden Punkte beziehen.

KPOSN analog zu KPOSA für die Datei 2

NFORMN analog zu NFORMA : formatierte bzw. unformatierte Ausgabe des interpolierten Feldes auf die Datei 2

MODEN analog zu MODEA, siehe S5

$\boxed{\text{S2}}$ Für NFORMN < 0 : Sprung nach S3

K5 : (Formatangabe zum Ausschreiben der interpolierten Werte)

FORMN(18) Formatangabe zum Ausschreiben der interpolierten Werte auf die Datei 2

$\boxed{\text{S3}}$ Für INDA > 0 (K1) : Sprung nach S5,

Für MODEA $\leq$ 0 (K2) : Sprung nach K9

$\left.\begin{array}{l}\text{K6}\\ \text{K7}\\ \text{K8}\end{array}\right\}$ Eingabe der $\left\{\begin{array}{c}X\\ Y\\ Z\end{array}\right\}$ - Koordinaten der Stützpunkte nach folgendem Schema:+) (hier nur dargestellt für die X-Koordinaten)

+) In der Schreibweise von Formel (5) werden hier also S_1, S_2, S_3 eingegeben, während S vom Programm bestimmt wird.

IANZ : Anzahl der folgenden Werte

X(1) : Anfangswert (i.allg. 0.0)

für i=2, (IANZ-1)/2

IX(i-1) : Anzahl der gleichmäßigen Schritte zwischen X(i-1) und X(i)

X(i) : Endkoordinate für die gleichmäßigen Schritte (X(i) < X(i+1) !)

$\boxed{S4}$ Sprung nach S5

K9, K10, K11: Eingabe der {X, Y, Z} - Koordinaten der Stützpunkte (für jede Achse in aufsteigender Folge, d.h. X(i) < X(i+1) usw.)

$\boxed{S5}$ Für MODEN $\leq$ 0 (K4) : Sprung nach K15

K12, K13, K14: Eingabe analog zu K6-K8 für die Koordinaten der zu interpolierenden Punkte.

$\boxed{S6}$ Sprung nach S7

K15, K16, K17: Eingabe analog zu K9-K11 für die Koordinaten der zu interpolierenden Punkte.

$\boxed{S7}$ Weiter:

K18 (Rechnen eines Folgefalles oder STOP)

Alphatext CONT : Eingabe für einen neuen Fall folgt (Karten K0 - K18)

beliebiger anderer Text : Ende der Eingabe, Programm stoppt.

Bemerkung:

Auf der Datei 1 müssen MG*MEA Sätze von je MZA*MSA Werten bereitstehen, auf der Einheit 2 muß Platz sein für MG*MEN Sätze von je MZN*MSN Worten. Falls nicht genug Kernspeicherplatz vorhanden ist, um eine ganze Gruppe im Kernspeicher zu halten, ist eine "direct access"Datei nötig (Dateinummer 3) mit MEA Sätzen der Länge MZA*MSA Wörter. Im Kernspeicher befinden sich dann nur die jeweiligen NZ Ebenen, die zur Interpolation benötigt werden. Die Dateinummern (1, 2, 3 und 5 = Standardeingabe, 6 = Standardausgabe) werden im BLOCK DATA-Unterprogramm gesetzt. Durch Änderung dieser Initialisierung können neue Dateinummern eingeführt werden.

5. Literatur

/1/ M. Abramowitz, I.A. Stegun (eds.)
Handbook of Mathematical Functions, National Bureau of Standards, Washington, D.C. (1964)

/2/ C. Günther
IPOL - Ein Fortran-Programm zur zweidimensionalen Interpolation (1975), KFK 2175

/3/ Sauer, Szabó
Mathematische Hilfsmittel des Ingenieurs, Teil III, Berlin-Heidelberg-New York (1968)

/4/ H.C. Thacher
Derivation of Interpolation Formulas in Several Independent Variables, Annals of the New York Academy of Sciences, Vol. 86 (1960), S. 758-775

/5/ H.C. Thacher, W.E. Milne
Interpolation in Several Variables J. SIAM, Vol. 8 (1960) S. 33-42

/6/ F.A. Willers
Methoden der praktischen Analysis, Berlin, 1957

/7/ F. Erwe
Differential- und Integralrechnung I, BI - Hochschultaschenbuch, Mannheim/Zürich (1962)

/8/ H.C. Thacher
Generalization of Concepts related to Linear Dependence, J. SIAM, Vol. 6 (1958), S. 288-299

/9/ J. Stoer
Einführung in die Numerische Mathematik I, Berlin-Heidelberg-New York (1972)

/10/ J.F. Steffensen
Interpolation (2nd ed.), Chelsea Publishing Company, New York, (1950)

/11/ W.E. Milne, W. Arntzen, N. Reynolds, J. Wheelock
Mathematics for Digital Computers, Vol. 1 Multivariate Interpolation WADC Technical Report 57-556, Wright Air Development Center, Wright-Patterson Air Force Base, Ohio (1960)

/12/ I.J. Schoenberg (ed.)
Approximations with Special Emphasis on Spline Functions, Academic Press, New York - London, (1969)

/13/ G.G. Lorentz (ed.)
Approximation Theory, Academic Press, New York - London (1973)

/14/ H.E. Salzer
Some New Divided Difference Algorithms for Two Variables, in:
R.E. Langer (ed.): On Numerical Approximation, The University of Wisconsin Press, Madison, (1959)

/15/ G.Birkhoff, H.L. Garabedian
Smooth Surface Interpolation, Journal of Mathematics and Physics, Vol. 39 (1960), S. 258-268

/16/ C. DeBoor
Bicubic Spline Interpolation, Journal of Mathematics and Physics, Vol. 41 (1962), S. 212-218

/17/ J. Ferguson
Multivariable Curve Interpolation, Journal of the ACM, Vol. 11 (1964) S. 221-228

/18/ A. Ženišek
Interpolation Polynomials on the Triangle, Numer. Math., Vol. 15 (1970) S. 283-296

/19/ P.B. Zwart
Multivariate Splines with nondegenerate partitions, SIAM J.Numer. Anal., Vol. 10 (1973), S. 665-673

/20/ G.M. Nielson
Multivariate Smoothing and Interpolating Splines, SIAM J. Numer. Anal., Vol. 11 (1974), S. 435-446

/21/ M.M. Schultz
Multivariate Spline Functions and Elliptic Problems, in /12/

/22/ J.W. Jerome
Topics in Multivariate Approximation Theory, in /13/

/23/ IBM System/360 and System/370, FORTRAN IV Language, IBM Form Nr. GC28-6515-8

/24/ D. Rhind
A Skeletal Overview of Spatial Interpolation Techniques, Comput. Appl., Vol.2 (1975), S. 293 - 309

