

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

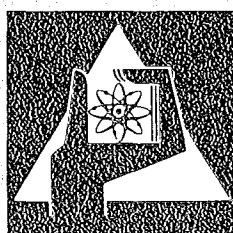
März 1976

KFK 2261

Institut für Reaktorbauelemente
Projekt Nukleare Sicherheit

**Ein FORTRAN-IV-Programm zur Lösung der
Wärmeleitungsgleichung in langen, dünnen Stäben**

W. Hame



**GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.**

KARLSRUHE

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 2261

Institut für Reaktorbauelemente
Projekt Nukleare Sicherheit

Ein FORTRAN-IV-Programm zur Lösung der Wärme-
leitungsgleichung in langen, dünnen Stäben.

W. Hame

Gesellschaft für Kernforschung mbH, Karlsruhe

Zusammenfassung

In der vorliegenden Arbeit wird ein FORTRAN-IV-Programm zur Lösung der Wärmeleitungsgleichung in langen, dünnen Stäben vorgestellt. Die Stäbe können in nahezu beliebiger Weise unterteilt sein in radiale und axiale Schichten. Die einzelnen Zonen können unterschiedliche physikalische Eigenschaften aufweisen und wahlweise beheizt werden. Das Programm wurde entwickelt zur Berechnung der instationären Temperaturverteilung in elektrisch beheizten Brennstabsimulatoren.

Im ersten Abschnitt wird das physikalisch-mathematische Modell erläutert und die für den Anwender nötige Information gegeben. Ferner werden die Eingabedaten für ein Testbeispiel dargestellt und die erhaltenen Ergebnisse interpretiert.

Im zweiten Abschnitt (ab Kapitel 6) werden die einzelnen Programmteile anhand von Strukturdiagrammen erläutert.

Dieses Programm wurde im Rahmen der Untersuchungen zur Wechselwirkung zwischen aufblähenden Zircaloy-Hüllen und einsetzender Kernnotkühlung (Forschungsvorhaben des Projektes Nukleare Sicherheit, GfK, Karlsruhe) erstellt.

A FORTRAN-IV program for the computation of temperature profiles in long rods.

Summary

In this paper a FORTRAN-IV program is presented which solves the equation governing the conduction of heat in a rod. The rod may be subdivided (in a nearly obvious manner) into zones of radial and axial direction with different physical properties. Heat may be generated in any of the zones. The main purpose of this program was to compute the unsteady temperature profile in an electrically heated simulator of a fuel rod.

In the first part the mathematical formulation is derived from a physical model and the most important informations for the use of this program are provided. In addition a program sample is presented with input data and the obtained results are discussed.

In the second part (starting with chapter 6) further and more detailed information is given by flow-charts.

This computer code was developed under the research program of the Projekt Nukleare Sicherheit, GfK, Karlsruhe: Investigation of the Interaction between Ballooning Zircaloy Claddings and Emergency Core Cooling.

Inhaltsverzeichnis

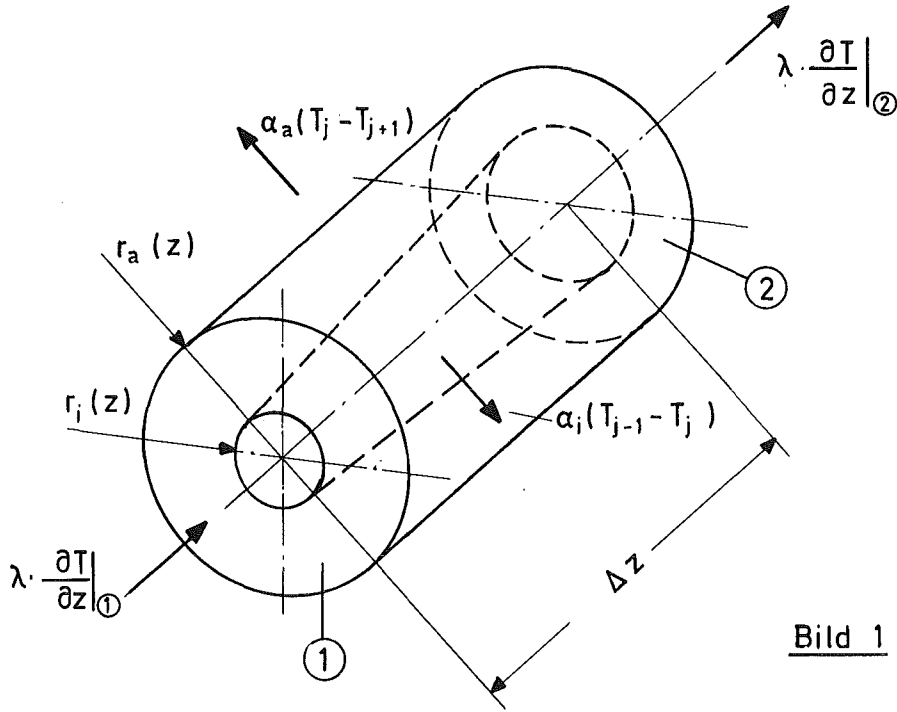
| | | |
|-----|--|----|
| 1. | Einleitung | 1 |
| 2. | Mathematische Formulierung | 2 |
| 3. | Beschreibung des Lösungsverfahrens | 9 |
| 4. | Beschreibung der Eingabedaten | 10 |
| 5. | Testbeispiel | 13 |
| 6. | Beschreibung der einzelnen Programmteile | 15 |
| 6.1 | Einlesen der Daten | 15 |
| 6.2 | Lösung der Differenzgleichung | 20 |
| 6.3 | Stoffwertfunktionen | 22 |
| 6.4 | Berechnung der Wärmeübergangszahlen | 22 |
| 6.5 | Weitere Hilfsprogramme | 25 |
| 7. | Zusammenfassung und Ausbaumöglichkeiten | 26 |
| | Anhang | 28 |
| | Programmliste | |

1. Einleitung

Für die Berechnung der Temperaturverteilung in langen, dünnen Stäben mit mehreren radialen Schichten wird ein Modell angesetzt, das die radiale Wärmeleitung von Schicht zu Schicht nicht durch den FOURIERSchen Ansatz beschreibt, sondern durch eine Wärmeübergangsbeziehung der Form $\alpha \cdot \Delta T$. Dieses Modell setzt voraus, daß die radialen Erstreckungen klein sind gegenüber den axialen Abmessungen, und bietet den Vorteil, daß pro radialer Schicht nur eine Stützstelle erforderlich ist. Die radialen Schichten können dabei beliebig durch Spalte getrennt werden oder direkt aneinanderstoßen.

2. Mathematische Formulierung

Für eine radiale Schicht "j" der Länge Δz , wie sie in Bild 1 dargestellt ist, folgt für die Energiebilanz:



$$\frac{d}{dt} \int \rho \cdot c \cdot T dV = - \int_{F_1} \lambda \frac{\partial T}{\partial z} dF + \int_{F_2} \lambda \frac{\partial T}{\partial z} dF - \int_{F_a} \alpha_a (T_j - T_{j+1}) dF$$

$$+ \int_{F_i} \alpha_i (T_{j-1} - T_j) dF + \int \dot{Q} dV . \quad (1)$$

Unter der Annahme, daß über den Integrationsbereichen konstante Temperaturwerte und damit konstante Stoffwerte vorliegen, ergibt sich hieraus

$$\frac{\partial}{\partial t} (\rho \cdot c \cdot T) = \frac{1}{V_j} \left(\left[\lambda \cdot \frac{\partial T}{\partial z} \right]_2 F_2 - \left[\lambda \cdot \frac{\partial T}{\partial z} \right]_1 F_1 \right)$$

$$+ \alpha_a \cdot (T_{j+1} - T_j) \cdot \frac{F_a}{V_j} + \alpha_i \cdot (T_{j-1} - T_j) \cdot \frac{F_i}{V_j} + \dot{Q} . \quad (2)$$

Für die Flächen F_1 und F_2 gilt nach Bild 1:

$$F_1 = \pi \cdot (r_{a1}^2 - r_{i1}^2), \quad (3.1)$$

$$F_2 = \pi \cdot (r_{a2}^2 - r_{i2}^2), \quad (3.2)$$

während die Flächen F_a und F_i , sowie das Zellvolumen V_j durch

$$F_a = \frac{1}{2} (r_{a2} + r_{a1}) \cdot 2 \pi \Delta z_j, \quad (4.1)$$

$$F_i = \frac{1}{2} (r_{i2} + r_{i1}) \cdot 2 \pi \Delta z_j, \quad (4.2)$$

$$V_j = \frac{\pi}{3} (r_{a2}^2 + r_{a1} \cdot r_{a2} + r_{a1}^2 - (r_{i2}^2 + r_{i1} \cdot r_{i2} + r_{i1}^2)) \cdot \Delta z_j, \quad (4.3)$$

approximiert werden, d.h. es wird zwischen r_{a1} und r_{a2} bzw. r_{i1} und r_{i2} ein linearer Verlauf angenommen.

Diskretisierung der Ableitungen

Es wird ein vollständig implizites Schema gewählt, d.h. die Zeitableitung wird an der Stelle $t + \Delta t$ diskretisiert mit $O(\Delta t)$:

$$\frac{\partial}{\partial t} (c \cdot T) \Big|_{t+\Delta t} \approx \frac{[c \cdot T]_{t+\Delta t} - [c \cdot T]_t}{\Delta t} \quad (5)$$

Die Raumableitungen bzw. der Wärmefluß durch die Zell-trennwände ① und ② wird nach Bild 2 mit $\sigma (\Delta z^2)$ approximiert.

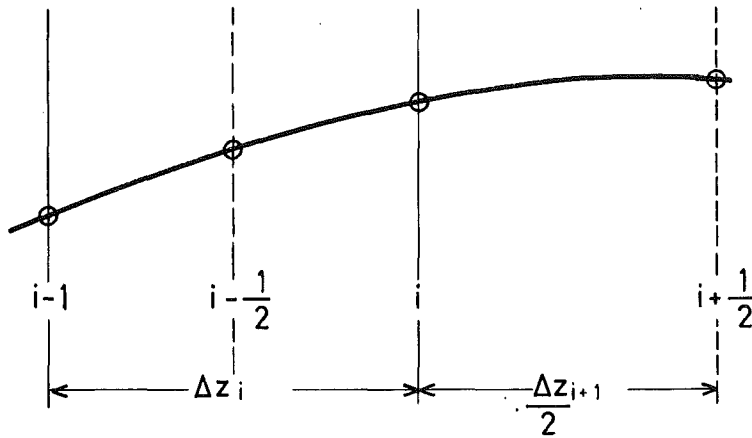


Bild 2

Es gilt

$$\frac{\partial T}{\partial z} \Big|_i = T_{i+\frac{1}{2}} \frac{2 \cdot \Delta z_i}{\Delta z_{i+1} (\Delta z_i + \Delta z_{i+1})} + T_i \frac{2 \cdot (\Delta z_{i+1} - \Delta z_i)}{\Delta z_{i+1} \cdot \Delta z_i} - T_{i-\frac{1}{2}} \frac{2 \cdot \Delta z_{i+1}}{\Delta z_i (\Delta z_{i+1} + \Delta z_i)}, \quad (6.1)$$

$$\frac{\partial T}{\partial z} \Big|_{i-1} = T_{i-\frac{1}{2}} \frac{2 \cdot \Delta z_{i-1}}{\Delta z_i (\Delta z_i + \Delta z_{i-1})} + T_{i-1} \frac{2 \cdot (\Delta z_i - \Delta z_{i-1})}{\Delta z_i \cdot \Delta z_{i-1}} - T_{i-\frac{3}{2}} \frac{2 \cdot \Delta z_i}{\Delta z_{i-1} (\Delta z_i + \Delta z_{i-1})}. \quad (6.2)$$

Für die Diskretisierung der Energiegleichung an der Stelle $i-\frac{1}{2}$ ergibt sich also in der radialen Schicht "j"

$$\begin{aligned} \rho \cdot \frac{[cT]_{j,i-\frac{1}{2}}^{k+1} - [cT]_{j,i-\frac{1}{2}}^k}{r \Delta t} &= \frac{1}{V_{j,i-\frac{1}{2}}} [q_{j,i}^{k+1} F_{ji} - q_{j,i-1}^{k+1} \cdot F_{j,i-1}] \\ -\alpha_{j+1,i-\frac{1}{2}} \cdot \frac{G_{j+1,i-\frac{1}{2}}}{V_{j,i-\frac{1}{2}}} \cdot (T_{j,i-\frac{1}{2}}^{k+1} - T_{j+1,i-\frac{1}{2}}^{k+1}) &+ \alpha_{j,i-\frac{1}{2}} \cdot \frac{G_{j,i-\frac{1}{2}}}{V_{j,i-\frac{1}{2}}} \cdot (T_{j-1,i-\frac{1}{2}}^{k+1} - T_{j,i-\frac{1}{2}}^{k+1}) \\ + \dot{Q}_{j,i-\frac{1}{2}}^{k+1} &. \end{aligned} \quad (7)$$

Die Wärmeflüsse $q_{j,i}^{k+1}$ und $q_{j,i-1}^{k+1}$ sind entsprechend (6.1) und (6.2) nach

$$q_{j,i}^{k+1} = \lambda_{j,i}^{k+1} \cdot \left(\frac{2 \cdot \Delta z_i}{\Delta z_{i+1} (\Delta z_i + \Delta z_{i+1})} T_{j,i+\frac{1}{2}}^{k+1} + \frac{2 \cdot (\Delta z_{i+1} - \Delta z_i)}{\Delta z_{i+1} \cdot \Delta z_i} T_{j,i}^{k+1} - \frac{2 \cdot \Delta z_i}{\Delta z_i (\Delta z_{i+1} + \Delta z_i)} T_{j,i-\frac{1}{2}}^{k+1} \right), \quad (8.1)$$

$$q_{j,i-1}^{k+1} = \lambda_{j,i-1}^{k+1} \cdot \left(\frac{2 \cdot \Delta z}{\Delta z_i (\Delta z_i + \Delta z_{i-1})} T_{j,i-\frac{1}{2}}^{k+1} + \frac{2 \cdot (\Delta z_i - \Delta z_{i-1})}{\Delta z_i \cdot \Delta z_{i-1}} T_{j,i-1}^{k+1} - \frac{2 \cdot \Delta z_i}{\Delta z_{i-1} (\Delta z_i + \Delta z_{i-1})} T_{j,i-\frac{3}{2}}^{k+1} \right). \quad (8.2)$$

zu berechnen. Die G_{jk} entsprechen den F_a und F_i aus (4).

