

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

Oktober 1973

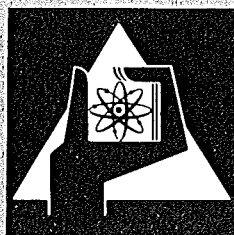
KFK 1869

Institut für Reaktorentwicklung
Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

A T T Ø W – K

Die Karlsruher IBM 360/65 – Version des
2d-Removal-Diffusion-Abschirmprogrammes **A T T Ø W**

G. Bönisch, H.W. Wiese



**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 1869

Institut für Reaktorentwicklung
Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

A T T Ø W - K

Die Karlsruher IBM360/65 - Version
des 2d-Removal-Diffusion-Abschirm-
programmes A T T Ø W

G. Bönisch

H.W. Wiese

Gesellschaft für Kernforschung mbH., Karlsruhe

Zusammenfassung:

Die IBM7090-Version des Removal-Diffusions-Abschirmcodes ATTØW für zweidimensionale Geometrie (x,y oder r,z) wurde auf die Karlsruher IBM360/65-370/165-Anlage übertragen. Es wurde die Möglichkeit geschaffen, Karlsruher Gruppenkonstantensätze direkt oder nach Speicherung als programmzugehörige Datei für den Diffusionsteil zu benutzen. Weiter wurde die SABINE-Removalquerschnittsbibliothek in das Programm integriert. Die Eingabebeschreibung wurde teilweise neu formuliert und mit zusätzlichen Erläuterungen versehen.

ATTØW-K: The Karlsruhe IBM360/65-Version of the 2d Removal Diffusion Shielding Code ATTØW.

Abstract:

The IBM7090-version of the removal-diffusion shielding code ATTØW for twodimensional geometry (x,y or r,z) was converted for the Karlsruhe IBM360/65-370/165 computer. The new version can utilize Karlsruhe multigroup constant sets either already in core storage or from external storage. Furthermore the SABINE removal cross section library was built in. The input discription has been partly reformulated and augmented by additional explanations.

<u>Inhalt:</u>	Seite
A1. Einleitung	2
A2. Möglichkeiten des Programms	2
A3. Zwischenspeicher	4
A4. Anordnung des Flusses und Ortsgitters auf Einheit 10	4
A5. Steuerung der Diffusionsrechnung durch den ØMEGA- Parameter und die ACCUR- und CØUNT-Parameter; Ausga- be von Kontrollgrößen	6
BO. Erläuterungen zur Eingabe	8
B1. Problemeingabe	9
Geometrie, Steuergrößen	9
Ortsgitter, Zonen	10
Quellen	12
Randbedingungen	13
Auswertung	14
Weitere Möglichkeiten	17
B2. Dateneingabe	20
Keine Bibliotheksdaten	20
RØACH-B-Bibliothek	21
KFKINR-Bibliothek	24
Daten mit NUSYS	25
B3. Removeeingabe	27
C. Datenbibliotheken	32
C1. RØACH-B-Bibliothek	32
C2. KFKINR-Bibliothek	32
C3. SABINE-Bibliothek	34
D. Reduzierung des Platzbedarfs durch Overlay	36
E. Eingabebeispiel mit JCL-Karten für die IBM360/65	37
F. Korrigiertes Removal-Diffusion Rechenbeispiel	38
G. Programmänderungen	52
H. Fortran-Liste von ATTØW-K	60
I. Neue Unterprogramme im Zusammenhang mit NUSYS- und SABINE-Querschnitten	109
K. RISLEY-Unterprogramme	116
L. Programm zum Lesen der gespeicherten Resultate	124
M. Literatur	59

A1. Einleitung:

Das Programm ATTØW wurde als IBM/7090-Version von der ENEA-Programmbibliothek übernommen und der Karlsruher Rechenanlage IBM360/65-370/165 angepaßt. Die vorgenommenen Änderungen werden in Abschnitt G beschrieben. Die Testbeispiele des Originalberichtes¹⁾ konnten mit sehr guter Übereinstimmung (Abweichungen in Flüssen und Removalquellen höchstens 1 %) nachgerechnet werden. Die Korrektur³⁾ eines Fehlers im Unterprogramm RRSCE ändert allerdings die Ergebnisse des Testbeispiels für Removal-Diffusions-Rechnungen um einen Faktor ≈ 1.9 , so daß es angebracht erscheint, dieses Testbeispiel mit den neuen Resultaten diesem Bericht anzufügen.

Als Datenbibliotheken können neben der originalen, eingebauten 16-Gruppen-RØACH-B-Bibliothek die Karlsruher 26-Gruppen-Dateien benutzt werden. Benutzer, die keinen Zugriff zu diesen Dateien haben, können andere 26-Gruppen-Datensätze bei geeignetem Format (s. Abschnitt C) dem Programm über externe Speicher zuführen. Eine spezielle Version des 26-Gruppen-KFKINR-Satzes²⁾ wurde als Test-Datensatz verwendet.

Die ursprüngliche ATTØW-Version enthielt keine Removalquerschnitte; sie mußten über Karten eingegeben werden. Dieser Engpaß wurde durch den Einbau der Removalquerschnitte des Programms SABINE⁴⁾ beseitigt (s. Abschnitt C).

Die Beschreibung der Eingabe wurde neu formuliert und, wo es angebracht erschien, mit zusätzlichem erläuterndem Text versehen.

Die Eingabe von RASH-Daten wird nicht beschrieben; sie ist nach dem Originalbericht¹⁾ vorzunehmen.

Der vorliegende Bericht enthält einen Vorschlag für eine Overlay-Struktur und die Liste des Quellprogramms der IBM370-Version ATTØW-K.

A2. Möglichkeiten des Programms:

Das Programm löst das System der inhomogenen Multigruppen-Diffusions-Gleichungen

Manuskript zum Druck eingereicht am 20. September 1973

$$\operatorname{div}(D_g \operatorname{grad} \varphi_g) - \sum_g^r \varphi_g + B_g = 0 \quad g = 1, 2, \dots, n_g$$

für eine beliebige Anzahl n_g von Gruppen in RZ- und XY-Geometrie für ca. 100 x 100 Maschenpunkte. Die Quelle B_g besteht aus der Streuquelle und einer äußeren Quelle:

$$B_g = Q_g + \sum_{\substack{h < g \\ g-h \leq n_{\text{scat}}}} \Sigma_{h,g} \varphi_h \quad g = 1, 2, \dots, n_g,$$

wobei in RZ-Geometrie die äußere Quelle Q_g neben einer Volumenquelle eine Removalquelle enthalten kann. Es ist n_{scat} die Maximalzahl von Einstreugruppen.

Durch Verwendung der allgemeinen linearen Randbedingung

$$a D \frac{d\varphi}{dn} + b\varphi + c = 0$$

für jede Gruppe und jeden Randpunkt ist es möglich, am Rande

- den Fluß $\varphi > 0$ vorzugeben
- den Strom in Normalenrichtung $D \frac{d\varphi}{dn}$ vorzugeben
($D \frac{d\varphi}{dn} = 0$: Symmetrie)
- den Fluß am extrapolierten Rand mit einer beliebigen Extrapolationslänge d Null zu setzen.

Quellen können gruppenabhängig in Mischungen und/oder Zonen angegeben werden.

Als Ausgabe ist möglich:

- der Fluß in ausgewählten Bereichen
- der Strom durch ausgewählte Maschen
- Raten (Dosisleistungen usw.) in ausgewählten Bereichen
- auf einem externen Speicher der Fluß gruppenweise ortsabhängig zur späteren Auswertung und für die Berechnung von Sekundärgammaquellen

A3. Zwischenspeicher:

Auf der IBM370/165-360/65 können statt Magnetbänder Trommel- und Plattenspeicher benutzt werden. Folgende Einheiten sind als Zwischenspeicher zu deklarieren:

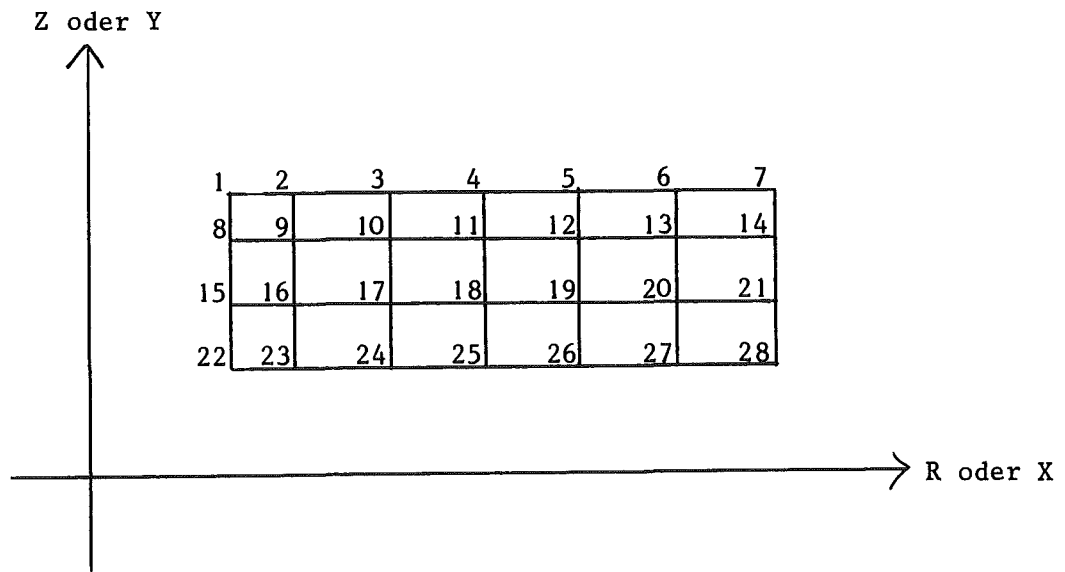
Nummer:	Auf der Einheit wird gespeichert:
1	Zwischenergebnisse
2	Removalquellen (nur nötig bei Removalrechnungen mit Speicherung der Removalquellen und bei Removal-Diffusions-Rechnungen)
3	früher berechnete Flüsse, die ausgewertet, verbessert oder vervollständigt werden sollen
4	RØACH-B-Bibliothek
7	Stanzen von Karten
8	Zwischenergebnisse
9	Zwischenergebnisse
10	Fluß und Ortsgitter am Ende der Rechnung
11	aufbereitete Eingabedaten
15	KFKINR-Bibliothek oder mit NUSYS bereitgestellte Daten

Richtwerte für die DD-Parameter der angegebenen Einheiten (mit Ausnahme von 15) sind UNIT=SYSDA, SPACE=(TRK,20), DCB=(BLKSIZE=2298,RECFM=VBS).

A4. Anordnung des Flusses und Ortsgitters auf Einheit 10:

Der ermittelte Fluß $\text{PHI}(I)$, $I = 1, 2, \dots, \text{NRØW} \cdot \text{NCØL}$, $I =$ Gitterpunktindex, wird in zwei Sätzen pro Gruppe auf den auf Einheit 10 liegenden Datenträger (Platte, Trommel oder Band) geschrieben.

NRØW ist die Zahl der Zeilen der Anordnung, NCØL die Zahl der Spalten. Die Nummerierung der Gitterpunkte erfolgt entsprechend dem skizzierten Beispiel ($\text{NRØW} = 4$, $\text{NCØL} = 7$).



Der Fluß wird für jede Gruppe mit den Anweisungen

```
MHALF(1) = NCØL*(NRØW/2)
MHALF(2) = NCØL*NRØW - MHALF(1)
IND = MHALF(1)
WRITE(10) (PHI(I), I = 1, IND)
IST = IND + 1
IND = IND + MHALF(2)
WRITE(10) (PHI(I), I = IST,IND)
```

zuerst für die obere und dann für die untere Hälfte der Anordnung gespeichert. Am Ende der Rechnung wird das Ortsnetz mit

```
WRITE(10) NCØL, (R(I), I = 1, NCØL)
WRITE(10) NRØW, (Z(I), I = 1, NRØW)
```

geschrieben.

Ein Beispiel für ein Leseprogramm, das durch geeignete Ergänzungen für zusätzliche Auswertungen, z.B. die Berechnung von Sekundärgammaquellen, eingesetzt werden kann, ist in Kapitel L zu finden.

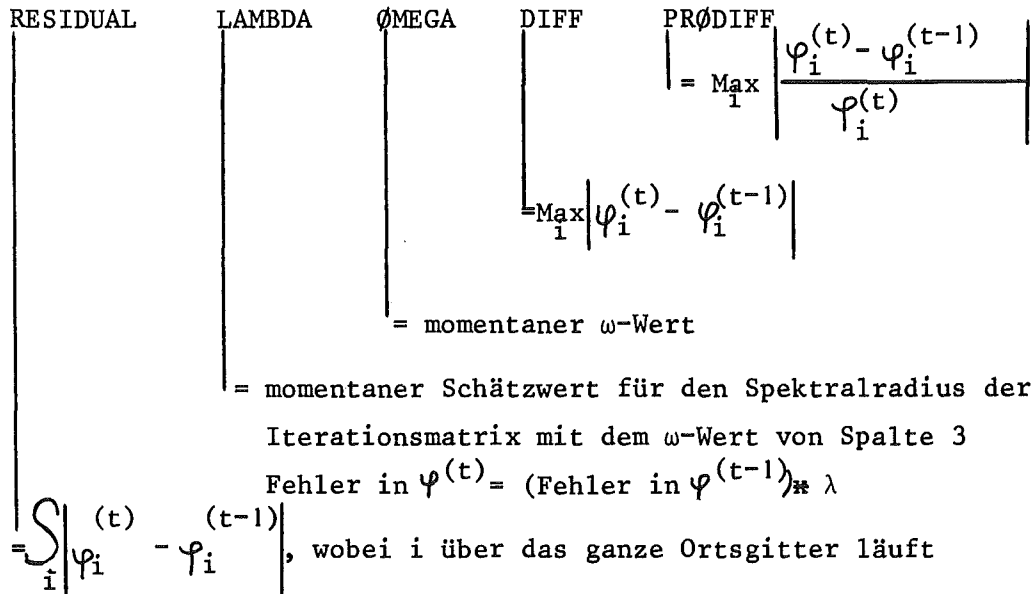
A5. Steuerung der Diffusionsrechnung durch den OMEGA-Parameter und die ACCUR- und COUNT-Parameter; Ausgabe von Kontrollgrößen

Die Multigruppen-Diffusionsgleichungen werden mit dem Zeilenüberrelaxationsverfahren gelöst (Einzelheiten s. Originalbericht¹⁾). Dabei ist der Beschleunigungsparameter ω bestimmend für den Verlauf der Rechnung (Rechenzeit!), während die Genauigkeit a_1, a_2, a_3 und die Maximalzahl m_g von Iterationen in der g -ten Gruppe für den Abbruch des Iterationsverfahrens sorgen (s. Bl, K 1.3).

Der Beschleunigungsparameter ω wird eingegeben oder 1. gesetzt und im Verlauf der Rechnung durch das Programm auf der Suche nach dem optimalen Wert verändert. Es ist möglich, durch Erraten des besten ω -Wertes die Rechenzeit auf 1/10 der Rechenzeit bei $\omega = 1$, zu reduzieren.

ω variiert von 1. bis über 1.9 für einen 101x101-Fall.

Bei jedem Iterationsschritt werden die untenstehenden Kontrollgrößen ausgedruckt. Mit $\varphi_i^{(t)}$ als Fluß im i -ten Gitterpunkt (vgl. A4) wird nach der t -ten Iteration folgende Information ausgegeben:



Außerdem werden jeweils die vier Größen l_1, l_2, l_3, l_4 ausgedruckt, wobei $l_j, j = 1, 2, 3$ angibt, daß für die $1., 2., \dots, l_j$ -te Spalte der Anordnung die Konvergenzbedingung

$$pd_i = \left| (\varphi_i^{(t)} - \varphi_i^{(t-1)}) / \varphi_i^{(t)} \right| \leq a_j$$

bereits erfüllt ist. l_4 ist als "Radius" von der linken unteren Ecke definiert und hängt von a_3 ab. Konvergenz ist erreicht, wenn $l_3 = ncol$ (Anzahl der Spalten), d.h. wenn $pd_i \leq a_3$ für alle Gitterpunkte i ist.

Die Rechnung für die g -te Gruppe wird abgebrochen, wenn Konvergenz im obigen Sinne erreicht ist oder wenn die Zahl der Iterationsschritte die als COUNT-Parameter angegebene Maximalzahl m_g erreicht hat. Im letzten Fall wird "ITERATIONS HAVE NOT CONVERGED" und der Stand der Rechnung ausgedruckt.

Nach Konvergenz oder Abbruch geht das Programm zur nächsten Gruppe.

BO. Erläuterungen zu Eingabe

Die ATTØW-Eingabe ist formatfrei. Die Zahl 123.0 z.B. kann als 123, 123., 123.0, .123E3, 1.23E02 usw. eingegeben werden. Ein Blank oder das Kartenende (Spalte 72) wird als Ende einer Zahl oder eines Wortes interpretiert.

Die Bedeutung eines eingegebenen Zeichens wird aus dessen Stellung hinter gewissen Schlüsselwörtern entnommen. Die Schlüsselwörter werden im folgenden mit ausschließlich großen Buchstaben geschrieben. Sie sind wie angegeben zu lochen, sie brauchen nicht in Spalte 1 zu beginnen.

Die Symbole K 1.1, K 2.4, S 2.3 usw., die nicht zu lochen sind, dienen zur Kennzeichnung von Datenblöcken und Verzweigungen. Was hinter z.B. K 1.4 steht, kann auf mehreren Karten abgelocht werden. Für klein geschriebene evtl. indizierte alphabetische Größen sind die aktuellen Werte zu lochen. Sonderzeichen wie das Gleichheitszeichen = , Klammern (), der *, das Komma, sind wie angegeben zu lochen; sie brauchen von Zahlen nicht durch Blanks getrennt zu werden. Punkte ... geben an, daß folgende aktuelle Werte nach dem angegebenen Schema weiter zu lochen sind.

B1. Problemeingabe:

- K 1.1 * Überschrift
Diese Titeltkarte muß in Spalte 1 mit einem * beginnen. Die Spalten 2 - 72 dürfen einen beliebigen alpha-numerischen Text einschließlich Sonderzeichen enthalten. Weitere Titeltkarten sind möglich und müssen mit * in Spalte 1 beginnen.
- S 1.1 Wenn eine Diffusionsrechnung (mit oder ohne Removalquelle) folgt: K 1.2 ; wenn eine Removalrechnung (mit oder ohne anschließende Diffusionsrechnung) folgt: K 3.1
- K 1.2 ATTØW
GRØUPS ng
CØNSTANTS nmat nscat
ng = Anzahl der Energiegruppen in der Diffusionsrechnung. Hinter dem Schlüsselwort GRØUPS wird nach einem oder mehreren Blanks die Anzahl der Energiegruppen gelocht.
nmat = Anzahl von Mischungen mit dem Schlüsselwort M (s.B2. Dateneingabe).
nscat = maximale Anzahl von Einstreu-
gruppen (höchstens 5).
Die Zeichen CØNSTANTS 7 5 (als Beispiel) können auf einer neuen Karte gelocht werden, können aber auch auf derselben Karte wie z.B. GRØUPS 9 erscheinen.
- S 2.2 Es folgt die Dateneingabe, beginnend mit K 2.1; nach der Dateneingabe geht es mit K 1.3 weiter.
- K 1.3 GEØMETRY XY
ACCUR a₁ a₂ a₃
(wenn dies fehlt, wird GEØMETRY RZ angenommen)
Rechengenauigkeiten (s.A5.) mit a₁ > a₂ > a₃ (eingebaute Werte: $\omega_1 = 0.2$, $\omega_2 = 1.E-6$, $\omega_3 = 1.E-4$)

\emptyset MEGA ω Beschleunigungsparameter für jede Gruppe zu Anfang der Rechnung (eingebaut $\omega = 1.$) (s.A5.)

C \emptyset UNT $m_1 m_2 \dots m_k$ m_i = Anzahl von Iterationen in Gruppe i für $i \leq k$. In Gruppen i mit $i > k$ wird m_k -mal iteriert. (Eingebaut $m_i = 50$ für alle i).
Wenn $m_{i+1} = m_{i+2} = \dots = m_{i+j}$ ist, kann die Folge $m_{i+1} m_{i+2} \dots m_{i+j}$ durch $j * m_{i+1}$ ersetzt werden. Bei C \emptyset UNT 5 * 40 5 * 60 75 (Blanks nicht vergessen!) wird in Gruppe 1 - 5 40mal, in Gruppe 6 - 10 60mal und in Gruppe 11 - ng 75mal iteriert.
Eingebaute Werte werden benutzt, wenn die entsprechenden Karten fehlen.
Wenn auf Einheit 2 Removalquellen zur Verfügung stehen:

REMOVAL S \emptyset URCES (Bei XY-Geometrie werden diese Removalquellen jedoch ignoriert)

K 1.4 Hier wird das Ortsgitter und die Anordnung der in der Dateneingabe definierten Mischungen in diesem Ortsgitter eingegeben.

MESH

MR $0=r_{\min} i_1=r_1 i_2=r_2 \dots i_{\max}=r_{\max}$

MZ $0=z_{\min} j_1=z_1 j_2=z_2 \dots j_{\max}=z_{\max}$

Dabei ist

i_n = Gitterindex beim Radius r_n (cm)
 j_n = Gitterindex bei $z = z_n$ (cm)
Die Gitterindizes müssen bei 0 beginnen, und es muß gelten:

$$i_n < i_{n+1}, r_n < r_{n+1}$$
$$j_n < j_{n+1}, z_n < z_{n+1}$$

Zwischen der i_n -ten und der i_{n+1} -ten Gitterlinie werden vom Programm $i_{n+1} - i_n - 1$ Gitterlinien eingesetzt, so daß gleichweite Maschen entstehen.

Entsprechendes gilt für j_n und z_n .

Bei XY-Geometrie ist einzugeben:

MESH

MX $0=x_{\min}$ $i_1=x_1$ $i_2=x_2$... $i_{\max}=x_{\max}$

MY $0=y_{\min}$ $j_1=y_1$ $j_2=y_2$... $j_{\max}=y_{\max}$

Es folgt die Zuordnung der Mischungen durch

ZØNE k_0

$(i_{11} j_{11} i_{12} j_{12}) k_1 (i_{21} j_{21} i_{22} j_{22}) k_2 \dots$

$\dots (i_{m1} j_{m1} i_{m2} j_{m2}) k_m \dots$

Die k_m sind Mischungsnummern, definiert durch die Dateneingabe. Die Wertepaare $i_{m1} j_{m1}$ und $i_{m2} j_{m2}$ in einer Klammer definieren zwei Punkte P_{m1} und P_{m2} in den Schnittpunkten der Gitterlinien i_{m1}, j_{m1} bzw. i_{m2}, j_{m2} . Durch P_{m1} und P_{m2} ist ein "Rechteck" R_m definiert, wenn man P_{m1} als untere rechte und P_{m2} als obere linke Ecke ansieht. Aus dieser Interpretation ergeben sich die Bedingungen

$$0 \leq i_{m1} < i_{m2} \leq i_{\max}$$

$$0 \leq j_{m1} < j_{m2} \leq j_{\max}$$

Die obige Eingabe bewirkt, daß zunächst das gesamte Gebiet $(0 \ 0 \ i_{\max} \ j_{\max})$ mit der Mischung k_0 gefüllt wird. Diese Zuordnung wird dann für das Rechteck R_1 ersetzt durch Zuordnung der Mischung k_1 zu R_1 ; dann wird R_2 mit k_2 gefüllt usw. Jede Neuzuordnung für ein Rechteck R_m

überschreibt die vorher getroffene Zuordnung für dieses Rechteck.

S 2.3 Wenn Quellen vorhanden sind, folgen K 1.5 und K 1.6; sonst sofort K 1.6

K 1.5 Quellen können gruppenabhängig in Mischungen und /oder gruppenabhängig in Zonen eingegeben werden

Nach
SOURCES IN folgt für jede Gruppe und jede Mischung die zur Quelle beitragen

GRØUP g BY MATERIALS k q_g
mit
g = Nummer der Gruppe
k = Nummer der Mischung, definiert in der Dateneingabe
 q_g = Quellstärke ($n \text{ cm}^{-3} \text{ sec}^{-1}$) in der g-ten Gruppe
Dann folgt für jede Gruppe die zur Zonenquelle beiträgt

GRØUP g BY ZONES O.

$(i_{11} \ j_{11} \ i_{12} \ j_{12}) \ q_{g1} (i_{21} \ j_{21} \ i_{22} \ j_{22}) \ q_{g2} \ \dots\dots$

mit
 q_{gm} = Quellstärke ($n \text{ cm}^{-3} \text{ sec}^{-1}$) im Rechteck R_m in Gruppe g

$(i_{m1} \ j_{m1} \ i_{m2} \ j_{m2})$ = Rechteck R_m wie bei K 1.4
Wenn sich bei der zonenweisen Quelleingabe für Gruppen $h \geq g$ nur die Quellstärken ändern und nicht die Zonen, spart man Rechenzeit, wenn man für $h > g$ eingibt:

GRØUP h SAME O.

$q_{h1} \ q_{h2} \ q_{h3} \ \dots\dots$

K 1.6 Die allgemeine lineare Randbedingung lautet

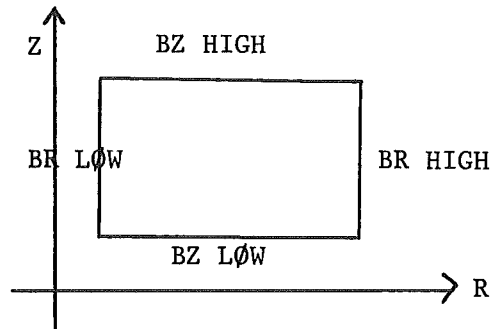
$$a D \frac{d\psi}{dn} + b\psi + c = 0$$

für jeden der $(i_{\max} + 1)(j_{\max} + 1)$ Gitterrandpunkte und für jede

Gruppe. Es lassen sich sechs Typen je nach dem Verschwinden einer oder mehrerer der Konstanten a,b,c definieren:

Typ	Null ges.	eingeben	resultierende Randbedingung
NØNZERØ	-	b/a c/a	
ØNEZERØ 1	a	c/b	Fluß $\psi = -c/b$
ØNEZERØ 2	b	c/a	Strom in Normalenrichtung $J_n = D \frac{d\psi}{dn} = -c/a$
ØNEZERØ 3	c	b/a	Fluß Null am extrapol. Rand im Abstand $d = \frac{D}{b/a}$ (normal $d = .7104 \lambda_{tr}$, also $b/a = .4692$)
TWØZERØ 1 3	a,c	-	Fluß $\psi = 0$
TWØZERØ 2 3	b,c	-	Strom in Normalenrichtung $J_n = D \frac{d\psi}{dn} = 0$ (Symmetrie)

Die Bezeichnung der Ränder erfolgt nach nebenstehender Figur. Bei XY-Geometrie ist R durch X und Z durch Y zu ersetzen. Die Bedingungen für einen Rand werden nach Eingabe des Schlüsselwortes punktweise mit aufsteigenden Gitterlinienindizes eingegeben. Für jeden Punkt werden der



Typ und dahinter die einzugebenden Konstanten b/a,c/a, c/b, c/a oder b/a gelocht. Bei einer Folge von gleichen Typen wird der Typ nur bei seinem ersten Auftreten eingegeben. Bei einer Folge von gleichen Konstanten oder Konstantenpaaren, z.B. b/a b/a b/a bzw. b/a c/a b/a c/a kann * 3 b/a bzw. * 2 b/a c/a gelocht werden. Hinter TWØZERØ m n steht entweder ALL für den ganzen Rand oder die Anzahl p der Punkte, für die die Bedingung erfüllt sein soll.

Für jede Gruppe wird eingegeben (als Beispiel):

```

BØUNDS BR LØW ØNEZERØ 1 .....
      BR HIGH ØNEZERØ 3 .....NØNZERØ .....
      BZ LØW TWØZERØ 2 3 ALL
      BZ HIGH NØNZERØ ....ØNEZERØ 2 .....
    
```

S 2.4 Das Programm bringt jetzt die bisherige Eingabe auf Einheit 11. Wird sie dort als temporärer Data-Set mit geeigneten DD-Parametern gespeichert, so kann sie in folgenden Steps von dort abgerufen werden. Wird sie als nicht-temporärer Data-Set (z.B. auf TSTLIB) gespeichert, so kann sie von dort in folgenden Jobs abgeholt werden.

Wenn nur die Eingabe auf Einheit 11 reserviert werden soll, folgt K 1.7; wenn in weniger als ng Gruppen (s K 1.2) gerechnet werden soll, folgt K 1.8; wenn mit der bisherigen Eingabe gerechnet werden soll, folgt K 1.9

- K 1.7 STØP Ende der Eingabe
- K 1.8 GRØUPS ng die Rechnung wird nur für die Gruppen
1, 2, ng ausgeführt.
Es folgt K 1.9
- K 1.9 SØLVE Wird weder STØP noch GRØUPS ng noch
SØLVE eingegeben, wird SØLVE angenommen.

S 2.5 Wenn nur die Flüsse ausgegeben werden sollen, folgt K 1.10;
sonst K 1.11

- K 1.10 END Ende der Eingabe
- K 1.11 Wenn die Flüsse in allen Gruppen und
Gitterpunkten noch einmal ausgedruckt
werden sollen:

ALLFLUXES

Wenn die Flüsse für ausgewählte Gebiete
ausgedruckt werden sollen:

SAMPLE R k(m)n Z p(q)r

Es wird

$$\Phi(r_i, z_j), \quad \begin{aligned} i &= k, k+m, k+2m, \dots, n \\ j &= p, p+q, p+2q, \dots, r \end{aligned}$$

ausgedruckt. Entsprechend für XY-Geo-
metrie.

Speziell: SAMPLe R O Z O(1)j_{max} er-
gibt alle Flüsse auf der z-Achse.

K 1.12

Wenn eine Rate berechnet und ausgedruckt werden soll:

TITLE
* Überschrift } kann entfallen

Wenn diese Rate für alle Gitterpunkte berechnet und ausgedruckt werden soll:

ACTIVATIØN AS SAMPLE

GRØUPS $n_1 n_2 , n_3 n_4 , n_5 , n_6 n_7$ (als Beispiel)

FACTØRS $\sigma_{n_1} \sigma_{n_1+1} \dots \sigma_{n_2}$ Es wird berechnet:
 $\sigma_{n_3} \sigma_{n_3+1} \dots \sigma_{n_4}$ $R = \sigma_{n_1} \Phi_{n_1} + \sigma_{n_1+1} \Phi_{n_1+1} + \dots + \sigma_{n_2} \Phi_{n_2}$
 σ_{n_5} $+ \sigma_{n_3} \Phi_{n_3} + \sigma_{n_3+1} \Phi_{n_3+1} + \dots + \sigma_{n_4} \Phi_{n_4}$
 $\sigma_{n_6} \sigma_{n_6+1} \dots \sigma_{n_7}$ $+ \sigma_{n_5} \Phi_{n_5}$
 $+ \sigma_{n_6} \Phi_{n_6} + \sigma_{n_6+1} \Phi_{n_6+1} + \dots + \sigma_{n_7} \Phi_{n_7}$

Wenn diese Rate für ausgewählte Punkte berechnet werden soll:

ACTIVATIØN R k(m)n Z p(q)r (s.K 1.11 oben)

GRØUPS wie oben

FACTØRS ...

Wenn mehrere Raten berechnet und ausgedruckt werden sollen:

Für jede Rate:

TITLE n n = 1,2, Nummer der Rate fortlaufend

* Überschrift

Dann wie vorher

ACTIVATIØN ...

GRØUPS ...

FACTØRS ...

Dann

REWIND

K 1.13

Wenn Ströme zwischen zwei Punkten in Achsenrichtungen ausgedruckt werden sollen:

CURRENTS (m₁ n₁) (m₂ n₂)

Es muß entweder m₁ = m₂ oder n₁ = n₂ sein. Es werden die Ströme durch jede Masche zwischen den Punkten (m₁, n₁) und (m₂, n₂) ausgedruckt.

Wenn Ströme zwischen mehreren Punkten ausgedruckt werden sollen:

CURRENTS (m₁ n₁) (m₁ n₂) (m₂ n₂) (m₂ n₃) ...,
(m₄ n₄) (m₅ n₄) (m₅ n₆)

Beispiel: CURRENTS (3 22) (7 22) (7 30),
(0 0) (0 10) (10 0)

K 1.14

Wenn die Flüsse als Anfangsschätzung in einer Folgerechnung verwendet werden sollen:

DUMP t

alle Flüsse werden auf Einheit t gespeichert; die Einheit t = 1,2 oder 3 muß durch geeignete DD-Parameter reserviert werden;

oder

DUMP t DØUBLE

interpolierte Flüsse auf der doppelten Anzahl von Gitterlinien in jeder Richtung werden gespeichert;

wenn die Flüsse aus Teilbereichen gespeichert werden sollen:

DUMP t X GE i₁ X LE i₂
Y GE j₁ Y LE j₂

es wird der Teilbereich

$$i_1 \leq i \leq i_2$$

$$j_1 \leq j \leq j_2$$

gespeichert; wird eine Grenze weglassen, wird der Fluß bis zum Rand hin gespeichert; das Hinzufügen von DØUBLE ist möglich.

Wenn die Flüsse der n-ten Spalte von Zeile m_1 bis m_2 (multipliziert mit -1) auf FORTRAN-Karten gestanzt zur Verwendung als Eingabe von Randbedingungen bei Folge-rechnungen ausgegeben werden sollen (im Format 6X, 1P6E11.3):

CØLUMN n m_1 m_2

wenn die Flüsse der m-ten Zeile von Spalte n_1 bis n_2 (multipliziert mit -1) auf Karten gestanzt ausgegeben werden sollen:

RØW m_1 n_1 n_2

S.2.6 Wenn ein weiterer Fall gerechnet werden soll, folgt K 1.1;
sonst K 1.15

K 1.15 END

Ende der Eingabe

Weitere Möglichkeiten des Programms

Erneute Auswertung:

Die Flüsse stehen am Ende der Rechnung automatisch auf Einheit 10. Wurden sie als nicht-temporärer Data-Set gespeichert, so kann in einem späteren Job eine weitere Auswertung (bei der Berechnung von Strömen mit CURRENTS werden zusätzlich Daten von Einheit 11 benötigt) vorgenommen werden. Man braucht dazu nur die Anzahl der Spalten $ncol = i_{max} + 1$ und die Anzahl der Zeilen $nrow = j_{max} + 1$ (s. K 1.4). Die Flüsse müssen bei der späteren Auswertung auf Einheit 3 stehen.

Die folgende Eingabe ist erforderlich:

* Überschrift

ATTØW

GRØUPS ng

CØNSTANTS nmat nscat

(nur nötig, wenn Ströme berechnet werden sollen; Daten von Einheit 11 !)

MESH

(wie K 1.4, jedoch nicht erforderlich, wenn im folgenden die Berechnung von CURRENTS erfolgt. Dann werden diese Daten von Einheit 11 gelesen).

ØUTPUT nrØwncol

Es folgen K 1.11 bis K 1.15

Fortsetzen oder Verbessern einer Rechnung:

Wenn die vorangegangene Rechnung für die Gruppen $1,2,\dots,nr \leq ng$ erfolgte und für die Gruppen $1,2, \dots,nk \leq nr$ konvergent war, so können je nach nr und nk folgende Fälle behandelt werden:

- (a) $nk = nr < ng$: unter Verwendung der ausiterierten Flüsse der Gruppen $1,2,\dots,nr$ für die Streuquelle werden die Flüsse für die Gruppen $nr+1, nr+2,\dots,ng$ berechnet (alte ACCUR-Parameter)
- (b) $nk < nr \leq ng$: unter Verwendung der ausiterierten Flüsse der Gruppen $1,2,\dots,nk$ für die Streuquelle wird die Rechnung für die Gruppen $nk+1,\dots,nr$ fortgesetzt, dann werden, wenn $nr < ng$, die Flüsse für die Gruppen $nr+1,\dots,ng$ berechnet (alte ACCUR-Parameter);
- (c) $nk = nr = ng$: die Rechnung wird unter erhöhter Genauigkeit fortgesetzt (neue ACCUR-Parameter)

Die Flüsse der Gruppen $1,2,\dots,nr$ aus der vorangegangenen Rechnung sind auf Einheit 3 zu legen; auf Einheit 11 wird der in der vorangegangenen Rechnung zu reservierende Eingabeblock, der auch die Querschnitte enthält, erwartet.

Die folgende Eingabe ist zu machen:

```
* Überschrift
ATTØW
GRØUPS ng
CØNSTANTS nmat nscat
GEØMETRY ..
ACCUR a1 a2 a3
```

```
ØMEGA ω
CØUNT nk*0 mnk+1..mng
RESTART nrØw*ncØ1 nr
CØUNT m1 m2..mng
RESTART nrØw*ncØ1 ng
SØURCES 1,2,.. gs
END
```

} in den Fällen (a) und (b)

} im Fall (c)

alle Gruppen mit Quellen
oder K 1.11 usw.

Eine Rechnung kann weiterhin unter Verwendung der Flüsse aus einem Dump nach K 1.14 als Anfangswerte fortgesetzt werden, wobei das Ortsnetz verfeinert und ein Teilgebiet ausgewählt werden kann. Dazu ist eine reguläre ATTØW-Rechnung - gegebenenfalls für das Teilgebiet mit verfeinertem Ortsnetz - zu starten, die auf K 1.3

USEDUMP nu nu = Gruppe, bis zu der die Flüsse
 als Anfangswerte benutzt werden,
enthält. Der Dump wird auf Einheit 3 erwartet.

B2. Dateneingabe:

K 2.1 MATERIALS CRAM

S 2.1 Wenn Bibliotheksdaten benutzt werden sollen, folgt S 2.4; wenn Konstanten aus eingegebenen makroskopischen oder mikroskopischen Konstanten für Isotope oder Verbindungen berechnet werden sollen, folgt K 2.3; wenn Konstanten für Mischungen eingegeben werden, folgt K 2.2

K 2.2 Für jede Mischung:

n	Mischungsnummer
$D_{n,1} D_{n,2} \dots D_{n,ng}$	Diffusionskonstanten
$\Sigma_{n,1} \Sigma_{n,2} \dots \Sigma_{n,ng}$	Removalquerschnitte (Diffusion!)
$\Sigma_{n,1,2} \Sigma_{n,1,3} \dots \Sigma_{n,1,nscat+1}$	
$\Sigma_{n,2,3} \Sigma_{n,2,4} \dots \Sigma_{n,2,nscat+2}$	Streuquerschnitte
\vdots	
$\Sigma_{n,ng-2,ng-1} \Sigma_{n,ng-2,ng} \Sigma_{n,ng-1,ng}$	

S 2.2 Es folgt K 1.3

K 2.3 Für jedes Isotop oder jede Verbindung:

In n = Nummer des Isotopes oder der Verbindung fortlaufend

$D_{n,1} D_{n,2} \dots D_{n,ng}$
 $\Sigma_{n,1}^r \Sigma_{n,2}^r \dots \Sigma_{n,ng}^r$
 $\Sigma_{n,1,2} \dots \Sigma_{n,ng-1,ng}$

Für jede Mischung, die sich aus den Isotopen oder Verbindungen In zusammensetzt:

Mk k = Nummer der Mischung fortlaufend, $k \leq nmat$
I1 $m_{k,1}$ I2 $m_{k,2}$... In $m_{k,n}$ Das Isotop In braucht nur aufgeführt zu werden, wenn $m_{k,n} > 0$ ist.

Es ist $m_{k,n}$ = Teilchenzahlendichte
 $\times 10^{-24}$

Aus den eingegebenen Daten wird berechnet:

$$\bar{D}_{k,g} = \frac{1}{\sum_i m_{k,i} \frac{1}{D_{i,g}}} \quad \text{Diffusionskonstante in der g-ten Gruppe für die k-te Mischung}$$

$$\bar{\Sigma}_{k,g}^r = \sum_i m_{k,i} \Sigma_{i,g}^r \quad \text{Removalquerschnitt}$$

$$\bar{\Sigma}_{k,g,h} = \sum_i m_{k,i} \Sigma_{i,g,h} \quad \text{Streuquerschnitt}$$

Gibt man als $D_{i,g}$, $\Sigma_{i,g}^r$, $\Sigma_{i,g,h}$ mikroskopische Daten und für die $m_{k,i}$ die Teilchenzahlendichte des Isotops oder der Verbindung I_i in der Mischung M_k ein, so sind $\bar{D}_{k,g}$, $\bar{\Sigma}_{k,g}^r$ und $\bar{\Sigma}_{k,g,h}$ die entsprechenden makroskopischen Gruppenkonstanten der Mischung M_k .

Dieselben makroskopischen Gruppenkonstanten bekommt man, wenn man

- a) für $D_{i,g}$ etc. makroskopische Werte bei der Dichte $\rho = 1 \text{ g.cm}^{-3}$ und für die $m_{k,i}$ Partialdichten $\rho_{k,i}$ in g.cm^{-3} eingibt,
- b) für $D_{i,g}$ etc. makroskopische Werte bei der Dichte ρ der Mischung M_k und für die $m_{k,i}$ Gewichtsanteile angibt.

S 2.3 Es folgt K 1.3

S 2.4 Wenn Daten mit NUSYS bereitgestellt werden, folgt K 2.11;
 wenn Daten der KFKINR-Bibliothek benutzt werden sollen : K 2.8;
 wenn unmodifizierte RØACH-B-Daten benutzt werden sollen: K 2.4;
 wenn modifizierte RØACH-B-Daten verwendet werden sollen: S 2.6;
 wenn RØACH-B-Daten kondensiert werden sollen, folgt K 2.7

K 2.4 LIB RØACH

Für jede Mischung:

M_k

k = Nummer der Mischung

I1 $m_{k,1}$ I2 $m_{k,2}$..Ii $m_{k,i}$.. nur beteiligte Isotope brauchen
aufgeführt zu werden

Hier sind I1, I2, .. die Namen der Isotope in der RØACH-B-Biblio-
thek (s. Cl. oder gleich Ref.1), die mikroskopische Querschnitte
in barn enthält. Analog zu K 2.3 wird berechnet:

$$D_{k,g} = \frac{1}{3 \sum_i m_{k,i} \sigma_{i,tr}}$$

$\sigma_{i,tr}$ = mikroskopischer Transport-
querschnitt des Isotops Ii

$$\Sigma_{k,g}^r = \sum_i m_{k,i} \sigma_{i,g}^r$$

$$\Sigma_{k,g,h} = \sum_i m_{k,i} \sigma_{i,g,h}$$

Für $m_{k,i}$ ist die Teilchenzahldichte $\times 10^{-24}$ des Isotops Ii
in der Mischung Mk einzugeben.

S 2.5 Es folgt K 1.3

S 2.6 Wenn die neuen Daten Diffusionskonstanten enthalten, folgt K 2.5;
wenn die neuen Daten Transportquerschnitte enthalten: K 2.6

K 2.5 LIB RØACH

Für jedes abzuändernde Isotop:

Ii

i = Nummer des Isotops in der Bib-
liothek, i = 1, 2, .. , 40.

$D_{i,1}$ $D_{i,2}$... $D_{i,16}$

Alle Werte dieses Isotops müssen
(geändert oder ungeändert) aufge-
führt werden.

$\Sigma_{i,1}^r$ $\Sigma_{i,2}^r$... $\Sigma_{i,16}^r$

$\Sigma_{i,1,2}$... $\Sigma_{i,15,16}$

Für jede Mischung:

Mk

I1 $m_{k,1}$ I2 $m_{k,2}$ Ii $m_{k,i}$..

S 2.7 Es folgt K 1.3

K 2.6 Es kann nur ein Isotop abgeändert werden, es muß den Namen I40
tragen.

I40

$$\Sigma_{40,1}^{tr} \quad \Sigma_{40,2}^{tr} \quad \dots \quad \Sigma_{40,16}^{tr}$$

$$\Sigma_{40,1}^r \quad \dots$$

Es gilt das zu K 2.5 Gesagte.

$$\Sigma_{40,1,2} \quad \dots$$

S 2.8 Es folgt K 2.5 oder K 2.4

K 2.7 LIB RØACH

Für jeden Satz von Gewichten:

$$S_k \quad S_{k,1} \quad S_{k,2} \quad S_{k,3} \quad \dots \quad S_{k,16}$$

mit k = Nummer des Satzes, $k \leq 5$

Neue Gruppen:

$$N_1 N_2, N_3 N_4, N_5, N_6, N_7 N_8 \quad (\text{als Beispiel})$$

Es gilt:

1. neue Gruppe = N_1 -te bis N_2 -te
alte Gruppe

2. neue Gruppe = N_3 -te bis N_4 -te
alte Gruppe

3. neue Gruppe = N_5 -te alte Gruppe
usw.

Zu beachten: Fallen alte und neue Gruppe zusammen, ist nur ein Index, der Index der alten Gruppe, von Kommas eingeschlossen zu lochen.

Für jede Fischung:

Mk

k = Nummer der Mischung, $k \leq nmat$

$$I1 \quad m_{k,1} \quad I2 \quad m_{k,2} \quad \dots \quad S_j$$

Für jede Mischung wird am Ende durch S_j der Gewichtssatz angegeben.

Zu beachten: Die $S_{k,i}$ gehen bei der Kondensation ungeändert als Gewichte ein. Es ist z.B.

$$D_1^{neu} = S_{1,1} D_1^{alt} + S_{1,2} D_2^{alt} + S_{1,3} D_3^{alt},$$

wenn mit S1 kondensiert wird und die ersten drei Gruppen zusammengefaßt werden. Will man hier wie in NUSYS oder wie bei der Kondensation der KFKINR-Bibliothek verfahren, ist die Summe der Gewichte in jeder neuen Gruppe auf 1 zu normieren. Die Removalquerschnitte werden hier wie die Diffusionskonstanten kondensiert, also anders als in NUSYS (s.K 2.9)

S 2.9 Es folgt K 1.3

K 2.8 LIB KFKINR Die KFKINR-Bibliothek muß auf Einheit 15 bereitgestellt werden.

S 2.10 Wenn kondensiert werden soll, folgt K 2.9; andernfalls K 2.10

K 2.9 Für jedes Kondensationsspektrum:

$S_j \quad S_{j,1} \quad S_{j,2} \quad S_{j,3} \quad \dots \quad S_{j,26}$ mit $j =$ Nummer des 26-Gruppen-Kondensationsspektrums, $j \leq 5$

Dann

CG $n_1 \quad n_2 \quad n_3 \quad \dots \quad n_{ng}$ Hinter dem Schlüsselwort CG folgt die Breite der neuen Gruppen; die i -te neue Gruppe umfaßt n_i alte Gruppen; es muß $\sum_i n_i = 26$ sein.