Anhang A

Programmliste von MODINT

```
C
C
C     ***** NAME :  M O D I N T
C
C     ***** VERSION I  VOM 28.FEBRUAR 1976
C
C     ***** AUTOR:K.KUEFNER
C     *****        KERNFORSCHUNGSZENTRUM KARLSRUHE
C     *****        INSTITUT F. NEUTRONENPHYSIK U. REAKTORTECHNIK
C
C     ***** ANSCHRIFT: POSTFACH 3640 , D-7500 KARLSRUHE
C     *****            BUNDESREPUBLIK DEUTSCHLAND
C
C     ***** ZWECK: INTERPOLATION VON FUNKTIONEN VON 2 UND 3
C     *****        UNABHAENGIGEN VERAENDERLICHEN.
C     *****        STUETZWERTE MUESSEN AUF EINEM GITTER LIEGEN,
C     *****        EBENSO DIE INTERPOLATIONSPUNKTE.
C
C     ***** METHODE: VERALLGEMEINERUNG DER METHODE NACH NEWTON
C     *****          (DIVIDIERTE DIFFERENZEN)
C
C     ***** DOKUMENTATION: LIEGT ALS KFK-BERICHT VOR (KFK     )
C
C
C     *****STEUERPROGRAMM ZUR INTERPOLATION IN MULTIGRUPPEN
C     *****FLUSS-FELDERN
C
      REAL ARB(1),DDNAME*8/'FT03F001'/,DDAT*8,DZEIT*8
      INTEGER IARB(1),CONT,WEITER/'CONT'/
      LOGICAL KLEIN
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *              NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1       /KONSTA/MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     *       /KONSTN/MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
     2       /FORMAT/ FORMA(18),FORMN(18),TEXT(18)
C
      EQUIVALENCE  (ARB(1),IARB(1))
C
      CALL DATUM(DDAT,DZEIT)
      WRITE(NFO,950) DDAT,DZEIT
      WRITE(NFO,957)
    1 KLEIN=.FALSE.
      IK=1
      CALL INPUT(IK)
      IF(IK .LT. 0) GOTO 300
C
C     ***** DYNAMISCHE FELDAUSDEHNUNG
C
      IFIX=MZA+MSA+MEA+MZN+MSN+MEN+4*INPG+IBUF*256+NEU+1
      MEX=MEA
   10 NEB=MEX*NALT
      K2=(IFIX+NEB+2)*4
      K2=2048*((K2+2047)/2048)
C
      CALL XTAREA(K1,K2,K3,ARB(1))
C
      IF(K2 .GT. 0) GOTO 50
      WRITE(NFO,953) MEX
      IF(MEX .LE. NZ) GOTO 180
      MEX=NZ
      KLEIN=.TRUE.
      GOTO 10
C
C     ***** POINTER-BERECHNUNG FUER FELDKOMPONENTEN
C
   50 NST=IBUF*1024
      K2=K2-NST
      IEND=K1+K2
      CALL REXTAR(IEND,NST)
      IFALT=(K1-K3)/4+1
      IFNEU=IFALT+NEB
      IXA  =IFNEU+NEU
      IYA  =IXA  +MZA
      IZA  =IYA  +MSA
      IXN  =IZA  +MEA
      IYN  =IXN  +MZN
      IZN  =IYN  +MSN
      IXS  =IZN  +MEN
      IYS  =IXS  +INPG
      IZS  =IYS  +INPG
      IFS  =IZS  +INPG
      IEND =IFS  +INPG-1
C
C     ***** BESETZUNG DER DREI KOORDINATENFELDER
C
      CALL CALSTE (ARB(IFALT),ARB(IFALT),ARB(IXA),IK,IFIX)
      IF(IK .LT. 0) GOTO 200
C
C     ***** AUFRUF DER INTERPOLATIONSROUTINE
C
      IF(MEA-2) 100,110,110
C
  100 WRITE(NFO,954)
      CALL IN2DIM(ARB(IFALT),ARB(IFNEU),ARB(IXA),ARB(IYA),
     1            ARB(IXN),ARB(IYN),ARB(IXS),ARB(IYS),ARB(IFS))
      WRITE(NFO,958)
      GOTO 200
C
  110 IF(.NOT.KLEIN) GOTO 150
      WRITE(NFO,952)
      CALL DDTEST(1,DDNAME,1,NST)
      CALL DINF(DDNAME,NBLK,NPRIQ)
      NBLK=NBLK/4
      CALL DEFI(NDA,NPRIQ,'U   ',NBLK,NASS)
C
  150 WRITE(NFO,954)
      CALL IN3DIM(ARB(IFALT),ARB(IFNEU),ARB(IXA),ARB(IYA),
     1            ARB(IZA),ARB(IXN),ARB(IYN),ARB(IZN),ARB(IXS),
     2            ARB(IYS),ARB(IZS),ARB(IFS),KLEIN)
      WRITE(NFO,958)
C
      GOTO 200
C
  180 WRITE(NFO,951)
C
```

```
  200 READ(NFI,955,END=300) CONT
      IF(CONT .NE. WEITER) GOTO 300
      WRITE(NFO,956)
      CALL REXTAR(K1,K2)
      REWIND NFA
      REWIND NFN
      GOTO 1
C
  300 WRITE(NFO,954)
      STOP
C
  950 FORMAT('1'/10('0'/),50X,20('*')/50X,'*',18X,'*'/50X,'*',
     1       18X,'*'/50X,'*',4X,'M O D I N T    *'/50X,'*',18X,
     2       '*'/50X,'*',18X,'*'/50X,'* DATUM   :',A8,' *'/50X,
     3       '*',18X,'*'/50X,'*',18X,'*'/50X,'* UHRZEIT:',A8,
     4       ' *'/50X,'*',18X,'*'/50X,20('*')/1H1)
  951 FORMAT('OREGIONANGABE FUER DEN JOB(-STEP) ERHOEHEN ',
     1       'ODER NZ VERKLEINERN')
  952 FORMAT('0'/'OOPTIMIERUNGSSTUFE II'/'0')
  953 FORMAT('0'/'OES PASSEN KEINE  ',I4,'  EBENEN IN DEN ',
     1       'KERNSPEICHER')
  954 FORMAT(1H1)
  955 FORMAT(A4)
  956 FORMAT('1',9X,'FOLGEFALL',5('   *')/'0'/10X,74('*'))
  957 FORMAT('OKONTROLLAUSDRUCK DER EINGABE:'/'0'/10X,74('*'))
  958 FORMAT('0'/'0'/'OINTERPOLATION ORDNUNGSGEMAESS BEENDET')
C
      END
```

```
C
C     ***** INITIALISIERUNG DES COMMON
C
      BLOCK DATA
C
      COMMON /FORMAT/ FORMA(18),FORMN(18),TEXT(18)
     1       /KONST/ MG,IBU,NWR,IN,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INP,NASS
     2       /KONSTA/ MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     3       /KONSTN/ MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
C
      DATA FORMA,FORMN,TEXT/54*'    '/,
     1 MG,IBU,NWR,IN,NX,NY,NZ,NFO,NFI,NDA/1,6,2*0,3*2,6,5,3/,
     2 NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INP,NASS/0,3,0,1,0,1,0,
     * 1,8,1/,
     3 MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA/3*1,3*0,1,1/,
     4 MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN/3*1,3*0,1,2/
C
      END
```

```
C
C     ***** BESETZEN DER KOORDINATENFELDER
C
      SUBROUTINE CALSTE(ARB,IARB,XA,IK,N)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1       /KONSTA/MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     2       /KONSTN/MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