Geordnet nach Gitterpunkterhalten wir also

$$\begin{aligned} \left[\frac{\rho c_{j,i-\frac{1}{2}}^{k+1}}{\Delta t} + \alpha_{j+1,i-\frac{1}{2}}^{k+1} \cdot \frac{G_{j+1,i-\frac{1}{2}}}{V_{j,i-\frac{1}{2}}} + \alpha_{j,i-\frac{1}{2}}^{k+1} \cdot \frac{G_{j,i-\frac{1}{2}}}{V_{j,i-\frac{1}{2}}} \right] T_{j,i-\frac{1}{2}}^{k+1} &= \\ \frac{1}{V_{j,i-\frac{1}{2}}} \xi [q_{j,i}^{k+1} F_{ji} - q_{j,i-1}^{k+1} F_{j,i-1}] & \\ + \alpha_{j+1,i-\frac{1}{2}}^{k+1} \cdot \frac{G_{j+1,i-\frac{1}{2}}}{V_{j,i-\frac{1}{2}}} T_{j+1,i-\frac{1}{2}}^{k+1} + \alpha_{j,i-\frac{1}{2}}^{k+1} \cdot \frac{G_{j,i-\frac{1}{2}}}{V_{j,i-\frac{1}{2}}} T_{j-1,i-\frac{1}{2}}^{k+1} & \\ + \frac{\rho \cdot c_{j,i-\frac{1}{2}}^k}{\Delta t} T_{j,i-\frac{1}{2}}^k + \dot{Q}_{j,i-\frac{1}{2}}^{k+1} &. \end{aligned} \quad (9)$$

Als Randbedingungen an den Stellen $i=1$ bzw. $i=IGES$ kann wahlweise die Temperatur selbst, der Wärmestrom oder eine Linearkombination von Temperatur und Wärmestrom in Form einer Wärmeübergangsbeziehung an ein umgebendes Medium vorgegeben werden. In der allgemeinen Form (und nur in dieser werden sie hier angegeben) lauten die Randbedingungen dann

$$2 \cdot \lambda_{j,i}^{k+1} \cdot \frac{T_{j,\frac{3}{2}}^{k+1} - T_{j,1}^{k+1}}{\Delta z_i} = \alpha_{j,i}^{k+1} \cdot (T_{j,i}^{k+1} - T_{K_{j,i}}^{k+1}) \quad (10.1)$$

$$2 \cdot \lambda_{j,IGES}^{k+1} \cdot \frac{T_{j,IGES}^{k+1} - T_{j,IGES-\frac{1}{2}}^{k+1}}{\Delta z_{IGES}} = \alpha_{j,IGES}^{k+1} \cdot (T_{j,IGES}^{k+1} - T_{K_{j,IGES}}^{k+1}) \quad (10.2)$$

Für den Wärmefluß in die Randzellen gilt entsprechend Gleichung (8)

$$q_{j,i}^{k+1} = 2 \cdot \frac{T_{j,\frac{3}{2}}^{k+1} - T_{j,1}^{k+1}}{\Delta z_i} \cdot \lambda_{j,i} \quad (10.3)$$

$$q_{j,IGES}^{k+1} = 2 \cdot \frac{T_{j,IGES}^{k+1} - T_{j,IGES-\frac{1}{2}}^{k+1}}{\Delta z_{IGES}} \cdot \lambda_{j,IGES} \quad (10.4)$$

Weitere Vereinfachungen

Mit dem Ziel, die Dateneingabe möglichst übersichtlich zu gestalten, wird gefordert, daß innerhalb gewisser axialer Bereiche Δz_l konstant sei, d.h. die Gleichungen (8) für den Wärmefluß vereinfachen sich zu

$$q_{j,i}^{k+1} = \lambda_{j,i}^{k+1} \cdot (T_{j,i+\frac{1}{2}}^{k+1} - T_{j,i-\frac{1}{2}}^{k+1}) \cdot \frac{1}{\Delta z_l} \quad (11.1)$$

$$q_{j,i-1}^{k+1} = \lambda_{j,i-1}^{k+1} \cdot (T_{j,i-\frac{1}{2}}^{k+1} - T_{j,i-\frac{3}{2}}^{k+1}) \cdot \frac{1}{\Delta z_l} \quad (11.2)$$

Die Temperaturen werden nur in den Zellmitten $i = 0,5$ berechnet, d.h. für die Werte in den Zelltrennflächen i bzw. $i-1$ ist außer am Rand eine geeignete Interpolationsformel einzugeben. Dies ist am einfachsten durch den linearen Ansatz

$$T_{j,i} = \frac{1}{2} \cdot (T_{j,i+\frac{1}{2}} + T_{j,i-\frac{1}{2}}) \quad (12)$$

möglich. An Trennstellen axialer Bereiche mit unterschiedlichen Δz_l ist jedoch

$$T = \frac{T_{j,i-\frac{1}{2}} \cdot \Delta z_{l+1} + T_{j,i+\frac{1}{2}} \cdot \Delta z_l}{\Delta z_{l+1} + \Delta z_l} \quad (13)$$

zu setzen.

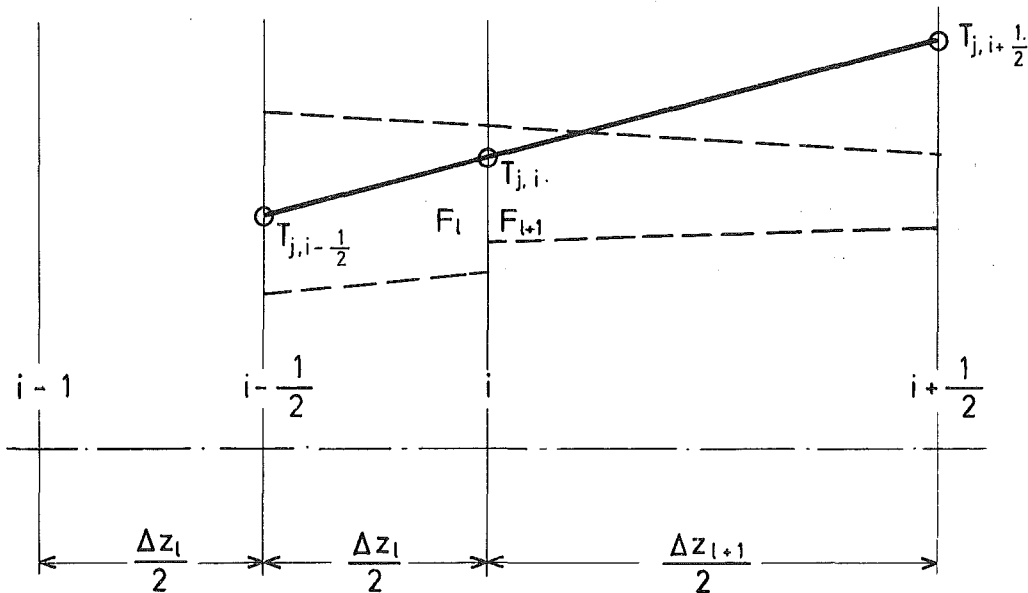


Bild 3

Eine Verbesserung der Genauigkeit kann erzielt werden, wenn neben der Stetigkeit der Temperaturen auch die Stetigkeit der Wärmeflüsse in den Zelltrennflächen gefordert wird. Es gilt nach Bild 3 für den Wärmefluß von links

$$q_{j,i}^{k+1} = 2 \lambda_{j,l}^{k+1} \cdot \frac{T_{j,i} - T_{j,i-\frac{1}{2}}}{\Delta z_l} \cdot F_l \quad (14.1)$$

und für den Wärmefluß nach rechts

$$q_{j,i}^{k+1} = 2 \lambda_{j,l+1}^{k+1} \cdot \frac{T_{j,i+\frac{1}{2}} - T_{j,i}}{\Delta z_{l+1}} \cdot F_{l+1} \quad (14.2)$$

Aus der Gleichheit dieser beiden Werte folgt dann

$$T_{j,i} = \frac{\frac{\lambda_{j,l+1}}{\Delta z_{l+1}} \cdot F_{l+1} \cdot T_{j,i+\frac{1}{2}} + \frac{\lambda_{j,l}}{\Delta z_l} \cdot F_l \cdot T_{j,i-\frac{1}{2}}}{\frac{\lambda_{j,l}^{k+1}}{\Delta z_l} \cdot F_l + \frac{\lambda_{j,l+1}^{k+1}}{\Delta z_{l+1}} \cdot F_{l+1}} \quad (15)$$

In dieser Gleichung ist sowohl eine Änderung der λ -Werte (z.B. anderes Material) als auch ein Sprung in den Flächen zugelassen. Es gilt aber hierbei, daß Endeffekte infolge des Querschnittsprungs vernachlässigbar klein sind. Diese Voraussetzung ist immer dann erfüllt, wenn die Radiendifferenz in Bezug auf die Radien selbst vernachlässigt werden kann.

3. Beschreibung des Lösungsverfahrens

Für eine radiale Schicht ist in Bild 4 die Zuordnung der Maschen- bzw. Zellmitten zu den Indizes des Lösungsfeldes dargestellt. Es sind für IGM2 Maschen also insgesamt

$$\text{IGES} = \text{IGM2} + 2$$

Lösungswerte zu berechnen.

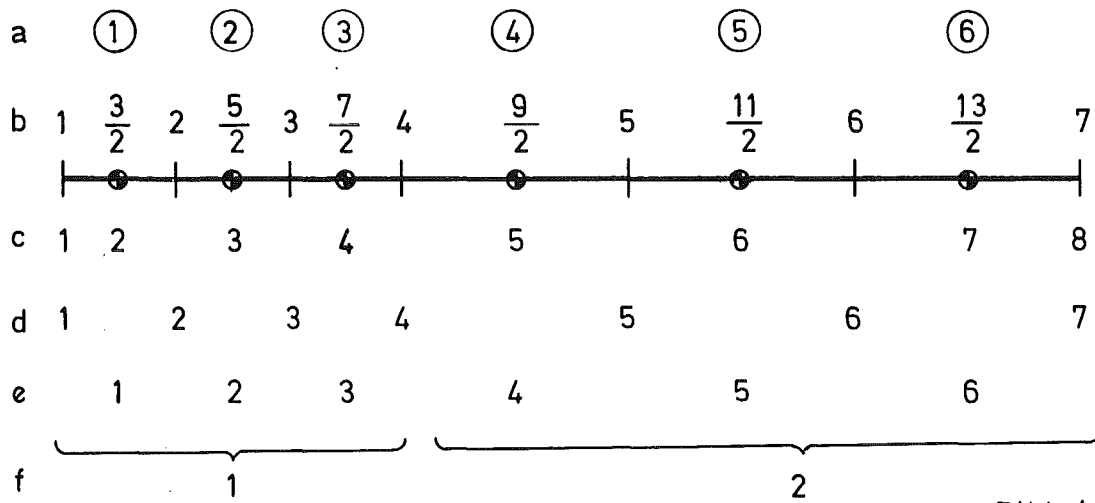


Bild 4

- a) Maschenindex
- b) Stützstelle
- c) Indizierung des Lösungsfeldes
- d) Indizierung der Flächen F
- e) Indizierung der Volumina V und der Flächen G
- f) Indizierung der axialen Schichten

Die Randwerte $T(J,1)$ und $T(J, \text{IGES})$ werden aus den Randbedingungen bestimmt. Danach wird das gesamte Innenfeld berechnet, wobei die Wärmeflüsse an dem linken und rechten Stabende aus den Gleichungen (10) folgen. Zur Lösung der Innengleichungen, die analog zum vollständig dreidimensionalen Problem ein Gleichungssystem mit pentadiagonal auf-

gebauter Koeffizientenmatrix darstellen, kann jedes bekannte Verfahren angewendet werden. Es wird hier ein Blockiterationsverfahren mit sukzessiver Überrelaxation eingesetzt, das neben dem Vorteil des schnellen Durchsatzes der Randbedingungen eine gute Koppelung der Iterationen über das Gleichungssystem und über die lösungsabhängigen Stoffwerte bietet.