Dann für jede Mischung:

Mk $k =$ Nummer der Mischung, $k \leq nmat$
I1 $m_{k,1}$ I2 $m_{k,2}$... Sj Hier sind I1, I2, ... die Namen der Isotope in der KFKINR-Bibliothek (s. C2); $m_{k,i}$ ist die Teilchenzahldichte $\times 10^{-24}$ Isotops Ii in der Mischung Mk; j ist die Nummer des Wichtungsspektrums, mit dem Mk kondensiert wird.

Es wird folgendermaßen kondensiert:

Seien I, J die Indizes der neuen Gruppen, g, h die Indizes der alten Gruppen. Ferner sei

$$\sum_{g(I)} = \text{Summe über alle alten Gruppen in der Gruppe I.}$$

Dann ist mit j als Index des zugeordneten Kondensationsspektrums und

$$w_h = S_{j,h} / \sum_{g(I)} S_{j,g} , \quad \text{wenn die I-te Gruppe die h-te Gruppe enthält,}$$

$$D_i = \sum_{g(I)} w_g D_g$$

$$\Sigma_I^r = \sum_{g(I)} \left[w_g \Sigma_g^r - \sum_{\substack{h(I) \\ h < g}} w_h \Sigma_{h \rightarrow g} \right]$$

$$\Sigma_{J \rightarrow I} = \sum_{g(I)} \sum_{h(J)} w_h \Sigma_{h \rightarrow g}$$

S 2.11 Es folgt K 1.3

K 2.10 Für jede Mischung:

M_k

I1 m_{k,1} I2 m_{k,2} ... SO

k = Nummer der Mischung, k ≤ nmat

Für die I_i und m_{k,i} gilt das zu K 2.9 Gesagte. Jede Mischung muß mit SO abgeschlossen werden

S 2.12 Es folgt K 1.3

K 2.11 LIB NUSYS

(s. anschließende Erläuterung)

S 2.13 Es folgt K 1.3

Erläuterung zur Datenerzeugung mit NUSYS:

Die Bereitstellung von Daten mit NUSYS erfolgt in zwei der ATTØW-Rechnung vorausgehenden Job-Steps.

1. Step: NUSYS-Eröffnung mit 00397;

Querschnittsberechnung mit 00446 für nmat Mischungen;

SIGMA-Block einschließlich DIFKØ auf LILI schreiben;

Kondensieren mit 00352 auf ng Gruppen (entfällt bei 26-Gruppen-Rechnungen);

SIGMA-Block mit 00451 auf Einheit 15 schreiben;

2. Step: SIGMA-Block mit dem Hauptprogramm NUSQU von Einheit 15

lesen, die Zahl der Einstreugruppen auf nscat reduzieren

und anschließend die Daten in ATTØW-Format wieder auf Einheit 15 schreiben. NUSQU benötigt im Format (3I10) die Eingabe

KØND = 1, wenn kondensiert wurde

= 0, wenn nicht kondensiert wurde

NSCAT = nscat = Maximalzahl von Einstreugruppen

(s. K 1.2)

NTP1 = Nummer der Speichereinheit, also hier 15

B3. Removal-Eingabe

K 3.1 REMØVAL

ENERGY BANDS n(m)k

Eingabe der Removalbänder durch
mit

n = obere Energiegrenze in MeV

m = Breite der Energiebänder in MeV

k = untere Energiegrenze in MeV

Man bekommt

$$(n-k)/m = nb \text{ Energiebänder,}$$

n-k muß ein ganzzahliges Vielfaches
von m sein.

Bei Verwendung der eingebauten
SABINE-Removalquerschnitte muß
18(1)0 eingegeben werden.

Es folgt

K 3.2 SHIELD RADIUS $\rho_1 \rho_2$
HEIGHT $h_1 h_2$

Sollen die Removalquellen in bestimm-
ten Punkten berechnet und ausge-
druckt werden (die Quellen werden
dann nicht gespeichert), so sind
diese Punkte einzugeben durch

SHIELD PØINTS GIVEN

R i $r_1 r_2 \dots r_i$

Z j $z_1 z_2 \dots z_j$

Es folgen die Core-Abmessungen:

CØRE RADIUS ρ HEIGHT $k_1 k_2$ mit

$$0 < \rho, \rho_1 \leq \rho \leq \rho_2,$$

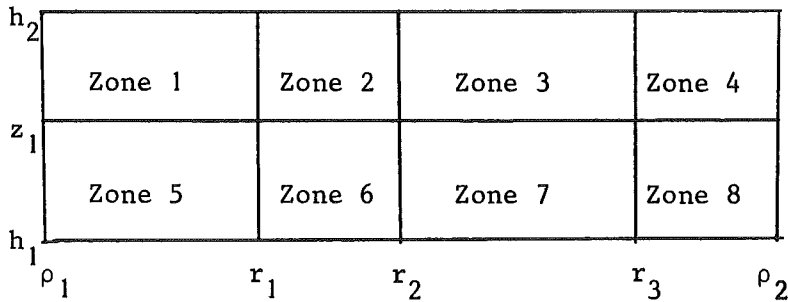
$$h_1 \leq k_1 < k_2 \leq h_2$$

PØINTS R n_r Z n_z T n_t

liefert die Anzahl der Punkte für die
Integration über das Core; n_t ist
die Anzahl der θ -Stützpunkte zwischen
0 und π .

ZONES R ρ_1 r_1 r_2 ... ρ_2
 Z h_1 z_1 z_2 ... h_2

beschreibt die Einteilung des Schildes einschließlich des zum Schild gerechneten Coreteiles in Zonen einheitlichen Materials. Diese Zonen werden von links nach rechts und von oben nach unten fortlaufend durchnummeriert:



K 3.3 COMPOSITION i_1 i_2
 i_n

Jeder dieser Zonen 1, 2, ..., n wird durch

eine Mischung i_k zugeordnet; es kann mehreren Zonen die gleiche Mischung zugeordnet werden.

Die Mischungen bzw. die Querschnitte für die Mischungen werden auf K 3.6 bzw. K 3.7 eingegeben.

Wenn die ersten ns Werte des eingebauten Spaltspektrums (für 18 Bänder je 1 MeV breit beginnend bei 18 MeV) überschrieben werden sollen, folgt

K 3.4 FS f_1 f_2 f_3 ... f_{ns}

K 3.5 POWER P

liefert die Leistung P des Cores; P ist in Watt anzugeben.

S 3.1 Wenn mit den eingebauten SABINE-Removalquerschnitten gerechnet werden soll, folgt K 3.6; bei Karteneingabe makroskopischer Removalquerschnitte folgt K 3.7

K 3.6 SABINE REMOVAL CROSS SECTIONS

Für jede Mischung auf K 3.3:

Mi

$$I1 n_{i,1} \quad I2 n_{i,2} \quad \dots$$

i = Nummer der Mischung

mit

Ik = Nummer des Isotops in der SABINE-Bibliothek (s. C3)

$n_{i,k}$ = Teilchenzahlendichte $\times 10^{-24}$ des Isotops Ik in der Mischung

Mi.

Es brauchen nur die in Mi enthaltenen Isotope aufgeführt zu werden.

Nach der letzten Mischung folgt

MO

Es folgt K 3.8

K 3.7 REMOVAL CROSS SECTIONS

Für jede Mischung auf K 3.3:

$$\begin{matrix} i \\ \text{rem} \end{matrix} \quad \begin{matrix} \text{rem} \\ \Sigma_{i,1} \end{matrix} \quad \begin{matrix} \text{rem} \\ \Sigma_{i,2} \end{matrix} \quad \dots \quad \begin{matrix} \text{rem} \\ \Sigma_{i,nb} \end{matrix}$$

i = Nummer der Mischung

mit $\Sigma_{i,k}^{\text{rem}}$ = makroskopischer Removalquerschnitt der Mischung i in der k-ten Gruppe, nb = Anzahl der Removalbänder (s. K 3.1)

Es folgt K 3.8

K 3.8

Das Programm rechnet mit einer Spaltquelle im Core der Form

$$Q(E, r, z) = FS(E)\varphi(r) \psi(z),$$

d.h. es verlangt neben der Separierbarkeit der Energieabhängigkeit die Separierbarkeit der Ortsabhängigkeit in radialer und axialer Richtung.

Die Normierung der Quelle erfolgt über die eingegebene Leistung.

$\varphi(r)$ und $\psi(z)$ werden punktweise eingegeben:

FISSION DENSITY DISTRIBUTION

RADIAL r_1 (δ_r) r_m φ_1 φ_2 .. φ_m

AXIAL z_1 (δ_z) z_n ψ_1 ψ_2 .. ψ_n

Die radialen Stützstellen sind

$r_1, r_1 + \delta_r, r_1 + 2\delta_r, \dots, r_m$

und es ist $\varphi_1 = \varphi(r_1), \varphi_2 = \varphi(r_1 + \delta_r), \dots \varphi_m = \varphi(r_m)$.

Entsprechend für die z-Richtung.

Es folgt

CONTINUE

Dann K 3.9

K 3.9

Eine folgende Diffusionsrechnung mit Removalquellen speist die Removalquelle des n-ten Removalbandes als zusätzliche ortsabhängige Quelle in die n-te Diffusionsgruppe ein. Es sind deshalb die Diffusionsgruppen geeignet zu wählen und die Removalbänder durch

COMBINE GROUPS

$n_1, n_2, n_3, n_4, n_5, n_6, n_7, \dots$

geeignet zusammenzufassen. Hierdurch werden die Removalquellen des n_1 -ten bis n_2 -ten Bandes, n_3 -ten bis n_4 -ten Bandes, des n_5 -ten Bandes usw. addiert und in die erste, zweite, dritte usw. Diffusionsgruppe zur Quelle hinzugefügt.

S 3.2 Wenn die Removalquellen auf Einheit 2 gebracht werden sollen, folgt

K 3.10; wenn die Removalquellen nur an den Punkten nach K 3.2

SHIELD POINTS .. berechnet und ausgedruckt werden sollen, folgt K 3.11

K 3.10

Für eine anschließende Diffusionsrechnung mit Removalquellen werden die Removalquellen für jede Zone durch Blynome in r und z dargestellt. Durch

ØRDER i * j

wird für alle Zonen der Grad dieser Polynome in r mit i und in z mit j festgelegt.

Durch

EXCEPT k₁ i₁ * j₁
 k₂ i₂ * j₂
 :
 :

wird diese Festlegung für die k_n-te Zone ersetzt durch i_n * j_n, k_n < k_{n+1}.

Soll für eine Zone kein Polynom gefunden werden (keine Removalquelle!), so ist 0 x 0 einzugeben.

0 x 1 und 1 x 0 sind nicht erlaubt.

Es folgt

CHEBY

CØNTINUE

S 3.3 Wenn sich eine Diffusionsrechnung anschließt, folgt K 1.1; wenn keine Diffusionsrechnung folgt, die Removalquellen jedoch auf Einheit 2 nach K 3.10 reserviert werden sollen, folgt K 3.12; wenn K 3.10 ignoriert werden soll und die Removalquellen nach K 3.2 SHIELD PØINTS .. berechnet und ausgedruckt werden sollen, folgt K 3.11

K 3.11 DUMMY

S 3.4 Wenn eine neue Rechnung begonnen werden soll, folgt K 1.1; sonst K 3.12

K 3.12 END

Ende der Eingabe

C. Datenbibliotheken:

C1. RØACH-B-Bibliothek:

Die eingebaute RØACH-B-Bibliothek enthält mikroskopische Wirkungsquerschnitte für 16 Gruppen. Die Gruppeneinteilung und die enthaltenen Isotope mit den zugehörigen Nummern, die hinter dem Schlüsselwort I zu lochen sind (s. z.B. K 2.4) sind dem Originalbericht¹⁾ zu entnehmen.

C2. KFKINR-Bibliothek:

Unter dem Namen KFKINR kann auf Einheit 15 eine 26-Gruppen-Datei bereitgestellt werden. Die z.Zt. benutzte Datei enthält Daten des KFKINR-Satzes²⁾ mit der Gruppeneinteilung des ABN-Satzes (s. z.B. KFK 770). Die folgende Tabelle enthält die Isotope dieser Bibliothek mit den zugehörigen Nummern. Isotope mit einem * sind unten erläutert.

Nummer	Isotop	Nummer	Isotop	Nummer	Isotop
1	AL270	21	K 390	41	SPP90 *
2	B 100	22	LI 60	42	SPU30 *
3	B 110	23	Li 70	43	SPU50 *
4	BE 90	24	MG 0	44	TA910
5	BI090	25	MØ960	45	TH320
6	C 120	26	N 140	46	TI480
7	CA 0	27	NA230	47	U 5A0 *
8	CD 0	28	NB930	48	U 5B0 *
9	CL350	29	NI590	49	U 8A0 *
10	CR520	30	Ø 160	50	U 8B0 *
11	CU640	31	PB070	51	U2330
12	C1/EO *	32	PU0b0 *-	52	U2340
13	D 20	33	PU2B0 *	53	U2350
14	EU 0	34	PU390	54	U2360
15	FE560	35	PU400	55	U2380
16	GD 0	36	PU410	56	V 510
17	H 10	37	PU420	57	W 0
18	HE 40	38	PU9A0 *	58	ZR910
19	HF 0	39	PU9B0 *-		
20	H1/EO *	40	SI280		

Erläuterung:

C1/EO	C ¹²		
H1/EO	H		mit einem "weichen" Spektrum gewichtet (vgl. ²⁾ s.20)
PUOBO	Pu ²⁴⁰	}	bei 2100 ⁰ K
PU2BO	Pu ²⁴²		
PU9BO	Pu ²³⁹		
U 5BO	U ²³⁵		
U 8BO	U ²³⁸		
PU9AO	Pu ²³⁹	}	bei 300 ⁰ K
U 5AO	U ²³⁵		
U 8AO	U ²³⁸		
SPP90	Pu ²³⁹ -	}	Spaltprodukte bei 900 ⁰ K
SPU30	U ²³³ -		
SPU50	U ²³⁵ -		

Die übrigen Pu- und U-Isotope sind auf 900⁰ K bezogen

Bei der Erstellung einer eigenen Datei (mit dem Namen KFKINR) ist zu beachten:

Die Daten müssen für jedes Isotop mit der Anweisung

WRITE(15) (D(I), I=1,26), (SR(I), I=1,26), (ST(I), I=1,125)

auf Einheit 15 gespeichert worden sein, um mit der Routine KFKINR durch den entsprechenden Lesebefehl gelesen werden zu können. Dabei ist

D(I) = mikroskopische Diffusionskonstante der I-ten Gruppe
 SR(I) = mikroskopischer Removalquerschnitt (Diffusion!) der I-ten Gruppe

ST(1) = $\sigma_{1 \rightarrow 2}$, ST(2) = $\sigma_{1 \rightarrow 3}$, ... , ST(5) = $\sigma_{1 \rightarrow 6}$,
 ST(6) = $\sigma_{2 \rightarrow 3}$, ST(7) = $\sigma_{2 \rightarrow 4}$, ... , ST(10) = $\sigma_{2 \rightarrow 7}$,

⋮

ST(101) = $\sigma_{21 \rightarrow 22}$, ST(102) = $\sigma_{21 \rightarrow 23}$, ... , ST(105) = $\sigma_{21 \rightarrow 26}$

ST(106) = $\sigma_{22 \rightarrow 23}$, ST(107) = $\sigma_{22 \rightarrow 24}$, ... , ST(109) = $\sigma_{22 \rightarrow 26}$,

ST(110) = 0.

ST(111) = $\sigma_{23 \rightarrow 24}$, ... , ST(113) = $\sigma_{23 \rightarrow 26}$, ST(114) = ST(115) = 0.

ST(116) = $\sigma_{24 \rightarrow 25}$, ST(117) = $\sigma_{24 \rightarrow 26}$, ST(118) = .. = ST(120) = 0.

ST(121) = $\sigma_{25 \rightarrow 26}$, ST(122) = = ST(125) = 0.

Die Isotope werden entsprechend ihrer Reihenfolge in der Datei identifiziert. Bei einer Mischung M1 I3 .5 I1 .2 z.B. werden aus dem dritten und ersten Satz der Datei makroskopische Querschnitte erzeugt.

C3. SABINE-Bibliothek (Removalquerschnitte):

Die in ATTØW-K eingebaute Removalquerschnittsbibliothek basiert auf experimentellen SABINE-Daten⁴⁾ von 1966. Da die Removalbänder in ATTØW-K und SABINE verschieden sind, mußten die SABINE-Querschnitte auf die ATTØW-Bänder umgerechnet werden.

Die SABINE-Querschnitte sind mit dem Spektrum einer Spaltneutronenquelle in Wasser gewichtet.

Tabelle: SABINE-Bandgrenzen und normiertes Wichtungsspektrum

Bandnummer i	Obere Bandgrenze E _i in MeV	Bruchteil χ_i von Spaltneutr. in Band i
1	18.	3.359 - 6
2	16.5	1.348 - 5
3	14.918	2.062 - 5
4	14.	5.080 - 5
5	13.	1.160 - 4
6	12. 11	2.630 - 4
7	11.	5.89 - 4
8	10.	1.310 - 3
9	9.	2.86 - 3
10	8.	6.15 - 3
11	7.	1.18 - 2
12	6.065	2.10 - 2
13	5.2	3.48 - 2
14	4.4	5.22 - 2
15	3.68	7.69 - 2
16	3.00	1.337 - 1
17	2.23	2.310 - 1
18	1.35	1.802 - 1
19	.821	1.147 - 1

Der Removalquerschnitt des i-ten Bandes ist

$$\sigma_i^{\text{rem}} = \int_{E_{i+1}}^{E_i} dE \sigma^{\text{rem}}(E) \chi(E) / \chi_i$$

Mit $\bar{E}_i = .5 (E_i + E_{i+1})$ und $\Delta_i E = E_i - E_{i+1}$ ist näherungsweise

$$\sigma^{\text{rem}}(\bar{E}_i) \chi(\bar{E}_i) = \sigma_i^{\text{rem}} \chi_i / \Delta_i E,$$

woraus durch quadratische Interpolation $\sigma^{\text{rem}}(\bar{E}'_I) \chi(\bar{E}'_I)$ gewonnen wurde. Gestrichene Größen mit Index I beziehen sich auf ATTØW-Removalbänder. Die ATTØW-Removalquerschnitte ergeben sich jetzt zu

$$\sigma_I^{\text{rem}} = \sigma^{\text{rem}}(\bar{E}'_I) \chi(\bar{E}'_I) \cdot \Delta'_I E / \chi'_I$$

Für die χ'_I werden die in ATTØW eingebauten Spaltspektrumswerte FS(I), I = 1, 2, ..., 18 genommen.

Bemerkung: Die in ATTØW realisierte Form des Removalkonzepts erlaubt nur eine grobe Korrektur der Diffusionsnäherung für große Eindringtiefen. Da außerdem i.a. die Removalquerschnitte nach einer Transportkontrollrechnung zu modifizieren sind, erfolgt die Umrechnung nach dem obigen Verfahren ausreichend genau.

Die Isotope in der SABINE-Bibliothek sind mit den zugehörigen Identifikationsnummern der folgenden Tabelle zu entnehmen. Die Bandgrenzen sind 18., 17., 16., ..., 1., 0. MeV mit einer konstanten Bandbreite von einem MeV.

Nummer	Element	Nummer	Element
1	Wasserstoff	12	Uran 238
2	Sauerstoff	13	Deuterium
3	Wasser	14	Silizium
4	Kohlenstoff	15	Phosphor
5	Lithium	16	Schwefel
6	Beryllium	17	Kalium
7	Bor	18	Calzium
8	Aluminium	19	Titan
9	Eisen	20	Vanadium
10	Blei	21	Chrom
11	Uran 235	22	Mangan

Nummer	Element	Nummer	Element
23	Kobalt	31	Cadmium
24	Nickel	32	Indium
25	Kupfer	33	Zinn
26	Zink	34	Barium
27	Natrium	35	Dysprosium
28	Magnesium	36	Wolfram
29	Zirkon	37	Gold
30	Molybdän		

D. Reduzierung des Platzbedarfs durch Overlay

Das Programm benötigt ohne Overlay-Struktur ca. 300K Bytes Kernspeicher. Benutzt man die untenstehende Overlay-Struktur, so reduziert sich der Platzbedarf auf ungefähr 230K.

Segment 1 (Rootsegment):

NØARG, ITDEC, INDEC, NEWDEC, FLDEC, WØRDEC, CHDEC, NEXTCH, QCHOUT, CVFN, CHTØWD, QSTOCH, QCHAR, KCHAR, QMØVE, QSHIFT, MESSAGE, INDEX, NEXCH, DBLDEC, DECTP, MAIN, SHIELD, Systemroutinen und CØMMØNs

Segment 2 :

REMØ, REMØIN, REMØQU, GPØINT, SETP, DENSIT, NØRMAL, SG, REMFUN, CHBY2

Segment 3:

SETUP, DATA, STØRE, MESH, NITER,

Segment 4:

CRASH, RASHCØ, PENDLE, MACRØX, STEP, RWTAPE, DLEN, FL

Segment 5:

CRAM, ISØTØP, RØACH, MIXES, SQUEEZE, ISØKFK, KFKNUS, KFKINR, VAL

Segment 6:

TESTIO

Segment 7:

SAUCE, BØUNDS

Segment 8:

CØEFS, SHUNT, NSEW, RRSCE, PØLY2, ACØEFS, XCØEFS, XACØEF

Segment 9:

ITERAT, SLØW, PRINT

Segment 10:

ØUTPUT, DUMPS, CURE1, PUNRØW, PUNCØL, PUNLAB, ØF, CURE2, PUW

E. Eingabebeispiel mit JCL-Karten für die IBM360/65

```
//INR238AT JOB (0238,101,P6M5A),WIESE,MSGLEVEL=(1,1),CLASS=A,
// REGION=300K,TIME=3
/*FORMAT PR,DDNAME=FT06F001,OVFL=ON
// EXEC FHG,LIB=NUSYS,NAME=ATTOWK
//G.FT01F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT02F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT03F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT04F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT08F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT09F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT10F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.FT11F001 DD UNIT=SYSDA,SPACE=(TRK,20),
// DCB=(BLKSIZE=2298,RECFM=VBS)
//G.SYSIN DD *
* CRAM COMPARISON CASE 23.2.73
ATTOW
GROUPS 6
CONSTANTS 3 1
OMEGA 1.5
ACCUR 5E-2 5E-3 5E-4
MATERIALS CRAM
1 1.09 .972 .942 .926 .787 .778
.03151 .04758 .05376 .02006 .0579 .000236
.03151 .04758 .05376 .02006 .05751
2 1.846 .970 .995 .913 .909 .918
.00136 .002765 .00307 .001195 .001107 .0056
.001359 .002765 .003066 .001178 .0004285
3 5.694 4.852 4.184 3.105 5.939 4.725
.00332 .004715 .005935 .002132 .002391 .01451
.00332 .004715 .005935 .002132 .0005585
MESH MR 0=0 2=4 23=56.5 MZ 0=0 10=30 14=38 24=68 28=76 38=106
ZONE 1 (0 10 5 28) 2 (0 14 2 24) 3
SOURCES IN GROUP 1 BY MATERIALS 3 .9803 GRUP 2 BY MATERIALS 3 .0197
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *39 1E9
BZ LOW ONEZERO 3 *24 1E9 BZ HIGH ONEZERO 3 *24 1E9
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *39 1E9
BZ LOW ONEZERO 3 *24 1E9 BZ HIGH ONEZERO 3 *24 1E9
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *39 1E9
BZ LOW ONEZERO 3 *24 1E9 BZ HIGH ONEZERO 3 *24 1E9
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *39 1E9
BZ LOW ONEZERO 3 *24 1E9 BZ HIGH ONEZERO 3 *24 1E9
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *39 1E9
BZ LOW ONEZERO 3 *24 1E9 BZ HIGH ONEZERO 3 *24 1E9
SOLVE END
/*
//
```

F. Korrigiertes Rem.-Diff. Rechenbeispiel

Die in Ref. 3, 2.2.4 a) angegebene Korrektur des Unterprogramms RRSCE, wonach auf vier Karten (Seite H58, Zeile 3-6) I durch IIA und J durch JJA zu ersetzen ist, führt zu Änderungen in den Resultaten bei Rem.-Diff.-Rechnungen. Reine Removalrechnungen und reine Diffusionsrechnungen bleiben unbeeinflusst. Bei dem Sample Problem des Originalreports¹⁾ unterscheiden sich die korrigierten Flüsse von den unkorrigierten um einen Faktor ≈ 1.9 . Im folgenden ist ein Auszug aus den Ergebnissen des korrigierten Rechenbeispiels gegeben. Weitere ausführliche Beispiele mit Eingabelisten sind in Ref.6 zu finden.

REMOVAL SOURCES FOR GROUP 2 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 1.0468E 07 3.5520E 07
17.472 1.8793E 07 5.9483E 07

REMOVAL SOURCES FOR GROUP 2 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 2.5833E 04 4.6528E 06
17.472 4.4168E 04 8.3436E 06

REMOVAL SOURCES FOR GROUP 2 , 43.88 LT R LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 9.4902E 02 1.0884E 03
17.472 1.5620E 03 1.8299E 03

REMOVAL SOURCES FOR GROUP 3 , 0 LT R LT 23.56 , 15.24 LT Z LT 30.48

21.982 11.780 1.578
29.459 2.7280E 08 3.1952E 08 2.1729E 08
22.860 4.5293E 08 5.6159E 08 3.9518E 08
16.261 5.5620E 08 6.7904E 08 4.7369E 08

REMOVAL SOURCES FOR GROUP 3 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 3.4166E 07 1.0481E 08
17.472 6.1001E 07 1.7774E 08

REMOVAL SOURCES FOR GROUP 3 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 9.5058E 04 1.7369E 07
17.472 1.6124E 05 3.0909E 07

REMOVAL SOURCES FOR GROUP 3 , 43.88 LT R LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 3.1512E 03 3.5811E 03
17.472 5.1439E 03 5.9733E 03

REMOVAL SOURCES FOR GROUP 4 , 0 LT R LT 23.56 , 15.24 LT Z LT 30.48

21.982 11.780 1.578
29.459 4.9183E 09 6.2434E 09 4.3632E 09
22.860 8.3026E 09 1.1038E 10 7.9257E 09
16.261 1.0120E 10 1.3286E 10 9.4484E 09

REMOVAL SOURCES FOR GROUP 4 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 7.2251E 08 1.8770E 09
17.472 1.2719E 09 3.2160E 09

REMOVAL SOURCES FOR GROUP 4 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 1.0250E 07 4.6600E 08
17.472 1.6472E 07 8.0833E 08

REMOVAL SOURCES FOR GROUP 4 , 43.88 LT R LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 4.8696E 05 5.4429E 05
17.472 7.4772E 05 8.5627E 05

REMOVAL SOURCES FOR GROUP 5 , 0 LT R LT 23.56 , 15.24 LT Z LT 30.48

21.982 11.780 1.578
29.459 6.9606E 09 9.0063E 09 6.3580E 09
22.860 1.1787E 10 1.5928E 10 1.1527E 10
16.261 1.4341E 10 1.9150E 10 1.3725E 10

REMOVAL SOURCES FOR GROUP 5 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 1.0587E 09 2.6031E 09
17.472 1.8534E 09 4.4664E 09

REMOVAL SOURCES FOR GROUP 5 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 2.1388E 07 7.3541E 08
17.472 3.3878E 07 1.2647E 09

REMOVAL SOURCES FOR GROUP 5 , 43.88 LT R LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 1.0334E 06 1.1479E 06
17.472 1.5623E 06 1.7787E 06

REMOVAL SOURCES FOR GROUP 6 , 0 LT R LT 23.56 , 15.24 LT Z LT 30.48

21.982 11.780 1.578
29.459 5.6553E 11 6.4555E 11 4.4099E 11
22.860 9.3155E 11 1.1294E 12 7.9847E 11
16.261 1.1471E 12 1.3676E 12 9.5941E 11

REMOVAL SOURCES FOR GROUP 6 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 6.8336E 10 2.1369E 11
17.472 1.2192E 11 3.6028E 11

REMOVAL SOURCES FOR GROUP 6 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 3.8076E 08 3.4885E 10
17.472 6.1527E 08 6.1802E 10

REMOVAL SOURCES FOR GROUP 6 , 43.88 LT P LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 1.5367E 07 1.7198E 07
17.472 2.3618E 07 2.6947E 07

REMOVAL SOURCES FOR GROUP 7 , 0 LT P LT 23.56 , 15.24 LT Z LT 30.48

21.982 11.780 1.578
29.459 2.0525E 12 2.5726E 12 1.7872E 12
22.860 3.4570E 12 4.5464E 12 3.2495E 12
16.261 4.2186E 12 5.4760E 12 3.8768E 12

REMOVAL SOURCES FOR GROUP 7 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 2.9417E 11 7.9013E 11
17.472 5.1949E 11 1.3522E 12

REMOVAL SOURCES FOR GROUP 7 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 3.3587E 09 1.8099E 11
17.472 5.4455E 09 3.1553E 11

REMOVAL SOURCES FOR GROUP 7 , 43.88 LT R LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 1.5548E 08 1.7434E 08
17.472 2.4104E 08 2.7685E 08

REMOVAL SOURCES FOR GROUP 8 , 0 LT R LT 23.56 , 15.24 LT Z LT 30.48

21.982 11.780 1.578
29.459 2.2750E 13 2.9602E 13 2.0973E 13
22.860 3.8553E 13 5.2346E 13 3.7989E 13
16.261 4.6883E 13 6.2916E 13 4.5220E 13

REMOVAL SOURCES FOR GROUP 8 , 23.56 LT R LT 28.64 , 15.24 LT Z LT 30.48

27.896 24.304
28.248 3.4630E 12 8.2297E 12
17.472 6.0441E 12 1.4119E 13

REMOVAL SOURCES FOR GROUP 8 , 28.64 LT R LT 43.88 , 15.24 LT Z LT 30.48

41.648 30.872
28.248 8.8144E 10 2.5333E 12
17.472 1.3811E 11 4.3340E 12

REMOVAL SOURCES FOR GROUP 8 , 43.88 LT P LT 46.42 , 15.24 LT Z LT 30.48

46.048 44.252
28.248 4.0361E 09 4.4535E 09
17.472 6.0325E 09 6.8246E 09

CONTINUE

===== INPUT-CARD =====

TIME = 1.506 MINUTES

```

*      SAMPL-PROBLEM NO. 2      DIFFUSION PART      ===== TITLE =====
ATTOW      ===== INPUT-CARD =====
GROUPS 8      ===== INPUT-CARD =====
CONSTANTS 4 1      ===== INPUT-CARD =====
REMOVAL SOURCES      ===== INPUT-CARD =====
ACCUR .01 .001 .00005      ===== INPUT-CARD =====
MATERIALS GRAM      ===== INPUT-CARD =====
1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 .7 .6 .4 .2 .1 .01 .01      ===== INPUT-CARD =====
2 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 .7 .6 .4 .2 .1 .01 .01      ===== INPUT-CARD =====
3 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 .7 .6 .4 .2 .1 .01 .01      ===== INPUT-CARD =====
4 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 .7 .6 .4 .2 .1 .01 .01      ===== INPUT-CARD =====
MESH MR 0=.0001 8=23.56 10=28.64 15=43.88 17=46.4199 MZ 0=15.2401 5=
30.4799      ===== INPUT-CARD =====
ZONE 1 (8 0 10 5)2(10 0 15 5)3(15 0 17 5)4      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====

```

RMESH

```

0.000    2.945    5.890    8.835    11.780    14.725    17.670    20.615    23.560    26.100    28.640    31.688    34.736
37.784    40.832    43.880    45.150    46.420

```

ZMESH

```

30.480    27.432    24.384    21.336    18.288    15.240

```

ZONES

```

1 1 1 1 1 1 1 1 1 2 2 3 3 3 3 3 4 4
1 1 1 1 1 1 1 1 1 2 2 3 3 3 3 3 4 4
1 1 1 1 1 1 1 1 1 2 2 3 3 3 3 3 4 4
1 1 1 1 1 1 1 1 1 2 2 3 3 3 3 3 4 4
1 1 1 1 1 1 1 1 1 2 2 3 3 3 3 3 4 4

```

```

BOUNDS BZ LOW TWOZERO 2 3 ALL BZ HIGH ONEZERO 3 *18 .871      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====
BOUNDS BZ LOW TWOZERO 2 3 ALL BZ HIGH ONEZERO 3 *18 .871      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====
BOUNDS BZ LOW TWOZERO 2 3 ALL BZ HIGH ONEZERO 3 *18 .871      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====
BOUNDS BZ LOW TWOZERO 2 3 ALL BZ HIGH ONEZERO 3 *18 .871      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====
BOUNDS BZ LOW TWOZERO 2 3 ALL BZ HIGH ONEZERO 3 *18 .871      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====
BOUNDS BZ LOW TWOZERO 2 3 ALL BZ HIGH ONEZERO 3 *18 .871      ===== INPUT-CARD =====
BOUNDS BR LOW TWOZERO 2 3 ALL BR HIGH ONEZERO 3 *6 1      ===== INPUT-CARD =====
SOLVE      ===== INPUT-CARD =====
END      ===== INPUT-CARD =====

```


RESIDUAL	LAMBDA	OMEGA	DIFF	PRODIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....				
0.2137242E 11	0.9280525E 00	1.000000000	0.60377514E 09	0.8941892E 00	1	1	1	0	
0.1972432E 11	0.9228864E 00	1.000000000	0.5321188E 09	0.1000000E 01	1	1	1	0	
0.1812182E 11	0.9187554E 00	1.000000000	0.4794391E 09	0.6739506E 00	1	1	1	0	
0.1650720E 11	0.9109017E 00	1.000000000	0.4318436E 09	0.5096307E 00	1	1	1	0	
0.1491064E 11	0.9032813E 00	1.262778282	0.3871798E 09	0.4097819E 00	1	1	1	0	
0.2074463E 11	0.1391263E 01	1.262778282	0.6570051E 09	0.3892471E 00	1	1	1	0	
0.1813393E 11	0.8741503E 00	1.262778282	0.5225797E 09	0.3230238E 00	1	1	1	0	
0.1551287E 11	0.8554608E 00	1.262778282	0.4019200E 09	0.2721378E 00	1	1	1	0	
0.1309712E 11	0.8442749E 00	1.262778282	0.3196887E 09	0.2235611E 00	1	1	1	0	
0.1096867E 11	0.8374872E 00	1.397206306	0.2569257E 09	0.1820993E 00	1	1	1	0	
0.1125125E 11	0.1025762E 01	1.397206306	0.2837668E 09	0.1794626E 00	1	1	1	0	
0.8965878E 10	0.7968784E 00	1.397206306	0.2241331E 09	0.1365052E 00	1	1	1	0	
0.7063572E 10	0.7878283E 00	1.397206306	0.1656340E 09	0.1082047E 00	1	1	1	0	
0.5481865E 10	0.7760754E 00	1.397206306	0.1199964E 09	0.8389950E-01	1	1	1	0	
0.4214874E 10	0.7688760E 00	1.463856697	0.8856371E 08	0.6502157E-01	1	1	1	0	
0.3513726E 10	0.8336490E 00	1.463856697	0.7189709E 08	0.6206948E-01	1	1	1	0	
0.2521524E 10	0.7176211E 00	1.463856697	0.5559091E 08	0.3880960E-01	1	1	1	0	
0.1840863E 10	0.7300594E 00	1.463856697	0.4045210E 08	0.2988058E-01	1	1	1	0	
0.1313331E 10	0.7134326E 00	1.463856697	0.2743501E 08	0.2156568E-01	11	1	1	0	
0.9271910E 09	0.7059840E 00	1.495955467	0.1886208E 08	0.1569927E-01	13	1	1	0	
0.6649503E 09	0.7171664E 00	1.495955467	0.1335706E 08	0.1358140E-01	13	1	1	0	
0.4328233E 09	0.6509107E 00	1.495955467	0.8884224E 07	0.7212222E-02	18	1	1	0	
0.2930381E 09	0.6770386E 00	1.495955467	0.6258688E 07	0.5406320E-02	18	1	1	0	
0.1916472E 09	0.6540011E 00	1.495955467	0.3936256E 07	0.3541589E-02	18	1	1	0	
0.1241418E 09	0.6477622E 00	1.509905815	0.2510848E 07	0.2392471E-02	18	10	1	0	
0.7923341E 08	0.6382491E 00	1.509905815	0.1601536E 07	0.1785040E-02	18	12	1	0	
0.4759226E 08	0.6006590E 00	1.509905815	0.9830400E 06	0.8771420E-03	18	18	1	0	
0.2980488E 08	0.6262548E 00	1.509905815	0.6225920E 06	0.6294847E-03	18	18	1	0	
0.1790770E 08	0.6008310E 00	1.509905815	0.3727360E 06	0.3741384E-03	18	18	1	2	
0.1070117E 08	0.5975737E 00	1.514961243	0.2129920E 06	0.2340674E-03	18	18	1	4	
0.6295149E 07	0.5882674E 00	1.514961243	0.1269760E 06	0.1511574E-03	18	18	1	5	
0.3641944E 07	0.5785318E 00	1.514961243	0.7782400E 05	0.7176399E-04	18	18	13	13	
0.2140384E 07	0.5877037E 00	1.514961243	0.4915200E 05	0.5197525E-04	18	18	17	16	
0.1191473E 07	0.5566632E 00	1.514961243	0.2867200E 05	0.2646446E-04	18	18	18	17	

FLUXES IN GROUP 1 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	8.116E 08	8.176E 08	8.222E 08	8.183E 08	8.011E 08	7.666E 08	7.115E 08	6.320E 08	5.216E 08	3.829E 08	2.793E 08
4 27.43	2.860E 09	2.873E 09	2.879E 09	2.856E 09	2.787E 09	2.657E 09	2.454E 09	2.167E 09	1.780E 09	1.346E 09	9.988E 08
3 24.38	4.571E 09	4.586E 09	4.587E 09	4.539E 09	4.418E 09	4.201E 09	3.871E 09	3.410E 09	2.802E 09	2.147E 09	1.611E 09
2 21.34	5.860E 09	5.875E 09	5.869E 09	5.801E 09	5.639E 09	5.355E 09	4.928E 09	4.338E 09	3.569E 09	2.753E 09	2.078E 09
1 18.29	6.664E 09	6.680E 09	6.669E 09	6.589E 09	6.401E 09	6.077E 09	5.591E 09	4.923E 09	4.055E 09	3.135E 09	2.373E 09
0 15.24	6.941E 09	6.957E 09	6.946E 09	6.861E 09	6.666E 09	6.329E 09	5.825E 09	5.131E 09	4.229E 09	3.270E 09	2.475E 09

FLUXES IN GROUP 1 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	1.956E 08	1.353E 08	9.010E 07	5.503E 07	2.701E 07	1.696E 07	7.393E 06
4 27.43	7.017E 08	4.859E 08	3.239E 08	1.984E 08	9.795E 07	6.139E 07	2.675E 07
3 24.38	1.138E 09	7.905E 08	5.279E 08	3.240E 08	1.604E 08	1.005E 08	4.380E 07
2 21.34	1.474E 09	1.026E 09	6.861E 08	4.215E 08	2.090E 08	1.310E 08	5.707E 07
1 18.29	1.686E 09	1.175E 09	7.864E 08	4.834E 08	2.397E 08	1.503E 08	6.548E 07
0 15.24	1.759E 09	1.226E 09	8.210E 08	5.047E 08	2.502E 08	1.569E 08	6.835E 07

FLUXES FOR GROUP 1 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PRODIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
0.1336919E 13	0.9293611E 00	1.000000000	0.3773504E 11	0.4989000E 00	1	1	1	0
0.1231789E 13	0.9213645E 00	1.000000000	0.3295373E 11	0.3550572E 00	1	1	1	0
0.1128598E 13	0.9162264E 00	1.000000000	0.2908376E 11	0.2726371E 00	1	1	1	0
0.1025066E 13	0.9082648E 00	1.000000000	0.2577839E 11	0.2359405E 00	1	1	1	0
0.9236439E 12	0.9010580E 00	1.260714531	0.2288583E 11	0.1963413E 00	1	1	1	0
0.1278375E 13	0.1384055E 01	1.260714531	0.3837670E 11	0.1965139E 00	1	1	1	0
0.1115975E 13	0.8729642E 00	1.260714531	0.3047817E 11	0.1641354E 00	1	1	1	0
0.9548904E 12	0.8556554E 00	1.260714531	0.2350036E 11	0.1399071E 00	1	1	1	0
0.8072975E 12	0.8454347E 00	1.260714531	0.1882371E 11	0.1163553E 00	1	1	1	0
0.6774865E 12	0.8392029E 00	1.396690369	0.1525658E 11	0.9628880E-01	1	1	1	0
0.6985678E 12	0.1031116E 01	1.396690369	0.1711676E 11	0.9923631E-01	1	1	1	0
0.5585734E 12	0.7995979E 00	1.396690369	0.1359236E 11	0.7506019E-01	1	1	1	0
0.4413127E 12	0.7900712E 00	1.396690369	0.1009228E 11	0.6085235E-01	1	1	1	0
0.3433683E 12	0.7780612E 00	1.396690369	0.7385514E 10	0.4783964E-01	1	1	1	0
0.2645836E 12	0.7705534E 00	1.464076996	0.5493490E 10	0.3752649E-01	1	1	1	0
0.2212600E 12	0.8362571E 00	1.464076996	0.4502061E 10	0.3662717E-01	1	1	1	0
0.1590293E 12	0.7187443E 00	1.464076996	0.3497066E 10	0.2263778E-01	1	1	1	0
0.1162004E 12	0.7306852E 00	1.464076996	0.2547384E 10	0.1767123E-01	1	1	1	0
0.8294865E 11	0.7138415E 00	1.464076996	0.1731133E 10	0.1279366E-01	12	1	1	0
0.5857674E 11	0.7061808E 00	1.496180534	0.1192165E 10	0.9350359E-02	18	1	1	0
0.4200638E 11	0.7171170E 00	1.496180534	0.8442348E 09	0.8141220E-02	18	1	1	0
0.2732003E 11	0.6503780E 00	1.496180534	0.5611192E 09	0.4297256E-02	18	1	1	0
0.1848412E 11	0.6765776E 00	1.496180534	0.3946578E 09	0.3243566E-02	18	1	1	0
0.1208057E 11	0.6535651E 00	1.496180534	0.2481848E 09	0.2122223E-02	18	1	1	0
0.7822733E 10	0.6475464E 00	1.510059357	0.1585971E 09	0.1436949E-02	18	10	1	0
0.4981727E 10	0.6368265E 00	1.510059357	0.1004667E 09	0.1069069E-02	18	15	1	0
0.2986011E 10	0.5993927E 00	1.510059357	0.6160384E 08	0.5294681E-03	18	18	1	0
0.1876538E 10	0.6284429E 00	1.510059357	0.3905946E 08	0.3746748E-03	18	18	1	0
0.1122476E 10	0.5981632E 00	1.510059357	0.2313421E 08	0.2249479E-03	18	18	1	2
0.6704686E 09	0.5973123E 00	1.515069962	0.1363149E 08	0.1410246E-03	18	18	1	4
0.3967780E 09	0.5917920E 00	1.515069962	0.7864320E 07	0.8732080E-04	18	18	1	5
0.2285773E 09	0.5760835E 00	1.515069962	0.4718592E 07	0.4661083E-04	18	18	18	17

FLUXES IN GROUP 2 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	4.748E 10	4.724E 10	4.647E 10	4.508E 10	4.302E 10	4.026E 10	3.681E 10	3.274E 10	2.818E 10	2.400E 10	2.002E 10
4 27.43	1.712E 11	1.703E 11	1.674E 11	1.624E 11	1.549E 11	1.449E 11	1.325E 11	1.178E 11	1.014E 11	8.652E 10	7.226E 10
3 24.38	2.793E 11	2.778E 11	2.731E 11	2.648E 11	2.525E 11	2.362E 11	2.158E 11	1.919E 11	1.653E 11	1.412E 11	1.181E 11
2 21.34	3.631E 11	3.611E 11	3.549E 11	3.440E 11	3.280E 11	3.067E 11	2.803E 11	2.492E 11	2.147E 11	1.837E 11	1.537E 11
1 18.29	4.161E 11	4.138E 11	4.067E 11	3.941E 11	3.757E 11	3.513E 11	3.210E 11	2.855E 11	2.460E 11	2.105E 11	1.763E 11
0 15.24	4.342E 11	4.318E 11	4.244E 11	4.113E 11	3.921E 11	3.666E 11	3.350E 11	2.979E 11	2.568E 11	2.198E 11	1.840E 11

FLUXES IN GROUP 2 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	1.569E 10	1.184E 10	8.457E 09	5.473E 09	2.811E 09	1.775E 09	7.753E 08
4 27.43	5.663E 10	4.275E 10	3.054E 10	1.977E 10	1.015E 10	6.412E 09	2.801E 09
3 24.38	9.261E 10	6.996E 10	4.999E 10	3.236E 10	1.663E 10	1.050E 10	4.587E 09
2 21.34	1.206E 11	9.117E 10	6.517E 10	4.220E 10	2.168E 10	1.370E 10	5.981E 09
1 18.29	1.384E 11	1.046E 11	7.481E 10	4.844E 10	2.490E 10	1.572E 10	6.868E 09
0 15.24	1.445E 11	1.093E 11	7.812E 10	5.059E 10	2.600E 10	1.642E 10	7.172E 09