     3       /FORMAT/ FORMA(18),FORMN(18),TEXT(18)
      DIMENSION ARB(1),IARB(1),XA(N),K1C(3)
      REAL ALTE/'ALTE'/,NEUE/'NEUE'/,XKE/'   X '/,
     1     YKE/'   Y '/,ZKE/'   Z '/
C
      IPOINT=1
      IALT=1
      IF2=1
      MODE=MODEA
      K1C(1)=MZA
      K1C(2)=MSA
      K1C(3)=MEA
    5 IF(MODE .GT. 0) GOTO 100
C
C     ***** EINLESEN DER SCHRITTWEITEN
C
      DO 10 I=1,3
      CALL LES(XA(IPOINT),K1C(I),IALT)
   10 IPOINT=IPOINT+K1C(I)
      GOTO 200
C
C     ***** BERECHNEN DER SCHRITTWEITEN
C
  100 DO 120 J=1,3
      READ(NFI,*) IANZ,ARB(2),(IARB(I+1),ARB(I+2),I=2,IANZ,2)
      IARB(1)=IANZ
      IH=K1C(J)
      DO 110 K=1,IANZ,12
      IE=MIN0(K+9,IANZ)+1
      WRITE(NFO,951)(ARB(I),I=K, IE)
      WRITE(NFO,952)
  110 CONTINUE
C
      CALL STEP(ARB,IARB,XA(IPOINT),IH,IFEHL)
C
      IPOINT=IPOINT+IH
C
      IF(IFEHL .NE. 1) GOTO 120
      WRITE(NFO,953)
      IF2=-1
C
  120 CONTINUE
C
C     ***** VORBEREITUNG AUF ZWEITEN DURCHLAUF FUER NEUE KOORD
C
  200 IF(IALT .EQ. 0) GOTO 250
      K1C(1)=MZN
      K1C(2)=MSN
      K1C(3)=MEN
```

```
      MODE=MODEN
      IALT=0
      GOTO 5
C
  250 WRITE(NFO,954)
C
C     ***** KONTROLLAUSDRUCK DER DIMENSIONEN UND
C     ***** DER WERTE DER EINZELNEN KOORDINATEN
C
      WRITE(NFO,950) TEXT
      WRITE(NFO,956) MG,XKE,YKE,ZKE,ALTE,MZA,MSA,MEA,
     1                            NEUE,MZN,MSN,MEN
      WRITE(NFO,958) NX,NY,NZ
      WRITE(NFO,955) ALTE,XKE,(XA(I),I=1,MZA)
      MA=MZA+1
      ME=MZA+MSA
      WRITE(NFO,955) ALTE,YKE,(XA(I),I=MA,ME)
      MA=ME+1
      ME=ME+MEA
      WRITE(NFO,955) ALTE,ZKE,(XA(I),I=MA,ME)
      MA=ME+1
      ME=ME+MZN
      WRITE(NFO,957)
      WRITE(NFO,955) NEUE,XKE,(XA(I),I=MA,ME)
      MA=ME+1
      ME=ME+MSN
      WRITE(NFO,955) NEUE,YKE,(XA(I),I=MA,ME)
      MA=ME+1
      ME=ME+MEN
      WRITE(NFO,955) NEUE,ZKE,(XA(I),I=MA,ME)
C
      IF(IF2 .LT. 0) IK=-1
      RETURN
C
  950 FORMAT(1H1,18A4/'0'/'0')
  951 FORMAT(9X,' *',5(I4,1PE10.3))
  952 FORMAT('+',T84,'*')
  953 FORMAT('0'/'0*** FALSCHE SCHRITTWEITENEINGABE - ABBRUCH')
  954 FORMAT(10X,74('*')/'0')
  955 FORMAT('0',2A4,' - KOORDINATEN :'/(' ',1P10G12.5))
  956 FORMAT('0'/'0ZAHL DER PUNKTE (BEI ',I4,' GRUPPEN) IN'/
     1       '0',7X,3(A4,2X),'  -  RICHTUNG'/
     2       ' ',24('-')/2(' ',A4,2X,3(I4,2X)/' ',24('-')/)/' ')
  957 FORMAT(1H0)
  958 FORMAT('0ZUR INTERPOLATION WERDEN IN X-RICHTUNG ',I4,
     1       ' ,IN Y-RICHTUNG ',I4,' ,UND IN Z-RICHTUNG ',I4,
     2       ' PUNKTE HERANGEZOGEN'/'0')
C
      END



C
C     ***** LESE-UND DRUCKROUTINE FUER DIREKTE
C     ***** KOORDINATENEINGABE
C
      SUBROUTINE LES (XA,MZA,IALT)
C
      DIMENSION XA(MZA)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1       /KONSTA/ K2(7),NFA
C
      IF(IALT.EQ.1 .AND. INDA.GT.0) GOTO 5
    1 READ(NFI,*) XA
      GOTO 6
    5 READ(NFA) XA
    6 DO 10 I=1,MZA,7
      IE=MINO(I+6,MZA)
      WRITE(NFO,951) (XA(K),K=I,IE)
      WRITE(NFO,952)
   10 CONTINUE
C
      RETURN
C
  951 FORMAT(9X,' *',1P7E10.3)
  952 FORMAT('+',T84,'*')
C
      END



C
C     ***** BERECHNEN DER KOORDINATEN
C
      SUBROUTINE STEP(A,IA,X,LX,IFEHL)
C
      DIMENSION A(1),IA(1),X(LX)
C
    1 IFEHL=0
      ISTEP=IA(1)
      X(1)=A(2)
      IF(LX.EQ.1 .OR. ISTEP.EQ.1) GOTO 50
      K=1
      DO 20 I=2,ISTEP,2
      IH=IA(I+1)
      IF(IH.GT.LX .OR. IH.LE.0) GOTO 30
      DX=(A(I+2)-A(I))/IH
      DO 10 J=1,IH
      K=K+1
   10 X(K)=X(K-1)+DX
   20 X(K)=A(I+2)
C
      IF(K.LT.LX) IFEHL=1
      GOTO 50
C
   30 IFEHL=1
C
   50 RETURN
C
      END
```

```
C
C     ***** DRUCKROUTINE FUER INTERPOLIERTES FELD
C
      SUBROUTINE DRUCK (A,IS,IG)
C
      COMMON /KONSTN/MZ,MS,K1(4),N,NFN
     1       /KONST/ K2(7),NFO,NFI,NDA,NEB,KA,K3(8)
C
      DIMENSION A(N)
      WRITE(NFO,901) IS,IG
      IZ=9
      IZM1=-8
C
      DO 15 I=1,MS,IZ
      IH=I-MS
      IF(IH .GT. IZM1) IZ=1-IH
      KE=I+IZ-1
      WRITE(NFO,903) (K,K=I,KE)
      WRITE(NFO,902)
C
      DO 15 J=MS,N,MS
      IA=J+IH
      CALL OUTP(J/MS,A(IA),IZ)
C
   15 CONTINUE
C
      WRITE(NFO,904)
      RETURN
C
  901 FORMAT('0'/'0INTERPOLIERTES FELD FUER EBENE',I4,
     1        ' IN GRUPPE',I4)
  902 FORMAT(1H )
  903 FORMAT(1H0,5X,9(5X,I3,5X))
  904 FORMAT(1H0)
C
      END
```

```
C
C     ***** EINLESEN EINES TEILS DER EINGABE
C
      SUBROUTINE INPUT(IK)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1  /KONSTA/MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     2  /KONSTN/MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
     3  /FORMAT/ FORMA(18),FORMN(18),TEXT(18)
C
      DIMENSION K1C(7),K2C(6),K3C(6)
      EQUIVALENCE (K1C(1),MG),(K2C(1),MZA),(K3C(1),MZN)
C
      READ(NFI,953) TEXT
      WRITE(NFO,954) TEXT
      WRITE(NFO,956)
C
      READ(NFI,*) K1C
      WRITE(NFO,952) K1C
      WRITE(NFO,956)
C
      READ(NFI,*) K2C
      WRITE(NFO,952) K2C
      WRITE(NFO,956)
C
C     ***** EINLESEN DER FORMATE FUER ALTE WERTE
C
      IF(NFORMA .LE. 0) GOTO 20
      READ(NFI,953) FORMA
      WRITE(NFO,954) FORMA
      WRITE(NFO,956)
C
   20 READ(NFI,*) K3C
      WRITE(NFO,952) K3C
      WRITE(NFO,956)
C
C     ***** EINLESEN DER FORMATE FUER NEUE WERTE
C
      IF(NFORMN .LE. 0 ) GOTO 40
      READ(NFI,953) FORMN