4. Beschreibung der Eingabedaten

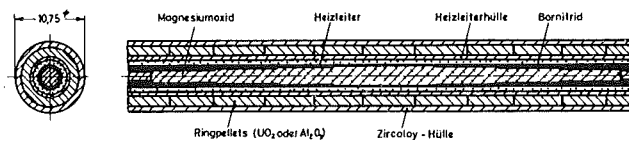
Geometrien -----

Der Lösungsbereich kann in der aktuellen Programmversion in maximal 9¹⁾ radiale Schichten und beliebig viele axiale Schichten eingeteilt werden. Es gilt jedoch die Einschränkung, daß die Zahl der axialen Maschen 53¹⁾ nicht übersteigen darf. Für das hier zu behandelnde Problem (Brennstabsimulator mit stufenförmig abgesetztem Heizleiter) sind die radialen Geometrien über gewissen axialen Bereichen konstant, d.h. für jeweils einen z-Bereich ist immer nur ein Satz Radien anzugeben; die z-Bereiche konstanter Geometrie und die Bereiche konstanter Maschenlängen werden hierbei einander gleichgesetzt.

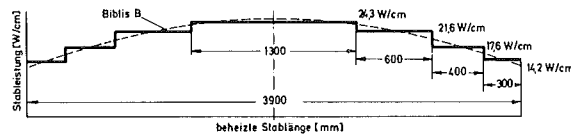
In Bild 5 ist die Geometrie des Testbeispiels dargestellt. Es ist zu beachten, daß Radien auch dann, wenn sie sich nicht ändern, von z-Bereich zu z-Bereich neu zu definieren sind. Wie bereits erwähnt, können die einzelnen radialen Schichten direkt aneinander angeschlossen werden, der Außenradius der inneren Schicht ist dann gleich dem Innenradius der äußeren Schicht, oder durch Spalte getrennt sein,

1) Diese Grenzwerte können natürlich jederzeit durch Änderung der Dimensionierungsvereinbarungen erweitert werden.

wobei die Spaltdicke durch die Radiendifferenz gegeben ist. Über den Spalt hinweg wird eine Wärmedurchgangszahl definiert, die auch den Einfluß der Strahlung miteinfaßt. Diese Zahl hängt natürlich entscheidend vom Medium im Spalt ab (in unserem Falle ist es gasförmiges Helium) und ist nur dann sinnvoll, wenn das Speichervermögen dieses Stoffes gering ist gegenüber dem Speichervermögen der Materialien in den angrenzenden radialen Schichten.



schematischer Aufbau



axiale Leistungsverteilung

| | | |
|-------------|--------------------|-------------------|
| GfK 1975 | Brennstabsimulator | IRB PNS - 4238 |
|-------------|--------------------|-------------------|

Bild 5 ^{+))}

Für die Geometriedaten und auch für die physikalischen Stoffgrößen gilt, daß sie grundsätzlich in beliebigen, kohärenten Einheiten angegeben werden können.

Werden jedoch die eingebauten Stoffwertfunktionen verwendet, so sind alle Längen in cm, die Zeit in sec, Wärmemengen in cal und Temperaturen in ^oK einzusetzen.

^{+))} Die dargestellte Geometrie entspricht der des Brennstabsimulators, wie er für die Untersuchungen im Rahmen des Forschungsvorhabens 4238 des Projektes Nukleare Sicherheit GfK, Karlsruhe entwickelt wurde. Eine eingehende Beschreibung dieses Brennstabstimulators wird im PNS-Jahresbericht, 1975, GfK, Karlsruhe gegeben.

In Bild 6 sind die Eingabedaten für das hier gerechnete Testbeispiel dargestellt. Die Reihenfolge der einzelnen Datenblöcke ist beliebig, lediglich die ersten beiden Datensätze (es sind dies die Verfahrenskonstanten) müssen stets an erster Stelle stehen und die Geometriedaten sollten bei Initialisierung vor allen anderen Datenblöcken eingelesen werden. Eine detaillierte Beschreibung des Einlesevorganges mit Erläuterungen zu den einzelnen Variablen wird in Abschnitt 6 mit der Beschreibung der einzelnen Programmteile gegeben.

```

00010 0.0 1.20 0.0010      2      5 3 3      2 0.8 .0637 .005 1.0
00020      0      0      8 100
00030      1
00040 10.0      0.01      10
00050      2
00060      6      6
00070 0.0      0.1487 .1487 0.175 .175      0.24 .24 0.3
00080 .305      0.46 .465 0.5375
00090 0.0      0.1487 0.1487 0.175 0.175      0.24 0.24 0.3
00100 0.305      0.46 0.465 0.5375
00110 0.0      0.1487 .1487 0.175 .175      0.24 .24 0.3
00120 .305      0.46 .465 0.5375
00130 0.0      0.1450 .1450 0.175 .175      0.24 .24 0.3
00140 .305      0.46 .465 0.5375
00150 0.0      0.1372 .1372 0.175 .175      0.24 .24 0.3
00160 .305      0.46 .465 0.5375
00170 0.0      0.1265 .1265 0.175 .175      0.24 .24 0.3
00180 .305      0.46 .465 0.5375
00190 0.0      2.5 5.0 65.0 125.0 105.0 195.0
00200      6      6 12 12 6 6
00210      3
00220      4 4 4 4 4 4
00230      8 8 8 8 8 8
00240      6 6 6 6 6 6
00250      8 8 8 8 8 8
00260      1 1 1 1 1 1
00270      3 3 3 3 3 3
00280      4
00290 0.0      0.0 0.0 0.0 0.0 0.0
00300 1000. 1000. 1000. 1000. 1000. 1000.
00310 1000. 1000. 1000. 1000. 1000. 1000.
00320 1000. 1000. 1000. 1000. 1000. 1000.
00330 1000. 1000. 1000. 1000. 1000. 1000.
00340 1000. 1000. 1000. 1000. 1000. 1000.
00350 0.0 0.0 0.0 0.0 0.0 0.0
00360      5
00370 0.0 0.0 0.0 0.0 0.0 0.0
00380 0.0 0.0 0.0 0.0 0.0 0.0
00390      6
00400 0.0 0.0
00410 0.0 0.0
00420 0.0 0.0
00430 0.0 0.0
00440 0.0 0.0
00450 0.0 0.0
00460      7
00470 573.2 573.2
00480 573.2 573.2
00490 573.2 573.2
00500 573.2 573.2
00510 573.2 573.2
00520 573.2 573.2
00530      8
00540 0.0 0.0 0.0 0.0 0.0 0.0
00550 217.14 217.14 217.14 171.64 113.21 75.81
00560 0.0 0.0 0.0 0.0 0.0 0.0
00570 0.0 0.0 0.0 0.0 0.0 0.0
00580 0.0 0.0 0.0 0.0 0.0 0.0
00590 0.0 0.0 0.0 0.0 0.0 0.0
00600      9
00610 573.2
00620      10
00630      1
00640 25.0 0.01 19
00650      4
00660 0.0 0.0 0.0 0.0 0.0
00670 1000. 1000. 1000. 1000. 1000. 1000.
00680 1000. 1000. 1000. 1000. 1000. 1000.
00690 1000. 1000. 1000. 1000. 1000. 1000.
00700 1000. 1000. 1000. 1000. 1000. 1000.
00710 0.0 1000. 1000. 1000. 1000. 1000.
00720 0.0 0.0 0.0 0.0 0.0
00730      10
00740      0

```

Bild 6

5. Testbeispiel

Für die in Bild 5 dargestellte Anordnung wurde mit den Eingabedaten nach Bild 6 ein Testlauf durchgeführt, dessen wesentliche Ergebnisse im folgenden zusammengestellt sind.

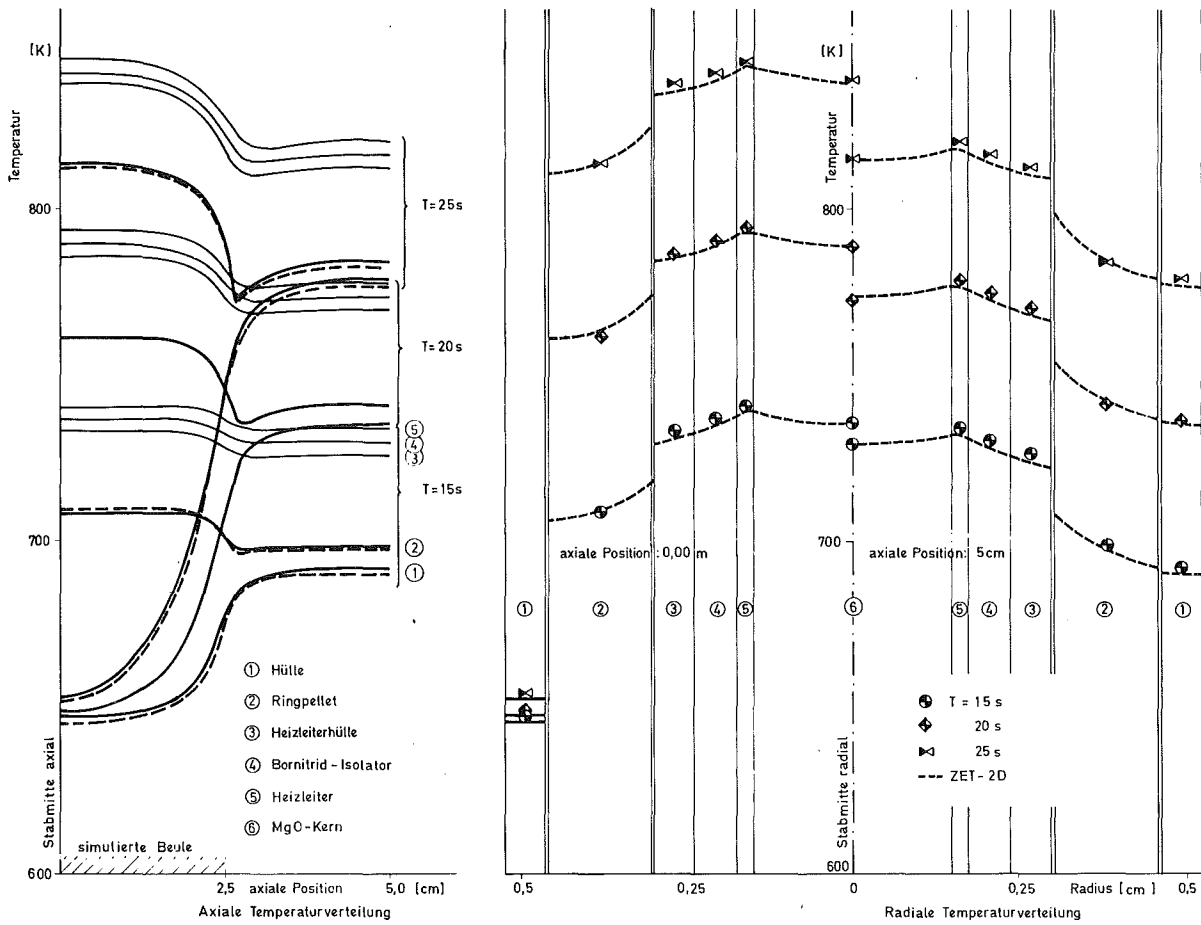


Bild 6.1

Das Bild 6.1 stellt links die axialen Temperaturprofile in den einzelnen Schichten eines Brennstabsimulators dar, bei dem nach einer ungestörten Aufheizphase von 10 sec die Ausbildung einer 5 cm (2.5 cm ab Mitte) langen, symmetrischen Beule simuliert wurde. Dies erfolgte durch Nullsetzen des Wärmeübergangskoeffizienten vom Ringpellet 2 zur Hülle 1. Man erkennt deutlich, wie sich über den Spalt mit zunehmender Zeit in Stabmitte ein immer größer werdendes Temperaturgefälle aufbaut. Der Hülle fließt an dieser Stelle Wärme nur noch in axialer Richtung zu. Ebenso deutlich ist zu erkennen, daß der Einfluß der Störung in axialer Richtung sehr schnell abklingt. Schon ab der Position 5 cm ist wieder ein ausgeglichenes, d.h. ungestörtes Temperaturprofil vorhanden.