FLUXES FOR GROUP 2 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PRODIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
0.6794902E 14	0.5289210E 00	1.000000000	0.1858982E 13	0.4847443E 00	1	1	1	0
0.6258036E 14	0.9209859E 00	1.000000000	0.1611828E 13	0.3385519E 00	1	1	1	0
0.5731581E 14	0.9159353E 00	1.000000000	0.1417365E 13	0.2548605E 00	1	1	1	0
0.5205210E 14	0.9080957E 00	1.000000000	0.1254674E 13	0.2168669E 00	1	1	1	0
0.4690181E 14	0.9010552E 00	1.260711670	0.1114230E 13	0.1768829E 00	1	1	1	0
0.6492856E 14	0.1384351E 01	1.260711670	0.1872775E 13	0.1746353E 00	1	1	1	0
0.5671587E 14	0.8735117E 00	1.260711670	0.1490571E 13	0.1430638E 00	1	1	1	0
0.4857583E 14	0.8564768E 00	1.260711670	0.1155581E 13	0.1200106E 00	1	1	1	0
0.4111515E 14	0.8464117E 00	1.260711670	0.9311135E 12	0.9847450E-01	1	1	1	0
0.3454600E 14	0.8402255E 00	1.397249222	0.7589708E 12	0.8068776E-01	1	1	1	0
0.3569755E 14	0.1033334E 01	1.397249222	0.8587754E 12	0.8323914E-01	1	1	1	0
0.2858420E 14	0.8007327E 00	1.397249222	0.6841916E 12	0.6197131E-01	1	1	1	0
0.2260886E 14	0.7909565E 00	1.397249222	0.5094307E 12	0.5014068E-01	1	1	1	0
0.1760564E 14	0.7787054E 00	1.397249222	0.3747569E 12	0.3928041E-01	1	1	1	0
0.1357387E 14	0.7709954E 00	1.464652061	0.2798450E 12	0.3075081E-01	1	1	1	0
0.1135446E 14	0.8364941E 00	1.464652061	0.2301498E 12	0.3013015E-01	1	1	1	0
0.8160003E 13	0.7186605E 00	1.464652061	0.1790297E 12	0.1838166E-01	1	1	1	0
0.5961131E 13	0.7305306E 00	1.464652061	0.1304544E 12	0.1439947E-01	1	1	1	0
0.4253419E 13	0.7135254E 00	1.464652061	0.8873470E 11	0.1040381E-01	14	1	1	0
0.3001575E 13	0.7056853E 00	1.496511459	0.6105229E 11	0.7602870E-02	18	1	1	0
0.2150032E 13	0.7163011E 00	1.496511459	0.4321811E 11	0.6625891E-02	18	1	1	0
0.1396670E 13	0.6496042E 00	1.496511459	0.2870582E 11	0.3475428E-02	18	1	1	0
0.9445226E 12	0.6762677E 00	1.496511459	0.2018299E 11	0.2631187E-02	18	1	1	0
0.6165643E 12	0.6527787E 00	1.496511459	0.1268358E 11	0.1718879E-02	18	1	1	0
0.3986954E 12	0.6466404E 00	1.510185242	0.8069841E 10	0.1162469E-02	18	10	1	0
0.2531945E 12	0.6350574E 00	1.510185242	0.5133828E 10	0.8659959E-03	18	18	1	0
0.1516255E 12	0.5988501E 00	1.510185242	0.3103785E 10	0.4253387E-03	18	18	1	0
0.9486264E 11	0.6256376E 00	1.510185242	0.1979711E 10	0.3058314E-03	18	18	1	0
0.5655088E 11	0.5961343E 00	1.510185242	0.1185939E 10	0.1810789E-03	18	18	1	2
0.3367363E 11	0.5954573E 00	1.514983177	0.6752829E 09	0.1132488E-03	18	18	1	4
0.1993984E 11	0.5921499E 00	1.514983177	0.3974103E 09	0.7224083E-04	18	18	1	5
0.1146598E 11	0.5752252E 00	1.514983177	0.2348810E 09	0.3653765E-04	18	18	18	17

FLUXES IN GROUP 3 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	2.327E 12	2.312E 12	2.266E 12	2.191E 12	2.086E 12	1.954E 12	1.798E 12	1.623E 12	1.433E 12	1.263E 12	1.092E 12
4 27.43	8.403E 12	8.348E 12	8.184E 12	7.911E 12	7.533E 12	7.057E 12	6.495E 12	5.861E 12	5.176E 12	4.563E 12	3.946E 12
3 24.38	1.376E 13	1.367E 13	1.340E 13	1.295E 13	1.233E 13	1.155E 13	1.063E 13	9.594E 12	8.473E 12	7.471E 12	6.461E 12
2 21.34	1.794E 13	1.782E 13	1.747E 13	1.689E 13	1.608E 13	1.506E 13	1.386E 13	1.251E 13	1.105E 13	9.742E 12	8.427E 12
1 18.29	2.059E 13	2.046E 13	2.005E 13	1.938E 13	1.846E 13	1.729E 13	1.591E 13	1.436E 13	1.268E 13	1.119E 13	9.676E 12
0 15.24	2.150E 13	2.136E 13	2.094E 13	2.024E 13	1.927E 13	1.805E 13	1.662E 13	1.500E 13	1.325E 13	1.168E 13	1.010E 13

FLUXES IN GROUP 3 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	8.905E 11	6.963E 11	5.121E 11	3.387E 11	1.763E 11	1.117E 11	4.884E 10
4 27.43	3.217E 12	2.515E 12	1.850E 12	1.224E 12	6.368E 11	4.035E 11	1.764E 11
3 24.38	5.268E 12	4.120E 12	3.030E 12	2.004E 12	1.043E 12	6.610E 11	2.890E 11
2 21.34	6.871E 12	5.373E 12	3.952E 12	2.615E 12	1.361E 12	8.622E 11	3.770E 11
1 18.29	7.890E 12	6.171E 12	4.539E 12	3.003E 12	1.563E 12	9.902E 11	4.329E 11
0 15.24	8.239E 12	6.444E 12	4.740E 12	3.136E 12	1.632E 12	1.034E 12	4.521E 11

FLUXES FOR GROUP 3 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PRODIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
0.2302061E 16	0.9290846E 00	1.000000000	0.6100387E 14	0.4803753E 00	1	1	1	0
0.2120782E 16	0.9212533E 00	1.000000000	0.5293225E 14	0.3331631E 00	1	1	1	0
0.1943200E 16	0.9162658E 00	1.000000000	0.4661202E 14	0.2490517E 00	1	1	1	0
0.1765327E 16	0.9084639E 00	1.000000000	0.4133825E 14	0.2106500E 00	1	1	1	0
0.1591365E 16	0.9014564E 00	1.261081696	0.3678720E 14	0.1705964E 00	1	1	1	0
0.2205440E 16	0.1385879E 01	1.261081696	0.6209019E 14	0.1676078E 00	1	1	1	0
0.1927945E 16	0.8741768E 00	1.261081696	0.4952527E 14	0.1363012E 00	1	1	1	0
0.1652489E 16	0.8571249E 00	1.261081696	0.3855646E 14	0.1135954E 00	1	1	1	0
0.1399695E 16	0.8470223E 00	1.261081696	0.3118495E 14	0.9265405E-01	1	1	1	0
0.1176841E 16	0.8407839E 00	1.397814751	0.2550405E 14	0.7554328E-01	1	1	1	0
0.1217159E 16	0.1034259E 01	1.397814751	0.2898539E 14	0.7788283E-01	1	1	1	0
0.9751809E 15	0.8011943E 00	1.397814751	0.2313377E 14	0.5749524E-01	1	1	1	0
0.7716599E 15	0.7912592E 00	1.397814751	0.1724993E 14	0.4641151E-01	1	1	1	0
0.6010560E 15	0.7789131E 00	1.397814751	0.1272465E 14	0.3625178E-01	1	1	1	0
0.4634605E 15	0.7710771E 00	1.465103149	0.9519258E 13	0.2831757E-01	1	1	1	0
0.3875405E 15	0.8361889E 00	1.465103149	0.7838852E 13	0.2777666E-01	1	1	1	0
0.2784050E 15	0.7183853E 00	1.465103149	0.6101269E 13	0.1680940E-01	1	1	1	0
0.2033148E 15	0.7302842E 00	1.465103149	0.4446365E 13	0.1318389E-01	1	1	1	0
0.1449913E 15	0.7131369E 00	1.465103149	0.3023925E 13	0.9507537E-02	18	1	1	0
0.1022623E 15	0.7053000E 00	1.496774673	0.2080375E 13	0.6948113E-02	18	1	1	0
0.7314381E 14	0.7152565E 00	1.496774673	0.1470489E 13	0.6048441E-02	18	1	1	0
0.4745573E 14	0.6488003E 00	1.496774673	0.9760313E 12	0.3164947E-02	18	1	1	0
0.3206945E 14	0.6757760E 00	1.496774673	0.6850473E 12	0.2395213E-02	18	1	1	0
0.2090532E 14	0.6518764E 00	1.496774673	0.4297652E 12	0.1564682E-02	18	1	1	0
0.1350256E 14	0.6458912E 00	1.510281563	0.2732673E 12	0.1056433E-02	18	11	1	0
0.8583610E 13	0.6357023E 00	1.510281563	0.1734093E 12	0.7861257E-03	18	18	1	0
0.5147113E 13	0.5996442E 00	1.510281563	0.1057636E 12	0.3845692E-03	18	18	1	0
0.3229794E 13	0.6274964E 00	1.510281563	0.6764573E 11	0.2789497E-03	18	18	1	0
0.1931420E 13	0.5580010E 00	1.510281563	0.4026532E 11	0.1623631E-03	18	18	1	1
0.1151108E 13	0.5959903E 00	1.515126228	0.2335388E 11	0.1037121E-03	18	18	1	3
0.6693553E 12	0.5814879E 00	1.515126228	0.1315334E 11	0.6711483E-04	18	18	1	5
0.3798980E 12	0.5675581E 00	1.515126228	0.8053064E 10	0.3188848E-04	18	18	18	17

FLUXES IN GROUP 4 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	7.706E 13	7.657E 13	7.511E 13	7.271E 13	6.942E 13	6.532E 13	6.051E 13	5.510E 13	4.924E 13	4.393E 13	3.848E 13
4 27.43	2.783E 14	2.766E 14	2.713E 14	2.626E 14	2.507E 14	2.359E 14	2.186E 14	1.990E 14	1.779E 14	1.587E 14	1.390E 14
3 24.38	4.559E 14	4.530E 14	4.444E 14	4.302E 14	4.107E 14	3.865E 14	3.580E 14	3.260E 14	2.913E 14	2.599E 14	2.277E 14
2 21.34	5.947E 14	5.910E 14	5.797E 14	5.611E 14	5.358E 14	5.041E 14	4.670E 14	4.253E 14	3.800E 14	3.391E 14	2.970E 14
1 18.29	6.830E 14	6.787E 14	6.657E 14	6.444E 14	6.153E 14	5.789E 14	5.363E 14	4.884E 14	4.365E 14	3.894E 14	3.411E 14
0 15.24	7.133E 14	7.088E 14	6.952E 14	6.730E 14	6.426E 14	6.046E 14	5.601E 14	5.101E 14	4.558E 14	4.067E 14	3.562E 14

FLUXES IN GROUP 4 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	3.185E 13	2.526E 13	1.880E 13	1.255E 13	6.568E 12	4.167E 12	1.823E 12
4 27.43	1.151E 14	9.125E 13	6.792E 13	4.535E 13	2.373E 13	1.505E 13	6.584E 12
3 24.38	1.885E 14	1.495E 14	1.113E 14	7.429E 13	3.887E 13	2.466E 13	1.079E 13
2 21.34	2.459E 14	1.950E 14	1.452E 14	9.691E 13	5.071E 13	3.217E 13	1.407E 13
1 18.29	2.824E 14	2.240E 14	1.667E 14	1.113E 14	5.824E 13	3.695E 13	1.616E 13
0 15.24	2.949E 14	2.339E 14	1.741E 14	1.162E 14	6.082E 13	3.859E 13	1.688E 13

FLUXES FOR GROUP 4 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PROOIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
C.3907410E 17	0.9293951E 00	1.000000000	0.1013685E 16	0.4785956E 00	1	1	1	0
0.3600599E 17	0.9215821E 00	1.000000000	0.8810119E 15	0.3309875E 00	1	1	1	0
0.3300674E 17	0.9165995E 00	1.000000000	0.7771416E 15	0.2467226E 00	1	1	1	0
0.2999656E 17	0.9088010E 00	1.000000000	0.6903859E 15	0.2081644E 00	1	1	1	0
0.2705057E 17	0.9017890E 00	1.261389732	0.6154087E 15	0.1680905E 00	1	1	1	0
0.3752149E 17	0.1387087E 01	1.261389732	0.1041890E 16	0.1648208E 00	1	1	1	0
0.3281757E 17	0.8746339E 00	1.261389732	0.8322444E 15	0.1336075E 00	1	1	1	0
0.2814197E 17	0.8575276E 00	1.261389732	0.6496224E 15	0.1110270E 00	1	1	1	0
0.2384672E 17	0.8473719E 00	1.261389732	0.5265802E 15	0.9030092E-01	1	1	1	0
0.2005710E 17	0.8410843E 00	1.398197174	0.4314595E 15	0.7344419E-01	1	1	1	0
0.2075214E 17	0.1034653E 01	1.398197174	0.4914731E 15	0.7568294E-01	1	1	1	0
0.1663080E 17	0.8014017E 00	1.398197174	0.3926330E 15	0.5562395E-01	1	1	1	0
0.1316218E 17	0.7914338E 00	1.398197174	0.2929898E 15	0.4482967E-01	1	1	1	0
0.1025284E 17	0.7789620E 00	1.398197174	0.2164320E 15	0.3495824E-01	1	1	1	0
0.7905433E 16	0.7710482E 00	1.465377808	0.1620577E 15	0.2726871E-01	1	1	1	0
0.6608078E 16	0.8358907E 00	1.465377808	0.1335048E 15	0.2675080E-01	1	1	1	0
0.4745539E 16	0.7181422E 00	1.465377808	0.1039210E 15	0.1611984E-01	1	1	1	0
0.3464743E 16	0.7301051E 00	1.465377808	0.7574175E 14	0.1264590E-01	1	1	1	0
0.2470127E 16	0.7129325E 00	1.465377808	0.5150954E 14	0.9114742E-02	18	1	1	0
0.1741410E 16	0.7049879E 00	1.496915817	0.3542919E 14	0.6650805E-02	18	1	1	0
0.1244884E 16	0.7148712E 00	1.496915817	0.2502677E 14	0.5793214E-02	18	1	1	0
0.8073500E 15	0.6485343E 00	1.496915817	0.1660005E 14	0.3022730E-02	18	1	1	0
0.5452123E 15	0.6753109E 00	1.496915817	0.1164366E 14	0.2292037E-02	18	1	1	0
0.3553092E 15	0.6516896E 00	1.496915817	0.7301444E 13	0.1495302E-02	18	1	1	0
0.2294667E 15	0.6458226E 00	1.510388374	0.4655745E 13	0.1010180E-02	18	14	1	0
0.1455222E 15	0.6342190E 00	1.510388374	0.2929168E 13	0.7507205E-03	18	18	1	0
0.8697995E 14	0.5976681E 00	1.510388374	0.1795296E 13	0.3696680E-03	18	18	1	0
0.5469708E 14	0.6288469E 00	1.510388374	0.1138166E 13	0.2636909E-03	18	18	1	0
0.3272367E 14	0.5982710E 00	1.510388374	0.6828998E 12	0.1575947E-03	18	18	1	1
0.1955027E 14	0.5974351E 00	1.515378952	0.3994320E 12	0.9799004E-04	18	18	1	3
0.1130274E 14	0.5781374E 00	1.515378952	0.2233383E 12	0.6294250E-04	18	18	1	5
0.6470335E 13	0.5724570E 00	1.515378952	0.1417339E 12	0.3212690E-04	18	18	18	17

FLUXES IN GROUP 5 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	1.290E 15	1.283E 15	1.259E 15	1.221E 15	1.169E 15	1.103E 15	1.026E 15	9.393E 14	8.443E 14	7.575E 14	6.673E 14
4 27.43	4.661E 15	4.633E 15	4.549E 15	4.410E 15	4.221E 15	3.985E 15	3.707E 15	3.393E 15	3.050E 15	2.736E 15	2.410E 15
3 24.38	7.636E 15	7.590E 15	7.452E 15	7.225E 15	6.915E 15	6.528E 15	6.072E 15	5.558E 15	4.996E 15	4.483E 15	3.949E 15
2 21.34	9.962E 15	9.901E 15	9.721E 15	9.426E 15	9.021E 15	8.516E 15	7.922E 15	7.251E 15	6.518E 15	5.848E 15	5.152E 15
1 18.29	1.144E 16	1.137E 16	1.117E 16	1.083E 16	1.036E 16	9.781E 15	9.098E 15	8.328E 15	7.486E 15	6.717E 15	5.917E 15
0 15.24	1.195E 16	1.188E 16	1.166E 16	1.131E 16	1.082E 16	1.021E 16	9.502E 15	8.698E 15	7.815E 15	7.015E 15	6.179E 15

FLUXES IN GROUP 5 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	5.561E 14	4.438E 14	3.321E 14	2.226E 14	1.167E 14	7.412E 13	3.243E 13
4 27.43	2.009E 15	1.603E 15	1.200E 15	8.042E 14	4.217E 14	2.678E 14	1.171E 14
3 24.38	3.291E 15	2.627E 15	1.965E 15	1.317E 15	6.909E 14	4.386E 14	1.919E 14
2 21.34	4.294E 15	3.427E 15	2.564E 15	1.719E 15	9.014E 14	5.722E 14	2.503E 14
1 18.29	4.931E 15	3.936E 15	2.945E 15	1.974E 15	1.035E 15	6.572E 14	2.875E 14
0 15.24	5.150E 15	4.110E 15	3.076E 15	2.062E 15	1.081E 15	6.864E 14	3.003E 14

FLUXES FOR GROUP 5 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PROCIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
0.3327919E 18	0.9296288E 00	1.000000000	0.8527937E 16	0.4778479E 00	1	1	1	0
0.3067723E 18	0.9218141E 00	1.000000000	0.7421433E 16	0.3300646E 00	1	1	1	0
0.2812572E 18	0.9168273E 00	1.000000000	0.6554386E 16	0.2457311E 00	1	1	1	0
0.2556681E 18	0.9090187E 00	1.000000000	0.5829151E 16	0.2071040E 00	1	1	1	0
0.2306115E 18	0.9019957E 00	1.261581421	0.5201515E 16	0.1670212E 00	1	1	1	0
0.3200524E 18	0.1387842E 01	1.261581421	0.8822687E 16	0.1636346E 00	1	1	1	0
0.2800135E 18	0.8748990E 00	1.261581421	0.7053230E 16	0.1324589E 00	1	1	1	0
0.2401813E 18	0.8577489E 00	1.261581421	0.5513518E 16	0.1099246E 00	1	1	1	0
0.2035664E 18	0.8475530E 00	1.261581421	0.4474433E 16	0.8928621E-01	1	1	1	0
0.1712449E 18	0.8412237E 00	1.398408890	0.3669757E 16	0.7253259E-01	1	1	1	0
0.1772068E 18	0.1034815E 01	1.398408890	0.4184947E 16	0.7472259E-01	1	1	1	0
0.1420280E 18	0.8014815E 00	1.398408890	0.3344852E 16	0.5479127E-01	1	1	1	0
0.1124141E 18	0.7914925E 00	1.398408890	0.2497128E 16	0.4412884E-01	1	1	1	0
0.8756799E 17	0.7789772E 00	1.398408890	0.1845736E 16	0.3437525E-01	1	1	1	0
0.6751860E 17	0.7710420E 00	1.465533257	0.1382705E 16	0.2679157E-01	1	1	1	0
0.5642719E 17	0.8357280E 00	1.465533257	0.1139438E 16	0.2628940E-01	1	1	1	0
0.4051594E 17	0.7180215E 00	1.465533257	0.8868936E 15	0.1580334E-01	1	1	1	0
0.2957472E 17	0.7299526E 00	1.465533257	0.6463067E 15	0.1240027E-01	1	1	1	0
0.2108022E 17	0.7127784E 00	1.465533257	0.4395298E 15	0.8930087E-02	18	1	1	0
0.1485819E 17	0.7048406E 00	1.497001648	0.3022970E 15	0.6516218E-02	18	1	1	0
0.1061638E 17	0.7145132E 00	1.497001648	0.2134427E 15	0.5675554E-02	18	1	1	0
0.6882041E 16	0.6482475E 00	1.497001648	0.1415621E 15	0.2961814E-02	18	1	1	0
0.4647000E 16	0.6752357E 00	1.497001648	0.9909349E 14	0.2238631E-02	18	1	1	0
0.3027762E 16	0.6515520E 00	1.497001648	0.6225985E 14	0.1464427E-02	18	1	1	0
0.1953988E 16	0.6453571E 00	1.510383606	0.3958242E 14	0.9899139E-03	18	18	1	0
0.1242000E 16	0.6356230E 00	1.510383606	0.2508261E 14	0.7349253E-03	18	18	1	0
0.7427746E 15	0.5980472E 00	1.510383606	0.1532444E 14	0.3594160E-03	18	18	1	0
0.4657369E 15	0.6270231E 00	1.510383606	0.9689446E 13	0.2595186E-03	18	18	1	0
0.2789216E 15	0.5988824E 00	1.510383606	0.5806796E 13	0.1525283E-03	18	18	1	1
0.1652768E 15	0.5925564E 00	1.514857292	0.3367254E 13	0.9697676E-04	18	18	1	3
0.9629612E 14	0.5826355E 00	1.514857292	0.1992865E 13	0.6222725E-04	18	18	1	5
0.5410719E 14	0.5618834E 00	1.514857292	0.1116691E 13	0.3069639E-04	18	18	18	17

FLUXES IN GROUP 6 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	1.091E 16	1.085E 16	1.066E 16	1.034E 16	9.915E 15	9.379E 15	8.745E 15	8.026E 15	7.235E 15	6.508E 15	5.747E 15
4 27.43	3.941E 16	3.918E 16	3.849E 16	3.736E 16	3.582E 16	3.388E 16	3.159E 16	2.899E 16	2.614E 16	2.351E 16	2.076E 16
3 24.38	6.456E 16	6.418E 16	6.306E 16	6.121E 16	5.867E 16	5.550E 16	5.174E 16	4.749E 16	4.281E 16	3.851E 16	3.401E 16
2 21.34	8.422E 16	8.373E 16	8.226E 16	7.985E 16	7.654E 16	7.240E 16	6.751E 16	6.196E 16	5.585E 16	5.024E 16	4.437E 16
1 18.29	9.673E 16	9.617E 16	9.448E 16	9.171E 16	8.791E 16	8.315E 16	7.753E 16	7.116E 16	6.415E 16	5.770E 16	5.096E 16
0 15.24	1.010E 17	1.004E 17	9.867E 16	9.578E 16	9.181E 16	8.684E 16	8.097E 16	7.431E 16	6.700E 16	6.026E 16	5.322E 16

FLUXES IN GROUP 6 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	4.805E 15	3.845E 15	2.884E 15	1.937E 15	1.017E 15	6.457E 14	2.825E 14
4 27.43	1.736E 16	1.389E 16	1.042E 16	6.997E 15	3.673E 15	2.332E 15	1.020E 15
3 24.38	2.843E 16	2.276E 16	1.707E 16	1.146E 16	6.017E 15	3.821E 15	1.672E 15
2 21.34	3.710E 16	2.969E 16	2.227E 16	1.495E 16	7.850E 15	4.985E 15	2.181E 15
1 18.29	4.261E 16	3.410E 16	2.558E 16	1.717E 16	9.016E 15	5.726E 15	2.505E 15
0 15.24	4.450E 16	3.561E 16	2.671E 16	1.794E 16	9.416E 15	5.980E 15	2.616E 15

FLUXES FOR GROUP 6 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PRODIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
0.2849849E 18	0.9297569E 00	1.000000000	0.7258078E 16	0.4776455E 00	1	1	1	0
0.2627398E 18	0.9219428E 00	1.000000000	0.6321986E 16	0.3297399E 00	1	1	1	0
0.2409200E 18	0.9169530E 00	1.000000000	0.5587555E 16	0.2453572E 00	1	1	1	0
0.2190291E 18	0.9091361E 00	1.000000000	0.4972533E 16	0.2066962E 00	1	1	1	0
0.1975879E 18	0.9021078E 00	1.261684418	0.4439575E 16	0.1666108E 00	1	1	1	0
0.2742987E 18	0.1388236E 01	1.261684418	0.7537552E 16	0.1631857E 00	1	1	1	0
0.2400200E 18	0.8750312E 00	1.261684418	0.6028249E 16	0.1320260E 00	1	1	1	0
0.2059019E 18	0.8578532E 00	1.261684418	0.4715492E 16	0.1095144E 00	1	1	1	0
0.1745254E 18	0.8476337E 00	1.261684418	0.3828813E 16	0.8891052E-01	1	1	1	0
0.1468295E 18	0.8412883E 00	1.398516655	0.3141571E 16	0.7219249E-01	1	1	1	0
0.1519490E 18	0.1034866E 01	1.398516655	0.3584202E 16	0.7436758E-01	1	1	1	0
0.1217897E 18	0.8015167E 00	1.398516655	0.2865293E 16	0.5448282E-01	1	1	1	0
0.9639796E 17	0.7915118E 00	1.398516655	0.2139375E 16	0.4386234E-01	1	1	1	0
0.7509149E 17	0.7789738E 00	1.398516655	0.1581884E 16	0.3415555E-01	1	1	1	0
0.5789530E 17	0.7709968E 00	1.465595245	0.1185192E 16	0.2661186E-01	1	1	1	0
0.4837912E 17	0.8356312E 00	1.465595245	0.9766412E 15	0.2611059E-01	1	1	1	0
0.3473344E 17	0.7179428E 00	1.465595245	0.7600374E 15	0.1568145E-01	1	1	1	0
0.2535335E 17	0.7299405E 00	1.465595245	0.5539563E 15	0.1230621E-01	1	1	1	0
0.1806937E 17	0.7127017E 00	1.465595245	0.3767760E 15	0.8857489E-02	18	1	1	0
0.1273378E 17	0.7047161E 00	1.497020721	0.2591025E 15	0.6466031E-02	18	1	1	0
0.9097930E 16	0.7144721E 00	1.497020721	0.1829312E 15	0.5629182E-02	18	1	1	0
0.5896612E 16	0.6481267E 00	1.497020721	0.1212340E 15	0.2933383E-02	18	1	1	0
0.3980454E 16	0.6750407E 00	1.497020721	0.8496304E 14	0.2223313E-02	18	1	1	0
0.2592376E 16	0.6512766E 00	1.497020721	0.5333490E 14	0.1450896E-02	18	1	1	0
0.1674046E 16	0.6457572E 00	1.510466576	0.3387441E 14	0.9803772E-03	18	18	1	0
0.1062803E 16	0.6348711E 00	1.510466576	0.2139323E 14	0.7276535E-03	18	18	1	0
0.6347433E 15	0.5972349E 00	1.510466576	0.1306529E 14	0.3592968E-03	18	18	1	0
0.3586691E 15	0.6280792E 00	1.510466576	0.8302172E 13	0.2561808E-03	18	18	1	0
0.2388129E 15	0.5990255E 00	1.510466576	0.4982162E 13	0.1524091E-03	18	18	1	1
0.1423644E 15	0.5961334E 00	1.515309334	0.2860448E 13	0.9542704E-04	18	18	1	3
0.8162683E 14	0.5733654E 00	1.515309334	0.1692217E 13	0.6031990E-04	18	18	1	5
0.4520859E 14	0.5538447E 00	1.515309334	0.9277129E 12	0.3057718E-04	18	18	18	17

FLUXES IN GROUP 7 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	9.313E 15	9.260E 15	9.103E 15	8.842E 15	8.483E 15	8.032E 15	7.498E 15	6.889E 15	6.217E 15	5.595E 15	4.946E 15
4 27.43	3.364E 16	3.345E 16	3.288E 16	3.193E 16	3.064E 16	2.901E 16	2.708E 16	2.488E 16	2.245E 16	2.021E 16	1.786E 16
3 24.38	5.510E 16	5.476E 16	5.385E 16	5.231E 16	5.018E 16	4.751E 16	4.435E 16	4.075E 16	3.678E 16	3.311E 16	2.926E 16
2 21.34	7.187E 16	7.147E 16	7.024E 16	6.823E 16	6.546E 16	6.198E 16	5.785E 16	5.316E 16	4.797E 16	4.319E 16	3.818E 16
1 18.29	8.254E 16	8.207E 16	8.067E 16	7.836E 16	7.518E 16	7.118E 16	6.644E 16	6.105E 16	5.510E 16	4.960E 16	4.385E 16
0 15.24	8.620E 16	8.571E 16	8.425E 16	8.183E 16	7.851E 16	7.434E 16	6.939E 16	6.376E 16	5.754E 16	5.180E 16	4.579E 16

FLUXES IN GROUP 7 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	4.139E 15	3.316E 15	2.490E 15	1.673E 15	8.786E 14	5.580E 14	2.441E 14
4 27.43	1.495E 16	1.198E 16	8.993E 15	6.043E 15	3.174E 15	2.016E 15	8.819E 14
3 24.38	2.449E 16	1.962E 16	1.473E 16	9.900E 15	5.199E 15	3.302E 15	1.445E 15
2 21.34	3.195E 16	2.560E 16	1.922E 16	1.292E 16	6.783E 15	4.308E 15	1.885E 15
1 18.29	3.670E 16	2.540E 16	2.208E 16	1.483E 16	7.791E 15	4.948E 15	2.165E 15
0 15.24	3.833E 16	3.071E 16	2.305E 16	1.549E 16	8.136E 15	5.167E 15	2.261E 15

FLUXES FOR GROUP 7 WRITTEN ON TAPE

RESIDUAL	LAMBDA	OMEGA	DIFF	PRODIF	THE PROPORTIONAL DIFFERENCE CRITERION IS SATISFIED AS FAR AS MESH LINE.....			
0.2589973E 18	0.9295523E 00	1.000000000	0.6630652E 16	0.4797187E 00	1	1	1	0
0.2387543E 18	0.9218410E 00	1.000000000	0.5788585E 16	0.3311664E 00	1	1	1	0
0.2189135E 18	0.9168983E 00	1.000000000	0.5121658E 16	0.2465250E 00	1	1	1	0
0.1990139E 18	0.9090987E 00	1.000000000	0.4559572E 16	0.2078254E 00	1	1	1	0
0.1795230E 18	0.9020626E 00	1.261643410	0.4070547E 16	0.1677462E 00	1	1	1	0
0.2491845E 18	0.1388036E 01	1.261643410	0.6906295E 16	0.1645218E 00	1	1	1	0
0.2180077E 18	0.8748847E 00	1.261643410	0.5520467E 16	0.1334029E 00	1	1	1	0
0.1869782E 18	0.8576686E 00	1.261643410	0.4312774E 16	0.1109144E 00	1	1	1	0
0.1584510E 18	0.8474305E 00	1.261643410	0.3497134E 16	0.9025800E-01	1	1	1	0
0.1332706E 18	0.8410840E 00	1.398375511	0.2865838E 16	0.7344419E-01	1	1	1	0
0.1378618E 18	0.1034450E 01	1.398375511	0.3263978E 16	0.7571840E-01	1	1	1	0
0.1104692E 18	0.8013037E 00	1.398375511	0.2607595E 16	0.5565488E-01	1	1	1	0
0.8742011E 17	0.7913530E 00	1.398375511	0.1945904E 16	0.4485846E-01	1	1	1	0
0.6808827E 17	0.7788627E 00	1.398375511	0.1437221E 16	0.3497362E-01	1	1	1	0
0.5249144E 17	0.7709322E 00	1.465468407	0.1075945E 16	0.2727664E-01	1	1	1	0
0.4386396E 17	0.8356403E 00	1.465468407	0.8860689E 15	0.2675730E-01	1	1	1	0
0.3149465E 17	0.7180074E 00	1.465468407	0.6895999E 15	0.1611513E-01	1	1	1	0
0.2299178E 17	0.7300217E 00	1.465468407	0.5025584E 15	0.1264060E-01	1	1	1	0
0.1638789E 17	0.7127717E 00	1.465468407	0.3417291E 15	0.9107888E-02	18	1	1	0
0.1155159E 17	0.7048854E 00	1.496961594	0.2350679E 15	0.6646693E-02	18	1	1	0
0.8255580E 16	0.7146707E 00	1.496961594	0.1659919E 15	0.5788624E-02	18	1	1	0
0.5352173E 16	0.6483098E 00	1.496961594	0.1100757E 15	0.3016829E-02	18	1	1	0
0.3614115E 16	0.6752611E 00	1.496961594	0.7709466E 14	0.2290964E-02	18	1	1	0
0.2355098E 16	0.6516390E 00	1.496961594	0.4847730E 14	0.1494229E-02	18	1	1	0
0.1519248E 16	0.6450891E 00	1.510304451	0.3076915E 14	0.1006305E-02	18	14	1	0
0.9645501E 15	0.6348863E 00	1.510304451	0.1936601E 14	0.7526875E-03	18	18	1	0
0.5775383E 15	0.5987645E 00	1.510304451	0.1186699E 14	0.3646612E-03	18	18	1	0
0.3618464E 15	0.6265323E 00	1.510304451	0.7529078E 13	0.2657771E-03	18	18	1	0
0.2159874E 15	0.5969035E 00	1.510304451	0.4518306E 13	0.1561642E-03	18	18	1	1
0.1291457E 15	0.5979317E 00	1.510304451	0.2611340E 13	0.9876490E-04	18	18	1	3
0.7513502E 14	0.5817847E 00	1.513737679	0.1537598E 13	0.6186962E-04	18	18	1	5
0.4310763E 14	0.5737355E 00	1.513737679	0.8933532E 12	0.3188848E-04	18	18	18	17

FLUXES IN GROUP 8 FOR COLUMNS 0 TO 10

Z / R	0.00	2.95	5.89	8.84	11.78	14.73	17.67	20.61	23.56	26.10	28.64
5 30.48	8.544E 15	8.504E 15	8.372E 15	8.143E 15	7.818E 15	7.400E 15	6.895E 15	6.311E 15	5.659E 15	5.049E 15	4.433E 15
4 27.43	3.081E 16	3.066E 16	3.017E 16	2.934E 16	2.816E 16	2.665E 16	2.483E 16	2.273E 16	2.039E 16	1.822E 16	1.600E 16
3 24.38	5.038E 16	5.013E 16	4.932E 16	4.795E 16	4.603E 16	4.356E 16	4.058E 16	3.715E 16	3.334E 16	2.981E 16	2.620E 16
2 21.34	6.564E 16	6.530E 16	6.425E 16	6.246E 16	5.995E 16	5.673E 16	5.286E 16	4.840E 16	4.344E 16	3.885E 16	3.416E 16
1 18.29	7.533E 16	7.494E 16	7.373E 16	7.168E 16	6.879E 16	6.509E 16	6.065E 16	5.554E 16	4.986E 16	4.461E 16	3.923E 16
0 15.24	7.865E 16	7.825E 16	7.698E 16	7.484E 16	7.182E 16	6.796E 16	6.333E 16	5.800E 16	5.207E 16	4.658E 16	4.096E 16

FLUXES IN GROUP 8 FOR COLUMNS 11 TO 17

Z / R	31.69	34.74	37.78	40.83	43.88	45.15	46.42
5 30.48	3.691E 15	2.946E 15	2.206E 15	1.479E 15	7.750E 14	4.921E 14	2.153E 14
4 27.43	1.333E 16	1.064E 16	7.965E 15	5.341E 15	2.800E 15	1.778E 15	7.779E 14
3 24.38	2.182E 16	1.742E 16	1.304E 16	8.748E 15	4.587E 15	2.913E 15	1.274E 15
2 21.34	2.846E 16	2.272E 16	1.702E 16	1.141E 16	5.984E 15	3.800E 15	1.662E 15
1 18.29	3.268E 16	2.609E 16	1.954E 16	1.311E 16	6.873E 15	4.364E 15	1.909E 15
0 15.24	3.413E 16	2.725E 16	2.041E 16	1.369E 16	7.178E 15	4.557E 15	1.994E 15

FLUXES FOR GROUP 8 WRITTEN ON TAPE

TIME = 0.109 MINUTES

G. Programmänderungen

Es waren Änderungen des Programms nötig,

- die sich aus der Umstellung von der IBM7090 auf die IBM360 ergaben,
- die Programmfehler beseitigten,
- die zusätzliche Rechenmöglichkeiten schaffen.

Weiter erschien es angebracht, die Papierausgabe zu straffen und neben den berechneten Flüssen auch Energiegruppenszahl und Ortsgitter auf Einheit 10 zu speichern. Unwesentliche Änderungen wie z.B. solche von FØRMAT-Anweisungen werden hier nicht einzeln aufgeführt.

Der Einbau der Unterprogramme SETMAT, SPARX, SPARE, SPARSE, SØRT, die in der bezogenen IBM7090-Version nicht aufgerufen werden und deren Aufgaben aufgrund fehlender Dokumentation unklar sind, bleibt einer späteren Version vorbehalten.

Die Unterprogramme REMØQU, KFKINR, KFKNUS und die Funktion VAL, die SABINE-Removalquerschnitte und KFK-Gruppenkonstanten einführen, wurden hinzugefügt. SABINE-Removalquerschnitte werden in einem DATA-Block bereitgestellt.

Die von A. Daneri u.a.³⁾ vorgeschlagenen Änderungen 2.2.1 bis 2.2.5 in Ref.3 wurden übernommen (s.F. Korrigiertes Rem.Diff. Sample Problem).

Im folgenden werden zunächst die maschinentechnischen Änderungen beschrieben; es folgt die Erläuterung der Modifizierungen in einzelnen Unterprogrammen. Die Angaben in Klammer bedeuten dabei die Seitenzahl im Anhang und die Zeilennummer dieser Seite, wo die angesprochene FØRTRAN-Anweisung oder entsprechende Größe zu finden ist.

1. Maschinentechnische Änderungen

1.1. Es wurden alle EQUIVALENCE-Anweisungen für indizierte Variable geändert. Alle derartigen Variablen innerhalb einer EQUIVALENCE-Anweisung wurden mit Indizes versehen, die angeben, ab welchem Wert den ein- oder mehrdimensionalen Feldern der gleiche Speicherplatz zugeordnet wird.

1.2. Alle DØ-Schleifen, die mit einer IF-Abfrage endeten, wurden so geändert, daß hinter die IF-Abfrage die Anweisung CØNTINUE

eingefügt wurde, da DØ-Schleifen nicht mit einer IF-Abfrage enden dürfen⁵⁾.

1.3. Das Entry LENARG der Assembler-Subroutine NØARG ermittelt die Länge einer Zeichenkette, die über die Subroutine MESSAGE oder deren Entry ERRØR als Nachricht ausgegeben werden soll. Die ursprüngliche Fassung von NØARG konnte nicht verwendet werden. Das neue Entry NØARG benötigt am Ende einer Zeichenkette das Zeichen & , um deren Länge zu ermitteln. Alle Anrufe von MESSAGE bzw. ERRØR wurden entsprechend abgeändert.

1.4. Alle Aufrufe der Funktion MAX wurden gemäß Ref. 5 abgeändert in AMAXO, AMAX1, MAXO, MAX1 je nach dem Typ des Arguments bzw. dem Typ des verlangten Funktionswertes. CALL ABS wurde teilweise durch CALL IABS ersetzt.

1.5. Anstatt der Subroutine CLØCK, die in der Risley-Bibliotheksroutine NEXCH aufgerufen wird, werden die Subroutine DATUM und die Funktion ZEIT verwendet. In NEXCH wurde die Anweisung REALx8 DDATUM, HZEIT (K4/35) eingefügt. Über den Aufruf CALL DATUM im Entry CØDEC erhält man mit DDATUM das Datum und mit HZEIT die Uhrzeit im Format A8. Beide werden zu Beginn einer Rechnung ausgedruckt. Gleichzeitig erfolgt mit PASTIM = ZEIT (0) die Initialisierung der Funktion ZEIT (K5/26). Am Ende der Rechnung ergibt der Wert TT = ZEIT(PASTIM) die benötigte CPU-Zeit in Sekunden.

2. Programmtechnische Änderungen

2.1. Subroutine REMØ

In der Subroutine REMØ wird nach Identifikation von ØRDER (H4/14) die Ordnung des zu bestimmenden zweidimensionalen Polynoms eingelesen (s. Removaleingabe K 3.10), und zwar stellt der Wert II (H4/17) die Ordnung in r-Richtung, JJ (H4/20) die Ordnung in z-Richtung dar. Bei anschließender Eingabe von EXCEPT können diese Werte für bestimmte Zonen überspeichert werden. Es wird eingelesen als ALTØRD(IP) die Zonenummer (H4/29), ALTØRD(IP+1) die Ordnung in z-Richtung (H4/33) und als ALTØRD(IP+2) die Ordnung des Polynoms in r-Richtung (H4/30) mit IP = 1,3,5,....

Später wird der Größe IAI der Wert II (H5/18) und der Größe JAJ der Wert JJ (H5/17) zugeordnet. Durch die anschließende IF-Abfrage (H5/19) wird ermittelt, ob für bestimmte Zonen diese Werte durch unter EXCEPT angegebene Werte überspeichert werden müssen. Wird

jedoch in der Eingabe auf EXCEPT verzichtet oder ist die unter EXCEPT angegebene größte Zonennummer kleiner als die Gesamtzonen-
zahl, dann sind die Werte ALTØRD überhaupt nicht oder für Zonen
größer als die unter EXCEPT angegebene größte Zonenzahl nicht
definiert. Die IF-Abfrage erfolgt mit Zufallswerten von ALTØRD;
sind diese größer als IJ, so werden auch den Größen IAI und JAJ
Zufallswerte zugeordnet.

Da anschließend IAI und JAJ als Endindizes für DØ-Schleifen benutzt
werden (H5/56 und H5/57) , erfolgt ein Überlaufen der in der
folgenden FØRTRAN-Anweisung (H5/58) enthaltenen indizierten Variablen
und der Job wird abgebrochen. Daher wurden in einer DØ-Schleife
(H4/1) vorweg alle Werte von ALTØRD mit 1000 definiert. Diese Werte
werden später teilweise durch aktuelle Werte überspeichert. Bei
nicht überspeicherten Werten werden jedoch über die IF-Abfrage
(H5/19) den Größen JAJ und IAI die unter ØRDER definierten Werte
zugeordnet.

2.2 Subroutine REMØIN

In der Subroutine REMØIN werden alle Anfangswerte von PHIREQ(I),
I = 1,..,20 mit 0. definiert (H7/13). Später werden in der Subroutine
DENSIT diese Werte durch berechnete Werte ersetzt, jedoch nur solche
PHIREQ(I) mit I < KR (H12/31, H12/34, H12/37). In der Subroutine
NØRMAL wird jedoch PHIREQ(I), I = 1,..,10 mit dem Wert FACTØR
multipliziert (H13/41), so daß für PHIREQ(I) mit I > KR Overflow
oder Underflow erfolgt, wenn der Anfangswert nicht mit 0. definiert
ist.

Bei Verwendung von SABINE-Removalquerschnitten wird in REMØIN
(H7/20) die Zeichenfolge SABI identifiziert. Dies führt zum Aufruf
CALL REMØQU (H9/37). In der Subroutine REMØQU erfolgt die Berechnung
der makroskopischen Removalquerschnitte.

2.3. Subroutine REMFUN

Bei der Nachrechnung des Rem.-Diff. Sample Problems unterschieden
sich die berechneten vonden in Ref. 1 angegebenen Removalquellen
um den konstanten Faktor 2.46. Dieser Faktor ist identisch mit $\bar{\nu}$ =
Anzahl von Spaltneutronen pro Spaltung. In der Subroutine REMØIN wurde
ANU = 2.46 gesetzt (H7/15); dieser Wert steht über CØMMØN in REMFUN

zur Verfügung (H14/30). Bei Berechnung der Removalquellen (H16/51) muß dieser Faktor in der Gleichung enthalten sein. Er wurde daher dort wieder eingefügt.

2.4. Function PØLY2

Es wurde eine IF-Abfrage eingefügt (H58/51), die das Auftreten des Index $J = 0$ verhindert.

2.5. Subroutine XACØEF

Mehrere Subroutinen enthalten wie auch XACØEF den Common /CØEF/ (H65/33). Alle diese Commons enthalten Größen mit den gleichen Bezeichnungen und in der gleichen Reihenfolge. Nur bei XACØEF war eine abweichende Reihenfolge angegeben, was zu Fehlern bei Rechnungen in xy-Geometrie führt. Die Reihenfolge wurde dort so geändert, daß sie mit allen anderen COMMON/CØEF/-Anweisungen identisch ist (vgl. z.B. Subroutine XCØEF, H65/33).

2.6. Subroutine CURE2

Es waren die Karten CPUW0700 (H83/23) bis CPUW0740 (H83/27) ursprünglich hinter Karte CPUW0610 (H83/14) angeordnet. Dies führte zu unsinnigen Ergebnissen bei der Berechnung von Neutronenströmen des Sample-Problems 1 (obwohl diese mit den übermittelten Resultaten übereinstimmten). Erst die richtige Reihenfolge, also Anordnung der erwähnten Karten hinter die Karte CPUW0690 (H83/22) ergab sinnvolle Ergebnisse.

Weiterhin wurde beim Einlesen der Ortsgitter-Ordinatenwerte von Einheit NT11 (H82/37) der 2. Laufindex IND zu NZST+NCØL berechnet, richtig muß es jedoch IND=NZST+NRØW heißen.

2.7. Subroutine CURE1

In dieser Subroutine wurde die Abfrage IF (IRET) 100, 150, 23 in IF (IRET) 100, 100, 232 geändert. Es erfolgte sonst ein zweimaliges Einlesen von Schlüsselwörtern ohne Identifikation des ersten in der Subroutine ØUTPUT (H72/43).

2.8. Subroutine ITERAT

Hier erfolgt bei einem RESTART das Einlesen bereits gerechneter Flüsse (H67/49,52). Bei der 2. Anweisung wurde jedoch der Laufindex IND mit IWO = NHALF(1)+MHALF(2) falsch angegeben. Richtig muß es IND = NUMIJ heißen (Berechnung von NUMIJ in STORE, H24/56).

2.9. Subroutine ØUTPUT

Eine erneute Auswertung bereits gerechneter und gespeicherter Neutronenflüsse ist durch die Anweisung ØUTPUT nrow * ncol möglich (s. Eingabe-Beschreibung: Erneute Auswertung). Durch Identifikation der folgenden Eingabe in ØUTPUT (H72/43) werden weitere Berechnungen durchgeführt. Es war jedoch so, daß für jede Energiegruppe alle verlangten Rechnungen durchgeführt wurden, anstatt nacheinander jede geforderte Berechnung für alle Energiegruppen. Daher wurde der Parameter IDEN (H72/33,40,44) eingeführt, welcher den Ablauf in ØUTPUT entsprechend steuert.

In diesem Zusammenhang waren noch einige weitere Anweisungen notwendig (H72/41,42,54,60, H73/6, H74/39).

Weiterhin wurden im Zusammenhang mit der folgenden Änderung die Argumentenlisten der Aufrufe CALL PUNRØW (H74/49) und CALL PUNCØL (H74/50) geändert.

2.10. Subroutinen PUNCØL, PUNRØW, PUNLAB

Die wahrscheinlich früher in ATTØW vorgesehene Möglichkeit, Neutronenflüsse bestimmter Ortsgitterpunkte zur Verwendung als Randbedingungen in Folgerechnungen auf Karten gestanzt auszugeben, wurde erneut aktiviert.

Dafür wurden die Subroutinen PUNCØL und PUNRØW erheblich geändert, DUNLAB neu geschrieben (bis dahin nur als DUMMY-Routine vorhanden).

2.11. Subroutine SETUP

Hier wurde bei der Eingabe von ØUTPUT nrow * ncol das * Zeichen nicht verstanden. Aus diesem Grund mußte die Anweisung IWD = ITDEC (4) in IWD = ITDEC(1) geändert werden.