      WRITE(NFO,954) FORMN
C
C     ***** POSITIONIERUNG VON FT01F001 UND FT02F001
C
   40 IF(KPOSA .LE.0) GOTO 60
      DO 50 I=1,KPOSA
   50 READ(NFA)
   60 IF(KPOSN .LE. 0) GOTO 80
      DO 70 I=1,KPOSN
   70 READ(NFN)
   80 NALT=MZA*MSA
      NEU=MZN*MSN
      KA=3
      IF(MEA .EQ. 1) KA=KA-1
      IF(MZA .GT. 1 .AND. MSA .GT. 1) GOTO 100
      WRITE(NFO,957)
C
      IK=-1
C
  100 IF(1.LE.NX.AND.NX.LE.MZA.AND.1.LE.NY.AND.NY.LE.MSA
     1   .AND.1.LE.NZ.AND.NZ.LE.MEA) GOTO 150
C
      WRITE(NFO,951)
      IK=-1
      GOTO 200
C
  150 INPG=NX*NY*NZ
      NXL=(NX-1)/2
      NXR=NX/2
      NYL=(NY-1)/2
      NYR=NY/2
      NZL=(NZ-1)/2
      NZR=NZ/2
C
  200 RETURN
```

```
C
  951 FORMAT('0'/'OFALSCHE EINGABE DER ANZAHLEN DER STUETZ',
     1       'STELLEN - ABBRUCH'/'0')
  952 FORMAT(9X,' *',18I4)
  953 FORMAT(18A4)
  954 FORMAT(9X,' *',18A4)
  956 FORMAT('+',T84,'*')
  957 FORMAT('0'/'OLAUT EINGABE GILT: MZA<=1 ODER MSA<=1',
     1       ' -  INTERPOLATION NICHT MOEGLICH ')
C
      END



C
C     ***** EIN-/AUSGABEROUTINEN FUER FELDER
C
      SUBROUTINE WIRE(A,N,IDA)
C
      DIMENSION A(N)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1       /FORMAT/ FORMA(18),FORMN(18),TEXT(18)
     2       /KONSTA/MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     3       /KONSTN/MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
C
      IF(IDA .GT. 0) GOTO 15
      IF(NFORMA .GT. 0) GOTO 10
      READ(NFA) A
      RETURN
C
   10 READ(NFA,FORMA) A
      RETURN
C
   15 READ(NDA'NASS) A
      RETURN
C
      ENTRY OUTPUT(A,N,IDA)
C
      IF(IDA .GT. 0) GOTO 25
      IF(NFORMN .GT. 0) GOTO 20
      WRITE(NFN) A
      RETURN
C
   20 WRITE(NFN,FORMN) A
      RETURN
C
   25 WRITE(NDA'NASS) A
      RETURN
C
C     ***** UMGEHUNG UEBERFLUESSIGER IBCOM#-AUFRUFE BEIM DRUCKEN
C
      ENTRY OUTP(J,A,N)
C
      WRITE(NFO,901) J,A
C
      RETURN
C
  901 FORMAT(I4,1X,1P9E13.5)
C
      END



C
C     ***** FINDET INDEX J, SO DASS X(J)<=Y<=X(J+1)
C     ***** FEHLERMELDUNG IER=1, FALLS X(J)>Y ODER X(N)<Y
C     ***** ODER FALLS J>N-1
C
      SUBROUTINE FIND(X,Y,N,IER,J,NXL,NX,IA,IE)
C
      DIMENSION X(N)
      DATA EM1,EP1 /0.99999,1.00001/
C
C     ***** X(J)>Y
C
      IF(X(J)*EM1 .GT. Y) GOTO 15
C
      N1=N-1
      IF(J .GT. N1) GOTO 15
      DO 10 I=J,N1
      IF(X(I)*EM1.LE.Y .AND. Y.LE.X(I+1)*EP1) GOTO 20
   10 CONTINUE
C
C     ***** X(N)<Y
C
   15 IER=1
      J=0
      RETURN
C
   20 IER=0
      J=I
C
      IA=J-NXL
      IF(IA .LE. 0) IA=1
      IE=IA+NX-1
      IF(IE .LE. N) GOTO 30
      IE=N
      IA=IE-NX+1
C
   30 RETURN
C
      END



C
C     ***** SUBROUTINE ZUR 2-DIMENSIONALEN INTERPOLATION
C
      SUBROUTINE IN2DIM(FALT,FNEU,XA,YA,XN,YN,X,Y,F)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
```

```
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1         /KONSTA/ MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     2         /KONSTN/ MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
C
      DIMENSION FALT(NALT),FNEU(NEU),XA(MZA),YA(MSA),XN(MZN),
     1          YN(MSN),X(INPG),Y(INPG),F(INPG),ARG(3)
      KS=0
C
C     ***** GRUPPENSCHLEIFE
C
      DO 100 IGRUPP=1,MG
C
      CALL WIRE(FALT,NALT,0)
      IZ=1
      I1=1
C
C     ***** SCHLEIFE UEBER DIE ZEILEN DES NEUEN FELDES
C
      DO 60 IZEIL=1,MZN
C
      ISPALT=0
      HZ=XN(IZEIL)
      ARG(1)=HZ
      CALL FIND(XA,HZ,MZA,IER,IZ,NXL,NX,IAX,IEX)
      IF(IER .EQ. 1) GOTO 150
      IA=0
      DO 15 I=1,NX
      H=XA(IAX+I-1)
      DO 10 J=1,NY
   10 X(IA+J)=H
   15 IA=IA+NY
      IAX=(IAX-1)*MSA-1
C
      IS=1
C
C     ***** SCHLEIFE UEBER DIE SPALTEN DES NEUEN FELDES
C
      DO 50 ISPALT=1,MSN
C
      HS=YN(ISPALT)
      ARG(2)=HS
      CALL FIND(YA,HS,MSA,IER,IS,NYL,NY,IAY,IEY)
      IF(IER .EQ. 1) GOTO 150
      IA=1
      DO 25 I=1,NY
      H=YA(IAY+I-1)
      DO 20 J=1,NX
      Y(IA)=H
   20 IA=IA+NY
   25 IA=IA-NX*NY+1
C
      IJ=1
      IND=IAX+IAY
      DO 35 I=1,NX
      DO 30 J=1,NY
      F(IJ)=FALT(IND+J)
   30 IJ=IJ+1
   35 IND=IND+MSA
C
      CALL INTPOL(X,Y,X,F,ARG,KS,FNEU,I1)
      IF(I1 .GT. NEU) GOTO 70
C
      IF(KS .EQ. 0) GOTO 50
      CALL FEHLER(KS,IGRUPP,1,IZEIL,ISPALT,X,Y,Z,F,ARG,FNEU,I1)
C
   50 CONTINUE
   60 CONTINUE
C
   70 CALL OUTPUT(FNEU,NEU,0)
      IF(NWRIT .GT. 0) CALL DRUCK(FNEU,1,IGRUPP)
C
  100 CONTINUE
      RETURN
C
  150 WRITE(NFO,951)IZEIL,ISPALT,IGRUPP
      RETURN
C
  951 FORMAT('OALTE UND NEUE KOORDINATEN UNVERTRAEGLICH FUER ',
     1       'DEN NEUEN PUNKT'/' IN DER ',I4,'-TEN ZEILE,',I4,
     2       '-TEN SPALTE UND DER ',I4,'-TEN GRUPPE - ABBRUCH')
C
      END
```



```
C
C     ***** SUBROUTINE ZUR 3-DIMENSIONALEN INTERPOLATION
C
      SUBROUTINE IN3DIM(FALT,FNEU,XA,YA,ZA,XN,YN,ZN,X,Y,Z,F,KLEIN)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     1         /KONSTA/ MZA,MSA,MEA,KPOSA,NFORMA,MODEA,NALT,NFA
     2         /KONSTN/ MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
C
      DIMENSION FALT(NEB),FNEU(NEU),XA(MZA),YA(MSA),ZA(MEA),
     1   XN(MZN),YN(MSN),ZN(MEN),X(INPG),Y(INPG),Z(INPG),
     2   ARG(3),F(INPG)
      LOGICAL KLEIN
      KS=0
C
C     ***** GRUPPENSCHLEIFE
C
      NXY=NX*NY
      DO 100 IGRUPP=1,MG