In der rechten Hälfte des Bildes sind für zwei axiale Positionen (maximal gestört bei 0 cm und ungestört bei 5 cm) die radialen Temperaturverteilungen aufgetragen.

Die eingezeichneten Punkte stellen die mit dem vorliegenden Programm errechneten Mitteltemperaturen dar, die gestrichelten Linien - hier sowohl als auch in der linken Bildhälfte - die Ergebnisse einer Vergleichsrechnung, die vom IKE, Stuttgart ²⁾ mit einem zweidimensionalen Differenzenverfahren (Programm ZET-2D) erzielt wurden.

Die Differenz der beiden Rechnungen liegt im Mittel bei etwa 2 grad, d.h. wenn man eine Bezugstemperatur von 400 °C zugrundelegt, bei ungefähr 0,5 %. Sie ist von der selben Größenordnung wie die vorgegebene Genauigkeitsschranke von 0,1 % bezogen auf 673 °K und damit vernachlässigbar.

Fazit: Der Vergleich mit den Rechnungen von IKE, Stuttgart und die physikalische Relevanz der Ergebnisse dieses und anderer Testläufe belegen die Richtigkeit des Programmes.

2) Institut für Kernenergetik der Universität Stuttgart.
Eine Beschreibung dieses Programmes liegt zur Zeit noch nicht vor.

6. Beschreibung der einzelnen Programmteile

6.1 Einlesen der Daten

In Bild 7 ist das Strukturdiagramm für das Einlesen der Daten dargestellt. Die Eingabeparameter sind in Gruppen eingeteilt, jeder Gruppe ist eine bestimmte Nummer zugeordnet. Abweichend hiervon sind die Verfahrenskonstanten, die bei jeder Initialisierung des Programms neu definiert werden müssen ohne Ordnungszahl. Diese Konstanten werden auf den stets ersten beiden Datenkarten angegeben und haben im einzelnen die folgende Bedeutung:

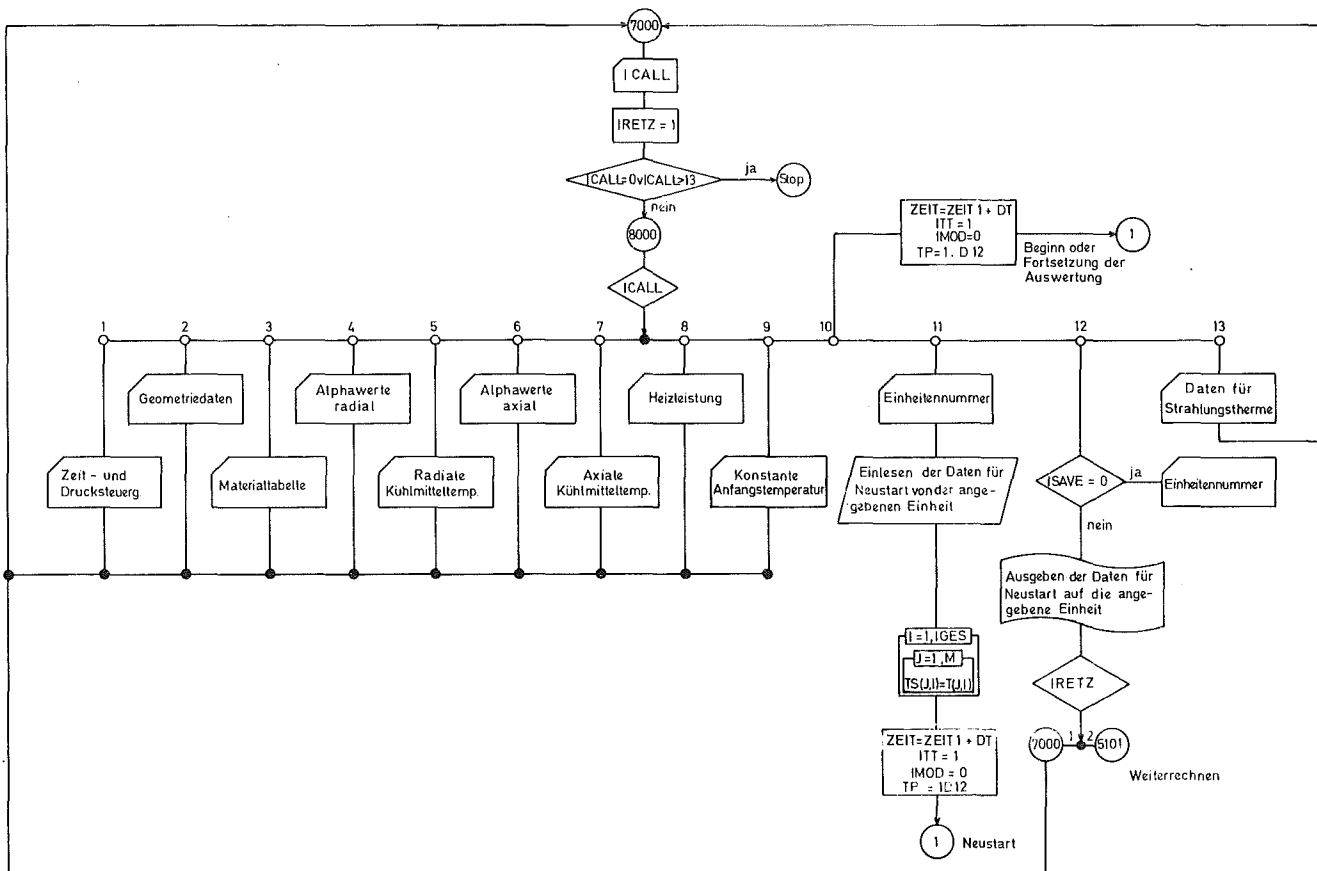


Bild 7

1. Datenkarte

| | |
|-------------------------|---|
| ZEIT 1 (real) | Startzeit (Problemzeit); |
| OMEGA (real) | Relaxationsfaktor (im allgemeinen ~ 1.5); |
| EPS (real) | Genauigkeitsschranke für Iteration und Zeitschritt- weitensteuerung; |
| JP, KP (integer) | Feldindizes eines Kontroll- punktes, für den die Ge- nauigkeitsabfrage durchgeführt werden soll; |
| NBC1, NBC2 (integer) | Steuervariablen für die Auswahl der Randbedingungen an den Stabenden 1: isotherm 2: Wärmefluß wird vorgegeben 3: Wärmeübergang an ein um- gebendes Medium; |
| V, EXP, DK (real) | Verdichtungsverhältnis, Exponent und Korngrößendurchmesser für MgO; |
| VEN (real) | Verdichtungsverhältnis für BN; |

2. Datenkarte

| | |
|-----------------|--|
| ICTRL (integer) | Druckersteuervariable: ICTRL= 0, nur die Temperaturfelder nach jedem durch IDR spezifizierten Zeitschritt werden ausgegeben. Für jeden anderen Wert von ICTRL werden auch Zwischenwerte ausgedruckt; |
|-----------------|--|

| | |
|-----------------|---|
| ISAVE (integer) | ISAVE=0, kein Aufbewahren des Lösungsfeldes nach jedem Zeitschritt. Für jeden anderen Wert von ISAVE werden die Daten für Neustart über die mit |
| IUNIT (integer) | spezifizierte Einheit ausgegeben. IUNIT sollte nicht auf 5 oder 6 gesetzt werden; |
| ITS (integer) | gibt an, nach welchen Iterationsschritten die Stoffdaten neu berechnet werden sollen. Es gilt, wenn ITR die Nummer des laufenden Iterationsschrittes ist, daß Neuberechnung für $\text{MOD}(\text{ITR}, \text{ITS}) = 0$ erfolgt. Wird z.B. ITS = 100 oder größer angegeben, so erfolgt keine Iteration über die Stoffwerte. |

Den beiden Datenkarten sind die Formate

2 F15, F10.2, 2I5, I3, I2, 4 F5.2

und

4I5

zugeordnet. Datenfelder, die nicht benötigt werden, können freibleiben. Die zugeordneten Variablen sind dann mit Null belegt.

Die Datenblöcke für Geometrie und physikalische Charakteristika werden mit der Variablen ICALL über ein computed GO TO angesteuert. Es gilt generell für Real-Variable das Format F8.2 und für Integer-Variable I5. Es bedeuten im einzelnen

ICALL = 0 : Ende der Auswertung;

ICALL = 1: ZEITN (real) Maximale Prozeßzeit bis zu der die Rechnung durchgeführt werden soll. Am Ende der Auswertung wird wieder ICALL verlangt;

DT (real) Anfangszeitritt, von dem ausgehend die Zeitschrittweitensteuerung arbeiten soll;

IDR (integer) es wird für ICTRL = 0 nur nach jedem IDR-ten Zeitschritt ausgedruckt.

ICALL = 2: M (integer) Anzahl der Schichten in r-Richtung

N (integer) Anzahl der Schichten in z-Richtung

R(2,M,K) (real) Innen- (Index 1) bzw. Außen - (Index 2) radius der M radialen Schichten für N axiale Schichten
K = 1 (1) N
Insgesamt sind hier also N Sätze zu $2 \cdot M$ Werten anzugeben.

L(N+1) (real) Axiale Einteilung; für N Abschnitte sind gemäß Bild 4 N+1 Längen anzugeben.

IM(N) (integer) Anzahl der äquidistanten Maschen innerhalb der jeweiligen axialen Schicht.

ICALL = 3: IMAT(J,N) Materialindex für $M \cdot N$
(integer) Positionen. Es gilt die Zuordnung
J= 1(1) M

| | |
|---|--------------------------------|
| 1 | UO ₂ |
| 2 | He |
| 3 | Zr - 4 |
| 4 | MgO |
| 5 | Vacromium |
| 6 | BN |
| 7 | AL ₂ O ₃ |
| 8 | Inconel 600 |

Weitere Materialien können beliebig hinzugenommen werden.

- ICALL = 4: ALRR(J,N)
(real)
J = 1(1) M+1
- Wärmeübergangszahlen zwischen den radialen Schichten bzw. am Innen-(J=1) und am Außen-(J=M+1) radius.
Wird ein Wert größer 999. angegeben, so wird α intern berechnet.
Es sind hier für M radiale Schichten M+1 Sätze zu je N Werten erforderlich.
- ICALL = 5: TKR(J,N) (real)
J = 1(1)2
- Kühlmitteltemperaturen am Innen- bzw. Außenradius für jeden axialen Abschnitt. Innerhalb der Abschnitte wird die Kühlmitteltemperatur als konstant angesehen.
- ICALL = 6: ALZZ(J,2) (real)
J = 1(1) M
- Wärmeübergangszahl an den Stabenden je radialer Schicht. Für NBC1 bzw. NBC2 gleich 2 wird mit ALZZ der zugeführte Wärmefluß angegeben. Für isotherme Enden ist ALZZ bedeutungslos.
- ICALL = 7: TKZ(J,2) (real)
J = 1(1) M
- Kühlmitteltemperaturen an den Stabenden je radialer Schicht. Für NBC1 oder NBC2 gleich 1 wird mit TKZ die Stabendtemperatur angegeben. Für NBC1 oder NBC2 gleich 2 ist TKZ bedeutungslos.

| | | |
|------------|-------------------------------|--|
| ICALL = 8: | QZ (J,N) (real) J = 1(1) M | Heizleistung für die M · N Positionen. Wird in einzelnen Positionen keine Heizleistung gewünscht, so ist 0.0 anzugeben. |
| ICALL = 9: | TKST (real) | Konstante Anfangstemperatur. |
| ICALL =10: | keine Eingabe | Beginn der Auswertung. |
| ICALL =11: | IUNIT (integer) | Einheitennummer für das Ein- lesen der Daten bei Neustart. |
| ICALL =12: | | Ausgabe der Daten über IUNIT, wenn ISAVE ≠ 0 angegeben war, sonst |
| | IUNIT (integer) | Einheitennummer zum Einlesen und Ausgeben der Daten für Neustart. |
| ICALL =13: | ISP (integer) | Materialnummer für Medium im Spalt |
| | CS (real) | Strahlungskonstante |
| | CEPS (real) | Strahlungsbeiwert ϵ |

Es brauchen nur die Datenblöcke angegeben zu werden, die auch wirklich erforderlich sind. Innerhalb der einzelnen Blöcke müssen jedoch stets alle Werte angegeben werden. Leerstellen bedeuten, daß die zugehörige Variable mit Null belegt wird.

6.2 Lösung der Differenzgleichung

In Bild 8 ist die Lösung der Differenzgleichung dargestellt. Dieser Programmteil ist im wesentlichen in zwei Abschnitte gegliedert:

1. Setzen der Koeffizientenmatrix
2. Auflösen des linearisierten Gleichungssystems.

Im allgemeinen Fall können die Stoffdaten von der Lösung selbst abhängen, d.h. es ist sowohl eine Iteration über das Gleichungssystem erforderlich, als auch eine Iteration über die Stoffdaten.