2.12. Subroutine RØACH

In RØACH wird für alle Isotope, deren mikroskopischer Transportquerschnitt σ_{tr} durch die ersten 16 Werte eines jeden Data-Blocks in DATA/PHI/ gegeben ist, die Größe $1/(3\sigma_{tr})$ berechnet (H30/54). Die entsprechende DØ-Schleife läuft von 1 bis 40; Werte sind jedoch nur für 39 Isotope gegeben. Für Isotop 40 ist es möglich, Werte einzugeben (s. Eingabebeschreibung B2 K 2.6). Wird von dieser Möglichkeit nicht Gebrauch gemacht, enthalten die entsprechenden Speicherplätze Zufalls-

größen und bei Bildung von $1/(3\sigma_{tr})$ können Divide Check oder Underflow auftreten. Sind Werte für Isotop 40 gegeben, so kann angenommen werden, daß für dieses Isotop $\sigma_{tr} > 1.E-4$ ist. Eine entsprechend IF-Abfrage stellt dies fest und setzt für alle Größen $< 1.E-4$ den Wert 1/3. Damit ist gewährleistet, daß dies nur geschieht, wenn keine Werte für Isotop 40 definiert sind.

2.13. Subroutine CRAM

In der Subroutine CRAM wird die Bereitstellung von Materialdaten gesteuert. Es sind folgende Fälle (gemäß Eingabebeschreibung) möglich:

- Es werden makroskopische Wirkungsquerschnitte direkt eingelesen. Dies erfolgt über Subroutine ISØTØP.
- Es wird die RØACH-B-Bibliothek benutzt.
- Es wird die KFKINR-Bibliothek benutzt. Dann wird in CRAM die Zeichenfolge KFKI identifiziert (H28/25).
- Makroskopische Wirkungsquerschnitte wurden durch einen vorherigen NUSYS-Lauf ermittelt. Dann wird in CRAM die Zeichenfolge NUSY identifiziert (H28/25).

In den letzten beiden Fällen, um die das Programm ATTØW erweitert wurde, werden die makroskopischen Wirkungsquerschnitte letztlich durch die Funktion VAL über CØMMØN /CRSECT/ bereitgestellt. Die Werte können nun ähnlich wie in Fall 1 eingelesen werden. Zu diesem Zweck wurde die Subroutine ISØKFK geschaffen, die i.w. wie ISØTØP aufgebaut ist, die Wirkungsquerschnitte einliest und auf die gleichen Speicherplätze setzt, wie dies in ISØTØP geschieht.

2.14. Function NEXCH

In Entry GØDEC dieser Function wurden aufgrund der unter "maschinentechnische Änderungen" beschriebenen notwendigen Einführung der IBM-Routinen DATUM und ZEIT verschiedene FØRTRAN-Anweisungen überflüssig, sodaß dieses Entry nur noch einen Teil der ursprünglichen Karten (z.T. leicht geändert) enthält.

In der Function NEXCH wird der Inhalt jeder Eingabekarte als Zeichenkette ausgedruckt, zusätzlich wird jetzt der Vermerk "TITLE CARD" oder "INPUT CARD" in der gleichen Zeile ausgegeben, was ein leichteres Auffinden der Eingabe im Ausdruck ermöglicht.

Das Entry ERRDEC wurde so geändert, daß zunächst über CALL ERRØR (2 H &) (K5/62) die folgende Nachricht als "ERRØR-MESSAGE" deklariert wird, bevor diese von ERRDEC ausgegeben wird. Weiterhin führt auch eine über ERRDEC ausgegebene Fehlernachricht (nur solche, die bei der Verarbeitung der Eingabe auftreten) zum Job-Abbruch, da sonst ein unkontrolliertes Fortsetzen der Rechnung möglich ist, oftmals unter Ausnutzung der gesamten, ausgegebenen Rechenzeit.

M. Literatur:

- 1) W. D. Collier, G.C. Curtis: A T T Ø W - A Two-Dimensional Shielding Program
TRG-Report 1466(R) (1967)
- 2) E. Kiefhaber : The KFKINR-Set of Group Constants; Nuclear Data Basis and First Results of its Application to the Recalculation of Fast Zero Power Reactors
KFK 1572 (März 1972)
- 3) A. Daneri et. al.: Codice A T T Ø W e sue versioni modificate
Doc. CEC(71) 3 (April 1971)
- 4) C. Ponti et al.: S A B I N E - A One-Dimensional Bulk Shielding Program
EUR 3636e (1967)
- 5) IBM System/360 FØRTRAN IV Language
IBM System Reference Library ØS 44 PS DØS
- 6) G. Bönisch: ATTØW - K : Ein zweidimensionales Removal-Diffusions-Programmsystem für Abschirmungsrechnungen (Karlsruher IBM960/65-Version des ATTØW-Codes).
Diplomarbeit am Lehrstuhl für Reaktortechnik der Universität (TH) Karlsruhe (August 1973)


```

*****
*           *
*  A N H A N G H  *
*           *
*****

```

FORTRAN-LISTEN DES PROGRAMMSYSTEMS ATTOW-K

----- INHALTSVERZEICHNIS -----

	SEITE
*** MAINPROGRAM ***	
MAINPROGRAM ATTOW	H3
*** SUBROUTINES ***	
SUBROUTINE ACCEFS	H59
SUBROUTINE BCUNDS	H48
SUBROUTINE CHBY2	H17
SUBROUTINE COEFS	H51
SUBROUTINE CRAM	H27
SUBROUTINE CRASH	H35
SUBROUTINE CURE1	H78
SUBROUTINE CURE2	H82
SUBROUTINE DATA	H23
SUBROUTINE DENSIT	H11
SUBROUTINE DUMPS	H76
SUBROUTINE GPCINT	H11
SUBROUTINE ISOTCP	H29
SUBROUTINE ITERAT	H66
SUBROUTINE MACRCX	H42
SUBROUTINE MESH	H25
SUBROUTINE MIXES	H31
SUBROUTINE NITER	H27
SUBROUTINE NORMAL	H13
SUBROUTINE NSEW	H56
SUBROUTINE OF	H81
SUBROUTINE OUTPUT	H72
SUBROUTINE PENDLE	H37
SUBROUTINE PRINT	H71
SUBROUTINE PUNCL	H80
SUBROUTINE PUNLAB	H81
SUBROUTINE PUNRCW	H80
SUBROUTINE PUW	H86
SUBROUTINE RASHCO	H35
SUBROUTINE REMFUN	H14

SEITE

SUBROUTINE	REMO	H3
SUBROUTINE	REMCIN	H7
SUBROUTINE	ROACH	H30
SUBROUTINE	RRSCE	H57
SUBROUTINE	RWTAPE	H42
SUBROUTINE	SAUCE	H45
SUBROUTINE	SETP	H11
SUBROUTINE	SETUP	H20
SUBROUTINE	SHIELD	H19
SUBROUTINE	SHUNT	H56
SUBROUTINE	SLOW	H68
SUBROUTINE	SQUEEZE	H33
SUBROUTINE	STORE	H24
SUBROUTINE	TESTIO	H44
SUBROUTINE	XACCEF	H65
SUBROUTINE	XCEFS	H60

*** SUBROUTINE-ENTRIES ***

ENTRY	CF	H81
ENTRY	MOVDAT	H81
ENTRY	PUNEND	H81
ENTRY	PUNNUM	H81
ENTRY	SLCR	H69

*** FUNCTIONS ***

FUNCTION	DLEN	H43
FUNCTION	FL	H43
FUNCTION	PCLY2	H58
FUNCTION	SG	H13
FUNCTION	STEP	H42

*** BLOCK DATAS ***

BLCK DATA	COMMON /TAPE/	H3
BLCK DATA	COMMON /NAME2/	H3
BLCK DATA	COMMON /GAUSS/	H87
BLCK DATA	COMMON /PHI/	H89

```

C          PROGRAMSYSTEM A T T O W - K          *** JULY 1973 ***    1
C          THE KARLSRUHER IBM360/65-VERSION OF THE    2
C          2D-REMOVAL-DIFFUSION-SHIELDING CODE ATTO    3
C          W          MAINPROGRAM ATTO    4
C          COMMON /TITLE/ TITLE(18)    5
10 WRITE (6,100)    6
100 FORMAT ('1')    7
CALL GODEC(TITLE)    8
3 JWD=ITDEC(4)    9
CALL INDEX(J,JWD,19HATTO*REMO*STOP*END&)    10
GO TO (4,5,7,7,8),J    11
4 CALL SHIELD    12
GO TO 10    13
5 CALL REMO(J)    14
GO TO (4,10), J    15
6 WRITE(6,9) JWD    16
9 FORMAT(34H EXPECTED ATTO OR REMOVAL. FOUND A4)    17
7 STOP    18
END    19

```

```

-----
BLOCK DATA    27
COMMON /TAPE/ NTP(8)    28
DATA NTP / 1,2,3,4,8,9,10,11 /    29
END    30

```

```

-----
BLOCK DATA    36
COMMON /NAME2/ SIGREN(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT    37
1 ,IHPR,IHPZ    38
DATA FS/.2E-5,.423E-5,.1153E-4,.209E-4,.594E-4,.1158E-3,.274E-3,    39
1 .603E-3,.001342,.002938,.006318,.01336,.02754,.05501,.1055,    40
2 .1893,.3,.2976/    41
END    42

```

```

-----
SUBROUTINE REMO(IJK)    48
DIMENSION BUFF(11)    49
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11    50
COMMON /WORK/FF(4096),COEFS(10,10)    51
COMMON /NAME1/ ERG(19),RAD,CORST,CCRTP,SHEST,SHEND,SHEBOT,RM(21),    52
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,    53
2 JR,JZ,MR,MZ,MTYPE(400)    54
COMMON /NAME2/ SIGREN(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT    55
1 ,IHPR,IHPZ    56
COMMON/NAME3/EPSIL,G    57
COMMON/GIVEN/ RR(40),ZZ(40) ,IGIVE    58
COMMON /CHUBBY/ CSM(40),CSN(40),F(10,10)    59
INTEGER G,ALTORD    60
DIMENSION ALTORD(183) ,NUGP(20)    61
DATA PI /0.14159265/    62

```

```

DO 2000 I=1,183    1
2000 ALTORD(I)=1000    2
C          REWIND NT2    3
CALL REMOIN    4
CALL GPDINT    5
CALL DENSIT    6
CALL NORMAL    7
DO 9 I=1,NG    8
9 NUGP(I)=I    9
NUGP(NG+1)=NG+1    10
NUG=NG    11
35 CALL WORDEC(WD,4)    12
36 CALL INDEX(J,W,34HORDE*COMB*CHEB*EXCE*ATTO*CONT*END&)    13
GO TO ( 37 , 41 , 15 , 11 , 55 , 10 , 10 , 40 ), J    14
16
37 II=INDEC(1)    17
IWD=NEWDEC(0)    18
IWD=ITDEC(1)    19
JJ=INDEC(0)    20
CALL WORDEC(WD,4)    21
MZ1=MZ-1    22
MR1=MR-1    23
IP=1    24
ALTORD(1)=MZ1*MR1 + 1    25
GO TO 36    26
11 IWD=NEWDEC(1)    27
IF(IWD)14,13,12    28
12 ALTORD(IP)=INDEC(1)    29
ALTORD(IP+2)=INDEC(1)    30
IWD=NEWDEC(0)    31
IWD=ITDEC(1)    32
ALTORD(IP+1)=INDEC(1)    33
IP=IP+3    34
GO TO 11    35
13 CALL ERROR (57HPROGRAM IDENTIFIED EXCEPT. NUMBER MUST FOLLOW. (REM    36
*0,13)E )    37
14 CALL WORDEC(WD,4)    38
GO TO 36    39
GROUP COMBINATIONS    40
41 IWD=ITDEC(4)    41
I=0    42
42 IWD=INDEC(1)    43
LWD=NEWDEC(1)    44
IF(LWD)48,43,44    45
43 LWD=ITDEC(1)    46
I=I+1    47
NUGP(I)=IWD    48
GO TO 42    49
44 LWD=INDEC(1)    50
I=I+1    51
NUGP(I)= IWD    52
IWD=NEWDEC(1)    53
IF(IWD)47,45,46    54
45 IWD=ITDEC(1)    55
GO TO 42    56
46 CALL ERROR (23HINPUT ERROR. (REMO,46)E )    57
47 NUGP(I+1)=LWD+1    58
NUG=I    59
GO TO 35    60
48 NUGP(I+1)=IWD    61

```

```

NUGP(I+2)=IWD+1
NUG=I+1
GO TO 35

REMOVAL SOURCES FOR COMBINED GROUPS

15 IG=0
WRITE (6,1080)
1080 FORMAT ('0')
28 IG=IG+1
IP=1
IJ=0
DO 17 JUP=1,MZ1
J=MZ1+1-JUP
DO 16 I=1,MR1
IJ=IJ+1
JAJ=JJ
IAI=II
IF(IJ.LT.ALTDORD(IP)) GO TO 18
JAJ=ALTDORD(IP+1)
IAI=ALTDORD(IP+2)
IP=IP+3
18 FM=IAI
IF(IAI.EQ.0) GO TO 161
FN=JAJ
M4=4*IAI
N4=4*JAJ

CALCULATE TABLE OF COSINES

THE=0.
DEL=PI/(2.*FM)
DO 20 K=1,M4
CSM(K)=COS(THE)
20 THE=THE+DEL
THE=0.
DEL=PI/(2.*FN)
DO 21 K=1,N4
CSN(K)=COS(THE)
21 THE=THE+DEL
GET POINTS AS REQUIRED BY CHEBY2

DO 22 L=1,JAJ
ZZ(L)=.5*(ZM(J)+ZM(J+1)+CSN(2*L)*(ZM(J+1)-ZM(J)) )
DO 23 K=1,IAI
23 RR(K)=.5*(RM(I)+RM(I+1)+CSM(2*K)*(RM(I+1)-RM(I)) )

GET FUNCTION VALUES AT CHEBY POINTS

DO 50 K=1,IAI
DO 50 L=1,JAJ
50 F(K,L)=0.
IST=NUGP(IG)
IND=NUGP(IG+1)-1
DO 24 G=IST,IND
DO 24 K=1,IAI
DO 24 L=1,JAJ
24 F(K,L)=F(K,L)+SG(RR(K),ZZ(L))
C
WRITE (6,1090)
1090 FORMAT (' ')
CALL MESSAGE( 56HREMOVAL SOURCES FOR GROUP * , * LT R LT * , * LT Z
1 LT *G,IG , RM(I),RM(I+1),ZM(J),ZM(J+1) )

```

```

1 WRITE(6,26)(RR(K),K=1,IAI) 1
2 DO 25 L=1,JAJ 2
3 25 WRITE(6,27) ZZ(L),(F(K,L),K=1,IAI) 3
4 26 FORMAT ('+',10X,10(F7.3,4X)) 4
5 27 FORMAT (' ',F7.3,1PE12.4,9E11.4) 5
6 FIT CHEBY 6
7 K=1 7
8 IF(K.EQ.1) GO TO 34 8
9 IMID=(IAI+1)/2 9
10 JMID=(JAJ+1)/2 10
11 A=ALOG(F(1,JMID)/F(IAI,JMID)) / (RR(IAI)-RR(1)) 11
12 B=ALOG(F(IMID,1)/F(IMID,JAJ)) / (ZZ(JAJ)-ZZ(1)) 12
13 DO 30 K=1,IAI 13
14 DO 31 L=1,JAJ 14
15 31 F(K,L)=F(K,L)/EXP(-A*RR(K)-B*ZZ(L)) 15
16 30 CONTINUE 16
17 17
18 WRITE(6,32) A,B 18
19 32 FORMAT(13H SOURCE=EXP(-1PE10.3,9H*H)*EXP(-E10.3,5H*Z)*G ) 19
20 DO 33 K=1,10 20
21 DO 33 L=1,10 21
22 33 COEFS(K,L)=0. 22
23 CALL CHB Y2(RM(I),RM(I+1),(RM(I)+RM(I+1))*0.5,ZM(J),ZM(J+1), 23
24 1 (ZM(J)+ZM(J+1))*0.5,IAI,JAJ,.0001,COEFS) 24
25 161 WRITE(NT2)MR1,MZ1,I,JUP,RM(I),RM(I+1),ZM(J),ZM(J+1),IAI,JAJ, 25
26 1 ((COEFS(IX,JX),IX=1,IAI),JX=1,JAJ) ,IG 26
27 16 CONTINUE 27
28 17 CONTINUE 28
29 IF(IG-NUG) 28,39,39 29
30 39 REWIND NT2 30
31 GO TO 35 31
32 FOR GIVEN POINTS ONLY 32
33 33
34 40 IG=0 34
35 3 IG=IG+1 35
36 WRITE(6,4)IG 36
37 4 FORMAT(///44X34HREMOVAL SOURCE DENSITIES IN GROUP I2) 37
38 IWA=1 38
39 IWB=MINO(11,JR) 39
40 57 WRITE(6,56){RR(I),I=IWA,IWB} 40
41 56 FORMAT(/7H Z / R = , 6X11(F8.3,2X)) 41
42 IST=NUGP(IG) 42
43 IND=NUGP(IG+1)-1 43
44 DO 5 J=1,JZ 44
45 DO 51 I=1,11 45
46 51 BUFF(I)=0. 46
47 IB=0 47
48 DO 6 I=IWA,IWB 48
49 IB=IB+1 49
50 DO 6 G=IST,IND 50
51 6 BUFF(IB)=SG(RR(I),ZZ(J)) + BUFF(IB) 51
52 5 WRITE(6,7) ZZ(J),(BUFF(I),I=1,IB) 52
53 7 FOR MAT (F8.3,3X,1P11E10.3) 53
54 IF(IWB-JR) 54,53,53 54
55 IWA=IWB+1 55
56 IWB=MINO(IWB+11,JR) 56
57 GO TO 57 57
58 53 IF(IG-NUG) 3,10,10 58
59 10 IJK=2 59
60 RETURN 60
61 55 IJK=1 61
62 RETURN 62
63 END 63

```

602

```

SUBROUTINE REMOIN
COMMON /NAME1/ ERG(19),RAD,CORST,CORSTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHE TOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
3 COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
4 ,IHPR,IHPZ
5 COMMON /NAME3/ EPSIL,G
6 COMMON/DENSE/PHIZEQ(20),PHIREQ(20),PCWER
7 COMMON/GIVEN/ RR(40),ZZ(40),IGIVE
8 INTEGER G
9
10 DO 2000 NULL=1,20
11 2000 PHIREQ(NULL)=0.
12
13 ANU=2.46
14 IRET=1
15 EPSIL=.0001
16 NMAT=0
17
18 100 CALL WORDEC(WD,4)
19 1011 CALL INDEX(J,WD,30HSABI*CORE*ENER*SHIE*ZCNE*REMO&)
20 GO TO (160,103,102,110,116,125,1012),J
21 1012 CALL INDEX(J,WD,26HNU*FISS*SPEC*EPSI*POWE*FS&)
22 GO TO (136,130,137,139,141,118,140),J
23
24 ENERGY BANDS
25
26 102 CALL WORDEC(WD,4)
27 NG=1
28 ERG(NG)=FLDEC(1)
29 1025 IWD=NEWDEC(0)
30 IF(IWD)1026,1022,1023
31 1026 NG=NG-1
32 GO TO 100
33 1023 CALL ERROR (27HINPUT ERROR. (REMOIN,1023)& )
34 1022 IWD=ITDEC(1)
35 CHA=FLDEC(1)
36 IWD=NEWDEC(0)
37 IWD=ITDEC(1)
38 ERGEND=FLDEC(1)
39 1024 NG=NG+1
40 ERG(NG)=ERG(NG-1)-CHA
41 IF(ERG(NG)-ERGEND)1025,1025,1024
42
43 CORE DIMENSIONS
44
45 103 JADD=0
46 104 CALL WORDEC(WD,4)
47 1041 CALL INDEX(J1,WD,26HR*Z*T*RADI*HEIG*FOIN*GIVE&)
48 J1=J1+JADD
49 GO TO (107,108,109,105,106,104,1011,1011,111,112,1023,114,115,104,
50 1151,1011 ),J1
51
52 105 RAD=FLDEC(0)
53 GO TO 104
54 106 CORST=FLDEC(0)
55 CORSTP=FLDEC(0)
56 GO TO 104
57
58 107 KR=INDEC(0)
59 GO TO 104
60 108 KZ=INDEC(0)
61 GOTO 104
62 109 KT=INDEC(0)
63

```

```

GO TO 104
SHIELD DIMENSICNS
110 JADD=8
GO TO 104
111 JR=INDEC(0)
GO TO 104
112 JZ=INDEC(0)
GO TO 104
114 SHEST=FLDEC(0)
SHEND=FLDEC(0)
GO TO 104
115 SHEBOT=FLDEC(0)
SHE TOP=FLDEC(0)
GO TO 104
GIVEN SHIELD POINTS
1151 IGIVE=1
1155 CALL WORDEC(WD,4)
CALL INDEX(J4,WD,4HR*Z&)
GO TO (1152,1154,1041),J4
1152 JR=INDEC(0)
DO 1153 I=1,JR
1153 RR(I)=FLDEC(0)
GO TO 1155
1154 JZ=INDEC(0)
DO 1156 I=1,JZ
1156 ZZ(I)=FLDEC(0)
GO TO 1155
116 CALL WORDEC(WD,4)
1161 CALL INDEX(J2,WD,9HR*Z*COMP&)
GO TO ( 117,120,123,1011),J2
SHIELD MAKE-UP
117 I=0
1171 I=I+1
ST=FLDEC(0)
IF(SHEST-ST)1177,1173,1023
1173 RM(I)=SHEST
1174 IWD=NEWDEC(0)
IF(IWD)1175,1023,1176
1175 IF(RM(I).GE.SHEND) GO TO 1178
I=I+1
RM(I)=SHEND
1178 MR=I
GO TO 116
1176 I=I+1
RM(I)=FLDEC(1)
GO TO 1174
1177 RM(1)=SHEST
RM(2)=ST
I=2
GO TO 1174
118 G=0
1181 G=G+1
IWD=NEWDEC(0)
IF(IWD)100,100,1182
1182 FS(G)=FLDEC(1)

```

```

GO TO 1181
120 I=0
    I=I+1
    ST=FLDEC(0)
    IF(SHEBOT-ST)1208,1202,1023
1202 ZM(I)=SHEBOT
1203 IWD=NEWDEC(0)
    IF(IWD)1204,1023,1207
1204 IF(ZM(I)-SHEBOT)1206,1205,1205
1205 MZ=I
    GO TO 116
1206 I=I+1
    ZM(I)=SHEBOT
    GO TO 1205
1207 I=I+1
    ZM(I)=FLDEC(1)
    GO TO 1203
1208 ZM(1)=SHEBOT
    ZM(2)=ST
    I=2
    GO TO 1203
123 MZ1=MZ-1
    MR1=MR-1
    ISUF=0
    DO 124 J=1,MZ1
    DO 124 I=1,MR1
    ISUF=ISUF+1
124 MTYPE(ISUF)=INDEC(0)
    GO TO 116

    REMOVAL CROSS-SECTIONS BY SABINE VALUES
160 DO 161 K=1,3
161 CALL WORDEC(WD,4)
    CALL REMOQU
    GO TO 100

    REMOVAL CROSS-SECTIONS BY INPUT
125 CALL WORDEC(WD,4)
    CALL INDEX(J3,WD,15HCROS*SECT*XSEC&)
    GO TO (125,127,127,
                                1023),J3
127 IWD=NEWDEC(1)
    IF(IWD)100,100,128
128 IMAT=INDEC(0)
    DO 129 I=1,NG
129 SIGREM(IMAT,I)=FLDEC(0)
    NMAT=MAX0(IMAT,NMAT)
    GO TO 127

    FLUX DISTRIBUTION
130 CALL WORDEC(WD,4)
131 CALL INDEX(J3,WD,25HAXIA*DENS*DIST*RADI*FORM&)
    GO TO (
                                134 , 130 , 130 , 132 ,1351,1011), J3
132 PHIR(1,1)=FLDEC(1)
    IWD=NEWDEC(0)
    IWD=ITDEC(1)
    DIST=FLDEC(1)

```

```

1 IWD=NEWDEC(0)
2 IWD=ITDEC(1)
3 DISTAN=FLDEC(0)
4 I=(DISTAN-PHIR(1,1))/DIST + 1.01
5 PHIR(1,2)=FLDEC(0)
6
7 1) = DISTANCE 2) = DENSITY DISTRIBUTION
8 DO 133 JI=2,I
9 PHIR(JI,1)=PHIR(JI-1,1)+DIST
10 133 PHIR(JI,2)=FLDEC(0)
11 IMPR=I
12 GO TO 130
13 134 PHIZ(1,1)=FLDEC(1)
14 IWD=NEWDEC(0)
15 IWD=ITDEC(1)
16 DIST=FLDEC(1)
17 IWD=NEWDEC(0)
18 IWD=ITDEC(1)
19 DISTAN=FLDEC(0)
20 I=(DISTAN-PHIZ(1,1))/DIST + 1.01
21 IHPZ=I
22 PHIZ(1,2)=FLDEC(0)
23 DO 135 JI=2,I
24 PHIZ(JI,1)=PHIZ(JI-1,1)+DIST
25 135 PHIZ(JI,2)=FLDEC(0)
26 GO TO 130
27
28 1351 IHPR=-1
29 DO 1352 I=1,8
30 1352 PHIR(I,1)=FLDEC(0)
31 GO TO 100
32
33 136 ANU=FLDEC(0)
34 GO TO 100
35 137 NG1=NG-1
36 DO 138 I=1,NG1
37 138 FS(I)=FLDEC(0)
38 GO TO 100
39 139 EPSIL=FLDEC(0)
40 GO TO 100
41 141 POWER=FLDEC(1)
42 GOTO 100
43 140 CONTINUE
44 CALL INDEX(J,WD,5HCONT&)
45 IF(J.EQ.1.AND.IRET.EQ.0) STCP
46 IF(J.EQ.1) GO TO 150
47 WRITE(6,99) WD
48 99 FORMAT(19H UNRECOGNISED WORD A4 )
49 IRET=0
50 98 IWD=NEWDEC(1)
51 IF(IWD)104,151,152
52 151 IWD=ITDEC(1)
53 GO TO 98
54 152 WD=FLDEC(1)
55 GO TO 98
56 150 RETURN
57 END
58
59
60
61
62
63

```

```

SUBROUTINE GPOINT
COMMON /WEIGHT/ AR(20),AZ(20),AT(20)
COMMON /NAME1/ ERG(19),RAD,CORST,CORTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
1 ,IHPR,IHPZ
DATA PI /3.14159265/

CALL SETP(RC,KR,1)
DO 1 I=1,KR
1 RC(I)=(RC(I)+1.)*RAD*.5
CALL SETP(ZC,KZ,1)
DO 2 I=1,KZ
2 ZC(I)=(ZC(I)*(CORTP-CORST)+CORTP+CORST)*.5
CALL SETP(THETA,KT,1)
DO 3 I=1,KT
3 THETA(I)=(THETA(I)+1.)*PI*.5
CALL SETP(AT,KT,2)
CALL SETP(AZ,KZ,2)
CALL SETP(AR,KR,2)
RETURN
END

```

```

SUBROUTINE SETP(X,M,IE)
COMMON /GAUSS/ G(2,8,16)
DIMENSION X(M)

SN=-1.
IF(IE.EQ.2) SN=1.
I=(M+1)/2 + 1
J=M+1
K=0
1 K=K+1
J=J-1
I=I-1
X(J)=G(IE,I,M)
X(K)=SN*X(J)
IF(I-1)2,2,1
2 RETURN
END

```

```

SUBROUTINE DENSIT
COMMON /NAME1/ ERG(19),RAD,CORST,CORTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
1 ,IHPR,IHPZ
COMMON/DENSE/PHIZEQ(20),PHIREQ(20),POWER

IF(IHPR.EQ.-1) GO TO 40
DO 26 JC=1,KZ
IT=(ZC(JC)-PHIZ(1,1)) / (PHIZ(2,1)-PHIZ(1,1)) +1.
P=(ZC(JC)-PHIZ(IT,1)) / (PHIZ(2,1)-PHIZ(1,1))
Q=1.-P

```

```

IF(IT.LE.1) GO TO 24
IF(IT.EQ.IHPZ-1) GO TO 25
IF(IT.GE.IHPZ) GO TO 36
PT=P*(P*P-1.)/3.
QT=Q*(Q*Q-1.)/3.
PHIZEQ(JC)=PHIZ(IT,2)*(Q-QT+PT*.5) + PHIZ(IT+1,2)*(P+QT*.5-PT)
1 + PHIZ(IT-1,2)*QT*.5 + PHIZ(IT+2,2)*PT*.5
GO TO 26
24 PHIZEQ(JC)=(PHIZ(1,2)*(PHIZ(2,1)-ZC(JC))+PHIZ(2,2)*(ZC(JC)-PHIZ(1,
1 1))) / (PHIZ(2,1)-PHIZ(1,1))
GO TO 26
25 PHIZEQ(JC)=PHIZ(IT+1,2)*P*(1.-.5*Q) + PHIZ(IT,2)*Q*(1.+P) -
1 PHIZ(IT-1,2)*P*Q*.5
GO TO 26
36 PHIZEQ(JC)=PHIZ(IHPZ,2)- (PHIZ(IHPZ-1,2)-PHIZ(IHPZ,2))*P
26 CONTINUE
27 DO 20 IC=1,KR
IT=(RC(IC)-PHIR(1,1))/(PHIR(2,1)-PHIR(1,1)) +1
IF(IT.EQ.0) GO TO 29
P=(RC(IC)-PHIR(IT,1))/(PHIR(2,1)-PHIR(1,1))
Q=1.-P
IF(IHPR.EQ.2) GO TO 23
IF(IT.EQ.1) GO TO 21
IF(IT.GE.IHPR) GO TO 30
IF(IT.GE.IHPR-1) GO TO 22
PT=P*(P*P-1.)/3.
QT=Q*(Q*Q-1.)/3.
PHIREQ(IC)=PHIR(IT,2)*(Q-QT+PT*.5) + PHIR(IT+1,2)*(P+QT*.5-PT)
1 +PHIR(IT-1,2)*QT*.5 + PHIR(IT+2,2)*PT*.5
GO TO 20
21 PHIREQ(IC)=PHIR(IT,2)*Q*(1.-.5*P)+PHIR(IT+1,2)*P*(1.+Q)-PHIR(IT+2,
1 2)*P*Q*.5
GO TO 20
22 PHIREQ(IC)=PHIR(IT+1,2)*P*(1.-.5*Q) + PHIR(IT,2)*Q*(1.+P) -
1 PHIR(IT-1,2)*P*Q*.5
GO TO 20
23 PHIREQ(IC)=(PHIR(1,2)*(PHIR(2,1)-RC(IC)) + PHIR(2,2)*(RC(IC)-PHIR(
1 1,1))) / (PHIR(2,1)-PHIR(1,1))
GO TO 20
30 PHIREQ(IC)=(1.+P)*PHIR(IT,2)-P*PHIR(IT-1,2)
GO TO 20
29 Q=(PHIR(1,1)-RC(IC))/(PHIR(2,1)-PHIR(1,1))
PHIREQ(IC)=(1.+Q)*PHIR(1,2)-Q*PHIR(2,2)
20 CONTINUE
28 RETURN
40 DO 41 JC=1,KZ
41 PHIZEQ(JC)=PHIR(5,1)*COS(PHIR(6,1)*ZC(JC)) + PHIR(7,1)*(EXP(ZC(JC)
1 *PHIR(8,1)) + EXP(-ZC(JC)*PHIR(8,1)))*.5
DO 42 IC=1,KR
42 PHIREQ(IC)=PHIR(1,1)*SIN(PHIR(2,1)*RC(IC))/RC(IC) + PHIR(3,1)*
1 (EXP(PHIR(4,1)*RC(IC)) - EXP(-PHIR(4,1)*RC(IC)))*.5
RETURN
END

```

15

```

SUBROUTINE NORMAL
COMMON /WEIGHT/ AR(20),AZ(20),AT(20)
COMMON /NAME1/ ERG(19),RAD,CORST,CORTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
1 ,IHPR,IHPZ
COMMON/DENSE/PHIZEQ(20),PHIREQ(20),PCWER
DIMENSION F(15),C(15)

PI=3.141592654
IF(IHPR.EQ.-1) GO TO 155

SGINT=0.
DO 3282 I=1,KR
3282 SGINT=SGINT+PHIREQ(I)*AR(I)*RC(I)
SGINT=SGINT*RAD/2.
SGENT=0.
DO 3283 I=1,KZ
3283 SGENT=SGENT+PHIZEQ(I)*AZ(I)
SGENT=SGENT*(CORTP-CORST)*.5
SGINT=SGINT*SGENT*2.*PI
GO TO 156

155 SGINT=2.*PI*(
PHIR(1,1)*(-COS(PHIR(2,1)*RAD)+1.)/PHIR(2,1)
+PHIR(3,1)*(EXP(PHIR(4,1)*RAD)+EXP(-PHIR(4,1)*
RAD)+2.)/2./PHIR(4,1)
3 ( PHIR(5,1)*(SIN(PHIR(6,1)*CORTP)-SIN(PHIR(6,1)*CORST))/
4 PHIR(6,1)
5 + PHIR(7,1)*(EXP(PHIR(8,1)*CORTP)-EXP(-PHIR(8,1)*CORTP)
6 -EXP(PHIR(8,1)*CORST)+EXP(-PHIR(8,1)*CORST)
7 /2. /PHIR(8,1)
GO TO 156

156 FACTOR = 7.626E10*POWER/SGINT
IF(IHPR.EQ.-1) GO TO 158
DO 157 I=1,IHPR
157 PHIR(I,2)=PHIR(I,2)*FACTOR
GO TO 159
158 PHIR(1,1)=FACTOR*PHIR(1,1)
PHIR(3,1)=FACTOR*PHIR(3,1)
159 CONTINUE
DO 160 I=1,10
160 PHIREQ(I)=PHIREQ(I)*FACTOR
RETURN
END

```

```

FUNCTION SG(R,Z)
COMMON /WEIGHT/ AR(20),AZ(20),AT(20)
COMMON /JIST/ JSTI,JSTI
COMMON /NAME1/ ERG(19),RAD,CORST,CORTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
1 ,IHPR,IHPZ
COMMON /NAME3/ EPSIL,G
COMMON /WORK/ F(4096),COEFS(10,10)
INTEGER G
DATA PI /3.14159265/

CALL REMFUN(R,Z,F,KR,KZ,KT)

```

```

SG=0.
I=0
DO 3 N=1,KT
SG2=0.
DO 2 M=1,KZ
SG1=0.
DO 1 L=1,KR
I=I+1
1 SG1=SG1+AR(L)*F(I)*RC(L)
2 SG2=SG2+AZ(M)*SG1
3 SG=SG+AT(N)*SG2
ICE=(MZ-JSTI-1)*(MR-1)+ISTI-1
IF(ISTI.EQ.1) ICE=ICE+1
IMAT=MTYPE(ICE)
SG=SG*RAD*PI*(CORTP-CORST)*.125
SG=SG*SIGREM(IMAT,G)*2.
RETURN
END

```

```

SUBROUTINE REMFUN(R,Z,F,L,M,N)
COMMON /WEIGHT/ AR(20),AZ(20),AT(20)
COMMON /NAME3/ EPSIL,G
COMMON /NAME1/ ERG(19),RAD,CORST,CORTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
1 ,IHPR,IHPZ
COMMON /JIST/ JSTI,JSTI
COMMON/DENSE/PHIZEQ(20),PHIREQ(20),PCWER
INTEGER G
REAL LAMBDA ,L2
DIMENSION F(L,M,N)
PI=3.141592654

DO 20 IC=1,KR
28 DO 20 JC=1,KZ
DO 20 KC=1,KT
F(IC,JC,KC)=0.
XSI=0.
IF(KC.NE.1) GO TO 29
RCI=RC(IC)
RCIP=RC(IC+1)
IF(IC.EQ.KR)RCIP=3.*RAD-2.*RCI
IF(R-RCI.GT.2.*(RCIP-R)) GO TO 29
ZCJ=ZC(JC)
ZCJP=ZC(JC+1)
IF(JC.EQ.KZ) ZCJP=3.*CORTP-2.*ZCJ
IF(Z-ZCJ.GT.2.*(ZCJP-Z)) GO TO 29
ZCJM=ZC(JC-1)
IF(JC.EQ.1) ZCJM=3.*CORST-2.*ZCJ
IF(ZCJ-Z.GT.2.*(Z-ZCJM)) GO TO 29
RCIM=RC(IC-1)
IF(IC.EQ.1) RCIM=-1.95*RCI
IF(RCI-R.GT.2.*(R-RCIM)) GO TO 29

R1=0.
ICM=IC-1
DO 40 I=1,ICM
40 R1=R1+AR(I)

```

```

R2=R1+AR(IC)
R1=RAD*.5*R1
R2=R2*RAD*.5
R12=R1*R1
R22=R2*R2
Z1=0.
JCM=JC-1
DO 41 J=1,JCM
41 Z1=Z1+AZ(J)
Z2=Z1+AZ(JC)
Z1=Z1*(CORTP-CORST)*.5
Z2=Z2*(CORTP-CORST)*.5
Z12=Z1*Z1
Z22=Z2*Z2
DTHE=AT(1)*PI*.5
SDT=SIN(DTHE)
DAV2=R*.5*(R12+R22)-1.333333333*R*(R22+R12+R1*R2)*SDT/(DTHE*(
1 R1+R2)) +Z*Z+.333333333*(Z22+Z12+Z1*Z2)-Z*(Z2+Z1)
DAV=SQRT(DAV2)
L2=DAV2
IGO=2
GO TO 32
53 ICE=(MZ-JST-1)*(MR-1)+IST-1
IMAT=MTYPE(ICE)
XSI=DAV*SIGREM(IMAT,G)
GO TO 42

29 ZH=Z-ZC(JC)
D2=RC(IC)*RC(IC) + R*R - 2.*R*RC(IC)*COS(THETA(KC))
D=SQRT(D2)
L2=D2+ZH*ZH
EL=SQRT(L2)
IGO=1

LAMBDA=0.
OLAM=0.
IF(ZH) 30,1,31
30 JH=1
JADD=1
GO TO 32

1 JH=0
JADD=0
GO TO 32

31 JH=0
JADD=-1

32 CONTINUE
IST=0
2 IST=IST+1
IF(RM(IST)-R) 2,3,3
3 JST=0
4 JST=JST+1
IF(ZM(JST)-Z)4,5,5
5 JST=JST-1
ISTI=IST
JSTI=JST
GO TO (10,53),IGO

10 ZETA=Z-LAMBDA*ZH
51 ISTUF=IABS(IST-1)
ARG=RM(ISTUF)*RM(ISTUF)*D2 - R*R*RC(IC)*RC(IC)*SIN(THETA(KC))*

```

```

1 SIN(THETA(KC))
2 IF(ARG.LT.0.) GO TO 50
3 RL1=R*(R-RC(IC)*COS(THETA(KC)))
4 RL2=SQRT(ARG)
5 RLM2=(RL1-RL2)/D2
6 RLM1=(RL1+RL2)/D2
7 RLAM=AMINI(RLAM1,RLAM2)
8 IF(RLAM.LE.OLAM) RLAM=AMAXI(RLAM1,RLAM2)
9 IF(RLAM.GT.1.) RLAM=1.
10 GO TO 33
11
12 THE LINE DOES NOT CUT WHAT APPEARS TO BE THE CYLINDER RM(IST-1)
13 THEREFORE WE MUST FIND OUT WHERE IT CUTS RM(IST) FOR THE SECOND TIME
14 IT MUST CUT RM(IST) BUT MAYBE WITH LAMBDA GT 1
15
16 50 IST=1-IST
17 GO TO 51
18
19 33 IZUF=JST+JH
20 IF(ABS(ZH).GT.1.E-20) GO TO 34
21 ZLAM = 1.
22 GO TO 36
23
24 34 ZLAM=(Z-ZM(IZUF)) / ZH
25 IF(ZLAM.GT.1.) ZLAM=1.
26
27 36 JT=JST
28 IT=IST
29 IF(RLAM-ZLAM) 8,7,6
30
31 6 LAMBDA=ZLAM
32 JST=JST+JADD
33 IST=IST-1
34 GO TO 9
35
36 7 LAMBDA=ZLAM
37 JST=JST+JADD
38 IST=IST-1
39 GO TO 9
40
41 8 LAMBDA=RLAM
42 IST=IST-1
43
44 9 PATH=LAMBDA-OLAM
45 OLAM=LAMBDA
46 IF(IST.EQ.1) IT=2
47 IF(IST.LT.0) IT=-IST +1
48 ICE=(MZ-JT-1)*(MR-1) + IT-1
49 IF(RLAM.EQ.ZLAM.AND.RLAM.EQ.1.) ICE=ICE-1
50 IMAT=MTYPE(ICE)
51 XSI=XSI+PATH*EL*SIGREM(IMAT,G)
52 IF(LAMBDA-1.) 10,42,42
53
54 42 CONTINUE
55
56 F(IC,JC,KC)=PHIZEQ(JC)*PHIREQ(IC)*FS(G)*ANU*EXP(-XSI) / (L2*.4*PI)
57
58 ***** AENDERUNG ***** 26.4.72 BOEN ISCH
59 FAKTOR ANU (DEFINIERT IN REMOIN, KARTE IREMO120) NACHTRAEGLICH
60 EINGEFUEGT.
61
62 20 CONTINUE
63 RETURN
64 END

```

101


```

SUBROUTINE CHBY2(X1,X2,X3,Y1,Y2,Y3,M,N,EPS,C)
CCHBY2 2-DIMENSIONAL POLYNOMIAL ROUTINE SELECTING OWN ORDERS
COMMON/CHUBBY/CSM(40),CSN(40),F(10,10)
DIMENSION C(10,10),CX(10,10),CXX(10,10),
IP(10,10),Q(10,10),R(10,10),S(10,10)
EQUIVALENCE (F(1,1),P(1,1),Q(1,1),R(1,1),S(1,1)),
1 (CX(1,1),CXX(1,1))

1 FM=M
FN=N
A=1.0/(FM*FN)
M4=4*M
N4=4*N
PI=3.141592654
CALCULATE CHEBYSHEV COEFFICIENTS
DO 11 I=1,M
AI=A
IF(I-1)99,12,13
13 AI=A+A
12 I2=2*I-2
DO 11 J=1,N
AIJ=AI
IF(J-1)99,14,15
15 AIJ=AI+AI
14 J2=2*J-2
C(I,J)=0.0
IK=I
DO 19 K=1,M
JL=J
DO 16 L=1,N
C(I,J)=C(I,J)+F(K,L)*CSM(IK)*CSN(JL)
JL=JL+J2
IF (JL-N4)16,16,17
17 JL=JL-N4
16 CONTINUE
IK=IK+I2
IF (IK-M4) 19,19,18
18 IK=IK-M4
19 CONTINUE
11 C(I,J)=AIJ*C(I,J)
INT=0
CMAX=0.
DO 60 K=1,M
DO 60 L=1,N
IF(ABS(C(K,L))-CMAX) 60,60,61
61 CMAX=C(K,L)
60 CONTINUE
DEL=CMAX*EPS
DO 62 L=1,N
IF(ABS(C(M,L))-DEL) 62,63,63
62 CONTINUE
INT=1
M=M-1
GO TO 64
63 DO 65 K=1,M
IF(ABS(C(K,N))-DEL) 65,70,70
65 CONTINUE
INT=1
N=N-1
GO TO 63
70 IF (INT) 1,66,1
TRANSFORM TO POWER SERIES
66 DO 22 L=1,N

```

```

1 Q(1,L)=0.0
2 22 Q(2,L)=0.0
3 Q(1,1)=1.0
4 Q(2,2)=1.0
5 DO 23 K=3,N
6 Q(K,1)=-Q(K-2,1)
7 DO 23 L=2,N
8 23 Q(K,L)=-Q(K-2,L)+2.0*Q(K-1,L-1)
9 DO 24 I=1,M
10 DO 24 J=1,N
11 CX(I,J)=0.0
12 DO 24 L=1,N
13 24 CX(I,J)=CX(I,J)+C(I,L)*Q(L,J)
14 DO 20 K=1,M
15 P(K,1)=0.0
16 20 P(K,2)=0.0
17 P(1,1)=1.0
18 P(2,2)=1.0
19 DO 21 L=3,M
20 P(1,L)=-P(1,L-2)
21 DO 21 K=2,M
22 21 P(K,L)=-P(K,L-2)+2.0*P(K-1,L-1)
23 DO 25 I=1,M
24 DO 25 J=1,N
25 C(I,J)=0.0
26 DO 25 K=1,M
27 25 C(I,J)=C(I,J)+P(I,K)*CX(K,J)
28 TRANSFORM XX=G+HX
29 G=Y2-Y1
30 H=2.0/G
31 G=(Y3+Y3-Y2-Y1)/G
32 DO 32 L=1,N
33 S(1,L)=0.0
34 S(1,1)=1.0
35 DO 33 K=2,N
36 S(K,1)=G*S(K-1,1)
37 DO 33 L=2,N
38 33 S(K,L)=G*S(K-1,L)+H*S(K-1,L-1)
39 DO 34 I=1,M
40 DO 34 J=1,N
41 CXX(I,J)=0.0
42 DO 34 L=1,N
43 34 CXX(I,J)=CXX(I,J)+C(I,L)*S(L,J)
44 G=X2-X1
45 H=2.0/G
46 G=(X3+X3-X2-X1)/G
47 DO 30 K=1,M
48 30 R(K,1)=0.0
49 R(1,1)=1.0
50 DO 31 L=2,M
51 R(1,L)=G*R(1,L-1)
52 DO 31 K=2,M
53 31 R(K,L)=G*R(K,L-1)+H*R(K-1,L-1)
54 DO 35 I=1,M
55 DO 35 J=1,N
56 C(I,J)=0.0
57 DO 35 K=1,M
58 35 C(I,J)=C(I,J)+R(I,K)*CXX(K,J)
59 CALL DVCHK(IDV)
60 RETURN
61 99 STOP
62 END