      IF(NWRIT.GT.0.AND.IGRUPP.GT.1) WRITE(NFO,952)
      IEA=-1
      IEN=1
C
C     ***** EINLESEN EINER GRUPPE (FALLS MOEGLICH)
C
      IF(KLEIN) GOTO 5
      IND=1
      DO 7 I=1,MEA
```

```
      CALL WIRE(FALT(IND),NALT,0)
    7 IND=IND+NALT
      GOTO 8
    5 NASS=1
      DO 3 I=1,MEA
      CALL WIRE(FALT,NALT,0)
    3 CALL OUTPUT(FALT,NALT,1)
C
C     ***** EBENENSCHLEIFE
C
    8 DO 80 IEBEN=1,MEN
      IZEIL=0
      ARG(3)=ZN(IEBEN)
      CALL FIND(ZA,ARG(3),MEA,IER,IEN,NZL,NZ,IAZ,IEZ)
      IF(IER .EQ. 1) GOTO 150
      IA=0
      DO 2 I=1,NZ
      H=ZA(IAZ+I-1)
      DO 1 J=1,NXY
    1 Z(IA+J)=H
    2 IA=IA+NXY
C
      IF(KLEIN) GOTO 9
C
C     ***** BERECHNUNG DER ANFANGSINDIZES DER ZU IEN
C     ***** GEHOERENDEN EBENEN (IN FALT)
C
      IND1=(IAZ-1)*NALT+1
      GOTO 20
C
C     ***** BEREITSTELLEN DER ZU IEN GEHOERENDEN EBENEN WENN NUR
C     ***** NZ EBENEN IN DIE KERNSPEICHERREGION PASSEN
C
    9 IF(IEN-IEA-1) 20,10,10
C
   10 NASS=IAZ
      J=1
      DO 4 I=1,NZ
      CALL WIRE(FALT(J),NALT,1)
    4 J=J+NALT
      IND1=1
C
   20 IEA=IEN
      IZ=1
      I1=1
C
C     ***** SCHLEIFE UEBER DIE ZEILEN DES NEUEN FELDES
C
      DO 60 IZEIL=1,MZN
      ISPALT=0
      ARG(1)=XN(IZEIL)
      CALL FIND(XA,ARG(1),MZA,IER,IZ,NXL,NX,IAX,IEX)
      IF(IER .EQ. 1) GOTO 150
      IA=0
      DO 25 J=1,NZ
      DO 25 I=1,NX
      H=XA(IAX+I-1)
      DO 25 K=1,NY
      IA=IA+1
      X(IA)=H
   25 CONTINUE
C
      IS=1
C
C     ***** SCHLEIFE UEBER DIE SPALTEN DES NEUEN FELDES
C
      DO 5C ISPALT=1,MSN
      ARG(2)=YN(ISPALT)
      CALL FIND(YA,ARG(2),MSA,IER,IS,NYL,NY,IAY,IEY)
      IF(IER .EQ. 1) GOTO 150
      DO 35 I=1,NY
      H=YA(IAY+I-1)
      IA=I
      DO 35 J=1,INPG,NY
      Y(IA)=H
      IA=IA+NY
   35 CONTINUE
C
C     ***** BESETZUNG DER STUETZWERTE ZUR INTERPOLATION
C
      IJ=0
      IND2=IND1-2+(IAX-1)*MSA
      DO 38 I=1,NZ
      IND=IND2+IAY
      DO 37 J=1,NX
      DO 36 K=1,NY
      IJ=IJ+1
   36 F(IJ)=FALT(IND+K)
   37 IND=IND+MSA
   38 IND2=IND2+NALT
C
C     ***** AUFRUF DER INTERPOLATIONSROUTINE
C
C
      CALL INTPOL(X,Y,Z,F,ARG,KS,FNEU,I1)
      IF(I1 .GT. NEU) GOTO 70
C
      IF(KS .EQ. 0) GOTO 50
      CALL FEHLER(KS,IGRUPP,IEBEN,IZEIL,ISPALT,X,Y,Z,F,
     1            ARG,FNEU,I1)
C
   50 CONTINUE
   60 CONTINUE
C
   70 CALL OUTPUT(FNEU,NEU,0)
      IF(NWRIT .GT. 0) CALL DRUCK(FNEU,IEBEN,IGRUPP)
C
   80 CONTINUE
C
      NASS=1
C
  100 CONTINUE
C
      RETURN
C
  150 WRITE(NFO,951)IZEIL,ISPALT,IEBEN
```

```
      RETURN
C
  951 FORMAT('OALTE UND NEUE KOORDINATEN UNVERTRAEGLICH FUER ',
     1       'DEN NEUEN PUNKT'/' IN DER ',I4,'-TEN ZEILE ,',I4,
     2       '-TEN SPALTE UND DER ',I4,'-TEN EBENE  -  ABBRUCH')
  952 FORMAT(1H1)
C
      END
```

```
C
C     ***** FEHLERBEHANDLUNG NACH FEHLERHAFTEM AUSFUEHREN
C     ***** VON   INTPOL
C
      SUBROUTINE FEHLER(KS,I,J,K,L,X,Y,Z,F,ARG,FN,I1)
C
      COMMON /KONST/K2(7),NFO,K1(12)
C
      DIMENSION X(1),Y(1),Z(1),F(1),ARG(3),FN(1)
C
      WRITE(NFO,950) I,J,K,L,KS
      KS=0
C
      RETURN
C
  950 FORMAT(' FALSCHE INTERPOLATION IN GRUPPE ',I4,' ,EBENE ',
     1       I4,' ,ZEILE ',I4,' ,SPALTE ',I4,' ;FEHLERCODE:',I4)
C
      END
```

```
C
C     ***** BERECHNUNG DER DIVIDIERTEN DIFFERENZEN ENTLANG
C     ***** EINER KOORDINATENACHSE
C
      SUBROUTINE DIVDIF(X,F,N,IV,KS)
C
      DIMENSION X(N),F(N)
C
      NM1=(N-1)*IV
      NM3=NM1+1
C
      DO 10 J=IV,NM1,IV
      I1=NM3+J
      DO 10 I=J,NM1,IV
      I2=I1-I
      H=X(I2)-X(I2-J)
      IF(ABS(H) .LT. 1.0E-12) GOTO 20
      F(I2)=(F(I2)-F(I2-IV))/H
   10 CONTINUE
C
      GOTO 30
C
   20 KS=1
C
   30 RETURN
C
C     ***** ABGEWANDELTES HORNERSCHEMA ZUR AUSWERTUNG DES
C     ***** NEWTON- POLYNOMS
C
      ENTRY HORNER(Z,X,F,N,P,IV)
C
      NM1=N-1
      I1=NM1*IV+1
      Q=F(N)
      IF(N .EQ. 1) GOTO 70
      DO 60 I=1,NM1
      I1=I1-IV
      I2=N-I
   60 Q=Q*(Z-X(I1))+F(I2)
C
   70 P=Q
C
      RETURN
C
      END
```

```
C
C     ***** INTERPOLATIONSROUTINE
C
      SUBROUTINE INTPOL(X,Y,Z,F,ARG,KS,FKT,IF1)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     1               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
C
      DIMENSION X(INPG),Y(INPG),Z(INPG),F(INPG),FKT(1),ARG(3)
C
    1 NXY=NX*NY
      IA=1
C
C     ***** BERECHNUNG DER DIVIDIERTEN DIFFERENZEN
C     ***** ENTLANG DER JEWEILIGEN KOORDINATENACHSE
C
      DO 20 I=1,NZ
      DO 10 J=1,NX
      CALL DIVDIF(Y,F(IA),NY,1,KS)
      IA=IA+NY
      IF(KS .NE. 0) GOTO 80
   10 CONTINUE
C
      DO 20 J=1,NY
      CALL DIVDIF(X,F((I-1)*NXY+J),NX,NY,KS)
      IF(KS .NE. 0) GOTO 80
   20 CONTINUE
C
      IF(KA .NE. 3) GOTO 40
C
      DO 30 I=1,NXY
      CALL DIVDIF(Z,F(I),NZ,NXY,KS)
```

```
      IF(KS .NE. 0) GOTO 80
   30 CONTINUE
C
C     ***** BERECHNUNG DES FUNKTIONSWERTS MIT EINEM
C     ***** HORNERSCHEMA-AEHNLICHEN ALGORITHMUS
C
   40 I1=1
      I2=0
      DO 60 J=1,NZ
      DO 50 I=1,NX
      CALL HORNER(ARG(2),Y,F(I1),NY,F(I+I2),1)
   50 I1=I1+NY
C
      I2=I2+1
      CALL HORNER(ARG(1),X,F(I2),NX,F(I2),NY)
   60 CONTINUE
      IF(KA .EQ. 3) GOTO 70
      FKT(IF1)=F(I2)
      GOTO 80
C
   70 CALL HORNER(ARG(3),Z,F,NZ,FKT(IF1),NXY)
   80 IF1=IF1+1
C
      RETURN
C
      END
```