Beide Iterationen sind ineinander verschachtelt, jedoch werden aus Zeitgründen nicht nach jedem Iterationsschritt die Stoffdaten neu berechnet, sondern nur für durch ITS spezifizierte Schritte. So bedeutet z.B. ITS = 6, daß für die Iterationsschritte

0, 5, 8, 9, 10, 11, 12 die Stoffwerte neu berechnet werden. Sollen die Stoffwerte nur am Anfang jedes Zeitschrittes mit den alten Temperaturwerten berechnet werden, so ist ITS sehr groß (z.B. 100) anzugeben.

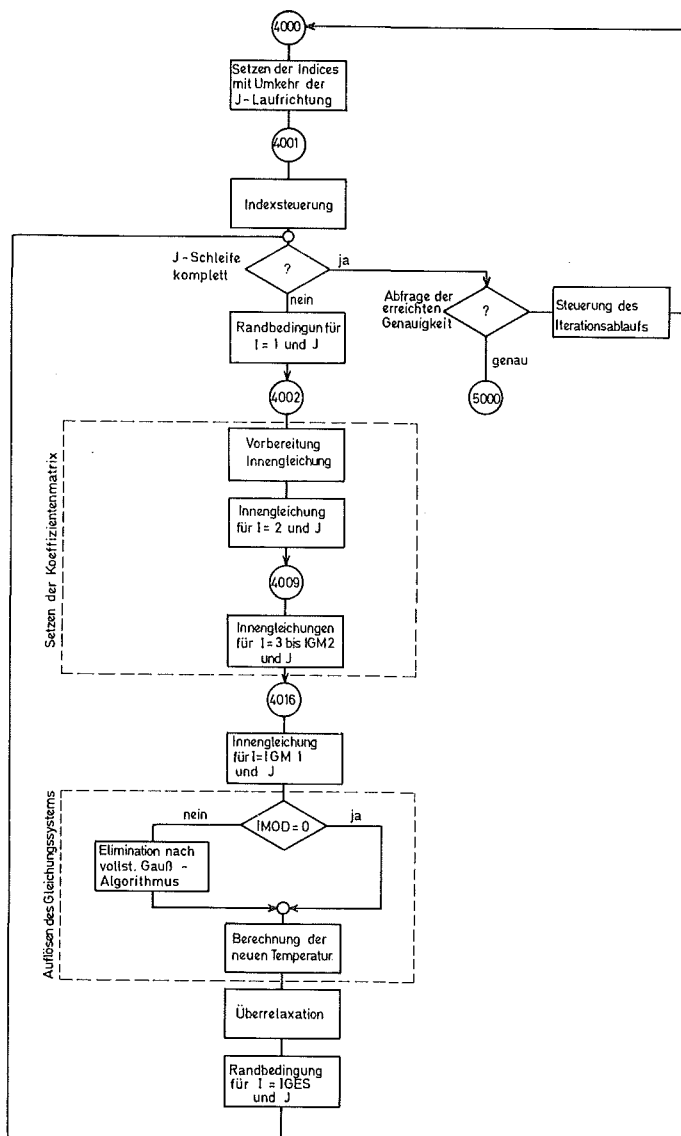


Bild 8

6.3 Stoffwertfunktionen

Die Stoffwertfunktionen für λ, ρ, c_p sind grundsätzlich so aufgebaut, daß nach Maßgabe der Materialtabelle die Stoffdaten über ein computed GO TO ausgewählt werden. Eine Erweiterung des Umfanges ist in beliebiger Weise möglich. Die Bilder 9.1 bis 9.3 zeigen die einzelnen Strukturdiagramme für die aktuell verwendeten Unterprogramme.

6.4 Berechnung der Wärmeübergangszahlen

Die Wärmeübergangszahlen α können wie in Abschnitt 6.1 angegeben explizit über die Eingabedaten gesetzt werden oder bei $\alpha > 999$. intern aus der folgenden Wärmeübergangsbeziehung berechnet werden ³⁾:

$$\alpha = \frac{1}{\frac{r_5}{r_2} \cdot \frac{r_2 - r_1}{\lambda_{j-1,i}} + \frac{r_5}{\lambda_{sp} + \frac{r_2 \cdot c_s^* \cdot (T_{j-1,i}^2 + T_{j,i}^2) \cdot (T_{j-1,i} + T_{j,i})}{\ln \frac{r_3}{r_2} + \frac{1}{\epsilon_{j-1}} + \frac{r_2}{r_3} \cdot \left(\frac{1}{\epsilon_j} - 1 \right)}} + \frac{r_4 - r_3}{\lambda_{j,i}} \cdot \frac{r_5}{r_3}} \quad (16)$$

$$c_s^* = 10^{-8} \cdot c_s$$

³⁾ Diese Gleichung wurde für den hier behandelten Fall eines Brennstabsimulators von Herrn Neitzel, IRB angegeben.

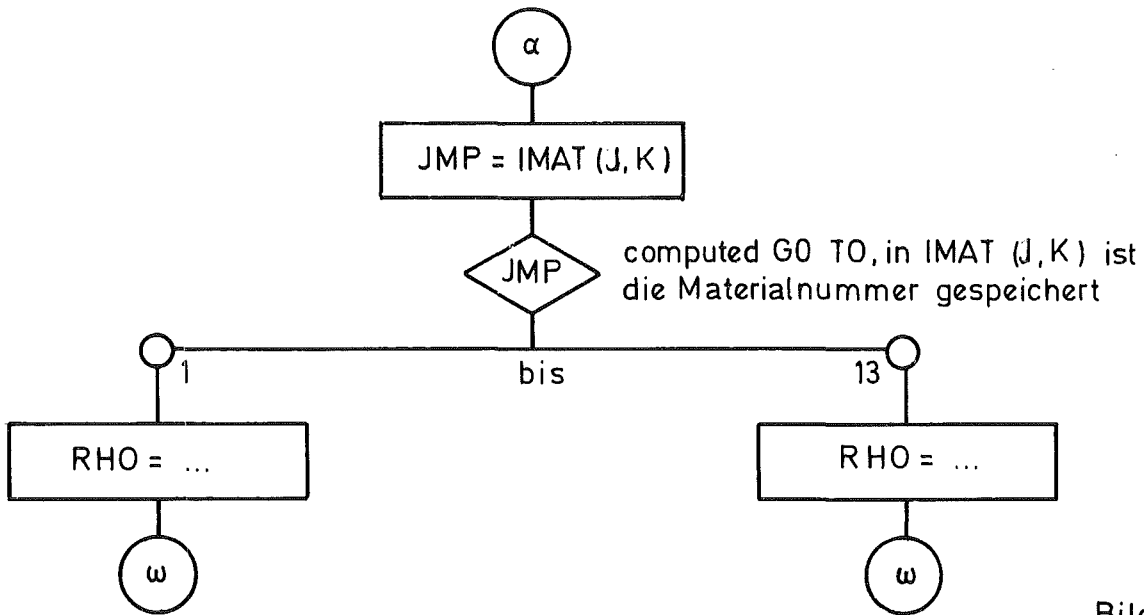


Bild 9.1

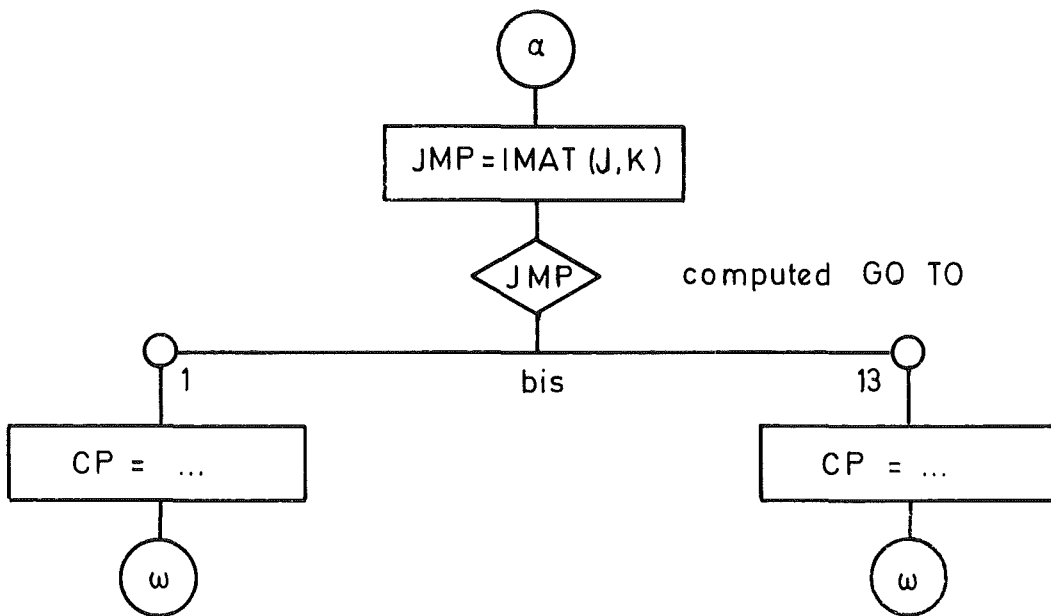


Bild 9.2

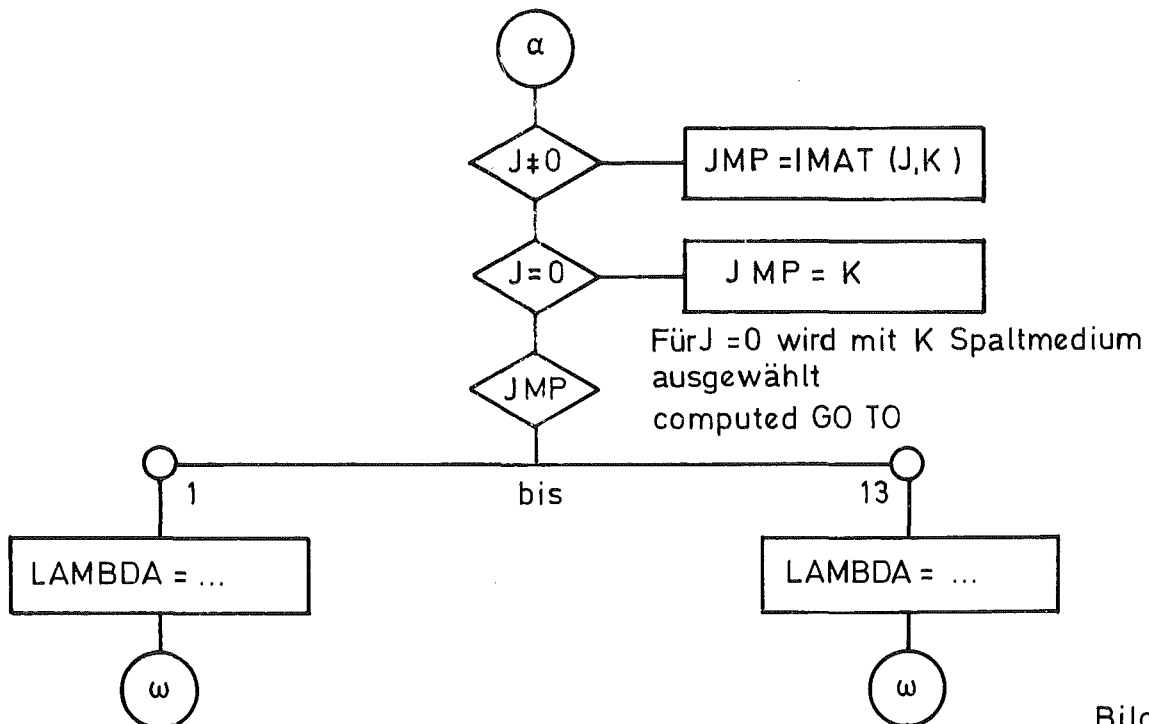


Bild 9.3

Die Radien r_1 bis r_4 sind in Bild 10 angegeben. r_5 wird wahlweise je nachdem ob α für den Übergang innen oder außen gilt, auf den Innenradius oder Außenradius der betreffenden Schicht gesetzt.