```

```

SUBROUTINE SHIELD
COMMON /TAPE/ NTP(8)
COMMON /COUNT/ COUNT(60)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NUM/ NBACK
COMMON /REST/ IREST
COMMON /COEF/ RO,OE,ON,OW,OS,AW,SE,SW,NE,NW,J,I
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1 ST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
COMMON/GEOM/GEOM
COMMON/PHI/PHI(16000)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDI ST

DIMENSION TMESH(1)
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1)
DIMENSION AE(1),AO(1),D(1),A(1),SG(1)
DIMENSION THETA(1),Y(1),BBB(1)
DIMENSION W(1),X(1)
INTEGER G ,GEOM
INTEGER SE,SW

EQUIVALENCE (PHI(1),TMESH(1))
EQUIVALENCE (NTP(4),NT4),(NTP(5),NT5),(NTP(8),NT11)
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1))
EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1))
EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1))
EQUIVALENCE (AO(1),W(1)),(AE(1),X(1))
DATA LCRAM / 1 /

ACCUR(4)=1.
DO 3 NI=1,8
N=NTP(NI)
3 REWIND N
IF(LCRAM.NE.1) GO TO 4
WRITE(NT4) (PHI(I),I=1,3783)
REWIND NT4
4 LCRAM=0
CALL SETUP
IF(GEOM.LT.25) GO TO 24
GEOM=GEOM-25
GO TO 25
24 IF(IREST.EQ.-1) GO TO 22
G=0
GO TO 21
22 CALLTEST10
CALL SAUCE
G=0
26 CALL INDEX(J,WD,15HSOLV*GROU*STOP&)
GO TO (30,31,32,33),J
30 CALL WORDEC(WD,4)
GO TO 21
31 NG=INDEC(O)
CALL WORDEC(WD,4)
GO TO 26
32 CALL EXIT
33 CALL MESSAGE(45HEXPECTED SOLVE OR STCP.PROGRAM ASSUMES SOLVE& )
21 IF(NPHIST.EQ.0) NG=1
IF(ACCUR(4)) 1002,1002,219
219 G=G+1
23 IST=NRST+1

```

```

IND=NRST+NCOL
READ(NT11)(RMESH(IRW),IRW=IST,IND)
1001 IST=NZST+1
IND=NZST+NROW
READ(NT11) (ZMESH(IRW),IRW=IST,IND)
IF(GEOM.EQ.1) CALL COEFS
IF(GEOM.EQ.2)CALL XCOEFS
CALL ITERAT
IND=MHALF(1)
WRITE(NT8) (PHI(I),I=1,IND)
10
IND=MHALF(1)+1
IND=NROW*NCOL
WRITE(NT8) (PHI(I),I=IST,IND)
12
WRITE(6,1) G
14
1 FORMAT(18H FLUXES FOR GROUP I2,16H WRITTEN ON TAPE )
15
2 IF(G.LT.NG) GO TO 219
16
25 CALL OUTPUT
17
200 RETURN
18
1002 CALL MESSAGE(18HEXECUTION DELETED&)
19
RETURN
20
END
21

-----
SUBROUTINE SETUP
COMMON /TAPE/ NTP(8)
COMMON /XTAPE/ NATOL
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NUM/ NBACK
COMMON /NST/ NBNDST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,NAS
1 T,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
COMMON /COEF/ SE,SW,NE,NW,RO,OE,ON,CW,OS
COMMON/SOURCE/SOURCE(100),ISSRCE(60)
COMMON /REST/ IREST
COMMON/GEOM/GEOM
INTEGER GEOM
COMMON/PHI/PHI(17000)
COMMON /IREMO/IREMO
COMMON /COUNT/ COUNT(60)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDI ST

DIMENSION TMESH(1)
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),AS2(1),AS(1)
DIMENSION SCATER(1),AE(1),AO(1),D(1),A(1),SG(1)
DIMENSION THETA(1),Y(1),BBB(1)
INTEGER COUNT

EQUIVALENCE (NTP(2),NT2)
EQUIVALENCE (PHI(1),TMESH(1))
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1))
EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1))
EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1))
EQUIVALENCE (AO(1),W(1)),(AE(1),X(1))
EQUIVALENCE (WD,IWD)

DO 1 IC=1,60
1 COUNT(IC)=50
NATOL=NT2
IREMO=0

```

100

```

NCOL=0
NROW=0
OMEGA=1.
ACCUR(1)=.2
ACCUR(2)=1.E-6
ACCUR(3)=1.E-4
NSLOR=50
NGAUSS=0
NBACK=1
IREST=-1
J4=0
GEOM=1

1001 IWD=ITDEC(4)
1002 CALL INDEX(J,IWD,85HCOUN*ACCU*GROU*OMEG*MATE*MESH*ZONE*CONS*REST*H
      10LD*GEOM*OUTP*USED*CHAN*TAP*REMO*ATCL&)
1003 GO TO ( 1030,1040,1050,1090,1070,1010,1020,1080,1300,1200,111
      *0,1130,1140,1001,1150,1160,1170,1026) , J

===== SET CONSTANTS =====

1030 CALL NITER(WD,WD)
      GO TO 1002

1040 DO 1041 I=1,3
1041 ACCUR(I)=FLDEC(0)
      GO TO 1001

1050 NG=INDEC(0)
      GO TO 1001

1070 CALL DATA
      GO TO 1002

1080 NMAT=INDEC(0)
      IF(NEWDEC(0)) 1081,1082,1004
1004 NBACK=INDEC(0)
      GO TO 1001
1081 IWD=ITDEC(4)
      NBACK=1
      GO TO 1002
1082 CALL ERROR (71HPROGRAM CANNOT IDENTIFY THE DECLARATIONS AFTER CONS
      *TANTS. (SETUP,1082)&)

1090 OMEGA=FLDEC(0)
      GO TO 1001

1110 IWD=ITDEC(4)
      CALL INDEX(GEOM,WD,6HRZ*XY&)
      IF(GEOM.GT.2) CALL ERROR (44HGEOMETRY NOT RZ OR XY DEFINED. (SETUP
      *,1082)&)
      GO TO 1001

1130 NROW=INDEC(1)
      IF(NEWDEC(0).NE.0) CALL ERRRC (62HGIVE VALUES FOR NROW*NCOL AFTER
      *DEFINING OUTPUT. (SETUP,1130)&)
      IWD=ITDEC(1)
      NCOL=INDEC(0)
      CALL STORE
      GEOM=GEOM+25
      NBACK=-NBACK
      IWD=ITDEC(4)
      RETURN

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

1140 NGAUSS=INDEC(0)
      GO TO 1001
1150 ITA=INDEC(0)
      CALL MESSAGE(15HNO HOLD OPTICN&)
      GO TO 1001

1160 IREMO=1
      IWD=ITDEC(4)
      DATA SWD/4HSOUR/
      IF(WD.EQ.SWD) GO TO 1001
      GO TO 1002

1170 NATOL=INDEC(0)
      NATOL=NTP(NATOL)
      GO TO 1001

1300 NROW=INDEC(1)
      IF(NEWDEC(0).NE.0) CALL ERROR (63HGIVE VALUES FOR NROW*NCOL AFTER
      *DEFINING RESTART. (SETUP,1300)&)
      IWD=ITDEC(1)
      NCOL=INDEC(0)
      IREST=INDEC(0)
      NR=NCOL
      NZ=NROW
      CALL STORE
      NGAUSS=IREST
      IF(NGAUSS.LT.0) NGAUSS=0
1301 DO 1302 I=1,60
1302 ISSRCE(I)=-1
      IWD=ITDEC(4)
      CALL INDEX(J,WD,5HSOUR&)
      GO TO (1303,1307),J
1303 IF(NEWDEC(0))1306,1305,1304
1304 IWD=INDEC(1)
      ISSRCE(IWD)=1
      GO TO 1303
1305 CALL ERROR (65HPROGRAM CANNOT IDENTIFY DECLARATIONS AFTER SOURCES.
      * (SETUP,1305)&)
1306 IWD=ITDEC(4)
1307 RETURN

1200 IF(NEWDEC(0)) 1005,1305,1201
1005 IWD=ITDEC(4)
      GO TO 1002
1201 IWD=INDEC(1)
      GO TO 1200

1010 CALL MESH
      GO TO 1002

=====
=====START ZONE DATA=====

1020 AAA=FLDEC(1)
      DO 1021 I=1,NTYPE
1021 ITYPE(I)=AAA
1025 IF(NEWDEC(0)) 1026,1023,9999
1023 IWD=ITDEC(1)
      IST=INDEC(0)+1
      JND=NROW-INDEC(0)-2
      IND=INDEC(0)

```

10

```

      JST=NROW-INDEC(1)-1
      MODI=NEWDEC(0)
      IWD=ITDEC(1)
      AAA=FLDEC(1)
      DO 1024 J=JST,JND
      IA=J*(NCOL-1)
      DO 1024 I=IST,IND
      IAA=IA+I
1024  ITYPE(IAA)=AAA
      GO TO 1025

9999 CALL ERROR(64HFOUND NUMBER INSTEAD OF WORD OR SPECIAL CHARACTER. (
      *SETUP,9999) & )
1026 IWD=ITDEC(4)
      RETURN

===== ON LEAVING SAUCE, WD CONTAINS THE NEXT DIRECTIVE=====

      END

```

```

SUBROUTINE DATA
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/DALINK/IAST,IDST,ISCA,IMAT,NCC,MCO,NGLIB,NSCAT,NISO,IB,
1  PATSTT,NEWGP(20),S(16,4),MTAPE
COMMON/PHI/PHI(17000)
COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1  G,NIT,IIDIST
DIMENSION D(1),A(1),SCA(1)
DIMENSION AWD(2)
INTEGER G
EQUIVALENCE (PHI(1),D(1),A(1),SCA(1))
EQUIVALENCE (WD,IWD)
EQUIVALENCE (IS,IBUFF,IX,IPOS),(ISUF,ISUFA),(NSCA,NND)

NISO=0
IAST=0
IDST=NMAT*NG
ISCA=2*IDST
96 IWD=ITDEC(4)
97 CALL INDEX(J,WD,16HCRAM*RASH*I*LIB&)
GO TO (100,200,300,400, 50), J

=====

100 CALL CRAM
GO TO 97

=====

200 CALL WORDEC(AWD,6)
CALL INDEX(J,AWD,17HREAD*READTH*CALC&,6)
GO TO (201,203,204,202),J
201 CALL CRASH
GO TO 50
202 CALL ERROR (59HAFTER RASH PROGRAM EXPECTS READ,READTH OR CALC. (DA
      *TA,202) & )
203 NG=-NG

```

```

204 CALL RASHCO
REWIND NT8
GO TO 50
300 CALL ERROR (56HAFTER MATERIAL PROGRAM EXPECTS CRAM OR RASH. (DATA,
      *300) & )
400 CALL ERROR (56HAFTER MATERIAL PRCGRAM EXPECTS CRAM OR RASH. (DATA,
      *400) & )
50 LIDST=IDST+1
LIDND=IDST+NMAT
LIAST=IAST+1
LIAND=IAST+NMAT
WRITE(NT8) (D(I),I=LIDST,LIDND),(A(I),I=LIAST,LIAND)
WRITE(NT8) (D(I),I=LIDST,LIDND),(A(I),I=LIAST,LIAND)
IF(NG.EQ.1) GO TO 24
IPOS=ISCA+1
DO 23 G=2,NG
NSCA=IPOS
DO 21 IE=1,2
IPOS=NSCA
IBOT=MAX(1,G-NBACK)
ITOP=G-1
DO 22 IG=IBOT,ITOP
IND=IPOS+NMAT-1
WRITE(NT8) (SCA(I),I=IPOS,IND)
22 IPOS=IPOS+NMAT
ISUFD=IDST+(G-1)*NMAT+1
ISUFA=ISUFD-IDST+IAST
LIDND=ISUFD+NMAT-1
LIAND=ISUFA+NMAT-1
21 WRITE(NT8) (D(I),I=ISUFD,LIDND),(A(I),I=ISUFA,LIAND)
23 CONTINUE
24 REWIND NT8
RETURN
END

```

```

SUBROUTINE STORE
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON /NST/ NBNDST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,NAS
1T,NSGST, NSCAST,NPHIST,NTHEST,NYST,NBBBST ,NTST
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1  G,NIT,IIDIST
C
1012 IND=NCOLST+NCOL-1
LHALF(1)=NROW/2
LHALF(2)=NROW-LHALF(1)
MHALF(1)=NCOL*LHALF(1)
MHALF(2)=NCOL*LHALF(2)
NHALF(1)=(NCOL-1)*LHALF(1)
NHALF(2)=(NCOL-1)*(LHALF(2)-1)
NTYPE=(NCOL-1)*(NROW-1)
NUMIJ=NCCL*NROW
NBST=NHALF(1)+NCOL-1
NBNDST=NBST+MHALF(2)
NRST=NBNDST+4*NCOL+4*NROW
NZST=NRST+NCOL
1014 NAS2ST=NZST+NROW
NASST=NAS2ST+NCOL
NWST=NASST+NCOL

```

```

NXST=NWST+NCOL      1
NTST=NXST+NCOL      2
NDST=NTST+NCOL      3
NAST=NDST+NMAT      4
NSGST=NAST+NMAT      5
THE FOLLOWING OVERWRITE W X D A SG      6
NSCAST=NWST          7
NPHIST=NCAST+NMAT    8
IF(NPHIST+MHALF(2).LT.17000) GO TO 1     9
CALL MESSAGE(52HSTORAGE WILL OVERFLOW DURING CALCULATION OF SCATTER 10
1& )
CALL MESSAGE(32HONLY ONE GROUP WILL BE ALLOWED & ) 11
NPHIST=0             12
PHI AND THE FOLLOWING OVERWRITE THE PREVIOUS QUANTITIES BUT SHOULD 13
STOP BEFORE REACHING AS2 AS W X D A SG    14
1 NTHEST=NUMIJ      15
NYST=NTHEST+NCOL   16
NBBBST=NYST+NCOL   17
=====CHECK THIS DOES NOT EXTEND BEYOND ALLOWED LIMITS===== 18
MAX2=NBBBST+NCOL   19
1015 IF(MAX2.GE.NAS2ST) CALL ERROR (35HMUST REORDER STORAGE. (STORE,101 20
*5)& )
RETURN              21
END                  22

```

```

SUBROUTINE MESH      32
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT 33
COMMON /NUM/ NBACK  34
COMMON /NST/ NBNDST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,NAS 35
IT,NSGST, NSCAST,NPHIST,NTHEST,NYST,NBBBST ,NTST
COMMON /COEF/ SE,SW,NE,NW,RO,OE,CN,CW,CS 36
COMMON/SOURCE/SOURCE(100),ISSRCE(60) 37
COMMON /REST/ IREST 38
COMMON/GEOM/GEOM 39
COMMON/PHI/PHI(17000) 40
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2), 41
1 G,NIT,IDI ST      42
DIMENSION TMESH(1) 43
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),AS2(1),AS(1) 44
DIMENSION SCATER(1),AE(1),AO(1),D(1),A(1),SG(1) 45
DIMENSION THETA(1),Y(1),BBB(1) 46
INTEGER GEOM 47
EQUIVALENCE (PHI(1),TMESH(1)) 48
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1), 49
1 AS2(1),AS(1),SCATER(1)) 50
EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1)) 51
EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1)) 52
EQUIVALENCE (AO(1),W),(AE(1),X) 53
EQUIVALENCE (WD,IWD) 54
===== START MESH CALCULATION ===== 55
NN=0 56
1010 IWD=ITDEC(4) 57
2001 CALL INDEX(J1,WD,12HMR*MZ*MX*MY&) 58
GO TO (1004,1005,1004,1005,1012),J1 59

```

```

1004 NCOLST=NN+1      1
GO TO 2010           2
1005 NROWST=NN+1     3
2010 NST=NN          4
NN=NN+1             5
LAST=INDEC(1)+1     6
106 I=NEWDEC(0)      7
102 CALL CHDEC(WD,1) 8
104 TME SH(NN)=FLEDC(1) 9
OLDMSH=TME SH(NN) 10
1103 IF (NEWDEC(0))1008,9995,1101 11
1101 NEXT=INDEC(1) 12
107 IF(NEWDEC(0))101,105,101 13
105 CALL CHDEC(WD,1) 14
1104 NEXT=NEXT+1    15
FARM SH=FLEDC(1)   16
IF(FARM SH.GT.OLDMSH.AND.NEXT.GT.LAST) GO TO 1105 17
101 CALL MESSAGE(27HDATA ERROR AT MESH POINT *&, NEXT-1) 18
CALL MESSAGE(44HWILL CCNTINUE TO READ DATA BUT NO EXECUTION& ) 19
ACCUR(4)=0.        20
1105 CONTINUE      21
ADD=(FARM SH-OLDMSH)/FLOAT(NEXT-LAST) 22
1102 NN=NN+1       23
TME SH(NN)=TME SH(NN-1)+ADD 24
IF(NN-NST.LT.NEXT) GO TO 1102 25
TME SH(NN)=FARM SH 26
LAST=NEXT          27
OLDMSH=FARM SH    28
GO TO 1103         29
9995 CALL CHDEC(NEXT,1) 30
CALL ERROR (59HREAD SPECIAL CHARACTER INSTEAD OF MESH NUMBER. (MES 31
*H,9995)& ) 32
1008 NEXT=ITDEC(4) 33
IWD=NEXT           34
GO TO (1009,1011,1009,1011),J1 35
1009 NCOL=NN -NROW 36
NR=NCOL           37
GO TO 2001        38
1011 NROW=NN -NCOL 40
NZ=NROW           41
GO TO 2001        42
1012 CALL STORE    43
DO 2012 I=1,NCOL 44
IT=NCOLST+I-1     45
IR=NRST+I         46
2012 RME SH(IR)=TME SH(IT) 47
DO 1013 I=1,NROW 48
IR=NZST+I         49
IT=NROWST+NROW-I 50
1013 ZME SH(IR)=TME SH(IT) 51
=====END OF MESH===== 53
RETURN            54
END               55

```

```

SUBROUTINE NITER(WD,IWD)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NUM/ NBACK
COMMON /COUNT/ COUNT(60)
INTEGER COUNT
DATA ISTAR /4H* /

IC=0
1 IF(NEWDEC(0)) 10,101,21
2 IBUFF=INDEC(1)
2 IF(NEWDEC(0)) 3,31,5
3 ISTO=ITDEC(1)
GO TO 32
3 ISTO=ITDEC(4)
32 IF(ISTO.EQ.ISTAR) GO TO 6
IC=IC+1
DO 4 IX=IC,NG
4 COUNT(IX)=IBUFF
GO TO 11
5 ISTO=INDEC(1)
IC=IC+1
IF(IC.GT.NG) GO TO 8
COUNT(IC)=IBUFF
IBUFF=ISTO
GO TO 2
6 ISTO=INDEC(0)
ID=IC+IBUFF
ID=MINO(ID,NG)
IC=IC+1
DO 7 IX=IC,ID
7 COUNT(IX)=ISTO
IC=ID
GO TO 1
8 CALL MESSAGE(27HCOUNTS FOR TCO MANY GROUPS&)
9 IF(NEWDEC(0))10,101,91
91 IWD=INDEC(1)
GO TO 9
10 IWD=ITDEC(4)
GO TO 12
101 IWD=ITDEC(1)
GO TO 12
11 IWD=ISTO
12 RETURN
END

```

```

SUBROUTINE CRAM
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /DALINK/ IAST,IDST,ISCA,IMAT,NCC,MCO,NGLIB,NSCAT,NISO,IB,
1 MATSTT,NEWGP(20),S(16,4),MTAPE
COMMON /PHI/ PHI(17000)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NUM/ NBACK
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIST
DIMENSION D(1),A(1),SCA(1)
INTEGER G
EQUIVALENCE (PHI(1),A(1),D(1),SCA(1))
EQUIVALENCE (WD,IWD)
EQUIVALENCE (IS,IBUFF,IX,IPOS),(ISUF,ISUFA),(NSCA,NND)

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
100 NCO=(NMAT*(NBACK*(2*NG-NBACK-1)+4*NG))/2
IF(NCO.GT.17000) GO TO 10
MCO=(NG-NBACK)*NBACK+(NBACK*(NBACK-1))/2 +2*NG
NCO=MCO
NGLIB=NG
NSCAT=NBACK
6 IF(NEWDEC(0))95,99,5
95 IWD=ITDEC(4)
GO TO 97
99 CALL ERROR (62HREAD SPECIAL CHARACTER WHILE READING MATERIAL DATA.
* (CRAM,99) &)
5 IMAT=INDEC(1)
CALL ISOTOP(1)
GO TO 97
10 CALL ERROR (25HTOO MUCH DATA. (CRAM,10) &)
96 IWD=ITDEC(4)
97 CALL INDEX(J,WD,6HI*LIB&)
GO TO (300,400,20),J
300 CALL ISOTOP(2)
GO TO 97
400 IWD=ITDEC(4)
CALL INDEX(J,WD,19HANL*ROAC*KFKI*NUSY&)
GO TO (401,401,402,403,401), J
401 CALL ROACH
NGLIB=16
NSCAT=5
NCO=97
NISO=40
GO TO 96
***** ISOKFK EINGEFUEGT FUER VERWENDUNG VON NUSYS-QUERSCHN. *****
***** ARGUMENT VON ISOKFK=1: KFKINR-DATA-SET *****
***** ARGUMENT VON ISOKFK=2: VCRHERIGER NUSYS-LAUF *****
402 CALL ISOKFK (1)
GO TO 96
403 CALL ISOKFK (2)
GO TO 96
=====
20 CALL INDEX(J,WD,8HM*I*Z*S&)
GO TO (40,300,99,30,
50),J
30 IS=INDEC(0)
DO 31 I=1,NGLIB
31 S(I,IS)=FLDEC(0)
IF(NEWDEC(0))201,331,332
201 IWD=ITDEC(4)
GO TO 20
331 IWD=ITDEC(1)
GO TO 34
332 WD=FLDEC(1)
33 I=0
34 I=I+1
IF(NEWDEC(0))1038,35,36
36 NEWGP(I)=INDEC(1)
IF(NEWDEC(0))38,371,39
35 ITCH=ITDEC(1)
NEWGP(I)=WD+.001

```

```

GO TO 37
371 IWD=I TDEC(1)
37 WD=FLDEC(1)
GO TO 34
39 CALL ERROR (23HINPUT ERROR. (CRAM,39)E )
1038 NEWGP(I)=WD+.001
38 IF(I.NE.NG) GO TO 39
IWD=I TDEC(4)
GO TO 20

=====
40 CALL MIXES
DO 71 I=1,MTAPE
READ(NT9) IMAT
BACKSPACE NT9
LIDST=IDST+IMAT
LIAND=IDST+IMAT+(NG-1)*NMAT
LIAND=IMAT+(NG-1)*NMAT
LISST=ISCA+IMAT
LISND=ISCA+(MCO-2*NG-1)*NMAT
READ(NT9) IMAT,(PHI(G),G=LIDST,LIDNO,NMAT),(PHI(G),G=IMAT,L IAND,NM
1 AT),(PHI(G),G=LISST,LISND,NMAT)
71 CONTINUE

REWIND NT9
50 RETURN
END

```

```

1 ITOP=MINO(NG,NBACK+G)
2 IBOT=G+1
3 IF(G.LE.NBACK) ISUF=ISUF+NMAT*((G*(G+1))/2 -1)
4 IF(G.GT.NBACK) ISUF=ISUF+NMAT*(NBACK*(G-NBACK)+ (NBACK*(NBACK+1) )/
5 1 2-1)
6 ADD=MINO(G,NBACK-1)
7 DO 4 IG=IBOT,ITOP
8 SCA( ISUF )=FLDEC(0)
9 ISUF=ISUF+ADD*NMAT
10 IF(ADD.LT.NBACK-1) ADD=ADD+1
11 4 CONTINUE
12 3 CONTINUE
13
14 6 IF(NEWDEC(0).GT.0)GO TO 51
15 IWD=I TDEC(4)
16 RETURN
17 51 IMAT=INDEC(1)
18 GO TO 5
19
20 =====
21
22 32 CONTINUE
23 30 IMAT=INDEC(0)
24 IND=I MAT*NCO
25 IST=IND-NCO+1
26 DO 31 I=IST,IND
27 31 PHI(I)=F LDEC(0)
28 NISO=MAX0(NISO,IMAT)
29
30 IWD=I TDEC(4)
31 IF(WD.EQ.DATI)GO TO 30
32 RETURN
33 END

```

-74-

```

SUBROUTINE ISOTOP(IGO)
COMMON/DALINK/IAST,IDST,ISCA,IMAT,NCC,MCO,NGLIB,NSCAT,N ISO, IB,
1 MATSTT,NEWGP(20),S(16,4) ,MTAPE
COMMON/PHI/PHI(17000)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDI ST
DIMENSION D(1),A(1),SCA(1)
INTEGER ADD
INTEGER G
EQUIVALENCE (PHI(1),A(1),D(1),SCA(1))
EQUIVALENCE (WD,IWD)
DATADATI /4HI /

GO TO (5,32),IGO
5 ISUF=I MAT+IDST
DO 1 IG=1,NG
D( ISUF )=FLDEC(0)
1 ISUF=ISUF+NMAT
ISUF=I MAT+IAST
DO 2 IG=1,NG
A( ISUF )=FLDEC(0)
2 ISUF=ISUF+NMAT
ISUF=I MAT+ISCA
IF(NG.EQ.1) GO TO 6
NG1=NG-1
DO 3 G=1,NG1
SCATTER IS OUT OF GROUP G
ISUF=I MAT+ISCA

```

```

SUBROUTINE ROACH
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/PHI/PHI(17000)
DATA IIN /0/
IF (IIN.EQ.0) READ(NT4) (PHI(I),I=1,3783)
DO 10 K=1,97
IF (PHI(3783+K).LE.1.E-04) PHI(3783+K) = 1./3.
10 CONTINUE

IIN=1
IST=1
IND=16
DO 1 J=1,40
DO 2 I=IST,IND
2 PHI(I)=1./(3.*PHI(I))
IST=IST+97
1 IND=IND+97
RETURN
END

```

63

57

```

SUBROUTINE MIXES
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/PHI/PHI(17000)
COMMON/DALINK/IAST,IDST,ISCA,IMAT,NCC,MCC,NGLIB,NSCAT,NISO,IB,
1 MATSTT,NEWGP(20),S(16,4),MTAPE
COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODF, LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIST
DIMENSION JORDER(20),IORDER(20)
DIMENSION AWD(4)
EQUIVALENCE (WD,IWD)
DATA AWD /4HMIXT,4HURE,4HMATE,4HRIAL/

(1)ENTRY POINT
PILOT FOR MIXTURES FROM
ISOTOPES(MADD=0)
TABULATES ORDER OF -
-PRESENTATION
INITIALISES-PHI=0

MADD=0
NGQ=NGLIB
MTAPE=0
IDO=0
MATSTT=NCO*NISO
301 CALL INDEX(J1,IWD,8HI*M*Z*S&)
GO TO (40,20,99),J1
20 IGO=1
IB1=IB+1
IF(MADD.EQ.0)IB=MATSTT+IDO*MCO
21 IDO=IDO+1
22 IMAT=INDEC(0)
IF(MADD.EQ.0)IORDER(IMAT)=IDO
JORDER(IDO)=IMAT
26 DO 23 IZ=1,NCO
IY=IB+IZ
23 PHI(IY)=0.
24 IWD=ITDEC(4)
IF(MADD)27,27,100
27 CALL INDEX(J1,IWD,8HI*M*Z*S&)
GO TO (40,28,28,99),J1
28 CALL SQUEEZE
GO TO 30

=====
(2)PERFORMS MIXING
40 ISO=INDEC(0)
IX=(ISO-1)*NCO
IF(MADD.EQ.1) IX=(IORDER(ISO)-1)*MCO+MATSTT
FCTION=F LDEC(0)
LCO=NCO
IF(MADD.EQ.1) LCO=MCO
DO 43 IZ=1,LCO
IY=IB+IZ
IX=IX+1
IF(IZ.LE.NGQ) GO TO 42
PHI(IY)=PHI(IY)+FCTION*PHI(IX)
GO TO 43
42 IF (PHI(IY).EQ.0.) PHI(IY) = 1.0E72
PHI(IY)=1./(1./PHI(IY)+FCTION/PHI(IX))
43 CONTINUE

```

```

1 GO TO 24
2
3 =====
4
5 (3)PRINTS OUT MIXTURES AND
6 MATERIALS
7
8 30 WRITE(6,31) AWD(2*MADD+1),AWD(2*MADD+2),IMAT
9
10 31 FORMAT(/1XA6,A4,I2 )
11 IB1=IB+1
12 IB2=IB+NG
13 IB2=IB1+NG -1
14 WRITE(6,32)(PHI(I),I=IB1,IB2)
15 IB1=IB2+1
16 IB2=IB2+NG
17 WRITE(6,32)(PHI(I),I=IB1,IB2)
18 32 FORMAT(1P10E12.5)
19 IB1=IB2+1
20 IB2=IB+MCO
21 WRITE(6,32)(PHI(I),I=IB1,IB2)
22
23 IF(MADD) 33,33,46
24 33 IF(MTAPE) 44,34,44
25 34 GO TO (98,70,50,35,98),J1
26 35 IWD=ITDEC(4)
27 CALL INDEX(J1,IWD,8HI*M*Z*S&)
28 GO TO 34
29
30 =====
31 (4.1.1)WRITES ON TAPE IF MIXTU
32 -RES WILL NOT FIT IN
33 CORE
34
35 44 JST=MATSTT+1
36 JND=MATSTT+MCO
37 DO 45 M=1,IDO
38 WRITE(NT9)JORDER(M),(PHI(I),I=JST,JND)
39 JST=JND+1
40 45 JND=JND+MCO
41 MTAPE=MTAPE+IDO
42 IDO=0
43 GO TO 34
44
45 =====
46 (4.2)WRITES MATERIALS ON
47 TAPE ANYWAY
48
49 46 WRITE(NT9) JORDER(1),(PHI(I),I=1,MCC)
50 MTAPE=MTAPE+1
51 IDO=0
52 GO TO 102
53
54 =====
55 (5)READS BACK MIXTURES IF
56 NECESSARY.OVERWRITES
57 ISOTOPES THEREFORE
58 MORE ROOM
59
60 50 MADD=1
61 IF(MTAPE.EQ.0) GO TO 54
62 REWIND NT9
63 MATSTT=MCO
64 JST=MATSTT+1

```

171


```

JND=MATSTT+MCO
DO 53 M=1,MTAPE
READ(NT9) IO,(PHI(I),I=JST,JND)
IORDER(IO)=M
JORDER(M)=IO
JST=JND+1
53 JND=JND+MCO
54 REWIND NT9

```

```

IB=0
NGQ=NG
MTAPE=0
IDO=0
WE KNOW WD=Z SC WE MUST
GO TO 20

```

```

=====
70 IF(IB+NCO.GT.17000) GO TO 44
GO TO 20

```

```

=====
100 CALL INDEX(J1,IWD,6HM*Z*S&)
GO TO (40,30,97,30),J1

```

```

102 CALL INDEX(J2,IWD,4HZ*S&)
GO TO (20,103,104),J2
103 IWD=ITDEC(4)
GO TO 102

```

```

104 REWIND NT9
RETURN

```

```

97 CALL ERROR (35HPROGRAM EXPECTS M OR Z. (MIXES,97)& )
98 CALL ERROR (37HPROGRAM EXPECTS M,Z OR S. (MIXES,98)& )
99 CALL ERROR (39HPROGRAM EXPECTS I,M,Z OR S. (MIXES,99)& )
STOP
END

```

```

-----
SUBROUTINE SQUEEZE
COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBAC
COMMON/DALINK/IAS,IDST,ISCA,IMAT,NCC,MCC,NGLIB,NSCAT,N ISO, IB,
1 MATSTT,NEWGP(20),S(16,4),MTAPE
COMMON/PHI/PHI(17000)

```

```

NBA=MINO(NSCAT,NG-1)
ID=IB
IX=IB
NGL=NG-1
IF(NG.EQ.NGLIB) GO TO 8
NS=INDEC(0)
DO 1 IE=1,2
DO 1 IG=1,NG
ID=ID+1

```

```

1 PHID=0.
2 NST=NEWGP(IG-1)+1
3 IF(IG.EQ.1)NST=1
4 NND=NEWGP(IG)
5 DO 3 IN=NST,NND
6 IX=IX+1
7 PHID=PHID+S(IN,NS)*PHI(IX)
8 PHI(ID)=PHID
9 NLIM=NGLIB-NSCAT
10 DO 2 IG=1,NG1
11 NTOP=MINO(IG+NBA,NG)
12 NBOT=IG+1
13 DO 2 JG=NBOT,NTOP
14 ID=ID+1
15 PHID=0.

```

```

16
17 =SCATTER IG,JG
18 MST=NEWGP(IG-1)+1
19 IF(IG.EQ.1)MST=1
20 MND=NEWGP(IG)
21 DO 4 KG=MST,MND
22 IX=NSCAT*(KG-1) +IB+2*NGLIB
23 IF(KG.GT.NLIM)IX=IX-((KG-NLIM)*(KG-NLIM-1))/2
24 NST=NEWGP(JG-1)+1
25 NND=MINO(NEWGP(JG),KG+NSCAT)
26 IF(NND-NST) 4,11,11

```

```

27 IX=IX+NST-KG-1
28 DO 10 LG=NST,NND
29 IX=IX+1
30 PHID=PHID+S(KG,NS)*PHI(IX)
31 4 CONTINUE
32 2 PHI(ID)=PHID
33

```

```

34 8 IF(NBA.EQ.NBACK) GO TO 12
35 ID=IB+2*NG
36 IX=ID
37 NBAC=NBACK-1
38 DO 5 IG=1,NG1
39 IF(NBAC.EQ.0) GO TO 9
40 NBB=MINO(NBAC,NG-IG)
41 DO 6 IA=1,NBB
42 ID=ID+1
43 IX=IX+1
44

```

```

45 6 PHI(ID)=PHI(IX)
46 IF(NBAC.EQ.NG-IG) GO TO 5
47 PHID=0.
48 NFIN=MINO(NBA-NBAC,NG-IG-NBAC)
49 DO 7 IC=1,NFIN
50 IX=IX+1
51 PHID=PHID+PHI(IX)
52 ID=ID+1
53 PHI(ID)=PHID
54 5 CONTINUE
55 12 RETURN
56 END

```

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55

```

SUBROUTINE CRASH
COMMON/DALINK/IAST,IDST,ISCA,IMAT,NCC,MCC,NGLIB,NSCAT,NISO,IB,
1 MATSTT,NEWGP(20),S(16,4),MTAPE
COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT,NBACK
1K
COMMON/PHI/PHI(17000)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRDIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIST

DIMENSION A(1),D(1),SCA(1)
INTEGER G
EQUIVALENCE (PHI(1),A(1),D(1),SCA(1))
EQUIVALENCE (WD,IWD)

10 NCO=(NMAT*(NBACK*(2*NG-NBACK-1)+4*NG))/2
IF(NCO.GT.16000) GO TO 98
IAST=0
IDST=NMAT*NG
ISCA=2*IDST
1 IF(NEWDEC(0))40,99,3
3 IMAT=INDEC(1)
ISUFD=IDST+IMAT
ISUFS=ISCA+IMAT
ISUFA=IAST+IMAT
DO 11 IG=1,NG
D(ISUFD)=FLDEC(0)
A(ISUFA)=D(ISUFD)*FLDEC(0)
SCA(ISUFS)=D(ISUFD)*FLDEC(0)
A(ISUFA)=A(ISUFA)+SCA(ISUFS)
ISUFD=ISUFD+NMAT
ISUFA=ISUFA+NMAT
11 ISUFS=ISUFS+NMAT
GO TO 1

99 CALL ERROR (52HREAD SPECIAL CHARACTER IN MATERIAL DATA. (CRASH,99)
*E )
98 CALL ERROR (38HTOO MUCH DATA.NC SORT YET. (CRASH,98)& )
40 IWD=ITDEC(4)
RETURN
END

```

```

SUBROUTINE RASHCO
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /XTAPE/ NATOL
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON/PHI/PHI(17000)
COMMON/LASTGP/LASGPS,NUMDN2,MATER,ICUT,ITEMP,ITHERM,IRTHEM ,I
COMMON/DALINK/IAST,IDST,ISCA,IMAT,NCC,MCC,NGLIB,NSCAT,NISO,IB,
1 MATSTT,NEWGP(20),S(16,4),MTAPE
COMMON ACCUR(4),WD,OMEGA,DIFF,PRDIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIST
DIMENSION A(27),SIGO(27),SIGS(27),SIGIS(27,2),SIGES(27,2),
1 GCOS(27,2),XSI(27,2),WEIGHT(50,27),ELETH(2)
DIMENSION EMUBAR(27)
DIMENSION SIGMAO(50),SIGTRA(50),SIGSSI(50),RHO(50)
DIMENSION TEMP(50),KMAT(50),ERG(60)
DIMENSION ACO(1),SCA(1),D(1)
DIMENSION DAS(3,60)

```

```

EQUIVALENCE (PHI(14901),A(1)),(PHI(14928),SIGO(1)),(PHI(14955),
1 SIGS(1)),(PHI(14982),SIGIS(1,1)),(PHI(15036),SIGES(1,1)),
2 (PHI(15090),GCOS(1,1)),(PHI(15144),XSI(1,1)),(PHI(15198),
3 WEIGHT(1,1)),(PHI(16548),ELETH(1)),(PHI(16550),NMIX)
4 EQUIVALENCE (PHI(16551),SIGMAO(1)),(PHI(16601),SIGTRA(1)),
5 (PHI(16651),SIGSSI(1)),(PHI(16701),RHO(1))
6 EQUIVALENCE (PHI(16751),TEMP(1)),(PHI(16801),KMAT(1)),
7 (PHI(16851),ERG(1))
8 EQUIVALENCE (PHI(16911),EMUBAR(1))
9 EQUIVALENCE (WD,IWD)
10 EQUIVALENCE (PHI(1),ACO(1),SCA(1),D(1))
11
12 ICUT=0
13 ITEMP=0
14 ITHERM=0
15 IF(NG.LT.0) IRTHEM=1
16 IF(NG.LT.0) NG=-NG
17 IF(NEWDEC(0))1,2,3
18 2 CALL ERROR (24HINPUT ERROR. (RASHCO,2)& )
19 1 IWD=ITDEC(4)
20 CALL INDEX(ITEMP,IWD,5HTEMP)
21 IF(ITEMP.NE.1)CALL ERROR (24HINPUT ERROR. (RASHCO,1)& )
22 3 IWD=INDEC(0)
23 DO 4 I=1,NMAT
24 KMAT(IWD)=INDEC(0)
25 IF(ITEMP.NE.1) GO TO 4
26 TEMP(IWD)=FLDEC(0) +273.
27 4 IWD=INDEC(0)
28 NG=IWD
29 NG1=NG-1
30 DO 5 I=1,NG1
31 5 ERG(I)=FLDEC(0)
32 IF(NEWDEC(0)) 6,2,8
33 6 IWD=ITDEC(4)
34 ICUT=1
35 GO TO 9
36 8 ERG(NG)=FLDEC(1)
37 5 IF(NEWDEC(0))10,2,11
38 10 IWD=ITDEC(4)
39 ITHERM=1
40 GO TO 12
41 11 ERG(NG+1)=FLDEC(1)
42
43 12 NMIX=0
44 16 IF(NEWDEC(0))15,151,14
45 14 IWD=INDEC(1)
46 RHO(IWD)=FLDEC(0)
47 J=0
48 13 J=J+1
49 IF(NEWDEC(0)) 2,17,191
50 191 WEIGHT(IWD,J)=FLDEC(1)
51 GO TO 19
52 17 JWD=ITDEC(1)
53 J=J-2
54 IB=WEIGHT(IWD,J+1) +.001
55 WE=FLDEC(0)
56 DO 18 IBB=1,IB
57 J=J+1
58 18 WEIGHT(IWD,J)=WE
59 19 IF(J.LT.27) GO TO 13
60 NMIX=MAX0(NMIX,IWD)
61 GO TO 16
62
63

```

```

151 IWD=ITDEC(1)
    GO TO 152
15 IWD=ITDEC(4)
152 READ(NATOL) (A(I),I=1,27)
    READ(NATOL) (SIGO(I),I=1,27)
    READ(NATOL) (SIGS(I),I=1,27)
    READ(NATOL) (EMUBAR(I),I=1,27)
    DO 20 I=1,NMIX
      SIGMAO(I)=0.
    DO 20 J=1,27
20 SIGMAO(I)=SIGMAO(I)+.6023*WEIGHT(I,J)*RHC(I)*SIGO(J)/A(J)
    CALL PENDLE
    REWIND NT8
    DO 22 I=1,NMAT
      ISUFD=IDST+I
      ISUFS=ISCA+I
      ISUFA=IAST+I
      READ(NT8)((DAS(J,IG),J=1,3),IG=1,NG)
      DO 21 IG=1,NG
        D(I SUFD)=DAS(1,IG)
        ACO(I SUFA)=D(I SUFD)*(DAS(2,IG)+DAS(3,IG) )
        SCA(I SUFS)=D(I SUFD)*DAS(3,IG)
        ISUFD=ISUFD+NMAT
        ISUFA=ISUFA+NMAT
21 ISUFS=ISUFS+NMAT
22 CONTINUE
    REWIND NT8
    RETURN
    END

```

```

SUBROUTINE PENDLE
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON /XTAPE/ NATOL
COMMON/PHI/PHI(17000)
COMMON/LASTGP/LASGPS,NUMON2,MATER,ICUT,ITEMP,ITHEM ,IRTHEM ,I
DIMENSION D(1),AKAP2(1),ALPH2(1)
DIMENSION A(27),SIGO(27),SIGS(27),SIGIS(27,2),SIGES(27,2),
1 GCOS(27,2),XSI(27,2),WEIGHT(50,27),ELETH(2)
DIMENSION EMUBAR(27)
DIMENSION SIGMAO(50),SIGTRA(50),SIGSSI(50),RHC(50)
DIMENSION TEMP(50),KMAT(50),ERG(60)
INTEGER G
EQUIVALENCE (WD,IWD),(PHI(1),D(1),AKAP2(1),ALPH2(1))
EQUIVALENCE (PHI(14901),A(1)),(PHI(14928),SIGO(1)),(PHI(14955),
1 SIGS(1)),(PHI(14982),SIGIS(1,1)),(PHI(15036),SIGES(1,1)),
2 (PHI(15090),GCOS(1,1)),(PHI(15144),XSI(1,1)),(PHI(15198),
3 WEIGHT(1,1)),(PHI(16548),ELETH(1)),(PHI(16550),NMIX)
EQUIVALENCE (PHI(16551),SIGMAO(1)),(PHI(16601),SIGTRA(1)),
1 (PHI(16651),SIGSSI(1)),(PHI(16701),RHC(1))
EQUIVALENCE (PHI(16751),TEMP(1)),(PHI(16801),KMAT(1)),
1 (PHI(16851),ERG(1))
EQUIVALENCE (PHI(16911),EMUBAR(1))

G=1
IBAND=1
ELE1=ALOG(14./ERG(1))
FRIG TO AVOID ALOG(CONST)

```

```

1 X14=14.
2 X100=100.
3 X40G0=40CO.
4 X1400=1400.
5 X64E5=.64E5
6 IF (ERG(1).GT.1.) ISKIP=50.*ELE1/ALCG(X14)
7 IF (ERG(1).LE.1..AND.ERG(1).GT.1.E-2) ISKIP=51.+100.*
8 C (ELE1-ALOG(X14))/ALOG(X100)
9 IF (ERG(1).LE.1.E-2.AND.ERG(1).GT.1.E-4) ISKIP=151.+50.*
10 C (ELE1-ALOG(X1400))/ALOG(X100)
11 IF (ERG(1).LE.1.E-4) ISKIP=201.+50.*(ELE1-ALOG(X64E5))/ALOG(X4000)
12 DO 1 I=1,ISKIP
13 READ(NATOL)
14 READ(NATOL)ELETH(1),(SIGES(J,1),J=1,27),(SIGIS(J,1),J=1,27)
15 1 (GCOS(J,1),J=1,27),(XSI(J,1),J=1,27)
16 READ(NATOL)ELETH(2),(SIGES(J,2),J=1,27),(SIGIS(J,2),J=1,27),
17 1 (GCOS(J,2),J=1,27),(XSI(J,2),J=1,27)
18 ISKIP=ISKIP+2
19 H2=ELETH(2)-ELETH(1)
20 H1=H2
21 IF(ELETH(1).EQ.ELE1) GO TO 2
22 INTERPOLATE INTO (1)
23 H1=ELETH(2)-ELE1
24 RAT1=(ELE1-ELETH(1))/H2
25 RAT2=H1/H2
26 DO 3 J=1,27
27 SIGES(J,1)=RAT2*SIGES(J,1)+RAT1*SIGES(J,2)
28 SIGIS(J,1)=RAT2*SIGIS(J,1)+RAT1*SIGIS(J,2)
29 GCOS(J,1)=RAT2*GCOS(J,1)+RAT1*GCOS(J,2)
30 XSI(J,1)=XSI(J,1)*RAT2+RAT1*XSI(J,2)
31 2 CALL MACROX(1)
32
33 12 IK=3*(G-NG)
34 DO 25 I=1,NMIX
35 IK=IK+3*NG
36 ID=IK-2
37 D(ID)=0.
38 25 AKAP2(IK)=0.
39 IK=3*(G-NG)
40 DO 4 I=1,NMIX
41 IK=IK+3*NG
42 ID=IK-2
43 D(ID)=D(ID)+H1/(6.*SIGTRA(I))
44 4 AKAP2(IK)=AKAP2(IK)+H1/(6.*SIGTRA(I))*SIGSSI(I)
45
46 CALL MACROX(2)
47
48 SMULT=(H1+STEP(ISKIP))/2.
49 IK=3*(G-NG)
50 DO 5 I=1,NMIX
51 IK=IK+3*NG
52 ID=IK-2
53 D(ID)=D(ID)+SMULT/(SIGTRA(I)*3.)
54 5 AKAP2(IK)=AKAP2(IK)+SMULT/(3.*SIGTRA(I))*SIGSSI(I)
55 ELE2=ALOG(14./ERG(G+1) )
56 START INTEGRATION OVER EVEN SIZED STEPS
57 6 ISKIP=ISKIP+1
58 READ(NATOL)ELETH(1),(SIGES(J,1),J=1,27),(SIGIS(J,1),J=1,27),
59 1 (GCOS(J,1),J=1,27),(XSI(J,1),J=1,27)
60
61 CALL MACROX(1)
62
63 SMULT=(STEP(ISKIP-1)+STEP(ISKIP))/2.