Anhang B

Ausdruck von MODINT für Testbeispiel 1 und 2 aus Kap. 3.d.

M O D I N T

DATUM :03.03.76

UHRZEIT:14.34.35

KONTROLLAUSDRUCK DER EINGABE:

```
****************************************************************************
*TESTBEISPIEL1 FUER MODINT (3D,DREIFACHE QUADRATISCHE INTERPOLATION)       *
*   1   8   0   0   3   3   3                                              *
*  20  20  20   3   0   1                                                  *
*  10  10  10   0   0   1                                                  *
*   3 0.0         19 1.900E+01                                             *
*   3 0.0         19 1.900E+01                                             *
*   3 0.0         19 1.900E+01                                             *
*   3 5.000E-01    9 1.850E+01                                             *
*   3 5.000E-01    9 1.850E+01                                             *
*   3 5.000E-01    9 1.850E+01                                             *
****************************************************************************
```

TESTBEISPIEL1 FUER MODINT (3D,DREIFACHE QUADRATISCHE INTERPOLATION)

ZAHL DER PUNKTE (BEI 1 GRUPPEN) IN

	X	Y	Z	- RICHTUNG
ALTE	20	20	20	
NEUE	10	10	10	

ZUR INTERPOLATION WERDEN IN X-RICHTUNG 3 ,IN Y-RICHTUNG 3 ,UND IN Z-RICHTUNG 3 PUNKTE HERANGEZOGEN

ALTE X - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Y - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Z - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

NEUE X - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Y - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Z - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

INTERPOLATICN ORDNUNGSGEMAESS BEENDET

```
*******************
*                 *
*    M O D I N T  *
*                 *
* DATUM  :03.03.76 *
*                 *
* UHRZEIT:14.35.16 *
*                 *
*******************
```

KONTROLLAUSDRUCK DER EINGABE:

```
***************************************************************************
*TESTBEISPIEL FUER MODINT (3D, DREIFACHE LINEARE INTERPOLATION)           *
*   1    8    0    0    2    2    2                                       *
*  20   20   20    3    0    1                                            *
*  10   10   10   10    0    1                                            *
*   3 0.0          19 1.900E+01                                           *
*   3 0.0          19 1.900E+01                                           *
*   3 0.0          19 1.900E+01                                           *
*   3 5.000E-01     9 1.850E+01                                           *
*   3 5.000E-01     9 1.850E+01                                           *
*   3 5.000E-01     9 1.850E+01                                           *
***************************************************************************
```

TESTBEISPIEL FUER MODINT (3D, DREIFACHE LINEARE INTERPOLATION)

ZAHL DER PUNKTE (BEI 1 GRUPPEN) IN

	X	Y	Z	- RICHTUNG
ALTE	20	20	20	
NEUE	10	10	10	

ZUR INTERPOLATION WERDEN IN X-RICHTUNG 2 ,IN Y-RICHTUNG 2 ,UND IN Z-RICHTUNG 2 PUNKTE HERANGEZOGEN

ALTE X - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Y - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Z - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

NEUE X - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Y - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Z - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

INTERPOLATION ORDNUNGSGEMAESS BEENDET

```
********************
*                  *
*   M O D I N T    *
*                  *
*  DATUM  :03.03.76 *
*                  *
* UHRZEIT:14.39.31 *
*                  *
********************
```

KONTROLLAUSDRUCK DER EINGABE:

```
****************************************************************************
*TESTBEISPIEL FUER MODINT (2D,ZWEIFACHE QUADRATISCHE INTERPOLATION)        *
*   2   8   1   1   3   3   1                                              *
*  20  20   1   0   0   0                                                  *
*  10  10   1   0   0   1                                                  *
* 0.0       1.000E+00 2.000E+00 3.000E+00 4.000E+00 5.000E+00 6.000E+00    *
* 7.000E+00 8.000E+00 9.000E+00 1.000E+01 1.100E+01 1.200E+01 1.300E+01    *
* 1.400E+01 1.500E+01 1.600E+01 1.700E+01 1.800E+01 1.900E+01              *
* C.0       1.000E+00 2.000E+00 3.000E+00 4.000E+00 5.000E+00 6.000E+00    *
* 7.000E+00 8.000E+00 9.000E+00 1.000E+01 1.100E+01 1.200E+01 1.300E+01    *
* 1.400E+01 1.500E+01 1.600E+01 1.700E+01 1.800E+01 1.900E+01              *
* 0.0                                                                      *
*    3 5.000E-01    9 1.850E+01                                            *
*    3 5.000E-01    9 1.850E+01                                            *
*    1 0.0                                                                 *
****************************************************************************
```

TESTBEISPIEL FUER MODINT (2D,ZWEIFACHE QUADRATISCHE INTERPOLATION)

ZAHL DER PUNKTE (BEI 2 GRUPPEN) IN

	X	Y	Z	- RICHTUNG
ALTE	20	20	1	
NEUE	10	10	1	

ZUR INTERPOLATION WERDEN IN X-RICHTUNG 3 ,IN Y-RICHTUNG 3 ,UND IN Z-RICHTUNG 1 PUNKTE HERANGEZOGEN

ALTE X - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Y - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Z - KOORDINATEN :
.0

NEUE X - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Y - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Z - KOORDINATEN :
.0

INTERPOLIERTES FELD FUER EBENE 1 IN GRUPPE 1

	1	2	3	4	5	6	7	8	9	10
1	4.37500E+00	1.78750E+01	3.93750E+01	6.88750E+01	1.06375E+02	1.51875E+02	2.05375E+02	2.66875E+02	3.36375E+02	4.13875E+02
2	1.58750E+01	3.53750E+01	6.28750E+01	9.83750E+01	1.41875E+02	1.93375E+02	2.52875E+02	3.20375E+02	3.95875E+02	4.79375E+02
3	3.53750E+01	6.08750E+01	9.43750E+01	1.35875E+02	1.85375E+02	2.42875E+02	3.08375E+02	3.81875E+02	4.63375E+02	5.52875E+02
4	6.28750E+01	9.43750E+01	1.33875E+02	1.81375E+02	2.36875E+02	3.00375E+02	3.71875E+02	4.51375E+02	5.38875E+02	6.34375E+02
5	9.83750E+01	1.35875E+02	1.81375E+02	2.34875E+02	2.96375E+02	3.65875E+02	4.43375E+02	5.28875E+02	6.22375E+02	7.23875E+02
6	1.41875E+02	1.85375E+02	2.36875E+02	2.96375E+02	3.63875E+02	4.39375E+02	5.22875E+02	6.14375E+02	7.13875E+02	8.21375E+02
7	1.93375E+02	2.42875E+02	3.00375E+02	3.65875E+02	4.39375E+02	5.20875E+02	6.10375E+02	7.07875E+02	8.13375E+02	9.26875E+02
8	2.52875E+02	3.08375E+02	3.71875E+02	4.43375E+02	5.22875E+02	6.10375E+02	7.05875E+02	8.09375E+02	9.20875E+02	1.04037E+03
9	3.20375E+02	3.81875E+02	4.51375E+02	5.28875E+02	6.14375E+02	7.07875E+02	8.09375E+02	9.18875E+02	1.03637E+03	1.16187E+03
10	3.95875E+02	4.63375E+02	5.38875E+02	6.22375E+02	7.13875E+02	8.13375E+02	9.20875E+02	1.03637E+03	1.15987E+03	1.29137E+03