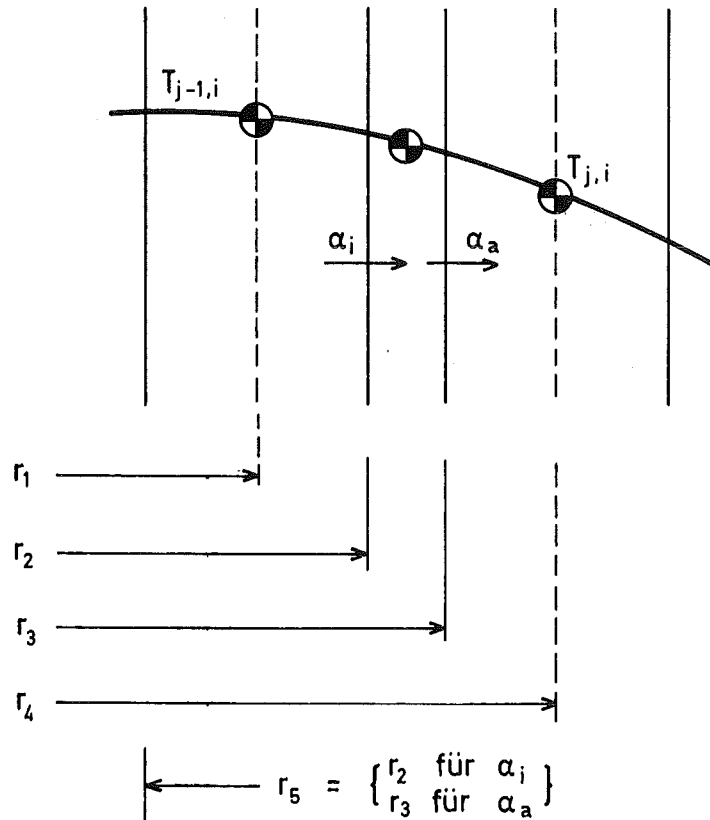


Bild 10

In Bild 11 ist das Strukturdiagramm des aktuell verwendeten Unterprogramms dargestellt. Es ist zu beachten, daß wegen Division durch die Strahlungsbeiwerte diese nicht Null sein dürfen. Soll der Strahlungseinfluß nicht berücksichtigt werden, so ist die Stefan-Boltzmann-Konstante c_s Null zu setzen.

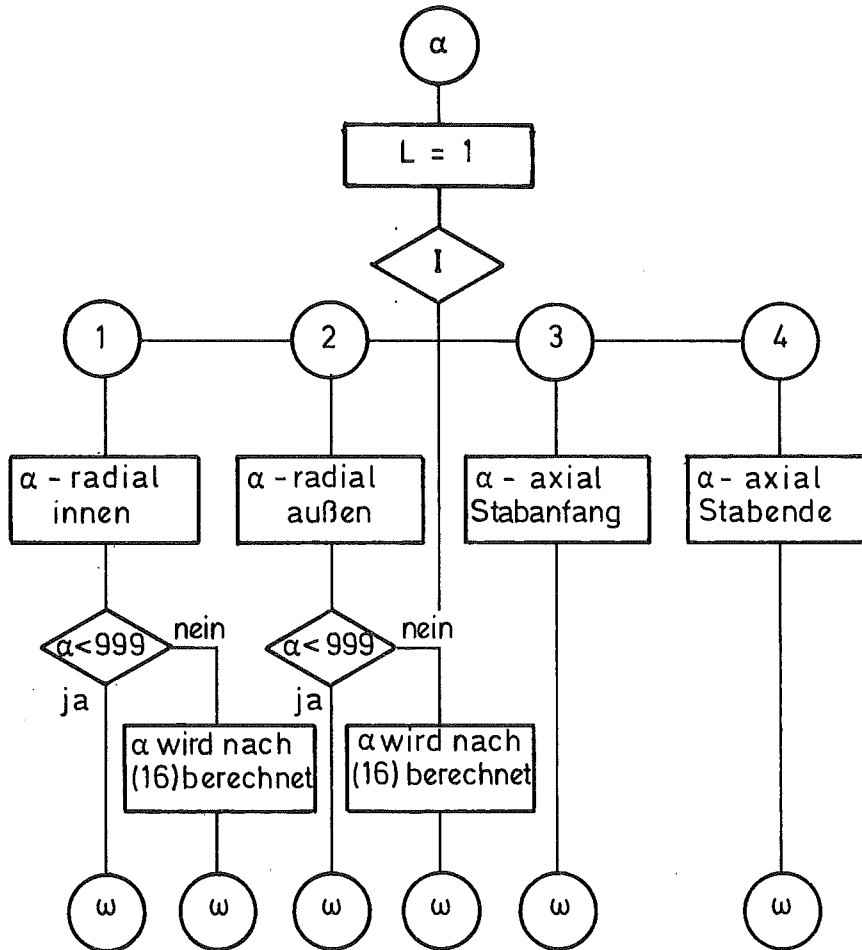


Bild 11

6.5 Weitere Hilfsprogramme

Die Unterprogramme TKUE und Q, die hier nur aus Gründen der Verallgemeinerungsfähigkeit als geschlossene Unterprogramme formuliert wurden, dienen nur dazu, die eingelesenen Werte bereitzustellen. Eine gesonderte Beschreibung kann der Einfachheit halber entfallen.

7. Zusammenfassung und Ausbaumöglichkeiten

Die Gliederung der Gesamtstruktur ist in Bild 12 dargestellt. Dieses Diagramm zeigt das Zusammenwirken der einzelnen Programmteile und gibt einen Überblick über die Möglichkeiten bzw. das Vorgehen bei der Steuerung eines Gesamtablaufes. Eine Erweiterung des Programms in Bezug auf Vergrößerung der maximalen Stützstellenanzahl bzw. der Anzahl der radialen Schichten ist in einfacher Weise durch eine Änderung der Dimensionierungsvereinbarungen möglich. Einschneidendere Änderungen, wie z.B. von z abhängige Heizleistung sind nur durch Änderung der zuständigen Programme möglich.

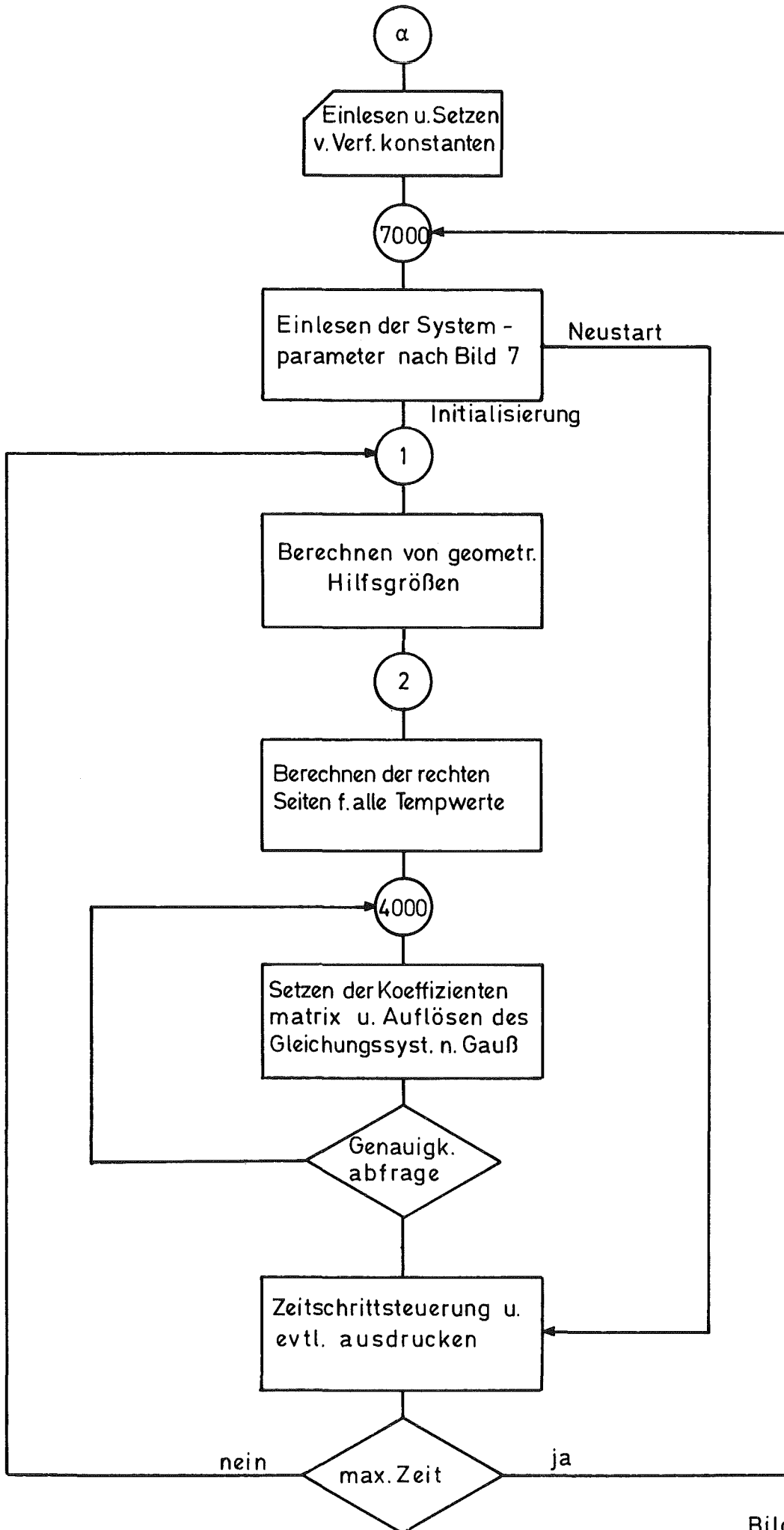


Bild 12

Anhang

Programmliste Stand Okt. 1975

Mit dem aufgelisteten Programm wurde
das in Abschnitt 5 angegebene Test-
beispiel gerechnet.

```

C *****
C PROGRAMM ZUR LOESUNG DER WAERMELEITUNGSGLEICHUNG
C *****
C
C      REAL R(2,9,7),L(7),TKR(2,7),TKZ(9,2),F1(9,53),F2(9,53),G1(9,53),
1      G2(9,53),ALRR(10,7),QZ(9,7),ALZZ(9,2),RR11(9,53),RR12(9,53),
2      RR21(9,53),RR22(9,53),FQ(9,53)
C      REAL*8 DT,DT02,DZ(7),TMIN,EQ(9,53),
1      T(9,55),TKUE,TU1,TU2,P11S,R12S,R21S,R22S,
2      TX,RHO,CPX,CP,ALPHA,LAMBDA,DZZ1,DZZ2,DZZ3,
5      TS(9,55),Q,R11,R12,R21,R22,
6      RB1,RB2,TP,FX1,FX2,S1(3),EPS
C      DIMENSION KK(53),DX(53),RX(2,9,54)
C      REAL*8 ALIX(9,53),ALAX(9,53),VJI,PCP(9,53),
1      RCPA(9,53),RM(2,9,53),
2      DKM1(53),DKM2(9,53),DKM3(9,53),DKM4(53),QW1(9,53),QW2(9,53),
3      RR1,RR2,RL1,RL2,TP1,TP1
C      INTEGER IA(7),IM(7),IE(7),IMAT(9,7)
C      COMMON /BLOCK1/ M,N,IMAT
C      COMMON /BLOCK2/ QZ,FQ
C      COMMON /BLOCK3/ ALRR,ALZZ,ISP,CS,DEPS(9)
C      COMMON /BLOCK4/ TKR,TKZ
C      COMMON /BLOCK5/ PM
C      COMMON /BLOCK6/ V,FXP,DK,VCN,VM1,VEXP,VEM1
C      EQUIVALENCE (TX1,S1(1)),(TX,S1(2)),(TX2,S1(3)),(RX,ALIX)
C *****
C      READ(1,7015) ZEIT1,OMEGA,EPS,JP,KP,NBC1,NBC2,ISP,V,EXP,DK,VCN
C      READ(1,7011) ICTRL,ISAVE,IUNIT,ITSO
C      IMOD = 0
C      ZEITO = ZEIT1
C      ZFIT = ZEITO
C      OMEGA1 = 1. - OMEGA
C      TP = 1.D12
C      VM1 = 1.-V
C      VEXP = V**EXP
C      VEM1 = 1.-VEXP
C      ITR = 0
C
C
C -----
C      IJKL = 1
C      ITM = 1
C -----
C
C      GO TO 7000
C
C *****
C PROGRAMMTEIL LEVEL 0
C *****
1      IGES = 2
C      IA(1) = 2
C      DO 2100 K = 1,N
C      DZ(K) = (L(K+1)-L(K))/IM(K)
C      IE(K) = IA(K) + IM(K) - 1
C      IGES = IGES + IM(K)
C      IF (K.EQ.N) GO TO 2100
C      IA(K+1) = IE(K) + 1