```

10

```

1 IF(ELE2.LT.ELETH(1)+STEP(ISKIP))SMULT=(STEP(ISKIP-1)+ELE2-ELETH(1) 1
2 )/2. 2
3 IK=3*(G-NG) 3
4 DO 7 I=1,NMIX 4
5 IK=IK+3*NG 5
6 ID=IK-2 6
7 D(ID)=D(ID)+SMULT/(3.*SIGTRA(I)) 7
8 AKAP2(IK)=AKAP2(IK)+SMULT/(3.*SIGTRA(I)*SIGSSI(I)) 8
9 IF(ELE2.GE.ELETH(1)+STEP(ISKIP)) GO TO 6 9
10 INTERPOLATE FOR LAST STEP 10
11 READ(NATOL)ELETH(2),(SIGES(J,2),J=1,27),(SIGIS(J,2),J=1,27), 11
12 (GCOS(J,2),J=1,27),(XSI(J,2),J=1,27) 12
13
14 H2=ELETH(2)-ELETH(1) 14
15 H1=ELE2-ELETH(1) 15
16 RAT1=H1/H2 16
17 RAT2=(ELETH(2)-ELE2)/H2 17
18 DO 8 J=1,27 18
19 SIGES(J,1)=RAT2*SIGES(J,1)+RAT1*SIGES(J,2) 19
20 SIGIS(J,1)=RAT2*SIGIS(J,1)+RAT1*SIGIS(J,2) 20
21 GCOS(J,1)=RAT2*GCOS(J,1)+RAT1*GCOS(J,2) 21
22 XSI(J,1)=RAT2*XSI(J,1)+RAT1*XSI(J,2) 22
23
24 CALL MACROX(1) 24
25
26 IK=3*(G-NG) 26
27 DO 9 I=1,NMIX 27
28 IK=IK+3*NG 28
29 ID=IK-2 29
30 D(ID)=D(ID)+H1/(SIGTRA(I)*6.) 30
31 AKAP2(IK)=AKAP2(IK)+H1/(6.*SIGTRA(I)*SIGSSI(I)) 31
32 TO FINISH THIS GROUP FOR ALL MIXTURES WE NEED TO FIND ALPHA**2 32
33 AND INVERT AKAP2 INTO KAPPA**2. 33
34 IK=3*(G-NG) 34
35 DO 10 I=1,NMIX 35
36 IK=IK+3*NG 36
37 IA=IK-1 37
38 ID=IK-2 38
39 ALPH2(IA)=.31812E-3*SIGMAO(I)*(1./SQRT(ERG(G+1))-1./SQRT(ERG(G))) 39
40 /D(ID) 40
41 AKAP2(IK)=1./AKAP2(IK) 41
42 D(ID)=D(ID)/(ELE2-ELE1) 42
43
44
45 IF(G.EQ.NG) GO TO 13 45
46 IF(G.EQ.NG-1.AND.ITHERM.EQ.1) GO TO 13 46
47 IF(G.EQ.NG-1.AND.IRTHEM.EQ.1) GO TO 13 47
48 IF(G.EQ.NG-2.AND.ICUT.EQ.1) GO TO 13 48
49 IF(G.EQ.NG-2.AND.ITEMP.EQ.1) GO TO 13 49
50 G=G+1 50
51 ELE1=ELE2 51
52 ELE2=ALOG(14./ERG(G+1)) 52
53 ISKIP=ISKIP+1 53
54 H2=ELETH(2)-ELETH(1) 54
55 H1=H2 55
56 IF(ELETH(1).EQ.ELE1) GO TO 11 56
57 H1=ELETH(2)-ELE1 57
58 GO TO 12 58
59
60 DO LAST TWO GROUPS AND PUT CN TAPE 60
61
62 WE HAVE ALLOWED 3*NG PLACES FOR EACH OF NMIX MIXTURES 62
63

```

```

1 NOW EACH MATERIAL WE DO ,IN TURN,USES THE SPACES EMBEDDED IN THAT- 1
2 -OVERWRITING WHERE NECESSARY THE PREVIOUS ONE 2
3
4 NUMON2=0 3
5 OLDG=G 4
6 OLDELE=ELE2 5
7 OSKIP=ISKIP 6
8 DO 14 I=1,NMAT 7
9 ISKIP=OSKIP 8
10 ELE2=OLDELE 9
11 G=OLDG 10
12 IGO=0 11
13 MATER=KMAT(I) 12
14 IF(G.EQ.NG) GO TO 21 13
15 IF(G.EQ.NG-1) GO TO 19 14
16 G=G+1 15
17 CALCULATE GROUP ABOVE THERMAL 16
18 IF(ICUT.EQ.0) GO TO 15 17
19 STEMP=2.419E-10*TEMP( I ) 18
20 ERG(NG)=STEMP 19
21 ELE1=ELE2 20
22 ELE2=ALOG(14./ERG(NG)) 21
23 ISKIP=ISKIP+1 22
24 LASGPS=0 23
25 CALL RWTAPE(1,1) 24
26 CALL RWTAPE(1,2) 25
27 IF(I.NE.1) CALL MACROX(1) 26
28 H2=ELETH(2)-ELETH(1) 27
29 H1=H2 28
30 IF(ELETH(2).NE.ELE1) H1=ELETH(2)-ELE1 29
31 IK=3*NG*MATER-3 30
32 ID=IK-2 31
33 D(ID)=H1/(6.*SIGTRA(MATER) ) 32
34 AKAP2(IK)=H1/(6.*SIGTRA(MATER)*SIGSSI(MATER) ) 33
35
36 CALL MACROX(2) 34
37
38 SMULT=(H1+STEP(ISKIP))/2. 35
39 D(ID)=D(ID)+SMULT/(3.*SIGTRA(MATER) ) 36
40 AKAP2(IK)=AKAP2(IK)+SMULT/(3.*SIGTRA(MATER)*SIGSSI(MATER) ) 37
41 ISKIP=ISKIP+1 38
42 CALL RWTAPE(2,1) 39
43
44 CALL MACROX(1) 40
45
46 SMULT=(STEP(ISKIP-1)+STEP(ISKIP))/2. 41
47 IF(ELE2.LT.ELETH(1)+STEP(ISKIP))SMULT=(STEP(ISKIP-1)+ELE2-ELETH(1) 42
48 )/2. 43
49 D(ID)=D(ID)+SMULT/(3.*SIGTRA(MATER) ) 44
50 AKAP2(IK)=AKAP2(IK)+SMULT/(3.*SIGTRA(MATER)*SIGSSI(MATER) ) 45
51 IF(ELE2.GE.ELETH(1)+STEP(ISKIP))GO TO 17 46
52
53 CALL RWTAPE(2,2) 47
54 H2=ELETH(2)-ELETH(1) 48
55 H1=ELE2-ELETH(1) 49
56 RAT1=H1/H2 50
57 RAT2=(ELETH(2)-ELE2)/H2 51
58 DO 18 J=1,27 52
59 SIGES(J,1)=RAT2*SIGES(J,1)+RAT1*SIGES(J,2) 53
60 SIGIS(J,1)=RAT2*SIGIS(J,1)+RAT1*SIGIS(J,2) 54
61 GCOS(J,1)=RAT2*GCOS(J,1)+GCOS(J,2)*RAT1 55
62 XSI(J,1)=RAT2*XSI(J,1)+RAT1*XSI(J,2) 56
63
64 CALL MACROX(1) 57
65

```

```

D(ID)=D(ID)+H1/(6.*SIGTRA(MATER))
AKAP2(IK)=AKAP2(IK)+H1/(6.*SIGTRA(MATER)*SIGSSI(MATER))
IA=IK-1
ALPH2(IA)=.31812E-3*SIGMAO(MATER)*(1./SQRT(ERG(G+1))-1./SQRT(ERG(G
1))) / D(ID)
AKAP2(IK)=1./AKAP2(IK)
D(ID)=D(ID)/(ELE2-ELE1)
IF(IGO.EQ.4) GC TO 21
19 CONTINUE
ELE1=ELE2
IK=3*NG*MATER
ID=IK-2
IA=IK-1
IF(ITHERM.EQ.1) GO TO 23
IF(IRTHERM.EQ.1) GO TO 22
D(ID)=0.
AKAP2(IK)=0.
ALPH2(IA)=0.
IGO=4
ELE1=ELE2
ELE2=ALOG(14./ERG(NG+1))
G=G+1
H1=ELETH(2)-ELE1
GO TO 20
22 D(ID)=WD
REWIND NT9
ERG(NG+1)=0.
ALPH2(IA)=0.
AKAP2(IK)=FLDEC(0)
CALL WORDEC(WD,4)
GO TO 21
23 SA=SIGMAO(MATER)/1.128
REWIND NT9
IF(ITEMP.EQ.1) SA=SA*SQRT(293./TEMP(I))
SS=0.
AMU=0.
DO 24 J=1,27
ADD=SIGS(J)*.6023*RHO(MATER)*WEIGHT(MATER,J)/A(J)
SS=SS+ADD
24 AMU=AMU+ADD*EMUBAR(J)
AMU=AMU/SS

AL=DLEN(SA,SS,AMU)
AL2=AL*AL
D(ID)=AL2*SA
AKAP2(IK)=1./AL2
ALPH2(IA)=0.
ERG(NG+1)=0.
GO TO 21

21 CONTINUE
REWIND NT9
IST=(MATER-1)*3*NG+1
IND=IST+3*NG-1
WRITE(NT8) (D(IRW),IRW=IST,IND)
WRITE(6,29) I
29 FORMAT(//25H MATERIAL (REGION) NUMBER I4/13H GROUP BOUNDS21X
1 1HD9X8HALPHA**2 5X8HKAPPA**2//)
DO 30 IG=1,NG
IND=IST+2
WRITE(6,31) ERG(IG),ERG(IG+1),(D(JK),JK=IST,IND)
31 FORMAT(1X1P2E11.4,4X3E13.5)

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

30 IST=IND+1
GO TO 14
14 CONTINUE

RETURN
END

-----
SUBROUTINE MACRCX(IGC)
COMMON/PHI/PHI(17000)
DIMENSION A(27),SIGO(27),SIGS(27),SIGIS(27,2),SIGES(27,2),
1 GCOS(27,2),XSI(27,2),WEIGHT(50,27),ELETH(2)
2 DIMENSION SIGMAO(50),SIGTRA(50),SIGSSI(50),RHO(50)
3 EQUIVALENCE (PHI(14901),A(1)),(PHI(14928),SIGO(1)),(PHI(14955),
1 SIGS(1)),(PHI(14982),SIGIS(1,1)),(PHI(15036),SIGES(1,1)),
2 (PHI(15090),GCOS(1,1)),(PHI(15144),XSI(1,1)),(PHI(15198),
3 WEIGHT(1,1)),(PHI(16548),ELETH(1)),(PHI(16550),NMIX)
EQUIVALENCE (PHI(16551),SIGMAO(1)),(PHI(16601),SIGTRA(1)),
1 (PHI(16651),SIGSSI(1)),(PHI(16701),RHC(1))
EQUIVALENCE (WD,IWD),(PHI(1),D,AKAP2,ALPH2)

DO 1 I=1,NMIX
SIGTRA(I)=0.
SIGSSI(I)=0.
DO 1 J=1,27
IF(WEIGHT(I,J).EQ.0.) GO TO 1
SIGTRA(I)=SIGTRA(I)+.6023*RHO(I)*WEIGHT(I,J)*((1.-GCOS(J,IGO)/3.)*
1 SIGS(J,IGO)+SIGIS(J,IGC))/A(J)
2 SIGSSI(I)=SIGSSI(I)+.6023*RHO(I)*WEIGHT(I,J)*(SIGIS(J,IGO)+SIGES(J
1 ,IGO))*XSI(J,IGC)/A(J)
1 CONTINUE
RETURN
END

-----
FUNCTION STEP(I SKIP)
THIS IS A FRIG TO AVOID EVALUATION OF CONSTANT FUNCTIONS
X=1.
STEP=ALOG(4000.*X)
IF(I SKIP.LE.200) STEP=ALOG(100.*X)
IF(I SKIP.LE.150) STEP=ALOG(10.*X)
IF(I SKIP.LE.50) STEP=ALOG(14.*X)
STEP=STEP/50.
RETURN
END

-----
SUBROUTINE RHTAPE(IGC,I21)
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /XTAPE/ NATOL
COMMON/PHI/PHI(17000)
COMMON/LASTGP/LASGPS,NUMON2,MATER,ICUT,ITEMP,ITHERM,IRTHERM,I
DIMENSION A(27),SIGO(27),SIGS(27),SIGIS(27,2),SIGES(27,2),
1 GCOS(27,2),XSI(27,2),WEIGHT(50,27),ELETH(2)

```

1
2
3
4
5
6
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
42
43
44
45
46
47
48
49
50
51
57
58
59
60
61
62
63

```

DIMENSION SIGMAO(50),SIGTRA(50),SIGSSI(50),RHC(50)
EQUIVALENCE (PHI(14901),A(1)),(PHI(14928),SIGO(1)),(PHI(14955),
1 SIGS(1)),(PHI(14982),SIGIS(1,1)),(PHI(15036),SIGES(1,1)),
2 (PHI(15090),GCOS(1,1)),(PHI(15144),XSI(1,1)),(PHI(15198),
3 WEIGHT(1,1)),(PHI(16548),ELETH(1)),(PHI(16550),NMIX)
EQUIVALENCE (PHI(16551),SIGMAO(1)),(PHI(16601),SIGTRA(1)),
1 (PHI(16651),SIGSSI(1)),(PHI(16701),RHC(1))
EQUIVALENCE (PHI(16911),EXTRA)
LASGPS=LASGPS+1
IF(IGO.EQ.1.AND. I .EQ.1) GO TO 1
ITA=NT9
IF(LASGPS.GT.NUMON2) ITA=NATOL
READ(ITA)ELETH(I21),(SIGES(J,I21),J=1,27),(SIGIS(J,I21),J=1,27),
1 (GCOS(J,I21),J=1,27),(XSI(J,I21),J=1,27)
IF(ITA.EQ.NT9) GO TO 10
1 NUMON2=NUMON2+1
WRITE(NT9) ELETH(I21),(SIGES(J,I21),J=1,27),(SIGIS(J,I21),J=1,27),
1 (GCOS(J,I21),J=1,27),(XSI(J,I21),J=1,27)
10 RETURN
END

```

```

FUNCTION DLEN(SA,SS,AMU)
GLEAST=1./(SA+SS)
AINC=4.
6 GUESS1=(1.+AINC)*GLEAST
GUESS2=(1.+AINC/2.)*GLEAST
1 F1=FL(SA,SS,AMU,GUESS1)
F2=FL(SA,SS,AMU,GUESS2)
GUESS3=(F1*GUESS2-F2*GUESS1)/(F1-F2)
IF(ABS(F2).LT..0001 .AND. ABS(GUESS3-GUESS2).LT..0001 ) GO TO 5
2 IF(GUESS3.GT.50.*GLEAST+1.) GO TO 4
3 GUESS1=GUESS2
GUESS2=GUESS3
GO TO 1
3 GUESS3=(GUESS3+GUESS2)/2.
GO TO 2
4 AINC=AINC/2.
IF(AINC.LT..01) GO TO 7
GO TO 6
5 DLEN=GUESS3
RETURN
7 CALL MESSAGE(30HFAILED TO SOLVE FOR DIFF LGTH&)
WRITE(6,8) SA,SS,AMU
8 FORMAT(3E18.8)
STOP
END

```

```

FUNCTION FL(SA,SS,AMU,AL)
SP=SS+SA
P=3.*AL*AL*SA*AMU
FL=AL*SS
FL=AL*SS*ALOG((AL*SP+1.)/(AL*SP-1.))/2.-(1.+P*SS)/(1.+P*SP)
RETURN
END

```

```

SUBROUTINE TEST10
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON /NST/ NBNDST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,NAS
1T,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST
COMMON/GEOM/GEOM
COMMON/PHI/PHI(17000)
COMMON ACCUR(4),LHALF(2),MHALF(2),NHALF(2),WD
COMMON INIT,NIT,IDI,OMEGA,DIFF,PRODIF
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),AS2(1),AS(1)
DIMENSION SCATER(1)
INTEGER GEOM
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1))
EQUIVALENCE (WD,IWD)
C
IST=NRST+1
IND=NRST+NCOL
IF(GEOM.EQ.1) CALL MESSAGE(6HRMESH&)
IF(GEOM.EQ.2) CALL MESSAGE(6HXMESH&)
WRITE(6,9)(RME SH(I),I=IST,IND)
9 FORMAT(13F9.3)
IST=NZST+1
IND=NZST+NROW
WRITE(6,93)
93 FORMAT(////)
IF(GEOM.EQ.1) CALL MESSAGE(6HZMESH&)
IF(GEOM.EQ.2) CALL MESSAGE(6HYMESH&)
WRITE(6,8)(ZME SH(I),I=IST,IND)
8 FORMAT(13F9.3)
N1=NROW-1
WRITE(6,7)
7 FORMAT(//6H ZONES)
ID=0
IZ=65/(NROW+4)
IST=1
IND=MINO(60,NCOL-1)
11 NST=IST -NCOL+1
ID=ID+1
IF (ID.EQ.1) WRITE (6,92)
92 FORMAT(1H1)
IF (ID.GT.IZ) ID=0
WRITE (6,91)
91 FORMAT ('0'//)
DO 1 J=1,N1
NST=NST+NCOL-1
NND=NST-IST+IND
1 WRITE(6,6) (ITYPE(I),I=NST,NND)
6 FORMAT(1X60I2)
IF(IND.EQ.NCOL-1) GO TO 10
IST=IND+1
IND=MINO(IND+60,NCOL-1)
GO TO 11
10 WRITE (6,94)
94 FORMAT ('0'//)
RETURN
END

```

```

SUBROUTINE SAUCE
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
IST, NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST
COMMON/COEF/ SE,SW,NE,NW,RO,OE,CN,CW,CS ,J,I
COMMON/SOURCE/SOURCE(100),ISSRCE(60)
COMMON/PHI/PHI(17000)

COMMON ACCUR(4),WD,CMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIIST

DIMENSION BOUND(1)
DIMENSION I TYPE(1),RMESH(1),ZMESH(1),SCA(1),D(1),A(1)
INTEGER SE,SW ,G
EQUIVALENCE (WD,IWD)
EQUIVALENCE (PHI(1),I TYPE(1),RMESH(1),ZMESH(1),SCA(1),D(1),
1 A(1),BOUND(1))

WD HAS BEEN READ WHEN SAUCE IS ENTERED FROM SETUP

IS=0
JEST=NAS2ST
G=1
1 CALL INDEX(J2,WD,38HIN*AT*BY*ZONE*SOME*ALL*VALU*SOUR*GROU&)
4 GO TO ( 2 ,2 ,10, 30 , 20 ,40 , 17 , 2 ,50,120),J2
2 IWD=ITDEC(4)
GO TO 1
10 IWD=ITDEC(4)
CALL INDEX(J3,WD,8HBY*MATE&)
GO TO (957,3,1),J3
3 DO 11 I=1,NTYPE
11 I TYPE(I)=1001*I TYPE(I)
17 DO 16 I=1,100
16 SOURCE(I)=0.
12 IF(NEWDEC(0))101,14,15
15 IWD=INDEC(1)
SOURCE(IWD)=FLDEC(0)
IS=IWD
GO TO 12
14 CALL ERROR (57HSPECIAL CHARACTER READ WHILE READING SOURCES. (SAUC
*E,14)&)

=====
20 DO 23 I=1,NMAT
23 SOURCE(I)=0.
IWD=ITDEC(4)
=POINTS
22 IF(NEWDEC(0)) 101,25,21
25 IWD=ITDEC(1)
GO TO 22
21 IWD=INDEC(1)
JWD=INDEC(1)
IWD=ITDEC(1)
=)
ISUF=(NROW-JWD)*(NCOL+1)+IWD+1
IS=IS+1
I TYPE(I SUF)=I TYPE(I SUF)+1000*IS
SOURCE(I S)=FLDEC(3)
IF(IS.EQ.101) GO TO 998

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

GO TO 22
=====
30 DO 35 I=1,100
35 SOURCE(I)=0.
IS=IS+1
DO 33 I=1,NTYPE
33 I TYPE(I)=I TYPE(I)+1000
SOURCE(I)=FLDEC(1)
34 IF(NEWDEC(3))101,31,999
31 IWD=ITDEC(1)
IST=INDEC(0)+1
JND=NROW-INDEC(0) -2
IND=INDEC(0)
JST=NROW-INDEC(1)-1
IWD=ITDEC(1)
IS=IS+1
SOURCE(I S)=FLDEC(1)
IF(IS.EQ.101) GO TO 998
DO 32 J=JST,JND
IA=J*(NCOL-1)
DO 32 I=IST,IND
IBB=IA+1
32 I TYPE( IBB)=MOD(I TYPE( IBB),1000)+1000*IS
GO TO 34

=====
40 IWD=ITDEC(4)
=POINTS
DO 41 I=1,NTYPE
IA=NTYPE+1-I
41 I TYPE( IA)=I TYPE( IA)+1000*INDEC(0)
GO TO 17

=====
50 IG=INDEC(0)
IS=0
51 IF(G.EQ.1)GO TO 151
IWD=ITDEC(4)
CALL INDEX(J7,IWD,5HSAME&)
IF(J7.GT.1) GO TO 52
55 IS=IS+1
IF(NEWDEC(0))56,997,54
54 SOURCE(I S)=FLDEC(1)
GO TO 55
52 IF(ISSRCE(G-1).LT.0)GO TO 151
DO 152 I=1,NTYPE
152 I TYPE(I)=MOD(I TYPE(I),1000)
151 IF(IG.GT.G) GO TO 100
IF(G.EQ.1) GO TO 2
GO TO 1
56 IS=IS-1
101 IWD=ITDEC(4)
100 WRITE(NT9) I S,(SOURCE(I),I=1,IS)
IT=1
DO 104 IE=1,2
IND=IT+NHALF(IE)-1
WRITE(NT9) (I TYPE(I),I=IT,IND)
104 IT=NHALF(1)+1
IF(IG.GT.G) ISSRCE(G)=-1

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

IF(IG.EQ.6) ISSRCE(G)= 1
G=G+1
IF(G.GT.NG) GO TO 110
IF(IG.G) 1,1,100
120 IG=NG+1
IF (G.EQ.1) GO TO 10G
GO TO 52

110 REWIND NT9
REWIND NT11
REWIND NT8
DO 201 G=1,NG
CALL INDEX(J6,WD,5HBOUND)
IF(J6.NE.1) CALL ERROR (29HEXPECTED BOUNDS. (SAUCE,110)& )
CALL BOUNDS
IST=NRST+1
IND=NRST+NCOL
WRITE(NT11) (RMESH(I),I=IST,IND)
IST=NZST+1
IND=NZST+NROW
WRITE(NT11) (ZMESH(I),I=IST,IND)
IST=NBNDST+1
IND=NBNDST+4*(NROW+NCOL)
WRITE(NT11) (BOUND(I),I=IST,IND)
READ(NT9) IS , (SOURCE(I),I=1,IS)
WRITE(NT11) IS, (SOURCE(I),I=1,IS)

IT=1
DO 204 IE=1,2
IND=IT+NHALF(IE)-1
READ (NT9) (ITYPE(I),I=IT,IND)
WRITE(NT11) (ITYPE(I),I=IT,IND)
IST=JDST+1
IND=JDST+NMAT
IF(G.EQ.1) GO TO 102
MG=MAXO(1,G-NBACK)
MG1=G-1
DO 103 JG=MG,MG1
READ (NT8) (SCA(I),I=IST,IND)
103 WRITE(NT11) (SCA(I),I=IST,IND)
102 IND=IND+NMAT
READ ( NT8) (D(I),I=IST,IND)
WRITE(NT11) (D(I),I=IST,IND)

204 IT=NHALF(I)+1
201 CONTINUE
REWIND NT9
REWIND NT11
REWIND NT8
RETURN

997 CALL ERROR (25HINPUT ERROR. (SAUCE,997)& )
999 CALL ERROR (25HINPUT ERROR. (SAUCE,999)& )
998 CALL ERROR(75HLIMIT CN NUMBER OF DIFFERENT SOURCES HAS BEEN PASSED
1. MAX=100. (SAUCE,998)& )
RETURN
END

```

```

1 SUBROUTINE BOUNDS 1
2 2
3 COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT 3
4 COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA 4
5 IST, NSGST, NSCAST,NPHIST, 5
6 2 NTHST,NYST,NBBBST 6
7 COMMON/COEF/ SE,SW,NE,NW,RO,OE,CN,CW,CS ,J,I 7
8 COMMON/PHI/PHI(17000) 8
9 9
10 COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2), 10
11 1 G,NIT,IDIST 11
12 12
13 DIMENSION TMESH(1) 13
14 DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1), 14
15 1 AS2(1),AS(1),SCATER(1) 15
16 DIMENSION AE(1),AO(1),D(1),A(1),SG(1) 16
17 DIMENSION THETA(1),Y(1),BBB(1) 17
18 LOGICAL SUMDUN 18
19 EQUIVALENCE (PHI(1),TMESH(1)) 19
20 EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1), 20
21 1 AS2(1),AS(1),SCATER(1)) 21
22 EQUIVALENCE(SCATER(1),AE(1),AO(1),D(1),A(1),SG(1)) 22
23 EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1)) 23
24 EQUIVALENCE (AO(1),W),(AE(1),X) 24
25 EQUIVALENCE (WD,IWD) 25
26 1000 SUMDUN=.FALSE. 26
27 MBNDST=NBNDST 27
28 LENGTH=NZ 28
29 1004 IF(NEWDEC(0)) 1001,1001,1104 29
30 1104 IWD=ITDEC(0) 30
31 GO TO 1004 31
32 1001 IWD=ITDEC(4) 32
33 CALL INDEX(J,IWD,21HBR*BZ*HIGH*LOW*BX*BY&) 33
34 GO TO (1004,1002,1003,1004,1004,1002,1040),J 34
35 35
36 1002 MBNDST=MBNDST+4*NZ 36
37 LENGTH=NR 37
38 GO TO 1004 38
39 1003 MBNDST=MBNDST+2*LENGTH 39
40 GO TO 1004 40
41 1040 CALL INDEX(J,IWD,19HHIG*NONZ*ONEZ*TWZ&) 41
42 IF(SUMDUN) GO TO 1041 42
43 IF(J.EQ.5) GO TO 1060 43
44 44
45 I=0 45
46 IA=MBNDST 46
47 IB=IA+LENGTH 47
48 1041 GO TO ( 9999, 1010, 1020, 1030 ,9999),J 48
49 49
50 ===== 50
51 NONZERO 51
52 1010 CONTINUE 52
53 IBB=1 53
54 IF(NEWDEC(0)) 1012,1015,1013 54
55 1015 ITCH=ITDEC(1) 55
56 IBB=INDEC(0) 56
57 1013 BB=FLDEC(0) 57
58 CC=FLDEC(0) 58
59 59
60 DO 1014 IC=1,IBB 60
61 I=I+1 61
62 IA=IA+1 62
63 IB=IB+1 63

```



```

BOUND(IB)=CC
1014 BOUND(IA)=BB
IF(I.LT.LENGTH) GO TO 1010
GO TO 1000

=====
                                ONEZERO
.020  IWD=INDEC(O)
READS 1,2 OR3 AS A,B OR C IS THE ZERC COEFFICIENT =====
GO TO (1021,1024,1027),IWD

1021 IBB=1
IF(NEWDEC(O)) 2012,1023,1051
1023 ITCH=ITDEC(1)
IBB=INDEC(O)
1051 CC=FLDEC(O)
DO 1052 IC=1,IBB
I=I+1
IA=IA+1
IB=IB+1
BOUND(IA)=1.E36
1052 BOUND(IB)=CC

IF(I.LT.LENGTH) GO TO 1021
GO TO 1000

=====
                                B=0.
1024 IBB=1
IF(NEWDEC(O)) 2012,1026,1054

1026 ITCH=ITDEC(1)
IBB=INDEC(O)
1054 CC=FLDEC(O)
DO 1055 IC=1,IBB
I=I+1
IA=IA+1
IB=IB+1
BOUND(IA)=0.
1055 BOUND(IB)=CC
IF(I.LT.LENGTH) GO TO 1024
GO TO 1000

=====
                                C=0.
1027 IBB=1
IF(NEWDEC(O)) 1012,1029,1057
1029 ITCH=ITDEC(1)
IBB=INDEC(O)
1057 BB=FLDEC(O)
DO 1058 IC=1,IBB
I=I+1
IA=IA+1
IB=IB+1
BOUND(IA)=BB
1058 BOUND(IB)=0.
IF(I.LT.LENGTH) GO TO 1027
GO TO 1000

=====
                                TWOZERO

```

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
1030 IWD=INDEC(O)
JWD=INDEC(O)
IWD=IWD+JWD-2
GO TO (9997,1031,1038),IWD

1031 IF(NEWDEC(O))1032,1032,1033
1032 IST=I+1
IWD=ITDEC(4)
IND=LENGTH
GOTO 1034
1033 IST=I+1
IWD=INDEC(O)
IND=IWD+I
1034 DO 1035 IC=IST,IND
IA=IA+1
IB=IB+1
I=I+1
BOUND(IA)=1.E36
1035 BOUND(IB)=0.
GO TO (1000,1036),J
1036 IF(I.EQ.LENGTH) GO TO 1000
SUMDUN=.TRUE.
GO TO 1004

1038 IF(NEWDEC(O))1039,1039,1071
1039 IST=I+1
IWD=ITDEC(4)
IND=LENGTH
GO TO 1072
1071 IST=I+1
IWD=INDEC(O)
IND=I+IWD
1072 DO 1073 IC=IST,IND
IA=IA+1
IB=IB+1
I=I+1
BOUND(IA)=0.
1073 BOUND(IB)=0.
C
IF(I.EQ.LENGTH) GO TO 1000
SUMDUN=.TRUE.
GO TO 1004
C
=====
C
1012 SUMDUN=.TRUE.
GO TO 1001
C
2012 SUMDUN=.TRUE.
GO TO 1001
C
9999 CALL ERROR (58HPROGRAM EXPECTS NCNZERC,ONEZERO OR TWOZERO. (BOUNDS
*,9999) & )
9997 CALL ERROR(74HDATA ERRCR. CANNOT HAVE A AND B ZERO IN BOUNDARY CON
IDITION. (BOUNDS,9997) & )
1060 RETURN
END

```

```

SUBROUTINE COEFS
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /COUNT/COUNT(60)
INTEGER COUNT
COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1ST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
COMMON /COEF/ RO,OE,ON,OW,OS,AW,SE,SW,NE,NW,J,I
COMMON /SOURCE/ SOURCE(100),ISSRCE(60)
COMMON /REST/ IREST
COMMON /PHI/PHI(17000)
COMMON /IREMO/IREMO
COMMON /LEFT/ NLEFT, NLEFT, NLINE, NFIRST
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIIST

DIMENSION TMESH(1)
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),AS2(1),AS(1),
1 SCATER(1),PHIT(1),T(1)
DIMENSION AE(1),AO(1),D(1),A(1),SG(1)
DIMENSION THETA(1),Y(1),BBB(1)
DIMENSION W(1),X(1)
INTEGER G
INTEGER SE, SW
LOGICAL SKIPS

EQUIVALENCE (PHI(1),TMESH(1))
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1),PHIT(1),T(1))
EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1))
EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1))
EQUIVALENCE (AO(1),W(1)),(AE(1),X(1))

NCOST=NTST+NCOL
NLEFT=17000-NCOST
NLINE=MINO(NROW,NLEFT/(5*NCOL))
NFIRST=NROW-NLINE
MG=MINO(NBACK,G-1)
SKIPS=COUNT(G).EQ.0.OR.COUNT(G).EQ.-1

IST=NBNDST+1
IND=NBNDST+4*NCOL+4*NROW
READ(NT11) (BOUND(IRW),IRW=IST,IND)

READ(NT11) IS,(SOURCE(IRW),IRW=1,IS)
ITAPE=NT8
NTAPE=NT1
IT=1
IRE=0
IF(G-2)8,1,2
1 REWIND ITAPE
GO TO 8
2 CALL SHUNT(NT8,IRE,1,G)
8 CONTINUE
DO 1003 IE=1,2
IST=IT
IND=IT+NHALF(IE)-1
READ(NT11) (ITYPE(IRW),IRW=IST,IND)
IST=NBST+1
IND=NBST+MHALF(IE)
DO 3001 I=IST,IND
3001 B(I)=C.

```

```

IF(G.EQ.1) GO TO 3002
IF(IRE.EQ.1) CALL SHUNT(NT8,IRE,1,G)
DO 13 IG=1,MG
IST=NSCAST+1
IND=NSCAST+NMAT
READ(NT11) (SCATER(IRW),IRW=IST,IND)
IST=NPHIST+1
IND=NPHIST+MHALF(IE)
JTAPE=ITAPE
IF(IE.EQ.2.AND.IG.EQ.MG) JTAPE=NT8
READ(JTAPE) (PHIT(I),I=IST,IND)
IF(IE.EQ.2.AND.IG.EQ.MG) BACKSPACE NT8
IF(IE.EQ.2.AND.IG.EQ.MG) WRITE(NT9) (PHIT(I),I=IST,IND)
14
7 IB=NPHIST
IF(SKIPS) GO TO 13
IA=NBST
IF(IE.EQ.2) GO TO 3004
IST=1
IND=LHALF(1)
GO TO 3005
3004 IST=LHALF(1)+1
IND=NROW
3005 CONTINUE
DO 1001 J=IST,IND
DO 1001 I=1,NCOL
CALL NSEW(NSCAST,IE)
IB=IB+1
IA=IA+1
BADD=0.
IF(I.NE.1.AND.J.NE.1)BADD=BADD+SCATER(NW)*OW*ON*(RO-OW/4.)
IF(I.NE.1.AND.J.NE.NROW)BADD=BADD+SCATER(SW)*OS*OW*(RO-OW/4.)
IF(I.NE.NCOL.AND.J.NE.1)BADD=BADD+SCATER(NE)*OE*ON*(RO+OE/4.)
IF(I.NE.NCOL.AND.J.NE.NROW)BADD=BADD+SCATER(SE)*OS*OE*(RO+OE/4.)
B(IA)=B(IA)+BADD*PHIT(IB)/4.
36
1001 CONTINUE
13 CONTINUE
38
3002 CONTINUE
B=B+TRUE SOURCE =====
IF(SKIPS) GO TO 2006
IF(ISSRCE(G).LT.0)GO TO 2006
IA=NBST
IF(IE.EQ.2) GO TO 3006
IST=1
IND=LHALF(1)
GO TO 3007
3006 IST=LHALF(1)+1
IND=NROW
3007 DO 3003 J=IST,IND
DO 3003 I=1,NCOL
CALL NSEW(-1,IE)
IA=IA+1
BADD =0.
IF(I.NE.1.AND.J.NE.1)BADD=BADD+SOURCE(NW)*CN*OW*(RO-OW/4.)
IF(I.NE.1.AND.J.NE.NROW)BADD=BADD+SOURCE(SW)*OS*OW*(RO-OW/4.)
IF(I.NE.NCOL.AND.J.NE.1)BADD=BADD+SOURCE(NE)*OE*ON*(RO+OE/4.)
IF(I.NE.NCOL.AND.J.NE.NROW)BADD=BADD+SOURCE(SE)*OS*OE*(RO+OE/4.)
B(IA)=B(IA)+BADD/4.
3003 CONTINUE
=====
2006 CONTINUE
IF(SKIPS) GO TO 42
63

```

-35-

```

IF(IREMO.NE.1) GO TO 42
CALL RRSCE(IE)
42 CONTINUE

=====

IST=NDST+1
JST=NA ST+1
IND=NDST+NMAT
JND=NA ST+NMAT

READ(NT11) (D(M),M=IST,IND),(A(M),M=JST,JND)
IF(SKIPS) GO TO 12

2005 IND=MHALF(IE)
REMOVAL SOURCES HAVE YET TO BE ADDED IN

IF(IE.EQ.1) IND=LHALF(1)
IF(IE.EQ.2) IND=NROW
  IF(IE.EQ.1) IST=1
  IF(IE.EQ.2) IST=LHALF(1) +1
  DO 1005 J=IST,IND
    CALL ACDEF5(IE)

    IF(J.NE.1) GO TO 1006
    DO 1007 I=1,NCOL
      IA=I+NBST
      IB=NBNDST+4*NROW+3*NCCL+I
      IC=NBNDST+4*NROW+2*NCCL+I
      IF(BOUND(IC).GE.1.E35) GO TC 3
      NCO=NRST+I
      RO=RMESH(NCO)
      OE=0.
      IF(I.NE.NCOL) OE=RMESH(NCO+1)-RO
      OW=0.
      IF(I.NE.1) OW=RO-RMESH(NCO)
      B(IA)=B(IA)-((RO+OE/4.)*OE+(RO-OW/4.)*OW)*BOUND(IB)/2.
      GO TO 1007
3 B(IA)=BOUND(IB)
1007 CONTINUE

1006 IF(J.NE.NROW) GO TO 1008
      DO 1009 I=1,NCOL
        IA=NBST+(LHALF(2)-1)*NCOL+I
        IB=NBNDST+4*NROW+NCOL+I
        IC=NBNDST+4*NROW+I
        IF(BOUND(IC).GE.1.E35) GO TC 4
        NCO=NRST+I
        RO=RMESH(NCO)
        OE=0.
        IF(I.NE.NCOL) OE=RMESH(NCO+1)-RO
        OW=0.
        IF(I.NE.1) OW=RO-RMESH(NCO)
        B(IA)=B(IA)-((RO+OE/4.)*OE+(RO-OW/4.)*OW)*BOUND(IB)/2.
        GO TO 1009
4 B(IA)=BOUND(IB)
1009 CONTINUE

1008 IF(IE.EQ.1) JA=J
      IF(IE.EQ.2) JA=J-LHALF(1)
      IA=(JA-1)*NCOL+1+NBST
      IB=NBNDST+2*NROW+1-J
      IC=NBNDST+NROW+1-J

```

```

1 IF(BOUND(IC).GE.1.E35) GO TC 5
2 IN=NBNDST+NROW
3 IF(J.EQ.1.AND.BOUND(IN).GE.1.E35)GO TO 2008
4 IF(J.EQ.NROW.AND.BOUND(NBNDST+1).GE.1.E35)GO TO 2008
5 NJO=NZST+J
6 ON=0.
7 IF(J.NE.1)ON=ZMESH(NJO-1)-ZMESH(NJO)
8 OS=0.
9 IF(J.NE.NROW)OS=ZMESH(NJO)-ZMESH(NJC+1)
10 B(IA)=B(IA)-RMESH(NRST+1)*(ON+OS)*BOUND(IB)/2.
11 GO TO 2008
12 5 B(IA)=BOUND(IB)
13 2008 CONTINUE
14 IA=NBST+JA*NCOL
15 IB=NBNDST+4*NROW+1-J
16 IC=NBNDST+3*NROW+1-J
17 IF(BOUND(IC).GE.1.E35) GO TO 6
18 IN=NBNDST+3*NROW
19 IF(J.EQ.1.AND.BOUND(IN).GE.1.E35) GO TO 2009
20 IN=NBNDST+2*NROW+1
21 IF(J.EQ.NROW.AND.BOUND(IN).GE.1.E35) GO TO 2009
22 NCO=NCOL+NRST
23 NJO=NZST+J
24 ON=0.
25 IF(J.NE.1)ON=ZMESH(NJO-1)-ZMESH(NJO)
26 OS=0.
27 IF(J.NE.NROW)OS=ZMESH(NJO)-ZMESH(NJC+1)
28 B(IA)=B(IA)-RMESH(NCO)*(ON+OS)*BOUND(IB)/2.
29 GO TO 2009
30 6 B(IA)=BOUND(IB)
31 2009 CONTINUE
32 IT=NTST+1
33 IW=NWST+1
34 IX=NXST+1
35
36 SPARE1=A0(IW)
37 SPARE2=AE(IX)
38 T(IT)=AE(IX)
39 W(IW)=0.
40 X(IX)=SPARE1
41 ISUF=NBNDST+NROW+1-J
42 JSUF=NBNDST+4*NROW+2*NCOL+1
43 KSUF=NBNDST+4*NROW+1
44 LSUF=ISUF+2*NROW
45 IF(BOUND(ISUF).GE.1.E35.OR.(J.EQ.1.AND.BOUND(JSUF).GE.1.E35).OR.
46 1 (J.EQ.NROW.AND.BOUND(KSUF).GE.1.E35)) GO TO 1012
47 GO TO 1013
48 1012 X(IX)=1.
49 T(IT)=0.
50
51 1013 DO 1010 I=2,NCOL
52 IW=IW+1
53 IX=IX+1
54
55 IT=IT+1
56 SPARE1=A0(IW)
57 SPARE3=SPARE2
58 SPARE2=AE(IX)
59 W(IW)=SPARE3/X(IX-1)
60 ICO=NBNDST+4*NROW+I
61 NCO=2*NCOL+ICO
62 IF(J.EQ.1.AND.BOUND(NCO).GE.1.E35)W(IW)=0.
63 IF(J.EQ.NROW.AND.BOUND(ICO).GE.1.E35)W(IW)=0.
  X(IX)=SPARE1-W(IW)*T(IT-1)

```

```

IF(J.EQ.1.AND.BOUND(NCO).GE.1.E35)X(IX)=1.
IF(J.EQ.NROW.AND.BOUND(ICD).GE.1.E35)X(IX)=1.
T(IT)=SPARE2
IF(I.EQ.NCOL.AND.BOUND(LSUF).LT.1.E35)T(IT)=1.
IF((J.EQ.1.AND.BOUND(NCO).GE.1.E35).OR.(J.EQ.NROW.AND.BOUND(ICD).GE.1.E35))T(IT)=0.
1010 CONTINUE

IF(BOUND(LSUF).GE.1.E35)W(IW)=0.
IF(BOUND(LSUF).GE.1.E35)X(IX)=1.