INTERPOLIERTES FELD FUER EBENE 1 IN GRUPPE 2

	1	2	3	4	5	6	7	8	9	10
1	-4.56250E+01	-3.21250E+01	-1.06250E+01	1.88750E+01	5.63750E+01	1.01875E+02	1.55375E+02	2.16875E+02	2.86375E+02	3.63875E+02
2	-3.41250E+01	-1.46250E+01	1.28750E+01	4.83750E+01	9.18750E+01	1.43375E+02	2.02875E+02	2.70375E+02	3.45875E+02	4.29375E+02
3	-1.46250E+01	1.08750E+01	4.43750E+01	8.58750E+01	1.35375E+02	1.92875E+02	2.58375E+02	3.31875E+02	4.13375E+02	5.02875E+02
4	1.28750E+01	4.43750E+01	8.38750E+01	1.31375E+02	1.86875E+02	2.50375E+02	3.21875E+02	4.01375E+02	4.88875E+02	5.84375E+02
5	4.83750E+01	8.58750E+01	1.31375E+02	1.84875E+02	2.46375E+02	3.15875E+02	3.93375E+02	4.78875E+02	5.72375E+02	6.73875E+02
6	9.18750E+01	1.35375E+02	1.86875E+02	2.46375E+02	3.13875E+02	3.89375E+02	4.72875E+02	5.64375E+02	6.63875E+02	7.71375E+02
7	1.43375E+02	1.92875E+02	2.50375E+02	3.15875E+02	3.89375E+02	4.70875E+02	5.60375E+02	6.57875E+02	7.63375E+02	8.76875E+02
8	2.02875E+02	2.58375E+02	3.21875E+02	3.93375E+02	4.72875E+02	5.60375E+02	6.55875E+02	7.59375E+02	8.70875E+02	9.90375E+02
9	2.70375E+02	3.31875E+02	4.01375E+02	4.78875E+02	5.64375E+02	6.57875E+02	7.59375E+02	8.68875E+02	9.86375E+02	1.11187E+03
10	3.45875E+02	4.13375E+02	4.88875E+02	5.72375E+02	6.63875E+02	7.63375E+02	8.70875E+02	9.86375E+02	1.10987E+03	1.24137E+03

INTERPOLATION ORDNUNGSGEMAESS BEENDET

```
********************
*                  *
*                  *
*    M O D I N T   *
*                  *
*                  *
* DATUM  :03.03.76 *
*                  *
*                  *
* UHRZEIT:14.39.42 *
*                  *
********************
```

KONTROLLAUSDRUCK DER EINGABE:

```
******************************************************************************
*TESTBEISPIEL FUER MODINT (2D, ZWEIFACHE LINEARE INTERPOLATION)              *
*   2   8   1   1   2   2   1                                                *
*  20  20   1   0   0   0                                                    *
*  10  10   1   2   0   1                                                    *
* 0.0        1.000E+00 2.000E+00 3.000E+00 4.000E+00 5.000E+00 6.000E+00     *
* 7.000E+00 8.000E+00 9.000E+00 1.000E+01 1.100E+01 1.200E+01 1.300E+01     *
* 1.400E+01 1.500E+01 1.600E+01 1.700E+01 1.800E+01 1.900E+01               *
* 0.0        1.000E+00 2.000E+00 3.000E+00 4.000E+00 5.000E+00 6.000E+00     *
* 7.000E+00 8.000E+00 9.000E+00 1.000E+01 1.100E+01 1.200E+01 1.300E+01     *
* 1.400E+01 1.500E+01 1.600E+01 1.700E+01 1.800E+01 1.900E+01               *
* 0.0                                                                        *
*   3 5.000E-01   9 1.850E+01                                                *
*   3 5.000E-01   9 1.850E+01                                                *
*   1 0.0                                                                    *
******************************************************************************
```

TESTBEISPIEL FUER MODINT (2D, ZWEIFACHE LINEARE INTERPOLATION)

ZAHL DER PUNKTE (BEI 2 GRUPPEN) IN

	X	Y	Z	- RICHTUNG
ALTE	20	20	1	
NEUE	10	10	1	

ZUR INTERPOLATION WERDEN IN X-RICHTUNG 2 ,IN Y-RICHTUNG 2 ,UND IN Z-RICHTUNG 1 PUNKTE HERANGEZOGEN

ALTE X - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Y - KOORDINATEN :

.0	1.0000	2.0000	3.0000	4.0000	5.0000	6.0000	7.0000	8.0000	9.0000
10.000	11.000	12.000	13.000	14.000	15.000	16.000	17.000	18.000	19.000

ALTE Z - KOORDINATEN :

.0

NEUE X - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Y - KOORDINATEN :

.50000	2.5000	4.5000	6.5000	8.5000	10.500	12.500	14.500	16.500	18.500

NEUE Z - KOORDINATEN :

.0

INTERPOLIERTES FELD FUER EBENE 1 IN GRUPPE 1

	1	2	3	4	5	6	7	8	9	10
1	4.87500E+00	1.83750E+01	3.98750E+01	6.93750E+01	1.06875E+02	1.52375E+02	2.05875E+02	2.67375E+02	3.36875E+02	4.14375E+02
2	1.63750E+01	3.58750E+01	6.33750E+01	9.88750E+01	1.42375E+02	1.93875E+02	2.53375E+02	3.20875E+02	3.96375E+02	4.79875E+02
3	3.58750E+01	6.13750E+01	9.48750E+01	1.36375E+02	1.85875E+02	2.43375E+02	3.08875E+02	3.82375E+02	4.63875E+02	5.53375E+02
4	6.33750E+01	9.48750E+01	1.34375E+02	1.81875E+02	2.37375E+02	3.00875E+02	3.72375E+02	4.51875E+02	5.39375E+02	6.34875E+02
5	9.88750E+01	1.36375E+02	1.81875E+02	2.35375E+02	2.96875E+02	3.66375E+02	4.43875E+02	5.29375E+02	6.22875E+02	7.24375E+02
6	1.42375E+02	1.85875E+02	2.37375E+02	2.96875E+02	3.64375E+02	4.39875E+02	5.23375E+02	6.14875E+02	7.14375E+02	8.21875E+02
7	1.93875E+02	2.43375E+02	3.00875E+02	3.66375E+02	4.39875E+02	5.21375E+02	6.10875E+02	7.08375E+02	8.13875E+02	9.27375E+02
8	2.53375E+02	3.08875E+02	3.72375E+02	4.43875E+02	5.23375E+02	6.10875E+02	7.06375E+02	8.09875E+02	9.21375E+02	1.04087E+03
9	3.20875E+02	3.82375E+02	4.51875E+02	5.29375E+02	6.14875E+02	7.08375E+02	8.09875E+02	9.19375E+02	1.03687E+03	1.16237E+03
10	3.96375E+02	4.63875E+02	5.39375E+02	6.22875E+02	7.14375E+02	8.13875E+02	9.21375E+02	1.03687E+03	1.16037E+03	1.29187E+03