```

```

2100 CONTINUE
    IGM1 = IGES - 1
    IGM2 = IGES - 2
    IGM3 = IGES - 3
1001 K = 1
    DX(1) = DZ(1)*.500
    KK(1) = 1
    DO 1002 J = 1,M
        R11 = R(1,J,1)
        R21 = R(2,J,1)
        FQ(J,1) = R21*R21 - R11*R11
        RX(1,J,1) = R11
1002    RX(2,J,1) = R21
    DO 1005 I = 2,IGM2
        IM1 = I - 1
        DO 1003 J = 1,M
            FQ(J,I) = FQ(J,IM1)
            RX(1,J,I) = R(1,J,K)
1003    RX(2,J,I) = R(2,J,K)
        IF (I.NE.IF(K)) GO TO 1105
        KM1 = K
        K = K + 1
        DZZ1 = DZ(KM1)
        DZZ2 = DZ(K)
        DZZ3 = DZZ1 + DZZ2
        DO 1004 J = 1,M
            R11 = R(1,J,K)
            R21 = R(2,J,K)
            FQ(J,I) = R21*R21 - R11*R11
            RX(1,J,I) = (DZZ2*RX(1,J,I)+DZZ1*R11)/DZZ3
1004    RX(2,J,I) = (DZZ2*RX(2,J,I)+DZZ1*R21)/DZZ3
1105    KK(I) = K
        DX(I) = DZ(K)*.500
1005 CONTINUE
    DO 1006 J = 1,M
        RX(1,J,IGM1) = R(1,J,N)
1006    RX(2,J,IGM1) = R(2,J,N)
    DO 1008 I = 1,IGM2
        IP1 = I + 1
        DO 1007 J = 1,M
            R11 = RX(1,J,I)
            R21 = RX(2,J,I)
            R12 = RX(1,J,IP1)
            R22 = RX(2,J,IP1)
            RM(1,J,I) = (R11+R12)*.500
            RM(2,J,I) = (R21+R22)*.500
            R11S = R11 * R11
            R21S = R21 * R21
            R12S = R12 * R12
            R22S = R22 * R22
            VJI = (R21S+R21*R22+R22S-(R11S+R11*R12+R12S))/3.
            G1(J,I) = 2.00*RM(1,J,I)/VJI
            G2(J,I) = 2.00*RM(2,J,I)/VJI
            FQ(J,I) = VJI/FQ(J,I)
            VJI = VJI*2.*DX(I)*DX(I)
            F1(J,I) = (R21S-R11S)/VJI
            F2(J,I) = (R22S-R12S)/VJI
1007 CONTINUE

```

III

```

1008 CONTINUE
      WRITE(6,6571) NBC1,NBC2,IGES,IGM1,(IE(K),K=1,N)
1      ,(IA(K),K=1,N)
6571 FORMAT(1H ,2I5,2X,2I5,2X,5I5,2X,5I5)
C*** BEGINN DER AUSWERTUNG FUER EINEN ZEITSCHRITT
2 JA = M
  JE = 1
  JS = -1
  ITR = 0
  ITS = ITS0
  DO 1009 I = 2,IGM1
    IM1 = I - 1
    KI = KK(IM1)
    DO 1009 J = 1,M
      TX = T(J,I)
1009   RCPA(J,IM1) = RHO(TX,J,KI)*CP(TX,J,KI)/DT*TX
C*** BEGINN EINES ITERATIONSSCHRITTES
4000 JJ = JA
  JA = JE
  JE = JJ
  JS = -JS
  J = JA
4001 IF ((J-JE)*JS.GT.0) GO TO 4029
  JP1 = J + 1
  JM1 = J - 1
C*** RANDBEDINGUNG FUER I = 1
  GO TO (4101,4201,4301),NBC1
4101 TS(J,1) = TKUE(ZEIT,3,J,1)
  GO TO 4002
4201 DI = DX(1)
  TX = TS(J,1)
  TS(J,1) = TS(J,2)-ALPHA(TX,3,J,1,1,1.)*DI/LAMBDA(TX,J,1)
  GO TO 4002
4301 DI = DX(1)
  TX = TS(J,1)
  RB1 = LAMBDA(TX,J,1)/DI
  RB2 = ALPHA(TX,3,J,1,1,1.)
  TS(J,1)=(RB2*TKUE(ZEIT,3,J,1)+RB1*TS(J,2))/(RB1+RB2)
C*** INNENGLICHTUNGEN FUER I = 2 BIS IGM1
4002 TX = TS(J,2)
  KI = KK(1)
  IF (J.NE.1) GO TO 4003
  TX1 = TKUE(ZEIT,1,J,KI)
  GO TO 4004
4003 TX1 = TS(JM1,2)
4004 IF (J.NE.M) GO TO 4005
  TX2 = TKUE(ZEIT,2,J,KI)
  GO TO 4006
4005 TX2 = TS(JP1,2)
4006 IF (IMOD.NE.0) GO TO 4008
  RM1 = RM(1,J,1)
  RM2 = RM(2,J,1)
  ALIX(J,1) = ALPHA(TX,1,J,KI,1,RM1)*G1(J,1)
  ALAX(J,1) = ALPHA(TX,2,J,KI,1,RM2)*G2(J,1)
  RCP(J,1) = RHO(TX,J,KI)*CP(TX,J,KI)/DT
  RCP(J,1) = ALIX(J,1) + ALAX(J,1) + RCP(J,1)
  QW1(J,1) = LAMBDA(TS(J,1),J,KI)*F1(J,1)
  TP1 = (TX*DX(2)+TS(J,3)*DX(1))/(DX(1)+DX(2))

```

IV

```

DO 4007 IJK = 1,IJKL
  RR1 = LAMBDA(TP1,J,KI)/DX(1)
  RR2 = LAMBDA(TP1,J,KK(2))/DX(2)
4007 TP1 = (RR1*TX+RR2*TS(J,3))/(RR1+RR2)
  RR12(J,1) = RR1/(RR1+RR2)
  RR22(J,1) = RR2/(RR1+RR2)
  QW2(J,1) = LAMBDA(TP1,J,KI)*F2(J,1)
  DKM2(J,1) = RCP(J,1)+QW2(J,1)*(1.-RR12(J,1))+QW1(J,1)
  DKM3(J,1) = -QW2(J,1)*RR22(J,1)
4008 DKM4(I) = QW1(J,1)*TS(J,1)+ALIX(J,1)*TX1+ALAX(J,1)*TX2
  1 +RCPA(J,1) + Q(I,ZEIT,J,KI)
  I = 3
4009 IF (I.GT.IGM2) GO TO 4016
  TX = TS(J,I)
  IM1 = I - 1
  IP1 = I + 1
  IM2 = I - 2
  KI = KK(IM1)
  IF (J.NE.1) GO TO 4010
  TX1 = TKUF(ZEIT,1,J,KI)
  GO TO 4011
4010 TX1 = TS(JM1,I)
4011 IF (J.NE.M) GO TO 4012
  TX2 = TKUE(ZEIT,2,J,KI)
  GO TO 4013
4012 TX2 = TS(JP1,I)
4013 IF (IMOD.NE.0) GO TO 4015
  RM1 = RM(1,J,IM1)
  RM2 = RM(2,J,IM1)
  ALIX(J,IM1) = ALPHA(TX,1,J,KI,IM1,RM1)*G1(J,IM1)
  ALAX(J,IM1) = ALPHA(TX,2,J,KI,IM1,RM2)*G2(J,IM1)
  RCP(J,IM1) = RHO(TX,J,KI)*CP(TX,J,KI)/DT
  RCP(J,IM1) = ALIX(J,IM1)+ALAX(J,IM1)+RCP(J,IM1)
  TM1 = (TX*DX(IM2)+TS(J,IM1)*DX(IM1))/(DX(IM2)+DX(IM1))
  TP1 = (TX*DX(I)+TS(J,IP1)*DX(IM1))/(DX(I)+DX(IM1))
  DO 4014 IJK = 1,IJKL
    RR1 = LAMBDA(TP1,J,KI)/DX(IM1)
    RR2 = LAMBDA(TP1,J,KK(I))/DX(I)
    RL1 = LAMBDA(TM1,J,KI)/DX(IM1)
    RL2 = LAMBDA(TM1,J,KK(IM2))/DX(IM2)
    TP1 = (RR1*TX+RR2*TS(J,IP1))/(RR2+RR1)
4014 TM1 = (RL1*TX+RL2*TS(J,IM1))/(RL1+RL2)
  QW1(J,IM1) = LAMBDA(TM1,J,KI)*F1(J,IM1)
  QW2(J,IM1) = LAMBDA(TP1,J,KI)*F2(J,IM1)
  RR12(J,IM1) = RR1/(RR1+RR2)
  RR22(J,IM1) = RR2/(RR1+RR2)
  RR11(J,IM1) = RL1/(RL1+RL2)
  RR21(J,IM1) = RL2/(RL1+RL2)
  DKM1(IM1) = -QW1(J,IM1)*RR21(J,IM1)
  DKM2(J,IM1) = RCP(J,IM1)+QW1(J,IM1)*(1.-RR11(J,IM1))
  1 +QW2(J,IM1)*(1.-RR12(J,IM1))
  DKM3(J,IM1) = -QW2(J,IM1)*RR22(J,IM1)
4015 DKM4(IM1) = ALIX(J,IM1)*TX1+ALAX(J,IM1)*TX2
  1 +RCPA(J,IM1) + Q(IM1,ZEIT,J,KI)
  I = I + 1
  GO TO 4009
4016 TX = TS(J,IGM1)
  KI = KK(IGM2)

```

```

      IF (J.NE.1) GO TO 4017
      TX1 = TKUE(ZEIT,1,J,KI)
      GO TO 4018
4017 TX1 = TS(JM1,IGM1)
4018 IF (J.NE.M) GO TO 4019
      TX2 = TKUE(ZEIT,2,J,KI)
      GO TO 4020
4019 TX2 = TS(JP1,IGM1)
4020 IF (IMGD.NE.0) GO TO 4022
      RM1 = RM(1,J,IGM2)
      RM2 = RM(2,J,IGM2)
      ALIX(J,IGM2) = ALPHA(TX,1,J,KI,IGM2,RM1)*G1(J,IGM2)
      ALAX(J,IGM2) = ALPHA(TX,2,J,KI,IGM2,RM2)*G2(J,IGM2)
      RCP(J,IGM2) = RHO(TX,J,KI)*CP(TX,J,KI)/DT
      RCP(J,IGM2) = ALIX(J,IGM2)+ALAX(J,IGM2)+RCP(J,IGM2)
      QW2(J,IGM2) = LAMBDA(TX,J,KI)*F2(J,IGM2)
      TM1 = (TX*DX(IGM3)+TS(J,IGM2)*DX(IGM2))/(LX(IGM3)+DX(IGM2))
      DC 4021 IJK = 1,IJKL
          RL1 = LAMBDA(TM1,J,KI)/DX(IGM2)
          RL2 = LAMBDA(TM1,J,KK(IGM3))/DX(IGM3)
4021 TM1 = (RL1*TX+RL2*TS(J,IGM2))/(RL1+RL2)
      RR11(J,IGM2) = RL1/(RL1+RL2)
      RR21(J,IGM2) = RL2/(RL1+RL2)
      QW1(J,IGM2) = LAMBDA(TM1,J,KI)*F1(J,IGM2)
      DKM2(J,IGM2) = RCP(J,IGM2)+QW2(J,IGM2)+QW1(J,IGM2)*
1          (1.-RR11(J,IGM2))
      DKM1(IGM2) = -QW1(J,IGM2)*RR21(J,IGM2)
4022 DKM4(IGM2) = QW2(J,IGM2)*TS(J,IGES)+ALIX(J,IGM2)*TX1
1 +ALAX(J,IGM2)*TX2+RCPA(J,IGM2)+Q(IGM2,ZEIT,J,KI)
C*** ES FOLGT DIE AUFLUESUNG DES GLEICHUNGSSYSTEMS
      IF (IMOD.NE.0) GO TO 4123
      DO 4023 I = 1,IGM3
          IP1 = I + 1
          EQ(J,I) = DKM1(IP1)/DKM2(J,I)
4023 DKM2(J,IP1) = DKM2(J,IP1) - DKM3(J,I)*EQ(J,I)
4123 DO 4223 I = 1,IGM3
          IP1 = I + 1
4223 DKM4(IP1) = DKM4(IP1) - DKM4(I)*EQ(J,I)
      DKM1(IGM2) = DKM4(IGM2)/DKM2(J,IGM2)
      I = IGM3
      IP1 = IGM2
4024 IF (I.LT.1) GO TO 4025
      DKM1(I) = (DKM4(I)-DKM3(J,I)*DKM1(IP1))/DKM2(J,I)
      IP1 = I
      I = I - 1
      GO TO 4024
C*** UEBERRELAXATION
4025 DO 4026 I = 1,IGM2
          IP1 = I + 1
          TS(J,IP1) = OMEGA1*TS(J,IP1) + OMEGA*DKM1(I)
          IF (TS(J,IP1).LT.TMIN) TS(J,IP1) = TMIN
4026 CONTINUE
      IF (ICTRL.GE.0) GO TO 4226
      DO 4126 I = 1,IGES
4126 WRITE(6,6002) I,(TS(JJ,I),JJ=1,M)
4226 CONTINUE
C*** RANDBEDINGUNG FUER I = IGES
      GO TO (4127,4227,4327),NBC2