      JST=NBST+(J-1)*NCOL+1
      IF(IE.EQ.2)JST=JST-MHALF(1)
LIWST=NWST+1
LIWND=NWST+NCOL
LIXST=NXST+1
LIXND=NXST+NCOL
LIAST=NASST+1
LIAND=NASST+NCOL
LIBST=JST
LIBND=JST+NCOL-1
LITST=NTST+1
LITND=NTST+NCOL
WRITE(NTAP)(W(IRW),IRW=LIWST,LIWND),(X(IRW),IRW=LIXST,LIXND),
1 (AS(IRW),IRW=LIAST,LIAND),(B(IRW),IRW=LIBST,LIBND),
2 (T(IRW),IRW=LITST,LITND)
1005 CONTINUE
12 CONTINUE
      IF(IE.EQ.2)GO TO 1003
IT=NCOL
IA=(NCOL-1)*(LHALF(1)-1)
NCO=NCOL-1
DO 1011 I=1,NCO
IA=IA+1
1011 ITYPE(I)=ITYPE(IA)
ITAPE=NT9
1003 CONTINUE
IRE=0
IF(G-1)14,15,16
15 REWIND NT9
GO TO 14
16 CALL SHUNT(NT9,IRE,-1,G+1)
14 IF(IRE.EQ.1)CALL SHUNT(NT9,IRE,1,G)
IF(SKIPS)GO TO 110
IF(NLINE-NROW/2)102,101,101
101 REWIND NT1
GO TO 104
102 DO 103 IX=1,NLINE
103 BACKSPACE NT1
GO TO 105
104 IF(NFIRST)105,105,106
106 DO 107 IX=1,NFIRST
107 READ(NT1)
105 IST=NCOST+1
108 DO 109 IX=1,NLINE
IND=IST+5*NCOL-1
READ(NT1)(PHI(IRW),IRW=IST,IND)
109 IST=IST+5*NCOL
110 REWIND NT1
RETURN
END

```

```

SUBROUTINE SHUNT(KTAPE,IRE,J,G)
INTEGER G
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NUM/ NBACK
IF(IRE)4,4,5
4 ITAPE=KTAPE
IF(NBACK.GE.G-1)GO TO 7
IQ=(G-1)/2
IF(NBACK-IQ)1,1,3
1 NB=NBACK+J
DO 2 I=1,NB
2 BACKSPACE ITAPE
RETURN
3 IRE=1
7 REWIND ITAPE
RETURN
5 NSKIP=G-1-NBACK
DO 6 I=1,NSKIP
6 READ(ITAPE)
IRE=0
RETURN
END
-----
SUBROUTINE NNEW(NNUMST,IE)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NST/ NBNDST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1ST,NSGST,NSCAST,NPHIST,
2 NTHEST,NYST,NBBST
COMMON/PHI/PHI(17000)
COMMON /COEF/ RO,OE,ON,OW,OS,AW,SE,SW,NE,NW,J,I
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDI ST
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1)
INTEGER SE,SW
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1))
IF(NNUMST.LT.0)GO TO 2003
MNW=(J-2)*(NCOL-1)+I-1
IF(IE.EQ.2)MNW=MNW-(LHALF(1)-1)*(NCOL-1)
IF(J.NE.1.AND.I.NE.1)NW=NNUMST+MOD(ITYPE(MNW),1000)
IF(J.NE.1.AND.I.NE.NCOL)NE=NNUMST+MOD(ITYPE(MNW+1),1000)
MSW=MNW*NCOL-1
IF(J.NE.NROW.AND.I.NE.1)SW=NNUMST+MOD(ITYPE(MSW),1000)
IF(J.NE.NROW.AND.I.NE.NCOL)SE=NNUMST+MOD(ITYPE(MSW+1),1000)
GO TO 2004
2003 MNW=(J-2)*(NCOL-1)+I-1
IF(IE.EQ.2)MNW=MNW-(LHALF(1)-1)*(NCOL-1)
IF(J.NE.1.AND.I.NE.1)NW=ITYPE(MNW)/1000
IF(J.NE.1.AND.I.NE.NCOL)NE=ITYPE(MNW+1)/1000
MNW=MNW*NCOL-1
IF(J.NE.NROW.AND.I.NE.1)SW=ITYPE(MNW)/1000
IF(J.NE.NROW.AND.I.NE.NCOL)SE=ITYPE(MNW+1)/1000

```

```

2004 IA=NRST+I
JA=NZST+J
RC=RSMESH(IA)
IF(J.NE.1) ON=ZMESH(JA-1)-ZMESH(JA)
IF(J.NE.NROW) OS=ZMESH(JA)-ZMESH(JA+1)
IF(J.EQ.1) ON=OS
IF(J.EQ.NROW) OS=ON
IF(I.NE.1) OW=RSMESH(IA)-RSMESH(IA-1)
IF(I.NE.NCOL) OE=RSMESH(IA+1)-RSMESH(IA)
IF(I.EQ.1) OW=OE
IF(I.EQ.NCOL) OE=OW
RETURN
END

```

```

SUBROUTINE RRSCE(IE)
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/PHI/PHI(17000)
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
IST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBST,NTST
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,ATYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NUM/ NBACK
COMMON ACCUR(4),WD,OMEGA,DIFF,PRCDIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIIST
20 DIMENSION RMESH(1),ZMESH(1),B(1)
21 DIMENSION REMO(10,10)
22 INTEGER G
23 EQUIVALENCE (PHI(1),RMESH(1),ZMESH(1),B(1))
24
25 ITS=NT2
26 IF(IE-1)17,18,16
16 JST=LHALF(1)+1
JND=NROW
GO TO 19
17 CALL ERROR (18HERROR. (RRSCE,17)E )
18 JST=1
JND=LHALF(1)
19 IA=NBST
IBA=0
NOBACK=0
20 JJA=JST
IST=1
21 READ(ITS) MR,MZ,IA,JA,R1,R2,Z1,Z2,K,L,((REMO(I,J),I=1,K),J=1,L),IG
RO=(R1+R2)*.5
ZC=(Z1+Z2)*.5
IF(IG-G)21,23,22
22 CALL ERROR (18HERROR. (RRSCE,22)E )
23 IZ=NZST+JJA
JSTOP = 0
IIA=IST
24 IR=NRST+IIA
IF(RMESH(IR).GT.R2) GO TO 28
IF(ZMESH(IZ).LT.Z1) GO TO 31
IF(K.EQ.0) GO TO 248
IA=(JJA-1)*NCOL+IIA+NBST-(IE-1)*LHALF(1)*NCOL
ON=ZMESH(IZ-1)-ZMESH(IZ)
OS=ZMESH(IZ)-ZMESH(IZ+1)
OE=RSMESH(IR+1)-RSMESH(IR)

```

```

1 OW=RSMESH(IR)-RSMESH(IR-1)
2 BADD=0.
3 IF(IIA.NE.1.AND.JJA.NE.1)BADD=BADD+(RMESH(IR)-OW/4.)*ON*OW
4 IF(IIA.NE.1.AND.JJA.NE.NROW)BADD=BADD+(RMESH(IR)-OW/4.)*OS*OW
5 IF(IIA.NE.NCOL.AND.JJA.NE.1)BADD=BADD+(RMESH(IR)+OE/4.)*ON*OE
6 IF(IIA.NE.NCOL.AND.JJA.NE.NROW)BADD=BADD+(RMESH(IR)+OE/4.)*OS*OE
7 B(IA)=B(IA)+BADD*POLY2(REMO,RMESH(IR)-RO,K,ZMESH(IZ)-ZO,L)/4.
8
9 24E IF(IIA-NCOL)25,26,26
10 25 IIA=IIA+1
11 GO TO 24
12 24E ISTOP = NCOL+1
13 IF(JJA-JND)27,35,35
14 27 JJA=JJA+1
15 GO TO 23
16 28 ISTOP=IIA
17 IF(JJA-JND)27,29,29
18 29 IBA=IBA+1
19 NZUF=NZST+JND+1
20 IF(ZMESH(NZUF).LT.Z1) NOBACK=1
21 30 JJA=JST
22 ISTOP=ISTOP
23 GO TO 21
24 31 IF(ISTOP-NCOL)33,33,32
25 32 IF (JSTOP.EQ.0) JSTOP = JJA
26 JST=JSTOP
27 GO TO 20
28 33 JSTOP=JJA
29 GO TO 30
30 35 IF(IE.EQ.2) GO TO 37
31 IF(NOBACK.EQ.1) GO TO 37
32 IBA=IBA+1
33 DO 36 IAB=1,IBA
34 36 BACKSPACE ITS
35 37 RETURN
36 END

```

```

FUNCTION POLY2 (C,R,K,Z,L)
42 DIMENSION C(10,10)
43 I=K
44 SUMR=C.
45
46 1 J=L
47 SUMZ=C(I,L)
48 IF(J-1) 3,3,2
49 2 J=J-1
50 SUMZ = SUMZ*Z + C(I,J)
51 IF(J-1) 3,3,2
52 3 SUMR=SUMR*R + SUMZ
53 I=I-1
54 IF(I) 4,4,1
55 4 POLY2=SUMR
56 RETURN
57 END

```

```

SUBROUTINE ACOEFS(IE)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NST/ NBNDST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,
1 NAST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST
COMMON /COEF/ RO,OE,ON,OW,OS,AW,SE,SW,NE,NW,J,I
COMMON/PHI/PHI(17000)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRCDIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIIST

DIMENSION TMESH(1)
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1)
DIMENSION AE(1),AO(1),D(1),A(1),SG(1)
DIMENSION THETA(1),Y(1),BBB(1)
DIMENSION W(1),X(1)
INTEGER SE,SW
EQUIVALENCE (PHI(1),TMESH(1))
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1))
EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1))
EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1))
EQUIVALENCE (AO(1),W(1)),(AE(1),X(1))

DO 1006 I=1,NCOL
CALL NSEW(NDST,IE)
IC=NAS2ST+I
IB=NASST+I
IW=NWST+I
IX=NXST+I
IF(J.NE.1)AS2(IC)=AS(IB)
AS(IB)=0.
AE(IX)=0.
AO(IW)=0.
RE4=R O+OE/4.
RW4=R O-OW/4.
IF(I.NE.NCOL.AND.J.NE.NROW) AS(IB)=AS(IB)+OE*RE4*(SE)/(2.*OS)
IF(I.NE.1.AND.J.NE.NROW) AS(IB)=AS(IB)+OW*RW4*(SW)/(2.*OS)

IF(I.NE.NCOL.AND.J.NE.1)AE(IX)=AE(IX)+(RO+OE/2.)*ON*O(NE)/(2.*OE)
IF(I.NE.NCOL.AND.J.NE.NROW) AE(IX)=AE(IX)+(RO+OE/2.)*OS*(SE)/(2.*O)
1 E)

NW=NW-NDST+NAST
NE=NE-NDST+NAST
SW=SW-NDST+NAST
SE=SE-NDST+NAST

IF(J.NE.NROW)AO(IW)=AO(IW)-AS(IB)
IF(J.NE.1) AO(IW)=AO(IW)-AS2(IC)
IF(I.NE.NCOL)AO(IW)=AO(IW)-AE(IX)
IF(I.NE.1) AO(IW)=AO(IW)-AE(IX-1)

IF(I.NE.1.AND.J.NE.1)AO(IW)=AO(IW)-OW*CN*RW4*A(NW)/4.
IF(I.NE.NCOL.AND.J.NE.1)AO(IW)=AO(IW)-OE*CN*RE4*A(NE)/4.
IF(I.NE.NCOL.AND.J.NE.NROW)AO(IW)=AO(IW)-OE*OS*RE4*A(SE)/4.
IF(I.NE.1.AND.J.NE.NROW)AO(IW)=AO(IW)-OW*CS*RW4*A(SW)/4.
1006 CONTINUE

ISUF=NBNDST+NROW+1-J
OX=0.
IF(J.NE.1)OX=OX+ON
IF(J.NE.NROW)OX=OX+OS

```

```

1 IF(BOUND(ISUF).GE.1.E35)GO TO 1
2 AC(NWST+1)=A(NWST+1)-BOUND(ISUF)*RMESH(NRST+1)*OX/2.
3 1 NCO=NRST+NCOL
4 ISUF=NBNDST+3*NROW+1-J
5 IF(BOUND(ISUF).GE.1.E35) GO TO 2
6 IW=NWST+NCOL
7 AO(IW)=AO(IW)-BOUND(ISUF)*OX*RMESH(NCO)/2.
8
9 2 IF(J.NE.1.AND.J.NE.NROW) GO TO 5
10 IF(J.EQ.1) NN=NBNDST+4*NZ+2*NR
11 IF(J.EQ.NROW) NN=NBNDST+4*NROW
12 DO 2003 I=1,NCOL
13 NN=NN+1
14 IW=I+NWST
15 NCO=NRST+I
16 IF(BOUND(NN).GE.1.E35) GOTC 2003
17 OE=0.
18 IF(I.NE.NCOL)OE=RMESH(NCO+1)-RMESH(NCO)
19 OW=0.
20 IF(I.NE.1)OW=RMESH(NCO)-RMESH(NCO-1)
21 AO(IW)=AO(IW)-((RMESH(NCO)+OE/4.)*OE+(RMESH(NCO)-OW/4.)*OW)*BOUND(
22 1 NN)/2.
23 2003 CONTINUE
24 5 RETURN
25 END

```

```

-----
SUBROUTINE XCOEFS
31 COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
32 COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
33 COMMON /NUM/ NBAC
34 COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
35 1ST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
36 COMMON /COEF/ RO,OE,ON,OW,OS,AW,SE,SW,NE,NW,J,I
37 COMMON/SOURCE/SOURCE(100),ISSRCE(60)
38 COMMON /REST/ IREST
39 COMMON/PHI/PHI(17000)
40 COMMON /COUNT/ COUNT(60)
41 COMMON /LEFT/ NCOL,NLEFT,NLINE,NFIRST
42
43 COMMON ACCUR(4),WD,OMEGA,DIFF,PRCDIF ,LHALF(2),MHALF(2),NHALF(2),
44 1 G,NIT,IDIIST
45
46 DIMENSION TMESH(1)
47 DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),AS2(1),AS(1),
48 1 SCATER(1),PHIT(1),T(1)
49 DIMENSION AE(1),AO(1),D(1),A(1),SG(1)
50 DIMENSION THETA(1),Y(1),BBB(1)
51 DIMENSION W(1),X(1)
52 INTEGER COUNT
53 INTEGER G
54 INTEGER SE,SW
55 LOGICAL SKIPS
56
57 EQUIVALENCE (PHI(1),TMESH(1))
58 EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
59 1 AS2(1),AS(1),SCATER(1),PHIT(1),T(1))
60 EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1))
61 EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1))
62 EQUIVALENCE (AO(1),W(1)),(AE(1),X(1))
63

```

```

NCO ST=NT ST+NCOL
NLEFT=17000-NCO ST
NLINE=MINO(NROW,NLEFT/(5*NCCL))
NFIRST=NROW-NLINE
MG=MINO(NBACK,G-1)

SKIPS=COUNT(G).EQ.0.CR.COUNT(G).EQ.-1

IST=NBND ST+1
IND=NBND ST+4*NCOL+4*NROW
READ(NT11) (BOUND(IRW),IRW=IST,IND)

READ(NT11) IS,(SOURCE(IRW),IRW=1,IS)
ITAPE=NT8
NTAP=NT1
IT=1
IRE=0
IF(G-2)8,1,2
1 REWIND ITAPE
GO TO 8
2 CALL SHUNT(NT8,IRE,1,G)
8 CONTINUE
DO 1003 IE=1,2
IST=IT
IND=IT+MHALF(IE)-1
READ(NT11) (ITYPE(IRW),IRW=IST,IND)
IST=NBST+1
IND=NBST+MHALF(IE)
DO 3001 I=IST,IND
3001 B(I)=0.
IF(G.EQ.1) GO TO 3002
IF(IRE.EQ.1) CALL SHUNT(NT8,IRE,1,G)
DO 13 IG=1,MG
IST=NCAST+1
IND=NCAST+NMAT
READ(NT11) (SCATER(IRW),IRW=IST,IND)
IST=NPHIST+1
IND=NPHIST+MHALF(IE)
JTAPE=ITAPE
IF(IE.EQ.2.AND.IG.EQ.MG) JTAPE=NT8
READ(JTAPE) (PHIT(I),I=IST,IND)
IF(IE.EQ.2.AND.IG.EQ.MG) BACKSPACE NT8
IF(IE.EQ.2.AND.IG.EQ.MG) WRITE(NT9) (PHIT(I),I=IST,IND)

7 IB=NPHIST
IF(SKIPS) GO TO 13
IA=NBST
IF(IE.EQ.2) GO TO 3004
IST=1
IND=LHALF(1)
GO TO 3005
3004 IST=LHALF(1)+1
IND=NROW
3005 CONTINUE
DO 1001 J=IST,IND
DO 1001 I=1,NCOL
CALL NSEW(NSCAT,IE)
IB=IB+1
IA=IA+1
BADD=0.
IF(I.NE.1.AND.J.NE.1) BADD=BADD+SCATER(NW)*OW*CN
IF(I.NE.1.AND.J.NE.NROW) BADD=BADD+SCATER(SW)*OW*OS

```

```

1 IF(I.NE.NCOL.AND.J.NE.1) BADD=BADD+SCATER(NE)*OE*ON 1
2 IF(I.NE.NCOL.AND.J.NE.NROW) BADD=BADD+SCATER(SE)*OE*OS 2
3 B(IA)=B(IA)+BADD*PHIT(IB)/4. 3
4 4
5 1001 CONTINUE 5
6 13 CONTINUE 6
7 3002 CONTINUE 7
8 IF(SKIPS) GO TO 2006 8
9 B=B+TRUE SOURCE ===== 9
10 IF(ISSRCE(G).LT.0) GO TO 2006 10
11 IA=NBST 11
12 IF(IE.EQ.2) GO TO 3006 12
13 IST=1 13
14 IND=LHALF(1) 14
15 GO TO 3007 15
16 3006 IST=LHALF(1)+1 16
17 IND=NROW 17
18 3007 DO 3003 J=IST,IND 18
19 DO 3003 I=1,NCOL 19
20 CALL NSEW(-1,IE) 20
21 IA=IA+1 21
22 BADD=0. 22
23 IF(I.NE.1.AND.J.NE.1) BADD=BADD+SOURCE(NW)*ON*OW 23
24 IF(I.NE.1.AND.J.NE.NROW) BADD=BADD+SOURCE(SW)*OS*OW 24
25 IF(I.NE.NCOL.AND.J.NE.1) BADD=BADD+SOURCE(NE)*ON*OE 25
26 IF(I.NE.NCOL.AND.J.NE.NROW) BADD=BADD+SOURCE(SE)*OS*OE 26
27 B(IA)=B(IA)+BADD/4. 27
28 3003 CONTINUE 28
29 ===== 29
30 2006 IST=NDST+1 30
31 JST=NAST+1 31
32 IND=NDST+NMAT 32
33 JND=NAST+NMAT 33
34 34
35 READ (NT11) (D(M),M=IST,IND),(A(M),M=JST,JND) 35
36 IF(SKIPS) GO TO 12 36
37 37
38 2005 IND=MHALF(IE) 38
39 REMOVAL SOURCES HAVE YET TO BE ADDED IN 39
40 40
41 IF(IE.EQ.1) IND=LHALF(1) 41
42 IF(IE.EQ.2) IND=NROW 42
43 IF(IE.EQ.1) IST=1 43
44 IF(IE.EQ.2) IST=LHALF(1) +1 44
45 DO 1005 J=IST,IND 45
46 46
47 CALL XACDEF(IE) 47
48 IF(J.NE.1) GO TO 1006 48
49 DO 1007 I=1,NCOL 49
50 IA=I+NBST 50
51 IB=NBNDST+4*NROW+3*NCOL+I 51
52 IC=NBNDST+4*NROW+2*NCOL+I 52
53 IF(BOUND(IC).GE.1.E35) GO TO 3 53
54 NCO=NRST+I 54
55 RO=RMESS(NCO) 55
56 OE=C. 56
57 IF(I.NE.NCOL) OE=RMESS(NCO+1)-RO 57
58 OW=C. 58
59 IF(I.NE.1) OW=RO-RMESS(NCO) 59
60 B(IA)=B(IA)-(OE+OW)*BOUND(IB)/2. 60
61 GO TO 1007 61
62 3 B(IA)=BOUND(IB) 62
63 1007 CONTINUE 63

```

1001

```

1006 IF(J.NE.NROW) GO TO 1008
      DO 1009 I=1,NCOL
        IA=NBNDST+(LHALF(2)-1)*NCOL+I
        IB=NBNDST+4*NROW+NCOL+I
        IC=NBNDST+4*NROW+I
        IF(BOUND(IC).GE.1.E35) GO TC 4
        NCO=NRST+I
        RO=RMESS(NCO)
        OE=0.
        IF(I.NE.NCOL)OE=RMESS(NCO+1)-RO
        OW=0.
        IF(I.NE.1) OW=RO-RMESH(NCO)
        B(IA)=B(IA)-(OE+OW)*BOUND(IB)/2.
        B(IA)=B(IA)-((RO+OE/4.)*OE+(RO-OW/4.)*OW)*BOUND(IB)/2.
        GO TO 1009
      4 B(IA)=BOUND(IB)
1009 CONTINUE

1008 IF(IE.EQ.1) JA=J
      IF(IE.EQ.2) JA=J-LHALF(1)
      IA=(JA-1)*NCOL+1+NBST
      IB=NBNDST+2*NROW+1-J
      IC=NBNDST+NROW+1-J
      IF(BOUND(IC).GE.1.E35) GO TO 5
      IN=NBNDST+NROW
      IF(J.EQ.1.AND.BOUND(IN).GE.1.E35)GO TC 2008
      IF(J.EQ.NROW.AND.BOUND(NBNDST+1).GE.1.E35)GO TO 2008
      NJO=NZST+J
      ON=0.
      IF(J.NE.1)ON=ZMESH(NJO-1)-ZMESH(NJO)
      OS=0.
      IF(J.NE.NROW)OS=ZMESH(NJO)-ZMESH(NJC+1)
      B(IA)=B(IA)-(ON+OS)*BOUND(IB)/2.
      GO TO 2008
      5 B(IA)=BOUND(IB)
2008 CONTINUE
      IA=NBNDST+JA*NCOL
      IB=NBNDST+4*NROW+1-J
      IC=NBNDST+3*NROW+1-J
      IF(BOUND(IC).GE.1.E35) GO TC 6
      IN=NBNDST+3*NROW
      IF(J.EQ.1.AND.BOUND(IN).GE.1.E35) GO TC 2009
      IN=NBNDST+2*NROW+1
      IF(J.EQ.NROW.AND.BOUND(IN).GE.1.E35) GC TC 2009
      NCO=NCOL+NRST
      NJO=NZST+J
      ON=0.
      IF(J.NE.1)ON=ZMESH(NJO-1)-ZMESH(NJO)
      OS=0.
      IF(J.NE.NROW)OS=ZMESH(NJO)-ZMESH(NJC+1)
      B(IA)=B(IA)-(ON+OS)*BOUND(IB)/2.
      GO TO 2009
      6 B(IA)=BOUND(IB)
2009 CONTINUE

      IT=NTST+1
      IW=NWST+1
      IX=NXST+1

      SPARE1=A0(IW)
      SPARE2=AE(IX)
      T(IT)=AE(IX)

```

```

1 W(IW)=0.
2 X(IX)=SPARE1
3 ISUF=NBNDST+NROW+1-J
4 JSUF=NBNDST+4*NROW+2*NCOL+1
5 KSUF=NBNDST+4*NROW+1
6 LSUF=ISUF+2*NROW
7 IF(BOUND(ISUF).GE.1.E35.OR.(J.EQ.1.AND.BOUND(JSUF).GE.1.E35).OR.
8 1 (J.EQ.NROW.AND.BOUND(KSUF).GE.1.E35)) GO TO 1012
9 GO TO 1013
1012 X(IX)=1.
11 T(IT)=0.
12 1013 DO 1010 I=2,NCOL
13 IW=IW+1
14 IX=IX+1
15 IT=IT+1
16 SPARE1=A0(IW)
17 SPARE3=SPARE2
18 SPARE2=AE(IX)
19 W(IW)=SPARE3/X(IX-1)
20 ICO=NBNDST+4*NROW+I
21 NCO=2*NCOL+ICO
22 IF(J.EQ.1.AND.BOUND(NCO).GE.1.E35)W(IW)=0.
23 IF(J.EQ.NROW.AND.BOUND(ICO).GE.1.E35)W(IW)=0.
24 X(IX)=SPARE1-W(IW)*T(IT-1)
25 IF(J.EQ.1.AND.BOUND(NCO).GE.1.E35)X(IX)=1.
26 IF(J.EQ.NROW.AND.BOUND(ICO).GE.1.E35)X(IX)=1.
27 T(IT)=SPARE2
28 IF(I.EQ.NCOL.AND.BOUND(LSUF).LT.1.E35)T(IT)=1.
29 IF((J.EQ.1.AND.BOUND(NCO).GE.1.E35).OR.(J.EQ.NROW.AND.BOUND(ICO)
30 1 .GE.1.E35)) T(IT)=0.
31 1010 CONTINUE
32
33 IF(BOUND(LSUF).GE.1.E35)W(IW)=0.
34 IF(BOUND(LSUF).GE.1.E35)X(IX)=1.
35
36 JST=NBNDST+(J-1)*NCOL+1
37 IF(IE.EQ.2) JST=JST-MHALF(1)
38 LIWST=NWST+1
39 LIWND=NWST+NCOL
40 LIXST=NXST+1
41 LIXND=NXST+NCOL
42 LIAST=NA SST+1
43 LIAND=NA SST+NCOL
44 LIBST=JST
45 LIBND=JST+NCOL-1
46 LITST=NTST+1
47 LITND=NTST+NCOL
48 WRITE(NTAP) (W(IRW),IRW=LIWST,LIWND),(X(IRW),IRW=LIXST,LIXND),
49 1 (AS(IRW),IRW=LIAST,LIAND),(B(IRW),IRW=LIBST,LIBND),
50 2 (T(IRW),IRW=LITST,LITND)
51 1005 CONTINUE
52 12 CONTINUE
53 IF(IE.EQ.2) GO TO 1003
54 IT=NCOL
55 IA=(NCOL-1)*(LHALF(1)-1)
56 NCO=NCOL-1
57 DO 1011 I=1,NCO
58 IA=IA+1
59 1011 ITYPE(I)=ITYPE(IA)
60 ITAPE=NT9
61 1003 CONTINUE
62 IRE=0
63 IF(G-1)14,15,16

```



```

15 REWIND NT9
GO TO 14
16 CALL SHUNT(NT9,IRE,-1,G+1)
14 IF(IRE.EQ.1) CALL SHUNT(NT9,IRE,1,G)
IF(SKIPS) GO TO 110
IF(NLINE-NROW/2) 102,101,101
101 REWIND NT1
GO TO 104
102 DO 103 IX=1,NLINE
103 BACKSPACE NT1
GO TO 105
104 IF(NFIRST) 105,105,106
106 DO 107 IX=1,NFIRST
107 READ( NT1)
105 IST=NCOST+1
108 DO 109 IX=1,NLINE
IND=IST+5*NCOL-1
READ(NT1)(PHI(IRW),IRW=IST,IND)
109 IST=IST+5*NCOL
REWIND NT1
110 RETURN
END

```

```

SUBROUTINE XACDEF (IE)
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1ST, NSGST, NSCAST,NPHIST,
2 NTHEST,NYST,NBBBST
COMMON /COEF/ RO,OE,ON,OW,OS,AW,SE,SW,NE,NW,J,I
COMMON/PHI/PHI(17000)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRDIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIIST
DIMENSION TMESH(1)
DIMENSION ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1)
DIMENSION AE(1),AO(1),D(1),A(1),SG(1)
DIMENSION THETA(1),Y(1),BBB(1)
DIMENSION W(1),X(1)
INTEGER SE,SW
EQUIVALENCE (PHI(1),TMESH(1))
EQUIVALENCE (PHI(1),ITYPE(1),BOUND(1),RMESH(1),ZMESH(1),B(1),
1 AS2(1),AS(1),SCATER(1))
EQUIVALENCE (SCATER(1),AE(1),AO(1),D(1),A(1),SG(1))
EQUIVALENCE (PHI(1), THETA(1),Y(1),BBB(1))
EQUIVALENCE (AO(1),W(1)),(AE(1),X(1))

DO 1006 I=1,NCOL
CALL NSEW(NDST,IE)
IC=NAS2ST+I
IB=NASST+I
IW=NWST+I
IX=NXST+I
IF(J.NE.1)AS2(IC)=AS(IB)
AS(IB)=0.
AE(IX)=0.
AO(IW)=0.
IF(I.NE.NCOL.AND.J.NE.NROW) AS(IB)=AS(IB)+OE*D(SE)/(2.*OS)
IF(I.NE.1.AND.J.NE.NROW) AS(IB)=AS(IB)+OW*D(SW)/(2.*OS)
IF(I.NE.NCOL.AND.J.NE.1) AE(IX)=AE(IX)+CN*D(NE)/(2.*OE)

```

```

IF(I.NE.NCOL.AND.J.NE.NROW) AE(IX)=AE(IX)+OS*D(SE)/(2.*OE)
NW=NW-NDST+NAST
NE=NE-NDST+NAST
SW=SW-NDST+NAST
SE=SE-NDST+NAST

IF(J.NE.NROW) AO(IW)=AO(IW)-AS(IB)
IF(J.NE.1) AO(IW)=AO(IW)-AS2(IC)
IF(I.NE.NCOL) AO(IW)=AO(IW)-AE(IX)
IF(I.NE.1) AO(IW)=AO(IW)-AE(IX-1)

IF(I.NE.1.AND.J.NE.1) AO(IW)=AO(IW)-CW*CN*A(NW) /4.
IF(I.NE.NCOL.AND.J.NE.1) AO(IW)=AO(IW)-OE*CN*A(NE)/4.
IF(I.NE.NCOL.AND.J.NE.NROW) AO(IW)=AO(IW)-OE*OS*A(SE)/4.
1006 IF(I.NE.1.AND.J.NE.NROW) AO(IW)=AO(IW)-CW*CS*A(SW) /4.

ISUF=NBNDST+NROW+1-J
OX=0.
IF(J.NE.1) OX=OX+ON
IF(J.NE.NROW) OX=OX+OS
IF(BOUND(ISUF).GE.1.E35) GO TO 1
AO(NWST+1)=A(NWST+1)-BOUND(ISUF)*CX/2.
1 ISUF=NBNDST+3*NROW+1-J
IF(BOUND(ISUF).GE.1.E35) GO TO 2
IW=NWST+NCOL
AO(IW)=A(IW)-BOUND(ISUF)*OX/2.

2 IF(J.NE.1.AND.J.NE.NROW) GO TO 5
IF(J.EQ.1) NN=NBNDST+4*NZ+2*NR
IF(J.EQ.NROW) NN=NBNDST+4*NROW
DO 2003 I=1,NCOL
NN=NN+1
IW=I+NWST
NCO=NRST+I
IF(BOUND(NN).GE.1.E35) GOTO 2003
OE=0.
IF(I.NE.NCOL) OE=RMESH(NCO+1)-RMESH(NCO)
OW=0.
IF(I.NE.1) OW=RMESH(NCO)-RMESH(NCO-1)
AO(IW)=AO(IW)-(OE+OW)*BOUND(NN)/2.
2003 CONTINUE
5 RETURN
END

```

```

SUBROUTINE ITERAT
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON /OMEGA/ OMEGA
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1ST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
COMMON/PHI/PHI(17000)
COMMON/IDI/ IDIST(3)
COMMON /COUNT/ COUNT(60)
COMMON ACCUR(4),WD,OMEGAC,DIFF,PRDIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IMMY
DIMENSION BBB(1),X(1),AS(1),AS2(1),Y(1),THETA(1),W(1),T(1)
DIMENSION JEAN(1)

```

```

INTEGER G
INTEGER COUNT
EQUIVALENCE (PHI(1),BBB(1),X(1),AS(1),AS2(1),Y(1),THETA(1),
1 W(1),T(1))
EQUIVALENCE (ACCUR(4),IRAD2)
EQUIVALENCE (PHI(1),JEAN(1))
IF(COUNT(G).EQ.0) GO TO 109
IF(COUNT(G).EQ.-1) GO TO 109
WRITE(6,1999)
1999 FORMAT (////' RESIDUAL',7X,' LAMBDA',11X,' OMEGA',8X,' DIFF',11X,
1 'PRODIF',16X,' THE PROPORTIONAL DIFFERENCE CRITERION'/85X,
2 'IS SATISFIED AS FAR AS MESH LINE.....')
OMEGA=OMEGAC
NCHA=0
IF(NROW-5) 108,108,100
100 IF(NROW-10) 106,106,101
101 IF(NROW-20) 105,105,102
102 IF(NROW-50) 104,104,103
103 N1=16
N2=7
OMAX=1.95
GO TO 109
104 N1=14
N2=6
OMAX=1.9
GO TO 109
105 N1=10
N2=5
OMAX=1.75
GOT 0 109
106 N1=7
N2=4
OMAX=1.7
GO TO 109
108 N1=6
N2=3
OMAX=1.6
109 CONTINUE
DO 1009 I=1,NUMIJ
1009 PHI(I)=0.
IF(NGAUSS.LE.0.OR.G.GT.NGAUSS) GO TO 1011
IND=MHALF(1)
READ(NT3) (PHI(I),I=1,IND)
IST=MHALF(1)+1
IND=MHALF(2)+MHALF(1)
READ(NT3) (PHI(I),I=IST,IND)
1011 IF(COUNT(G).GT.-1) GO TO 1012
IND=MHALF(1)
READ(NT10) (PHI(I),I=1,IND)
IST=MHALF(1)+1
IND=NUMIJ
READ(NT10) (PHI(I),I=IST,IND)
1012 CONTINUE
SUM2=0.
IF(COUNT(G).EQ.0.OR.COUNT(G).EQ.-1) GO TO 1005
NSLOR=IABS(COUNT(G))
IF(COUNT(G).LT.0) BACKSPACE 10
IF(COUNT(G).LT.0) BACKSPACE 10
CALL SLOW(NCOL,BBB(NBBBST+1),X(NXST+1),W(NWST+1),AS(NASST+1),AS2(N
1 AS2ST+1), Y(NYST+1),THETA(NTHEST+1),T(NTST+1),SUM2,IABC)
DO 1001 NI T=1,NSLOR
DIFF=0.
PRODIF=0.

```

```

1 SUM1=SUM2
2 SUM2=0.
3 IRAD2=NCOL*NCOL+NRWC*NROW
4 CALL SLOR
5 DO 10 ITEN=1,NUMIJ
6 IF(PHI(ITEN)) 11,10,10
7 11 PHI(ITEN)=0.
8 10 CONTINUE
9 IST=MHALF(1)+1
10
11 IF(NIT-2) 2010,2010,1002
12
13 1002 ORAT=RAT
14 RAT=SUM2/SUM1
15 OMEGA0=OMEGA
16 IF(ORAT+.001.LE.RAT) GO TO 51
17 IF(NIT.LT.N1) GO TO 51
18 IF(NCHA.LT.N2) GO TO 51
19 IF(RAT.LE.OMEGA-1.) GO TO 1006
20 RHO=(RAT+OMEGA-1.)/(OMEGA*SQRT(RAT))
21 IF(RHO.GT.1.) CALL MESSAGE(19H DIVERGENCE. RHO= *%,RHO)
22 IF(RHO.GT.1.) GO TO 1006
23 OMEGAB=2./(1.+SQRT(1.-RHO**2))
24 OMEGA=(OMEGA+OMEGAB)/2.
25 OMEGA=AMINI(OMEGA,OMAX)
26 NCHA=0
27 GO TO 1006
28 51 NCHA=NCHA+1
29
30 1006 WRITE(6,1007) SUM2,RAT,OMEGA,DIFF,PRODIF,IDI ST
31 1007 FORMAT(' ',2E15.7,F15.9,2E15.7,8X,3I8)
32 IRAD2=SQRT(FLOAT(IRAD2))
33 WRITE(6,2001)IRAD2
34 2001 FORMAT('+',107X,I8)
35 1008 IF(PRODIF.LT.ACCUR(3)) GO TO 1005
36 2010 JSS=3
37 1001 CONTINUE
38 CALL MESSAGE(31H ITERATIONS HAVE NOT CONVERGED& )
39 IEG1=NBBBST+NCOL+1
40 IEG2=IEG1-1+NROW
41 WRITE(6,2222)(JEAN(IEGER),IEGER=IEG1,IEG2)
42 2222 FORMAT(45H INDIVIDUAL ROWS HAVE REACHED FOLLOWING STATE /
43 1 3X2H0)10I4/2X3H10)10I4/2X3H20)10I4/5H 30)10I4/(5X10I4))
44 1005 CALL PRINT
45 RETURN
46 END
-----
52 SUBROUTINE SLOW(/,NCOL/,B,X,W,AS,AS2,Y,THETA,T,/SUM/,/IABC/)
53 COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
54 COMMON /LEFT/ NLEFT,NLINE,NFIRST
55 COMMON /NUM/ NZ,NR,NROW,NDOC,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT,NBAC
56 COMMON /NST/ NBNST,NRST,NZST,NBST,NAS2ST,NASST,NWST,NXST,NDST,NAS
57 1T,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST ,NTST
58 COMMON /OMEGA/ OMEGA
59 COMMON/PHI/PHI(17000)
60 COMMON/IDI/ IDIST(3)
61
62 COMMON ACCUR(4),WD,OMEGC,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
63 1 G,NIT,IMMY

```

1001

```

DIMENSION B(NCOL),X(NCOL),W(NCOL),AS(NCOL),Y(NCOL),THETA(NCOL),
1 AS2(NCOL),T(NCOL)
DIMENSION JEAN(1),MD(3)
EQUIVALENCE (ACCUR(4),IRAD2)
EQUIVALENCE (PHI(1),JEAN(1))
RETURN
===== ENTRY SLOR =====
ENTRY SLOR
DO 100 I=1,3
100 IDIST(I)=NCOL
IP=0
NTAP=NT1
DO 1001 J=1,NROW
MD(1)=NCOL
MD(2)=NCOL
MD(3)=NCOL
IF(NFIRST-J) 3002,2002,2002
2002 READ(NTAP) (W(I),I=1,NCOL),(X(I),I=1,NCOL),(AS(I),I=1,NCOL),
1 (B(I),I=1,NCOL),(T(I),I=1,NCOL)
IF(NFIRST.EQ.J) REWIND NT1
DO 1003 I=1,NCOL
IP=IP+1
THETA(I)=-B(I)*OMEGA+(1.-CMEGA)*X(I)*PHI(IP)
IF(T(I).EQ.0.) GO TO 1003
IF(I.NE.1) THETA(I)=THETA(I)+(1.-OMEGA)*W(I)*(X(I-1)*PHI(IP-1)+
1 T(I-1)*PHI(IP))
IF(I.NE.NCOL) THETA(I)=THETA(I)+(1.-CMEGA)*(T(I)*PHI(IP+1))
IQ=IP-NCOL
IF(J.NE.1) THETA(I)=THETA(I) - OMEGA*AS2(I)*PHI(IQ)
IQ=IP+NCOL
IF(J.NE.NROW) THETA(I)=THETA(I) - OMEGA*AS(I)*PHI(IQ)
1003 CONTINUE
Y(1)=THE TA(1)
DO 1004 I=2,NCOL
1004 Y(I)=THE TA(I)-W(I)*Y(I-1)
PHIIP=PHI(IP)
PHI(IP)=Y(NCOL)/X(NCOL)
SUM=SUM+ABS(PHIIP-PHI(IP))
IF(PHI(IP).LT.1.E-35.OR.NIT.EQ.1) GO TO 1006
DIFF=AMAX1(ABS(PHIIP-PHI(IP)),DIFF)
PRODIF =AMAX1(ABS(1.-PHIIP/PHI(IP)),PRODIF)
1006 IY=NCOL
IZ=NCOL
DO 1005 I=2,NCOL
IP=IP-1
IY=IY-1
PHIIP=PHI(IP)
PHI(IP)=(Y(IY)-T(IY)*PHI(IP+1))/X(IY)
SUM=SUM+ABS(PHIIP-PHI(IP))
IF(PHI(IP).LT.1.E-35.OR.NIT.EQ.1) GO TO 1005
DIFF=AMAX1(ABS(PHIIP-PHI(IP)),DIFF)
PROD=ABS(1.-PHIIP/PHI(IP))
1007 PRODIF=AMAX1(PROD,PRODIF)
DO 1008 ID=1,3
1008 IF(PROD.GT.ACCUR(ID)) MD(ID)=IY
1005 CONTINUE
DO 1002 I=1,NCOL
1002 AS2(I)=AS(I)

```

```

1 3005 DO 1009 ID=1,3
2 1009 IDIST(ID)=MINO(MD(ID),IDIST(ID))
3 IF(NIT.LT.NSLOR) GO TO 2223
4 IEGER=NBBSST+NCOL+NROW+1-J
5 JEAN(IEGER)=MD(3)
6 2223 IRADIU=(MD(3)-1)*(MD(3)-1) + (NROW-J)*(NROW-J)
7 IRAD2=MINO(IRADIU,IRAD2)
8 IP=IP-1+NCOL
9 GO TO 1001
10
11 3002 IW=NCOST+(J-NFIRST-1)*5*NCOL
12 IX=IW+NCOL
13 IS=IX+NCOL
14 IB=IS+NCOL
15 IT=IB+NCOL
16 3003 DO 3005 I=1,NCOL
17 IP=IP+1
18 IX=IX+1
19 IS=IS+1
20 IB=IB+1
21 IT=IT+1
22 IW=IW+1
23 THETA(I)=-PHI(IB)*OMEGA+(1.-OMEGA)*PHI(IX)*PHI(IP)
24 IF(PHI(IT).EQ.0.)GO TO 3005
25 IF(I.NE.1) THETA(I)=THETA(I)+(1.-OMEGA)*PHI(IW)*PHI(IX-1)*
26 PHI(IP-1)+PHI(IT-1)*PHI(IP)
27 IF(I.NE.NCOL) THETA(I)=THETA(I)+THETA(I)+(1.-OMEGA)*(PHI(IT)*PHI(IP+1))
28 IQ=IP-NCOL
29 IF(J.NE.1) THETA(I)=THETA(I)-OMEGA*AS2(I)*PHI(IQ)
30 IQ=IP+NCOL
31 IF(J.NE.NROW) THETA(I)=THETA(I)-OMEGA*PHI(IS)*PHI(IQ)
32 3005 CONTINUE
33 Y(1)=THE TA(1)
34 IW=IW+1-NCOL
35 DO 3006 I=2,NCOL
36 IW=IW+1
37 3006 Y(I)=THE TA(I)-PHI(IW)*Y(I-1)
38 PHIIP=PHI(IP)
39 PHI(IP)=Y(NCOL)/PHI(IX)
40 SUM=SUM+ABS(PHIIP-PHI(IP))
41 IF(PHI(IP).LT.1.E-35.OR.NIT.EQ.1)GO TO 3007
42 DIFF=AMAX1(ABS(PHIIP-PHI(IP)),DIFF)
43 PRODIF=AMAX1(ABS(1.-PHIIP/PHI(IP)),PRODIF)
44 3007 IY=NCOL
45 IZ=NCOL
46 DO 3008 I=2,NCOL
47 IP=IP-1
48 IY=IY-1
49 IX=IX-1
50 IT=IT-1
51 PHIIP=PHI(IP)
52 PHI(IP)=(Y(IY)-PHI(IT)*PHI(IP+1))/PHI(IX)
53 SUM=SUM+ABS(PHIIP-PHI(IP))
54 IF(PHI(IP).LT.1.E-35.OR.NIT.EQ.1)GO TO 3008
55 DIFF=AMAX1(ABS(PHIIP-PHI(IP)),DIFF)
56 PROD=ABS(1.-PHIIP/PHI(IP))
57 PRODIF=AMAX1(PROD,PRODIF)
58 DO 3010 ID=1,3
59 3010 IF(PROD.GT.ACCUR(ID))MD(ID)=IY
60 3008 CONTINUE
61 ISUF=IS-NCOL
62 DO 3004 I=1,NCOL
63 ISUF=ISUF+1

```

16-1

```

3004 AS2(I)=PHI(I,SUF)
GO TO 30C9
1001 CONTINUE
RETURN
END

```

1
2
3
4
5

```

WRITE(NT10) NCOL,(RMESH(J),J=IIS,IIN)
IIS = NZST+1
IIN = NZST+NROW
WRITE(NT10) NROW,(ZMESH(J),J=IIS,IIN)
RETURN
END

```

1
2
3
4
5
6

```

SUBROUTINE PRINT
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/GEOM/GEOM
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1ST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
COMMON/PHI/PHI(17000)
COMMON /COUNT/ COUNT(60)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDI ST
DIMENSION ZMESH(1),RMESH(1)
INTEGER G,GEOM
INTEGER COUNT
EQUIVALENCE (PHI(1),RMESH(1),ZMESH(1))

```

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

IND=MHALF(1)
IF(COUNT(G).NE.-1)
1WRITE(NT10) (PHI(I),I=1,IND)
IST=MHALF(1)+1
IND=MHALF(1)+MHALF(2)
IF(COUNT(G).NE.-1)
1WRITE(NT10) (PHI(I),I=IST,IND)
20 IST=1
IND=MINO(11,NCOL)
21 ISTM=IST-1
INDM=IND-1
IIS=NRST+IST
IIN=NRST+IND
IF (GEOM.EQ.1) WRITE (6,97) G,ISTM,INDM,(RMESH(J),J=IIS,IIN)
97 FORMAT (////' FLUXES IN GROUP',I3,' FOR COLUMNS',I3,' TO',I3//
*' Z / R',F8.2,10F10.2)
IF (GEOM.EQ.2) WRITE (6,98) G,ISTM,INDM,(RMESH(J),J=IIS,IIN)
98 FORMAT (////' FLUXES IN GROUP',I3,' FOR COLUMNS',I3,' TO',I3//
*' Y / X',F8.2,10F10.2)
WRITE (6,1000)
1000 FORMAT (' ')
DO 23 J=1,NROW
IJ=NROW-J
JI=NZST+J
NST=(J-1)*NCOL+IST
NND=NST-IST+IND
23 WRITE(6,99)IJ,ZMESH(JI),(PHI(N),N=NST,NND)
99 FORMAT(' ',I3,F7.2,2X,1P11E10.3)
IF(IND.EQ.NCOL)GO TO 6
IST=IND+1
IND=MINO(IND+11,NCOL)
GO TO 21
6 WRITE(6,9)
9 FORMAT(////)
IF (G.LT.NG) RETURN

IIS = NRST+1
IIN = NRST+NCOL

```

```

SUBROUTINE OUTPUT
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/PUCU/ PUCU
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON/NUM/NBACK
COMMON/NST/NBNDST,NRST,NZST,NXST(14)
COMMON/GEOM/GEOM
COMMON /ISW/ISW
COMMON /COUNT/ COUNT(60)
COMMON /PHI/PHI(17000)
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDI ST
DIMENSION ZMESH(1),RMESH(1),IREL(4)
DIMENSION SAMBO(50,2),NAGRP(50),FACTOR(50),ACTIV(1)
DIMENSION SAMPLE(50,2),BUFF(12),DATWRD(4),DATWD(3)
DIMENSION TITLE(18)
INTEGER GEOM,SAMPLE,G,SAMBO
INTEGER COUNT
EQUIVALENCE (PHI(1),RMESH(1),ZMESH(1)),(PHI(1),ACTIV(1))
EQUIVALENCE (WD,IWD)

```

12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

IDEN = 100
IFTA=NTLC
IF(NBACK.LT.0) IFTA=NT3
80 REWIND NT11
REWIND IFTA
ISW=0
PUCU=0
1 IF (IDEN.LT.50) GO TO 2
CALL INDEX (J,IWD,4HEND&)
IF (J.EQ.1) GO TO 300
CALL INDEX(J,IWD,4HDUMP*ALLF*SAMP*RCW*COLU*ACT I*ITL*CURRE*PUCU&)
IDEN = 0
IF(J.NE.1) GO TO 12
JTA=INDEC(0)
CALL DUMPS(1,JTA)
ISW=ISW+1
GO TO 1
12 IF (J.NE.5) GO TO 13
K=INDEC(0)+1
JMIN=INDEC(0)
JMA X=INDEC(0)
IWD=ITDEC(4)
13 IF (J.NE.4) GO TO 14
KROW=INDEC(0)+1
JMIN=INDEC(0)
JMA X=INDEC(0)
IWD=ITDEC(4)
14 IF (J.EQ.2) IWD=ITDEC(4)
IF (J.EQ.3)GO TO 200
IF (J.EQ.6)GO TO 60
IF (J.EQ.7)GO TO 190

```

```

IF(J.EQ.8) GO TO 180
IF(J.EQ.9) PUCU=1
IF(J.EQ.10) GO TO 2
IF(J.EQ.9) GO TO 301
ISW=ISW+2*(J-1)
GO TO 2
301 IWD=ITDEC(4)
GO TO 1
190 CALL GODEC(TITLE)
ISW=ISW+256
GO TO 301
180 CALL CURE1
ISW=ISW+128
GO TO 1
200 IWD=NEWDEC(0)
IWD=ITDEC(4)
NSOW=1
NSOL=1
DO 231 I=1,2
DO 231 J=1,50
231 SAMPLE(J,I)=-1
235 CALL INDEX(J,IWD,26HR*Z*AS*SAMP*GROU*FACT*XYE)
IF(J1.GT.2) J1=J1-2
IF(J1.EQ.3)GO TO 236
I=1
SAMPLE(I,J1)=INDEC(1)
232 MWD=NEWDEC(0)
IF(MWD)237,233,49
237 IWD=ITDEC(4)
GO TO 235
49 CALL ERROR (19HERROR. (OUTPUT,49)E )
233 IWD=ITDEC(1)
ISKIP=INDEC(1)
IWD=ITDEC(1)
NEXT=INDEC(1)
234 I=I+1
SAMPLE(I,J1)=SAMPLE(I-1,J1)+ISKIP
IF(J1.EQ.1) NSOL=I
IF(J1.EQ.2) NSOW=I
IF(SAMPLE(I,J1)-NEXT)234,232,234
236 ISW=ISW+4
GO TO 1
60 NACTST=NZST+NROW
ISW=ISW+32
NAOW=1
NAOL=1
DO 64 I=1,50
DO 64 J=1,2
64 SAMBO(I,J)=0
IWD=NEWDEC(0)
109 IWD=ITDEC(4)
108 CALL INDEX(J,IWD,26HR*Z*AS*SAMP*GROU*FACT*XYE)
IF(J.EQ.7.OR.J.EQ.8) J=J-6
IF(J.EQ.9) J=7
GO TO (110,110,109,130,140,150,120),J
130 DO 65 I=1,50
DO 65 J=1,2
65 SAMBO(I,J)=SAMPLE(I,J)
NAOW=NSOW
NAOL=NSOL
GO TO 109
110 I=1
SAMBO(I,J)=INDEC(1)

```