INTERPOLIERTES FELD FUER EBENE 1 IN GRUPPE 2

	1	2	3	4	5	6	7	8	9	10
1	-4.51250E+01	-3.16250E+01	-1.01250E+01	1.93750E+01	5.68750E+01	1.02375E+02	1.55875E+02	2.17375E+02	2.86875E+02	3.64375E+02
2	-3.36250E+01	-1.41250E+01	1.33750E+01	4.88750E+01	9.23750E+01	1.43875E+02	2.03375E+02	2.70875E+02	3.46375E+02	4.29875E+02
3	-1.41250E+01	1.13750E+01	4.48750E+01	8.63750E+01	1.35875E+02	1.93375E+02	2.58875E+02	3.32375E+02	4.13875E+02	5.03375E+02
4	1.33750E+01	4.48750E+01	8.43750E+01	1.31875E+02	1.87375E+02	2.50875E+02	3.22375E+02	4.01875E+02	4.89375E+02	5.84875E+02
5	4.88750E+01	8.63750E+01	1.31875E+02	1.85375E+02	2.46875E+02	3.16375E+02	3.93875E+02	4.79375E+02	5.72875E+02	6.74375E+02
6	9.23750E+01	1.35875E+02	1.87375E+02	2.46875E+02	3.14375E+02	3.89875E+02	4.73375E+02	5.64875E+02	6.64375E+02	7.71875E+02
7	1.43875E+02	1.93375E+02	2.50875E+02	3.16375E+02	3.89875E+02	4.71375E+02	5.60875E+02	6.58375E+02	7.63875E+02	8.77375E+02
8	2.03375E+02	2.58875E+02	3.22375E+02	3.93875E+02	4.73375E+02	5.60875E+02	6.56375E+02	7.59875E+02	8.71375E+02	9.90875E+02
9	2.70875E+02	3.32375E+02	4.01875E+02	4.79375E+02	5.64875E+02	6.58375E+02	7.59875E+02	8.69375E+02	9.86875E+02	1.11237E+03
10	3.46375E+02	4.13875E+02	4.89375E+02	5.72875E+02	6.64375E+02	7.63875E+02	8.71375E+02	9.86875E+02	1.11037E+03	1.24187E+03

INTERPOLATION ORDNUNGSGEMAESS BEENDET

Anhang C Alternatives Interpolationsunterprogramm

Für Funktionen in zwei Veränderlichen kann (23) aus Kapitel 2 auf eine einfache Form gebracht werden. Sei zur Abkürzung

$$p = \frac{x-x_o}{x_1-x_o}, \qquad q = \frac{y-y_o}{y_i-y_o}.$$

Dann ist (Bezeichnungen wie in Kap. 2):

$$
\begin{aligned}
f[0;0] &= F(x_o,y_o)\\
(y-y_o)f[0;1] &= (y-y_o)\, f\left[x_o;y_o,y_1\right] = q(F(x_o,y_1) - F(x_o,y_o))\\
(x-x_o)f[1;0] &= p(F(x_1,y_o) - F(x_o,y_o))\\
(x-x_o)(y-y_o)\, f[1;1] &= (x-x_o)(y-y_o)\, f\left[x_o,x_1;y_o,y_1\right] =\\
&= (x-x_o)\, q(f\left[x_o,x_1;y_1\right] - f\left[x_o,x_1;y_o\right]\\
&= p\cdot q\ (F(x_1,y_1)-F(x_o,y_1) - F(x_1,y_o) + F(x_o,y_o)).
\end{aligned}
$$

Damit kann (23) ungeformt werden:

$$
\begin{aligned}
F(x,y)^{+)} &= F(x_o,y_o) + p(F(x_1,y_o) - F(x_o,y_o)) + q(F(x_o,y_1) - F(x_o,y_o))\\
&\quad + p\, q\ (F(x_1,y_1) - F(x_o,y_1) - F(x_1,y_o) + F(x_o,y_o))\\
&\left.\begin{aligned}
&= (1-p)(1-q)F(x_o,y_o) + (1-p)\cdot q\ F(x_o,y_1)\\
&\quad + p\ (1-q)F(x_1,y_o) + p\ q\ F(x_1,y_1
\end{aligned}\right\} \qquad (C1)
\end{aligned}
$$

Analog dazu kann für Funktionen von drei Veränderlichen Formel (26) dargestellt werden:

$$
F(x,y,z)^{+)} \left.\begin{aligned}
&= (1-r)(1-p)(1-q)F(x_o,y_o,z_o) + (1-r)(1-p)\cdot q\ F(x_o,y_1,z_o)\\
&\quad + (1-r)\cdot p\cdot (1-q)\ F(x_1,y_o,z_o) + (1-r)\cdot p\cdot q\ F(x_1,y_1,z_o)\\
&\quad + r\cdot (1-p)(1-q)\ F(x_o,y_o,z_1) + r\ (1-p)\cdot q\ F(x_o,y_1,z_1)\\
&\quad + r\cdot p\cdot (1-q)\ F(x_1,y_o,z_1) + r\cdot p\cdot q\ F(x_1,y_1,z_1)
\end{aligned}\right\} \qquad (C2)
$$

+) Restglied weggelassen

(C1) und (C2) sind im nachfolgenden FORTRAN Unterprogramm implementiert.

Testbeispiel 1.c. zeigt, daß erhebliche Einsparungen an Rechenzeit gegenüber der Standardversion (mit NX = NY = NZ = 2) möglich sind.

```
C
C      ***** UNTERPROGRAMM ZUR ZWEI- UND DREIDIMENSIONALEN
C      ***** INTERPOLATION (LINEAR ENTLANG JEDER ACHSE)
C      ***** ZUM ANSCHLUSS AN DAS PROGRAMM MODINT
C
      SUBROUTINE INTPOL(X,Y,Z,F,ARG,KS,FKT,I1)
C
      COMMON /KONST/ MG,IBUF,NWRIT,INDA,NX,NY,NZ,NFO,NFI,NDA,
     *               NEB,KA,NXL,NXR,NYL,NYR,NZL,NZR,INPG,NASS
     2        /KONSTN/ MZN,MSN,MEN,KPOSN,NFORMN,MODEN,NEU,NFN
      DIMENSION X(INPG),Y(INPG),Z(INPG),F(INPG),ARG(3),FKT(NEU)
C
      F2(P,Q,F1,F2,F3,F4)=
     1   (1-P)*((1-Q)*F1+Q*F2)+P*((1-Q)*F3+Q*F4)
C
      F3(P,Q,R,F1,F2,F3,F4,F5,F6,F7,F8)=
     1   (1-R)*((1-P)*((1-Q)*F1+Q*F2)+P*((1-Q)*F3+Q*F4))+
     2      R  *((1-P)*((1-Q)*F5+Q*F6)+P*((1-Q)*F7+Q*F8))
C
      KS=0
      IF(KA .EQ. 3) GOTO 300
C
C      ***** ZWEIDIMENSIONALE INTERPOLATION (ZWEIFACH LINEAR)
C
      P=(ARG(1)-X(1))/(X(3)-X(1))
      Q=(ARG(2)-Y(1))/(Y(2)-Y(1))
      FKT(I1) =F2(P,Q,F(1),F(2),F(3),F(4))
      I1=I1+1
C
      GOTO 960
C
C      ***** DREIDIMENSIONALE INTERPOLATION (DREIFACH LINEAR)
C
  300 P=(ARG(1)-X(1))/(X(3)-X(1))
      Q=(ARG(2)-Y(1))/(Y(2)-Y(1))
      R=(ARG(3)-Z(1))/(Z(5)-Z(1))
      FKT(I1) =F3(P,Q,R,F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8))
      I1=I1+1
C
  960 RETURN
C
      END
```