```


VI

```

4127 TS(J,IGES) = TKUE(ZEIT,4,J,N)
      GO TO 4028
4227 TX = TS(J,IGES)
      DI = DX(IGM2)
      TS(J,IGES) = TS(J,IGM1)-ALPHA(TX,4,J,N,IGM2,1.)*DI/LAMBDA(TX,J,N)
      GO TO 4028
4327 TX = TS(J,IGES)
      DI = DX(IGM2)
      RB1 = LAMBDA(TX,J,N)/DI
      RB2 = ALPHA(TX,4,J,N,IGM2,1.)
      TS(J,IGES) = (RB2*TKUE(ZEIT,4,J,N)+RB1*TS(J,IGM1))/(RB1+RB2)
4028 J = J + JS
      GO TO 4001
C
C*****
C ABRAGE DER ERREICHTEN GENAUGKEIT
C*****
C
4029 TX = TS(JP,KP)
      IMOD = 1
      IF (DABS(TX-TP)/TP.LT.EPS.AND.ITR.GT.ITM) GO TO 5000
      ITR = ITR + 1
      IF (MOD(ITR,ITS).NE.0) GO TO 4129
      IF (ITS.GT.1) ITS = ITS - 1
      IMOD = 0
4129 TP = DABS(TX)
      IF (ICTRL.EQ.0) GO TO 4000
      WRITE(6,6004) TP
      DO 4030 I = 1,IGES
4030 WRITE(6,6002) I,(TS(J,I),J=1,M)
      GO TO 4000
C
C
C*****
C NAECHSTER ZEITSCHRITT
C*****
C
5000 IF (ISAVE.EQ.0) GO TO 5101
      IRETZ = 2
      GO TO 7112
5101 IF (MOD(ITT,IDR).EQ.0) GO TO 6000
5001 TU1 = TS(JP,KP)
      TU2 = T(JP,KP)
      TX = DABS((TU2-TU1)/TU2)*5.
C      IF (TX.LT.EPS) DT = DT + 0.1*DT
      TX = TX/50.
      IF (TX.LT.EPS) GO TO 5002
C      DT = DT - 0.2*DT
C      ZEIT = ZEIT1 + DT
C      GO TO 5004
5002 DO 5003 I = 1,IGES
      DO 5003 J = 1,M
5003     T(J,I) = TS(J,I)
      ITT = ITT + 1
      ZEIT1 = ZEIT
      ZEIT = ZEIT1 + DT
      IF (ZEIT.GT.ZEITN) GO TO 7000
5004 TP = 1.D12

```

VII

```

      DO 51004 J = 1,M
        DO 51004 I = 1,ICES
51004      TS(J,I) = T(J,I)
          IMOD = 0
          GO TO 2

```

C

C*****

C AUSDRUCKEN DES ERGEBNISSES

C*****

C

```

6000 WRITE(6,6002) ITT,ZEIT
      WRITE(6,6003)
6021 DO 6001 I = 1,ICES
      WRITE(6,6002)I,(TS(J,I),J=1,M)
6001 CONTINUE
      WRITE(6,6003)
6002 FORMAT(1H ,I5,11F10.3)
6003 FORMAT(1H ,'*****')
6004 FORMAT(1H ,F10.5)
      GO TO 5001

```

C

C

C*****

C ENDE DER AUSWERTUNG ODER NEU EINLESEN

C*****

```

7000 READ(1,7011) ICALL
      IRETZ = 1
      WRITE(6,7666) ICALL
      IF (ICALL.EQ.0.OR.ICALL.GT.10) STOP
8000 GO TO (7001,7002,7003,7004,7005,7006,7007,7008,7009,7010,
1  7111,7112),ICALL
7001 READ(1,7012) ZEITN,DT, IDR
      GO TO 7000
7002 READ(1,7013) M,N
      WRITE(6,7666) M,N
      DO 7102 K = 1,N
      READ(1,7016) (R(1,J,K),R(2,J,K),J=1,M)
7102 WRITE(6,7667) (R(1,J,K),R(2,J,K),J=1,M)
      MPI = M + 1
      NPI = N + 1
      READ(1,7014) (L(K),K=1,NPI)
      WRITE(6,7668) (L(K),K=1,NPI)
      READ(1,7013) (IM(K),K=1,N)
      WRITE(6,7666) (IM(K),K=1,N)
      GO TO 7000
7003 DO 7103 J = 1,M
      READ(1,7013) (IMAT(J,K),K=1,N)
7103 WRITE(6,7666) (IMAT(J,K),K=1,N)
      GO TO 7000
7004 DO 7104 J = 1,MPI
      READ(1,7014) (ALRR(J,K),K=1,N)
7104 WRITE(6,7668) (ALRR(J,K),K=1,N)
      GO TO 7000
7005 DO 7105 J = 1,2
      READ(1,7014) (TKR(J,K),K=1,N)
7105 WRITE(6,7668) (TKR(J,K),K=1,N)
      GO TO 7000
7006 DO 7106 J = 1,M

```

VIII

```

      READ(1,7014) (ALZZ(J,K),K=1,2)
7106 WRITE(6,7668) (ALZZ(J,K),K=1,2)
      GO TO 7000
7007 DO 7107 J = 1,M
      READ(1,7014) (TKZ(J,K),K=1,2)
7107 WRITE(6,7668) (TKZ(J,K),K=1,2)
      GO TO 7000
7008 DO 7108 J = 1,M
      READ(1,7014) (QZ(J,K),K=1,N)
7108 WRITE(6,7668) (QZ(J,K),K=1,N)
      GO TO 7000
7009 READ(1,7014) TKST
      TMIN =TKST
      DO 7109 I = 1,55
      DO 7109 J = 1,9
      T(J,I) = TKST
7109 TS(J,I) = TKST
      GO TO 7000
7010 ZEIT = ZEIT1+DT
      IIT = 1
      IMOD = 0
      TP = 1.D12
      GO TO 1
7111 READ(1,7011) IUNIT
      READ(IUNIT,7017) M,N,IGES,ZEIT1
      READ(IUNIT,7017) (((R(JK,J,K),JK=1,2),J=1,M),K=1,N)
      NP1 = N + 1
      READ(IUNIT,7017) (L(K),K=1,NP1)
      READ(IUNIT,7017) (IM(K),K=1,N)
      READ(IUNIT,7017) ((IMAT(J,K),J=1,M),K=1,N)
      MP1 = M + 1
      READ(IUNIT,7017) ((ALRR(J,K),J=1,MP1),K=1,N)
      READ(IUNIT,7017) ((TKR(J,K),J=1,2),K=1,N)
      READ(IUNIT,7017) ((ALZZ(J,K),J=1,M),K=1,2)
      READ(IUNIT,7017) ((TKZ(J,K),J=1,M),K=1,2)
      READ(IUNIT,7017) ((QZ(J,K),J=1,M),K=1,N)
      READ(IUNIT,7017) ((T(J,I),J=1,M),I=1,IGES)
      DO 7211 I = 1,IGES
      DO 7211 J = 1,M
7211 TS(J,I) = T(J,I)
      GO TO 7010
7112 IF (ISAVE.EQ.0) READ(1,7011) IUNIT
      REWIND IUNIT
      WRITE(IUNIT,7017) M,N,IGES,ZEIT1
      WRITE(IUNIT,7017) (((R(JK,J,K),JK=1,2),J=1,M),K=1,N)
      WRITE(IUNIT,7017) (L(K),K=1,NP1)
      WRITE(IUNIT,7017) (IM(K),K=1,N)
      WRITE(IUNIT,7017) ((IMAT(J,K),J=1,M),K=1,N)
      WRITE(IUNIT,7017) ((ALRR(J,K),J=1,MP1),K=1,N)
      WRITE(IUNIT,7017) ((TKR(J,K),J=1,2),K=1,N)
      WRITE(IUNIT,7017) ((ALZZ(J,K),J=1,M),K=1,2)
      WRITE(IUNIT,7017) ((TKZ(J,K),J=1,M),K=1,2)
      WRITE(IUNIT,7017) ((QZ(J,K),J=1,M),K=1,N)
      WRITE(IUNIT,7018) ((T(J,K),J=1,M),K=1,IGES)
      GO TO (7000,5101),IRETZ
7113 READ(1,7014) CS
      READ(1,7014) (DEPS(J),J=1,M)
      DO 7213 J = 1,M

```

```
7213   DEPS(J) = 1./DEPS(J)
      GO TO 7000
7017  FORMAT(9Z8)
7018  FORMAT(4(Z16,2X))
7011  FORMAT(10I5)
7012  FORMAT(F10.2,F10.2,I5)
7013  FORMAT(14I5)
7014  FORMAT(8F9.2)
7015  FORMAT(2F5.2,F10.2,2I5,I3,I2,I5,4F5.2)
7016  FORMAT(4(F8.2,2X,F8.2))
7666  FORMAT(1H ,15I8)
7667  FORMAT(1H ,6(F8.2,2X,F8.2))
7668  FORMAT(1H ,12F10.2)
12345 FORMAT(1H ,2I5,F15.7)
      END
```

X

```
BLOCK DATA
REAL*8 RM
COMMON /BLOCK1/ M,N, IMAT(9,7) /BLOCK2/ QZ(9,7),FQ(9,53)
1 /BLOCK3/ ALRR(10,7),ALZZ(9,2),ISP,CS,DEPS(9)
2 /BLOCK4/ TKR(2,7),TKZ(9,2) /BLOCK5/ RM(2,9,53)
3 /BLOCK6/ W(7)
DATA M/9/,N/7/,IMAT/63*13/,ISP/2/,QZ/63*0.0/,FQ/477*1.0/,
1 ALRR/70*0.0/,ALZZ/18*0.0/,TKR/14*0.0/,TKZ/18*0.0/,
2 RM/954*0.0/,W/7*1.0/,CS/0.0/,DEPS/9*0.5/
END
```

XI

```

C
C
C *****:
C  WAERME DURCHGANGSZAHLEN
C *****:
  REAL FUNCTION ALPHA*(T,I,J,K,IK,R5)
  REAL*8 T(1),ALX,LS, TM1, TM, TM2, L1, L2, LAMBDA, RM(2,9,53)
  REAL*8 R1,R2,R3,R4
  REAL ALRR(10,7),ALZZ(9,2)
  COMMON /BLOCK3/ ALRR,ALZZ,ISP,CS,DEPS(9)
  COMMON /BLOCK5/ RM
C *****:
  L = 1
  GO TO (1,2,3,4),I
  1 JMI = J - 1
    ALX = ALRR(J,K)
    IF (ALX.GT.999.) GO TO 10
    ALPHA = ALX
    RETURN
  2 JPI = J + 1
    ALX = ALRR(JPI,K)
    IF (ALX.GT.999.) GO TO 20
    ALPHA = ALX
    RETURN
  3 ALPHA = ALZZ(J,1)/R5
    RETURN
  4 ALPHA = ALZZ(J,2)/R5
    RETURN
  10 TM1 = T(L-1)
    TM2 = T(L)
    TM = .5*(TM1+TM2)
    JPI = J
    R1 = RM(1,JMI,IK)
    R2 = RM(2,JMI,IK)
    R3 = RM(1,J,IK)
    R4 = RM(2,J,IK)
  1000 IF (R1.GT..5D-4) GO TO 1001
    GO TO 1002
  1001 R1 = .5D0*(R2+R1)
  1002 R4 = .5D0*(R4+R3)
  102 L1 = LAMBDA(TM,JMI,K)/(R2-R1)*R2
    I1 = 2
  103 IF (DABS(R3/R2-1.).GT..5D-4) GO TO 105
  104 I2 = 1
    GO TO 106
  105 LS = LAMBDA(TM,C,ISP)/DLOG(R3/R2)
    LS = LS + R2*CS/((DEPS(JMI)+R2/R3*(DEPS(JPI)-1.))*
  1 (TM1*TM1+TM2*TM2)*(TM1+TM2)
    I2 = 2
  106 L2 = LAMBDA(TM,JPI,K)/(R4-R3)*R3
  107 GO TO (108,109),I2
  108 ALPHA = 1.00/((1.00/L1+1.00/L2)*R5)
    RETURN
  109 ALPHA = 1.00/((1.00/L1+1.00/LS+1.00/L2)*R5)
    RETURN
  20 TM1 = T(L)
    TM2 = T(L+1)
    TM = .5*(TM1+TM2)

```

```
JM1 = J  
R1 = RM(1,J,IK)  
R2 = RM(2,J,IK)  
R3 = RM(1,JP1,IK)  
R4 = RM(2,JP1,IK)  
GC TO 1000  
END
```

```
C
C
C*****
C  KUEHLMITTEL TEMPERATUREN
C*****
  REAL FUNCTION TKUE*8(Z,I,J,K)
  REAL TKR(2,7),TKZ(9,2)
  COMMON /BLOCK4/ TKR,TKZ
C*****
C  GO TO (1,2,3,4),I
1  TKUE = TKR(1,K)
   RETURN
2  TKUE = TKR(2,K)
   RETURN
3  TKUE = TKZ(J,1)
   RETURN
4  TKUE = TKZ(J,2)
   RETURN
END
```



```
C
C
C*****
C HEIZLEISTUNG
C*****
  REAL FUNCTION Q*8(I,Z,J,K)
  REAL QZ(9,7)
  COMMON /BLOCK2/ QZ,FQ(9,53)
C*****
  Q = QZ(J,K)/FQ(J,I)
  RETURN
  END
```