```

1 111 MWD=NEWDEC(0)
2 IF(MWD)109,113,49
3 113 IWD=ITDEC(1)
4 ISKIP=INDEC(1)
5 IWD=ITDEC(1)
6 NEXT=INDEC(1)
7 112 I=I+1
8 SAMBO(I,J)=SAMBO(I-1,J)+ISKIP
9 IF(J.EQ.2)NAOW=I
10 IF(J.EQ.1)NAOL=I
11 IF(SAMBO(I,J)-NEXT)112,111,112
12 140 I=1
13 NAGPP(I)=INDEC(1)
14 144 IF(NEWDEC(0))143,142,145
15 145 Ibuff=INDEC(1)
16 141 I=I+1
17 NAGRP(I)=NAGRP(I-1)+ 1
18 IF(Ibuff.EQ.NAGRP(I))GO TO 144
19 GO TO 141
20 142 I=I+1
21 Ibuff=ITDEC(1)
22 NAGRP(I)=INDEC(1)
23 GO TO 144
24 143 IWD=ITDEC(4)
25 NAGRP(I+1)=NG+1
26 NAC=NACTST+I
27 NSC=NACTST+NAOL*NAOW
28 DO 66 I=NAC,NSC
29 66 ACTIV(I)=0.
30 GO TO 108
31 150 IG=0
32 151 IF(NEWDEC(0))109,3280,152
33 152 IG=IG+1
34 FACTOR(IG)=FLDEC(1)
35 GO TO 151
36 120 IG=1
37 GO TO 1
38
39 2 IDEN = 100
40 DO 3 G=1,NG
41 IND=MHALF(1)
42 READ(IFTA) (PHI(I),I=1,IND)
43 IST=MHALF(1)+1
44 IND=MHALF(1)+MHALF(2)
45 READ(IFTA) (PHI(I),I=IST,IND)
46 4 IF(MOD(ISW,2).EQ.1)CALL DUMPS(2,JTA)
47 5 IF(MOD(ISW,4)/2.EQ.1)GO TO 20
48 6 IF(MOD(ISW,8)/4.EQ.1)GO TO 30
49 10 IF(MOD(ISW,16)/8.EQ.1) CALL PUNROW (KROW,IMIN,IMAX)
50 8 IF(MOD(ISW,32)/16.EQ.1)CALL PUNCOL(K,JMIN,JMAX)
51 9 IF(MOD(ISW,64)/32.EQ.1)GO TO 160
52 IF(MOD(ISW,256)/128.EQ.1) CALL CURE2
53 GO TO 3
54 20 IST=1
55 IND=MINO(11,NCOL)
56 21 ISTM=IST-1
57 INDM=IND-1
58 IIS=NRST+IST
59 IIN=NRST+IND
60 IF (GEOM.EQ.1) WRITE (6,97) G,ISTM,INDM,(RMESH(J),J=IIS,IIN)
61 97 FORMAT (////' FLUXES IN GROUP',I3,' FOR COLUMNS',I3,' TO',I3//
62 *' Z / R',F8.2,10F10.2)
63 IF (GEOM.EQ.2) WRITE (6,98) G,ISTM,INDM,(RMESH(J),J=IIS,IIN)

```

```

98 FORMAT (////' FLUXES IN GROUP',I3,' FOR COLUMNS',I3,' TO',I3//
*' Y / X',F8.2,10F10.2)
WRITE (6,1000)
1000 FORMAT (' ')
DO 23 J=1,NROW
IJ=NROW-J
JI=NZST+J
NST=(J-1)*NCOL+IST
NND=NST-IST+IND
23 WRITE(6,99)IJ,ZMESH(JI),(PHI(N),N=NST,NND)
99 FORMAT(' ',I3,F7.2,2X,1P11E10.3)
IF(IND.EQ.NCOL)GO TO 6
IST=IND+1
IND=MIN0(IND+11,NCOL)
GO TO 21
30 IB=0
42 WRITE(6,37)G
37 FORMAT (////5X,'SAMPLE OF FLUXES IN GROUP',I3)
DO 38 IJ=1,NSOW
JSUF=NSOW+1-IJ
JA=SAMPLE(JSUF,2)
J=NROW-JA
39 I=IB
ISUF=(J-1)*NCOL+1
53 I=I+1
IS=SAMPLE(I,1)
IF(IS.LT.0)GO TO 54
IIS=ISUF+IS
ID=I-IB
BUFF(ID)=PHI(IIS)
=====
IF(I.LT.IB+12) GO TO 53
54 IF(IS.LT.0) I=I-1
IF(IJ.GT.1) GO TO 45
IST=IB+1
IND=I
WRITE(6,46)(SAMPLE(IC,1),IC=IST,IND)
46 FORMAT (7X,'Z',I9,11I10)
45 JND=I-IB
IH=NZST+J
WRITE(6,48)JA,ZMESH(IH),(BUFF(IA),IA=1,JND)
48 FORMAT (' ',I3,F6.1,1X,1P12E10.3)
38 CONTINUE
IS=SAMPLE(I+1,1)
IF(IS.LT.0)GO TO 10
IB=I
GOTO 42
160 IF(NAGRP(IG).NE.G)GO TO 3
DO 70J=1,NAOW
DO 70 I=1,NAOL
ISUF=NACTST +(J-1)*NAOL+I
JA=NAOW+1-J
JB=NROW-SAMBO(JA,2)
JSUF=(JB-1)*NCOL+SAMBO(I,1) +1
70 ACTIV(ISUF)=ACTIV(ISUF)+FACTOR(IG)*PHI(JSUF)
IG=IG+1
3 CONTINUE
IF(MOD(ISW,64)/32.NE.1)GO TO 71
IF(MOD(ISW,512)/256.NE.1) GO TO 79
WRITE(6,69) TITLE
69 FORMAT(1H118A4)
79 WRITE(6,75)

```

```

75 FORMAT (////5X,'DETECTOR ACTIVATIONS / REACTION RATES'//)
ISA=0
LEN=MIN0(NAOL,12)
74 IW=ISA+1
WRITE(6,76) (SAMBO(I,1),I=IW,LEN)
76 FORMAT (3X,12I10)
NAC=NACTST+ISA+1
DO 73 J=1,NAOW
NSC=NAC-1+LEN -ISA
JA=NAOW-J+1
WRITE(6,72)SAMBO(JA,2),(ACTIV(I),I=NAC,NSC)
72 FORMAT (' ',I4,2X,1P12E10.3)
73 NAC=NAC+NAOL
IF(LEN.EQ.NAOL) GO TO 71
ISA=LEN
LEN=MIN0(NAOL,LEN+12)
GO TO 74
71 CALL INDEX(J,IWD,7HREWIND&)
IF (J-1) 80,81,80
81 IWD=ITDEC(4)
GO TO 80
300 RETURN
3280 IWD=ITDEC(4)
WRITE(6,3281) IWD
3281 FORMAT(15H DATA ERROR.WD=A4,70H PROGRAM WILL ATTEMPT TO CONTINUE 0
IUTPUT OPTIONS SPECIFIED BEFORE THIS )
ISW=ISW-32
GO TO 2
END
-----
SUBROUTINE DUMPS(IGO,JTA)
COMMON/CHOPIN/NCHOP,IST,IND,JST,JND
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMI,J,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON /PHI/PHI(17000)
COMMON /ISW/ ISW,NROW,MCOL ,LROW,LCOL
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODF ,LHALF(2),MHALF(2),NHALF(2),
I G,NIT,IOIST
EQUIVALENCE (WD,IWD)
GO TO (100,200),IGO
100 IST=1
IND=NCOL
JST=I
JND=NROW
REWIND JTA
C
104 IWD=ITDEC(4)
CALL INDEX(J,WD,8HR*X*Z*YE)
GO TO (101,101,102,102,103),J
101 IWD=ITDEC(4)
CALL INDEX(J,WD,6HLE*GE&)
IF(J.EQ.1) IND=INDEC(0)+1
IF(J.EQ.2)IST=INDEC(0)+1
GO TO 104
102 IWD=ITDEC(4)
CALL INDEX(J,WD,6HLE*GE&)
IF(J.EQ.1)JST=NROW-INDEC(0)
IF(J.EQ.2)JND=NROW-INDEC(0)
GO TO 104
103 CALL INDEX(J,IWD,5HDOUR&)

```

```

IF(J.NE.1) GO TO 400
ISW=ISW+64
IWD=ITDEC(4)
GO TO 400

200 CALL OF(JTA)
IF(MOD(ISW,128)/64.NE.1)GO TO 300
MROW=(JND-JST)*2+1
MROW2=MROW/2
MCOL=(IND-IST)*2+1
IND1=IND-1
IND2=IND-2
JM=0
DO 201 J=JST,JND
ISUF=(J-1)*NCOL+IST-1
DO 202 I=IST,IND1
ISUF=ISUF+1
PHIG=(PHI(ISUF)+PHI(ISUF+1))/2.
CALL MOVDAT(PHI(ISUF),1)
202 CALL MOVDAT(PHI(ISUF+1),1)
CALL MOVDAT(PHI(ISUF+1),1)
JM=JM+1
IF(J.EQ.JND)GO TO 201
IF(JM.NE.MROW2.OR.2*JM.NE.MROW) GO TO 203
CALL CF
CALL OF(JTA)
203 ISUF=(J-1)*NCOL+IST
KSUF=ISUF+NCOL
PHIE=(PHI(ISUF)+PHI(KSUF))/2.
PHIG=(PHI(ISUF+1)+PHI(KSUF+1))/2.
PHIF=(PHIE+PHIG)/2.

DO 204 I=IST,IND1
CALL MOVDAT(PHIE,1)
CALL MOVDAT(PHIF,1)
PHIE=PHIG
ISUF=ISUF+1
KSUF=KSUF+1
PHIG=(PHI(ISUF+1)+PHI(KSUF+1))/2.
204 PHIF=(PHIE+PHIG)/2.
CALL MOVDAT(PHIE,1)
JM=JM+1
IF(JM.NE.MROW2.OR.2*JM.EQ.MROW) GO TO 201
CALL CF
CALL OF(JTA)
201 CONTINUE
CALL CF
GO TO 400
300 MROW=JND-JST+1
MROW2=MROW/2
MCOL=IND-IST+1
DO 301 J=1,MROW2
ISUF=(J-2+JST)*NCOL+IST
301 CALL MOVDAT(PHI(ISUF),IND-IST+1)
CALL CF
CALL OF(JTA)
MROW3=MROW2+1
DO 302 J=MROW3,MROW
ISUF=(J-2+JST)*NCOL+IST
302 CALL MOVDAT(PHI(ISUF),IND-IST+1)
CALL CF
400 RETURN
END

```

```

1 SUBROUTINE CURE1 1
2 COMMON/CURRENT/ISTORE(50) 2
3 COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA 3
4 IST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST 4
5 COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2), 5
6 1 G,NIT,IDIIST 6
7 EQUIVALENCE (WD,IWD) 7
8 8
9 IRET=0 9
10 DO 30 I=1,50 10
11 30 ISTORE(I)=-100 11
12 ITOP=50 12
13 IBOT=1 13
14 100 IF(NEWDEC(0))23,1,231 14
15 1 IWD=ITDEC(1) 15
16 I1=INDEC(0)+1 16
17 J1=INDEC(1)+1 17
18 ITCH=NEWDEC(0) 18
19 IWD=ITDEC(1) 19
20 ITCH=NEWDEC(0) 20
21 IWD=ITDEC(1) 21
22 I2=INDEC(0)+1 22
23 J2=INDEC(1)+1 23
24 ITCH=NEWDEC(0) 24
25 IWD=ITDEC(1) 25
26 IF(J1-J2) 2,4,3 26
27 2 ISTORE(ITOP)=J1 27
28 ISTORE(ITOP-1)=I1 28
29 ISTORE(ITOP-2)=J2 29
30 ISTORE(ITOP-3)=I2 30
31 ITOP=ITOP-4 31
32 GO TO 11 32
33 3 ISTORE(IBOT)=I1 33
34 ISTORE(IBOT+1)=J1 34
35 ISTORE(IBOT+2)=I2 35
36 ISTORE(IBOT+3)=J2 36
37 IBOT=IBOT+4 37
38 GO TO 15 38
39 4 IF(NEWDEC(0))31,5,311 39
40 31 IRET=1 40
41 IWD=ITDEC(4) 41
42 GO TO 6 42
43 311 IWD=INDEC(1) 43
44 IRET=1 44
45 GO TO 6 45
46 5 IWD=ITDEC(1) 46
47 CALL INDEX(J,IWD,4H(*,C)) 47
48 IF(J-2)7,6,23 48
49 6 ISTORE(IBOT)=I1 49
50 ISTORE(IBOT+1)=J1 50
51 ISTORE(IBOT+2)=I2 51
52 ISTORE(IBOT+3)=J2 52
53 IBOT=IBOT+4 53
54 IF(IRET)100,100,232 54
55 7 I3=INDEC(0)+1 55
56 J3=INDEC(1)+1 56
57 ITCH=NEWDEC(0) 57
58 IWD=ITDEC(1) 58
59 IF(J2-J3)9,8,10 59
60 8 I2=I3 60
61 J2=J3 61
62 GO TO 4 62
63 9 ISTORE(ITOP)=J1 63

```

top

```

ISTORE(I TOP-1)=I1
ISTORE(I TOP-2)=J2
ISTORE(I TOP-3)=I2
ISTORE(I TOP-4)=J3
ISTORE(I TOP-5)=I3
ITOP=I TOP-6
I2=I3
J2=J3
GO TO 11
10 ISTORE(IBOT)=I1
   ISTORE(IBOT+1)=J1
   ISTORE(IBOT+2)=I2
   ISTORE(IBOT+3)=J2
   ISTORE(IBOT+4)=I3
   ISTORE(IBOT+5)=J3
   IBOT=IBOT+6
   I2=I3
   J2=J3
   GO TO 15
11 IF(NEWDEC(0))23,12,231
12 IWD=ITDEC(1)
   CALL INDEX(J,IWD,4H(*,6)
   IF(J-2)17,18,232
18 ISTORE(ITOP)=-100
   ITOP=ITOP-1
   GO TO 100
17 I1=I2
   J1=J2
   I2=INDEC(0)+1
   J2=INDEC(1)+1
   ITCH=NEWDEC(0)
   IWD=ITDEC(1)
   IF(J2-J1)14,13,13
13 ISTORE(ITOP)=J2
   ISTORE(I TOP-1)=I2
   ITOP=ITOP-2
   GO TO 11
14 ISTORE(ITOP)=-100
   ITOP=ITOP-1
   GO TO 3
15 IF(NEWDEC(0))23,16,231
16 IWD=ITDEC(1)
   CALL INDEX(J,IWD,4H(*,8)
   IF(J-2)20,19,23
19 ISTORE(IBOT)=-100
   IBOT=IBOT+1
   GO TO 100
20 I1=I2
   J1=J2
   I2=INDEC(0)+1
   J2=INDEC(1)+1
   ITCH=NEWDEC(0)
   IWD=ITDEC(1)
   IF(J2-J1)21,21,22
21 ISTORE(IBOT)=I2
   ISTORE(IBOT+1)=J2
   IBOT=IBOT+2
   GO TO 15
22 ISTORE(IBOT)=-100
   IBOT=IBOT+1
   GO TO 2

```

```

1      23 IWD=ITDEC(4)
2      GO TO 232
3      231 IWD=INDEC(1)
4      232 RETURN
5      END
6
7
8
9
10
11      SUBROUTINE PUNRCW(KROW,IMIN,IMAX)
12      COMMON /PHI/PHI(17000)
13      COMMON ACCUR(4),WD,OMEGA,DIFF,PRCDIF ,LHALF(2),MHALF(2),NHALF(2),
14      1      G,NIT,IDIST
15      COMMON/NST/NBNDST,NRST,NZST,NXST(13)
16      COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
17      COMMON/NUM/NBACK
18      DIMENSION ZMESH(1),RMESH(1)
19      INTEGER G
20      EQUIVALENCE (PHI(1),RMESH(1),ZMESH(1))
21      90 CALL PUNLAB (IMIN,IMAX)
22      KI=KROW-1
23      M=NZST+NROW+1-KROW
24      WRITE(6,201) KI,ZMESH(M),G,IMIN,IMAX
25      201 FORMAT (/' ROH',I3,' ( Z(Y) =',F8.3,' )      FLUXES IN GROUP ',
26      * I2,5X,' COLUMN',I3,' TO',I3/)
27      J3=(NROW-KROW)*NCOL
28      DO 203 I=1,NCOL
29      J3=J3+1
30      ABUFF=-PHI(J3)
31      203 CALL PUNNUM (ABUFF,I)
32      CALL PUNEND
33      RETURN
34      END
35
36
37
38
39
40      SUBROUTINE PUNCOL(K,JMIN,JMAX)
41      COMMON /PHI/PHI(17000)
42      COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
43      1      G,NIT,IDIST
44      COMMON/NST/NBNDST,NRST,NZST,NXST(13)
45      COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLDR,NMAT
46      COMMON/NUM/NBACK
47      DIMENSION ZMESH(1),RMESH(1)
48      INTEGER G
49      EQUIVALENCE (PHI(1),RMESH(1),ZMESH(1))
50      50 CALL PUNLAB (JMIN,JMAX)
51      KI=K-1
52      M=NRST+K
53      WRITE(6,29) KI,RMESH(M),G,JMIN,JMAX
54      29 FORMAT (/' COLUMN',I3,' ( R(X) =',F8.3,
55      * ' ) OF FLUXES IN GROUP ',I2,5X,' RCW',I3,' TO',I3/)
56      DO 28 J=1,NROW
57      J3=(NROW-J)*NCOL+K
58      ABUFF=-PHI(J3)
59      28 CALL PUNNUM (ABUFF,J)
60      CALL PUNEND
61      RETURN
62      END
63

```



```

SUBROUTINE PUNLAB (J1,J2)
DIMENSION FLUX (10)
DATA NPAGE /-1/
IF (NPAGE.LT.0) WRITE (6,90)
90 FORMAT ('1')
NPAGE=10
JMIN=J1+1
JMAX=J2+1
L=0
RETURN
C
===== ENTRY PUNNUM =====
ENTRY PUNNUM (ABUFF,J)
IF (J.LT.JMIN.OR.J.GT.JMAX) RETURN
L=L+1
FLUX(L)=ABUFF
IF (L.LT.6) RETURN
WRITE (7,100) (FLUX(I),I=1,6)
WRITE (6,100) (FLUX(I),I=1,6)
100 FORMAT (6X,1P6E11.3)
L=0
RETURN
C
===== ENTRY PUNEND =====
ENTRY PUNEND
IF (L.EQ.0) GO TO 200
WRITE (7,100) (FLUX(I),I=1,L)
WRITE (6,100) (FLUX(I),I=1,L)
200 WRITE (6,2000)
2000 FORMAT (/8X,'( BOUNDARY CONDITION CARDS HAVE BEEN PUNCHED )'//)
RETURN
END

```

```

SUBROUTINE OF(NT)
COMMON /PHI/ PHI(11000),USE(6000)
N=NT
3 IF(IOPEN.EQ.12345) CALL ERROR(34HFILE NOT CLOSED CORRECTLY. (OF,3)
*E )
L=0
IOPEN=12345
RETURN
C
===== ENTRY MOVDAT =====
ENTRY MOVDAT(A,M)
DIMENSION A(M)
1 IF(IOPEN.NE.12345) CALL ERROR(34HFILE ACCESSED WHEN CLOSED. (OF,1)
*E )
DO 2 I=1,M
L=L+1
2 USE(L)=A(I)
RETURN
C
===== ENTRY CF =====
ENTRY CF
IF(IOPEN.NE.12345) GO TO 1
WRITE(N) (USE(I),I=1,L)
IOPEN=0
RETURN
END

```

```

SUBROUTINE CURE2
COMMON /TAPE/ NT1,NT2,NT3,NT4,NT8,NT9,NT10,NT11
COMMON/CURENT/ISTORE(50)
COMMON/PHI/PHI(17000)
COMMON /GEOM/ GEOM
COMMON /NUM/ NZ,NR,NROW,NCOL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
COMMON/NUM/NBACK
COMMON /NST/ NBNDST,NRST,NZST, NBST,NAS2ST,NASST,NWST,NXST,NDST,NA
1ST,NSGST,NSCAST,NPHIST,NTHEST,NYST,NBBBST,NTST
COMMON ACCUR(4),WD,OMEGA,DIFF,PRODIF ,LHALF(2),MHALF(2),NHALF(2),
1 G,NIT,IDIIST
DIMENSION RMESH(1),ZMESH(1),D(1),ITYPE(1)
DIMENSION DAWD(4)
INTEGER SW,SE,G,GECM
EQUIVALENCE (PHI(1),D(1),RMESH(1),ZMESH(1),ITYPE(1))
EQUIVALENCE (NSGST,NTYST)
DATA DAWD(1),DAWD(2),DAWD(3),DAWD(4) /' JZ=', ' JR=', ' JY=', ' JX=' /
C
IF (GEOM.LT.2) GO TO 104
DAWD(1)=DAWD(3)
DAWD(2)=DAWD(4)
104 NT2ST=NTYST+NCOL-1
IBOT=1
WRITE(6,101) G
101 FORMAT (///51X,'----- CURRENTS FOR GROUP',I3,
* ' -----'/)
I1=ISTORE(IBOT)
J1=ISTORE(IBOT+1)
I2=ISTORE(IBOT+2)
J2=ISTORE(IBOT+3)
IBOT=IBOT+4
IST=NRST+1
IND=NRST+NCOL
READ(NT11) (RMESH(I),I=IST,IND)
IST=NZST+1
IND=NZST+NROW
READ(NT11) (ZMESH(I),I=IST,IND)
IST=NBNDST+1
IND=NBNDST+4*NCOL+4*NROW
READ(NT11) (PHI(I),I=IST,IND)
NSKIP=MINO(G-1,IABS(NBACK))
MSKIP=NSKIP+2
DO 1 ISKIP=1,MSKIP
1 READ(NT11)
IST=NDST+1
IND=NDST+2*NMAT
READ(NT11) (D(I),I=IST,IND)
DO 2 ISKIP=1,MSKIP
2 BACKSPACE NT11
102 IST=NTYST+1
IND=NTYST+NHALF(1)
READ(NT11) (ITYPE(I),I=IST,IND)
IE=1
IF (I1.LT.0) GO TO 18
103 IF(LHALF(2)-J1+1.GT.0.AND.IE.EQ.1) GO TO 200
IF(LHALF(2)-J1+1.LE.0.AND.IE.EQ.2) GO TO 210
3 IST=NTYST+1
IND=NTYST+NHALF(IE) +[(IE-1)*(NCOL-1)]
DO 4 J=IST,IND
4 ITYPE(J)=MOD(ITYPE(J),100)
ISTAR=0
17 NROWJ=NROW-J1
IF(J1.EQ.1.OR.J1.EQ.NROW) GO TO 30

```

```

IF(I1.EQ.1.OR.I1.EQ.NCOL) GC TO 50
NFLAP=NROWJ+NZST
NFLIP=NRST+I1
ON=ZMESH(NFLAP)-ZMESH(NFLAP+1)
OW=RMESSH(NFLIP)-RMESH(NFLIP-1)
OS=ZMESH(NFLAP+1)-ZMESH(NFLAP+2)
OE=RMESSH(NFLIP+1)-RMESH(NFLIP)
IPO=NROWJ*NCOL+I1
8 IPN=IPO-NCOL
IPS=IPO+NCOL
IPW=IPO-1
IPE=IPO+1
NOW=NTYST+(NROWJ-1)*(NCOL-1)+I1-(IE-1)*(NHALF(1)-NCOL+1)
NW=ITYPE(NOW-1)+NDST
NE=ITYPE(NOW)+NDST
NOW=NOW+NCOL-1
SW=ITYPE(NOW-1)+NDST
SE=ITYPE(NOW)+NDST
106 IF(ISTAR.EQ.1) GO TO 11
IF(J2-J1)11,6,11
6 CJZ= .5*((OE*D(SE)+OW*D(SW))*(PHI(IFS)-PHI(IPO))/OS -
1 (OE*D(NE)+OW*D(NW))*(PHI(IPN)-PHI(IPO))/ON)/(OE+OW)
CALL PUM(CJZ,DAWD(1),I1,J1,IPO,G)
C IF(I2-I1)LEFT,THERE,RIGHT
IF(I2-I1) 9, 13, 7
7 I1=I1+1
OW=OE
NFLIP=NRST+I1
OE=RMESSH(NFLIP+1)-RMESH(NFLIP)
IPO=IPO+1
GO TO 8
9 I1=I1-1
OE=OW
NFLIP=NRST+I1
OW=RMESSH(NFLIP)-RMESH(NFLIP-1)
IPO=IPO-1
GO TO 8
11 CJX= -.5*((ON*D(NE)+OS*D(SE))*(PHI(IPE)-PHI(IPO))/OE -
1 (ON*D(NW)+OS*D(SW))*(PHI(IPW)-PHI(IPO))/OW)/(OS+ON)
CALL PUM(CJX,DAWD(2),I1,J1,IPC,G)
ISTAR=1
IF(J2-J1)12,13,5
5 CALL ERROR (17HERROR. (CURE2,5)E' )
12 J1=J1-1
IF(LHALF(2)-J1+1.GT.0.AND.IE.EQ.1) GC TO 200
IF(LHALF(2)-J1+1.LE.0.AND.IE.EQ.2) GC TO 210
GO TO 17
13 ISTAR=0
IF(IBOT.GT.49) GO TO 26
I2=ISTORE(IBOT)
IF(I2)18,14,14
14 J2=ISTORE(IBOT+1)
IBOT=IBOT+2
GO TO 17
C
18 IBOT=IBOT+1
I1=ISTORE(IBOT)
IF(I1.LT.0.AND.IBOT.LT.50) GO TO 18
IBOT=IBOT+1
IF(IBOT.GT.48) GO TO 26
J1=ISTORE(IBOT)
I2=ISTORE(IBOT+1)
J2=ISTORE(IBOT+2 )

```

```

1 IBOT=IBOT+3
2 GO TO 103
3 30 IF(ISTAR.EQ.1) GO TO 31
4 IF(J1-J2) 31,35,31
5 31 NROWJ=NROW-J1
6 NFLAP=NROWJ+NZST
7 NFLIP=NRST+I1
8 OE=RMESSH(NFLIP+1)-RMESH(NFLIP)
9 OW=RMESSH(NFLIP)-RMESH(NFLIP-1)
10 IPO=NROWJ*NCOL+I1
11 IPE=IPO+1
12 IPW=IPO-1
13 NOW=NTYST+(NROWJ+IE-2)*(NCOL-1)+I1-(IE-1)*NHALF(1)
14 IF(J1-1)32,32,33
15 32 NW=NDST+ITYPE(NOW-1)
16 NE=NDST+ITYPE(NOW )
17 SW=NW
18 SE=NE
19 OS=0.
20 NZZ=NZST+NROW
21 ON=ZMESH(NZZ-1)-ZMESH(NZZ)
22 IPS=IPO
23 IPN=IPO-NCOL
24 GO TO 106
25 33 NOW=NOW+NCOL-1
26 SW=NDST+ITYPE(NOW-1)
27 SE=NDST+ITYPE(NOW)
28 NW=SW
29 NE=SE
30 ON=0.
31 OS=ZMESH(NZST+1)-ZMESH(NZST+2)
32 IPN=IPO
33 IPS=IPO+NCOL
34 GO TO 106
35 IPO=NROWJ*NCOL + I1
36 IF(J1-1)36,36,41
37 36 ISUF=NBNDST+4*NROW+I1
38 IF(PHI(ISUF).GE.1.E35) GO TC 37
39 JSUF=ISUF+NCOL
40 CJZ=-PHI(IPO)*PHI(ISUF)-PHI(JSUF)
41 CALL PUM(CJZ,DAWD(1),I1,J1,IPO,G)
42 IF(I1-I2) 38,13,39
43 I1=I1+1
44 GO TO 35
45 39 I1=I1-1
46 GO TO 35
47 37 IPN=IPO-NCOL
48 NOW=NTYST + (NROWJ+IE-2)*(NCOL-1) + I1 - (IE-1)*NHALF(1)
49 NW=NDST+ITYPE(NOW-1)
50 NE=NDST+ITYPE(NOW)
51 NFLIP=NRST+I1
52 NZZ=NZST+NROW
53 ON=ZMESH(NZZ-1)-ZMESH(NZZ)
54 OE=RMESSH(NFLIP+1)-RMESH(NFLIP)
55 OW=RMESSH(NFLIP)-RMESH(NFLIP-1)
56 CJZ=(PHI(IPO)-PHI(IPN))*(D(NW)*OW+D(NE)*OE) / (ON*(OE+OW))
57 GO TO 40
C
41 ISUF=NBNDST + 4*NROW + 2*NCCL + I1
42 IF(PHI(ISUF).GE.1.E35) GO TC 42
43 JSUF=ISUF+NCOL
44 CJZ=PHI(IPO)*PHI(ISUF) + PHI(JSUF)
45 GO TO 40

```

101

```

42 IPS=IPO+NCOL
   NOW=NTYST + (NROWJ+IE-1)*(NCOL-1) + I1 - (IE-1)*NHALF(1)
   SW=NDST+ITYPE(NOW-1)
   SE=NDST+ITYPE(NOW)
   OE=R MESH(NFLIP+1)-R MESH(NFLIP)
   OW=R MESH(NFLIP)-R MESH(NFLIP-1)
   CJZ=(PHI(IPS)-PHI(IPN))*(D(SW)*OW+D(SE)*OE) / (OS*(OE+OW))
   GO TO 40
50 IF(I1-I2) 51,55,51
51 NROWJ=NR OW-J1
   NFLAP=NR OWJ+NZ ST
   ON=Z MESH(NFLAP-1)-Z MESH(NFLAP)
   OS=Z MESH(NFLAP)-Z MESH(NFLAP+1)
   IPO=NROWJ*NCOL+I1
   IPN=IPO-NCOL
   IPS=IPO+NCOL
   NOW=NTYST+(NROWJ+IE-2)*(NCOL-1)+I1-(IE-1)*NHALF(1)
   IF(I1-1) 52,52,53
52 NE=NDST+ITYPE(NOW)
   NOW=NOW+NCOL-1
   SE=NDST+ITYPE(NOW)
   SW=SE
   NW=NE
   OW=0.
   OE=R MESH(NRST+2)-R MESH(NRST+1)
   IPW=IPO
   IPE=IPO+1
   GO TO 6
53 NW=NDST+ITYPE(NOW-1)
   NOW=NOW-2+NCOL
   SW=NDST+ITYPE(NOW)
   NE=NW
   SE=SW
   OE=0.
   NRR=NRST+NCOL
   OW=R MESH(NRR)-R MESH(NRR-1)
   IPE=IPO
   IPW=IPO-1
   GO TO 6
55 NROWJ=NR OW-J1
   IPO=NROWJ*NCOL+I1
   IF(I1-1) 56,56,61
56 ISUF=NBNDST+J1
   IS PHI GIVEN ON BX LCW
   IF(PHI(ISUF).GE.1.E35) GO TO 57
   JSUF=ISUF+NROW
   CJX=-PHI(IPO)*PHI(ISUF)-PHI(JSUF)
60 CALL PUM(CJX,DAWD(2),I1,J1,IPO,G)
   IF(J1-J2) 13,13,58
58 J1=J1-1
   IF(LHALF(2)-J1+1.LE.0.AND.IE.EQ.2) GC TO 210
   IF(LHALF(2)-J1+1.GT.0.AND.IE.EQ.1) GC TO 200
   GO TO 55
57 IPE=IPO+1
   NOW=NTYST+(NROWJ+IE-2)*(NCOL-1) - (IE-1)*NHALF(1) + I1
   NE=NDST+ITYPE(NOW)
   NOW=NOW+NCOL
   SE=NDST+ITYPE(NOW)
   OE=R MESH(NRST+2)-R MESH(NRST+1)

```

```

1   NFLAP=NZ ST+NROWJ
2   ON=Z MESH(NFLAP)-Z MESH(NFLAP+1)
3   OS=Z MESH(NFLAP+1)-Z MESH(NFLAP+2)
4   CJX=(PHI(IPE)-PHI(IPO))*(D(NW)*ON+D(SW)*OS) / (OW*(ON+OS))
5   GO TO 60
6
7   C 61 ISUF=NBNDST+2*NROW+J1
8     IF(PHI(ISUF).GE.1.E35) GO TO 62
9     JSUF=ISUF+NROW
10    CJX=PHI(IPO)*PHI(ISUF) + PHI(JSUF)
11    GO TO 60
12
13   C 62 IPW=IPO-1
14     NOW=NTYST+(NROWJ+IE-1)*(NCOL-1) - (IE-1)*NHALF(1)
15     NW=NDST+ITYPE(NOW-1)
16     NOW=NOW+NCOL-1
17     SW=NDST+ITYPE(NOW-1)
18     NFLAP=NZ ST+NROWJ
19     OS=Z MESH(NFLAP+1)-Z MESH(NFLAP+2)
20     ON=Z MESH(NFLAP)-Z MESH(NFLAP+1)
21     NRR=NRST+NCOL
22     OW=R MESH(NRR)-R MESH(NRR-1)
23     CJX=(PHI(IPW)-PHI(IPO))*(D(SW)*OS + D(NW)*ON) / (OW*(OS+ON))
24     GO TO 40
25
26   200 NCOL1=NCOL-1
27     DO 201 I=1,NCOL1
28       NTB=NTYST+NHALF(1)-NCOL1+I
29       NTL=NTYST+I
30       201 ITYPE(NTI)=ITYPE(NTB)
31         MSKIP=NSKIP+1
32         DO 202 I SKIP=1,MSKIP
33           202 READ(NT11)
34             IS2=NT2ST+1
35             IN2=NT2ST+NHALF(2)
36             READ(NT11) (ITYPE(IM2),IM2=IS2,IN2)
37             IE=2
38             GO TO 103
39       210 MSKIP=NSKIP+3
40         DO 211 I SKIP=1,MSKIP
41           211 BACKSPACE NT11
42           GO TO 102
43       26 IF(IE.EQ.1) MSKIP=2*NSKIP+3
44         IF(IE.EQ.2) MSKIP=NSKIP+1
45         DO 27 I SKIP=1,MSKIP
46           27 READ(NT11)
47           RETURN
48           END
49
50 -----
51
52
53   SUBROUTINE PUM(CJX,DAWD,I1,J1,IPO,IG)
54   COMMON /PHI/ PHI(17000)
55   8 CJPLUS=PHI(IPO)/4. + CJX/2.
56   CJMINS=PHI(IPO)/4. - CJX/2.
57   II=I1-1
58   JJ=J1-1
59   WRITE(6,10)II,JJ,PHI(IPO),DAWD ,CJX,CJPLUS,CJMINS
60   10 FORMAT (11X,'COL',I3,' RCW',I3,5X,'FLUX=',IPE11.4,3X,A4,
61     * E11.4,5X,'J+='',E11.4,' J-='',E11.4)
62   RETURN
63   END

```

BLOCK DATA
COMMON/GAUSS/ G1(2,8),G2(2,8),G3(2,8),G4(2,8),G5(2,8),G6(2,8),
1 G7(2,8),G8(2,8),G9(2,8),G10(2,8),G11(2,8),G12(2,8),G13(2,8),
2 G14(2,8),G15(2,8),G16(2,8)

DATA G1 /
1 16*0. /

DATA G2 /
A.577350269 , 1.0 ,
2 14*0. /

DATA G3 /
A.0 , 0.888888889 ,
A.774596669 , 0.555555556 ,
3 12*0. /

DATA G4 /
A.339981044 , 0.652145154 ,
A.861136312 , 0.347854845 ,
3 12*0. /

DATA G5 /
A.0 , 0.568888889 ,
A.538469310 , 0.478628670 ,
A.906179846 , 0.236926885 ,
4 10*0. /

DATA G6 /
A.238619186 , 0.467913935 ,
A.661209386 , 0.360761573 ,
A.932469514 , 0.171324492 ,
4 10*0. /

DATA G7 /
A.0 , 0.417959184 ,
A.405845151 , 0.381830051 ,
A.741531186 , 0.279705391 ,
A.949107912 , 0.129484966 ,
5 8*0. /

DATA G8 /
A.183434642 , 0.362683783 ,
A.525532410 , 0.313706646 ,
A.796666477 , 0.222381034 ,
A.960285856 , 0.101228536 ,
5 8*0. /

DATA G9 /
A.0 , 0.330239355 ,
A.324253423 , 0.312347077 ,
A.613371433 , 0.260610696 ,
A.836031107 , 0.180648161 ,
A.968160240 , 0.081274388 ,
6 6*0. /

DATA G10 /
A.148874339 , 0.295524225 ,
A.433395394 , 0.269266719 ,
A.679409568 , 0.219086363 ,
A.865063367 , 0.149451349 ,
A.973906529 , 0.066671344 ,
6 6*0. /

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

DATA G11 /

A.0 , 0.272925087 ,
A.269543156 , 0.262804545 ,
A.519096129 , 0.233193765 ,
A.730152006 , 0.186290211 ,
A.887062600 , 0.125580369 ,
A.978228658 , 0.055668567 ,
7 4*0. /

DATA G12 /

A.125233409 , 0.249147046 ,
A.367831499 , 0.233492537 ,
A.587317954 , 0.203167427 ,
A.769902674 , 0.160078329 ,
A.904117256 , 0.106939326 ,
A.981560634 , 0.047175336 ,
7 4*0. /

DATA G13 /

A.0 , 0.232551553 ,
A.230458316 , 0.226283180 ,
A.448492751 , 0.207816048 ,
A.642349339 , 0.178145981 ,
A.801578091 , 0.038873510 ,
A.517598359 , 0.092121500 ,
A.984183055 , 0.040484005 ,
8 2*0. /

DATA G14 /

A.108054949 , 0.215263853 ,
A.319112369 , 0.205198464 ,
A.515248636 , 0.185538397 ,
A.687292905 , 0.157203167 ,
A.827201315 , 0.121518571 ,
A.928434884 , 0.080158087 ,
A.586283809 , 0.035119460 ,
8 2*0. /

DATA G15 /

A.0 , 0.202578242 ,
A.201194094 , 0.198431485 ,
A.394151347 , 0.186161000 ,
A.570972173 , 0.166269206 ,
A.724417731 , 0.139570678 ,
A.848206583 , 0.107159220 ,
A.937273392 , 0.070366047 ,
A.987992518 , 0.030753242 /

DATA G16 /

A.095012510 , 0.189450610 ,
A.281603551 , 0.182603415 ,
A.458016778 , 0.169156519 ,
A.617876244 , 0.149595989 ,
A.755404408 , 0.124628971 ,
A.865631202 , 0.095158512 ,
A.944575023 , 0.062253524 ,
A.989400935 , 0.027152459 /

END

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

BLOCK DATA

COMMON/PHI/T1(97),T2(97),T3(97),T4(97),T5(97),T6(97),T7(97),T8(97)
 1 ,T9(97),T10(97),T11(97),T12(97),T13(97),T14(97),T15(97),
 2 T16(97),T17(97),T18(97),T19(97),T20(97),T21(97),T22(97)
 3 ,T23(97), T24(97),T25(97),T26(97),T27(97),T28(97)
 4 ,T29(97),T30(97),T31(97),T32(97),T33(97),T34(97),T35(97)
 5 ,T36(97),T37(97),T38(97),T39(97),T40(97),PHI(13120)

U235 REVISED RCH A

DATA T 1/
 X 4.250, 4.500, 4.650, 5.200, 7.900, 12.400, 15.100,
 X 21.100, 37.200, 68.000, 100.000, 94.000, 49.000, 90.000,
 X 234.000, 621.000, 3.050, 2.730, 2.350, 1.780, 1.740,
 X 3.200, 5.550, 11.150, 27.250, 58.050, 90.050, 84.050,
 X 39.050, 80.050, 224.040, 611.000, 0.270, 0.370, 0.650,
 X 0.440, 0.060, 0.240, 0.670, 0.450, 0.070, 0. ,
 X 0.550, 0.400, 0.070, 0. , 0. , 0.350, 0.080,
 X 0. , 0. , 0. , 0.080, 0. , 0. , 0. ,
 X 0. , 0.050, 0. , 0. , 0. , 0. , 0.050,
 X 0. , 0. , 0. , 0. , 0.050, 0. , 0. ,
 X 0.050, 0. , 0. , 0. , 0. , 0.050, 0. ,
 X 0. , 0. , 0. , 0.050, 0. , 0. , 0. ,
 X 0.050, 0. , 0. , 0.050, 0. , 0.040/

U238 REVISED RCH A

DATA T 2/
 X 4.070, 4.400, 4.500, 5.250, 8.200, 12.000, 14.000,
 X 15.000, 22.000, 59.000, 65.000, 119.000, 9.400, 9.550,
 X 10.000, 11.440, 2.816, 2.575, 1.594, 0.720, 0.240,
 X 0.550, 0.760, 2.060, 11.050, 50.060, 56.060, 110.060,
 X 0.460, 0.610, 1.050, 2.440, 0.330, 0.460, 0.790,
 X 0.530, 0.070, 0.350, 0.960, 0.640, 0.090, 0. ,
 X 0.800, 0.550, 0.100, 0. , 0. , 0.500, 0.080,
 X 0. , 0. , 0. , 0.080, 0. , 0. , 0. ,
 X 0. , 0.100, 0. , 0. , 0. , 0. , 0.060,
 X 0. , 0. , 0. , 0. , 0.060, 0. , 0. ,
 X 0.060, 0. , 0. , 0. , 0. , 0.060, 0. ,
 X 0. , 0. , 0. , 0.060, 0. , 0. , 0. ,
 X 0.060, 0. , 0. , 0.060, 0. , 0.050/

PU239 REVISED RCH A

DATA T 3/
 X 4.250, 4.500, 4.800, 5.700, 8.400, 13.200, 13.500,
 X 16.700, 35.100, 93.000, 143.000, 53.000, 33.000, 234.000,
 X 2023.000, 990.000, 3.200, 3.080, 2.710, 2.150, 1.890,
 X 2.350, 3.540, 6.740, 25.140, 83.060, 133.070, 43.060,
 X 23.070, 224.080, 2013.060, 980.000, 0.200, 0.270, 0.450,
 X 0.310, 0.040, 0.180, 0.500, 0.350, 0.050, 0. ,
 X 0.450, 0.300, 0.060, 0. , 0. , 0.290, 0.050,
 X 0. , 0. , 0. , 0.050, 0. , 0. , 0. ,
 X 0. , 0.050, 0. , 0. , 0. , 0. , 0.040,
 X 0. , 0. , 0. , 0. , 0.040, 0. , 0. ,
 X 0. , 0. , 0.040, 0. , 0. , 0. , 0. ,
 X 0.060, 0. , 0. , 0. , 0. , 0.070, 0. ,
 X 0. , 0. , 0. , 0.060, 0. , 0. , 0. ,
 X 0.070, 0. , 0. , 0.080, 0. , 0.060/

PU240 REVISED RCH A

DATA T 4/
 X 4.250, 4.500, 4.650, 5.450, 8.200, 11.450, 11.700,
 X 13.000, 22.000, 71.000, 43.000, 39.000, 7149.400, 936.200,

X 151.000, 263.030, 3.220, 2.880, 2.410, 1.380, 0.330, 1
 X 0.500, 0.750, 2.050, 11.050, 60.080, 31.090, 24.100, 2
 X 6645.260, 898.550, 140.070, 252.030, 0.240, 0.330, 0.550, 3
 X 0.400, 0.050, 0.220, 0.580, 0.400, 0.060, 0. , 4
 X 0.500, 0.350, 0.060, 0. , 0. , 0.400, 0.060, 5
 X 0. , 0. , 0. , 0.050, 0. , 0. , 0. , 6
 X 0. , 0.050, 0. , 0. , 0. , 0. , 0.050, 7
 X 0. , 0. , 0. , 0. , 0.050, 0. , 0. , 8
 X 0. , 0. , 0.050, 0. , 0. , 0. , 0. , 9
 X 0.080, 0. , 0. , 0. , 0. , 0.090, 0. , 10
 X 0. , 0. , 0. , 0.100, 0. , 0. , 0. , 11
 X 3.860, 0. , 0. , 0.350, 0. , 0.070/

C REVISED RCH

DATA T 5/
 X 1.230, 1.420, 2.260, 2.930, 3.590, 4.250, 4.440, 16
 X 4.340, 4.340, 4.340, 4.340, 4.440, 4.440, 4.440, 17
 X 4.440, 4.440, 0.515, 0.314, 0.856, 0.604, 0.433, 18
 X 0.401, 0.428, 0.428, 0.428, 0.603, 0.662, 0.616, 19
 X 0.677, 0.813, 0.537, 0.003, 0.515, 0. , 0. , 20
 X 0. , 0. , 0.314, 0. , 0. , 0. , 0. , 21
 X 0.856, 0. , 0. , 0. , 0. , 0.604, 0. , 22
 X 0. , 0. , 0. , 0.433, 0. , 0. , 0. , 23
 X 0. , 0.401, 0. , 0. , 0. , 0. , 0.428, 24
 X 0. , 0. , 0. , 0. , 0.428, 0. , 0. , 25
 X 0. , 0. , 0.428, 0. , 0. , 0. , 0. , 26
 X 0.603, 0. , 0. , 0. , 0. , 0.662, 0. , 27
 X 0. , 0. , 0. , 0.616, 0. , 0. , 0. , 28
 X 0.677, 0. , 0. , 0.813, 0. , 0.536/

O REVISED RCH

DATA T 6/
 X 1.330, 1.180, 3.230, 3.630, 3.710, 3.260, 3.550, 33
 X 3.640, 3.640, 3.640, 3.640, 3.640, 3.640, 3.640, 34
 X 3.640, 3.641, 0.464, 0.191, 0.902, 0.556, 0.337, 35
 X 0.231, 0.255, 0.270, 0.270, 0.380, 0.414, 0.380, 36
 X 0.414, 0.498, 0.331, 0.000, 0.424, 0. , 0. , 37
 X 0. , 0. , 0.191, 0. , 0. , 0. , 0. , 38
 X 0.902, 0. , 0. , 0. , 0. , 0.556, 0. , 39
 X 0. , 0. , 0. , 0.337, 0. , 0. , 0. , 40
 X 0. , 0.231, 0. , 0. , 0. , 0. , 0.255, 41
 X 0. , 0. , 0. , 0. , 0.270, 0. , 0. , 42
 X 0. , 0. , 0.270, 0. , 0. , 0. , 0. , 43
 X 0.380, 0. , 0. , 0. , 0. , 0.414, 0. , 44
 X 0. , 0. , 0. , 0.380, 0. , 0. , 0. , 45
 X 0.414, 0. , 0. , 0.498, 0. , 0.331/

NA REVISED RCH

DATA T 7/
 X 1.607, 2.260, 2.990, 3.119, 3.107, 3.884, 6.089, 50
 X 4.545, 3.010, 3.010, 3.030, 3.069, 3.099, 3.210, 51
 X 3.350, 3.747, 0.470, 0.556, 1.191, 0.649, 0.196, 52
 X 0.189, 0.302, 0.235, 0.160, 0.227, 0.254, 0.251, 53
 X 0.295, 0.394, 0.388, 0.447, 0.440, 0. , 0. , 54
 X 0. , 0. , 0.556, 0. , 0. , 0. , 0. , 55
 X 1.191, 0. , 0. , 0. , 0. , 0.604, 0.036, 56
 X 0.008, 0. , 0. , 0.195, 0. , 0. , 0. , 57
 X 0. , 0.188, 0. , 0. , 0. , 0. , 0.301, 58
 X 0. , 0. , 0. , 0. , 0.234, 0. , 0. , 59
 X 0. , 0. , 0.155, 0. , 0. , 0. , 0. , 60
 X 0.216, 0. , 0. , 0. , 0. , 0.236, 0. , 61
 X 0. , 0. , 0. , 0.219, 0. , 0. , 0. , 62
 X 0.238, 0. , 0. , 0.294, 0. , 0.198/

FE REVISED RCH							
DATA T 8/							
X	2.247,	2.200,	1.960,	2.290,	2.440,	2.230,	5.650,
X	7.114,	10.968,	11.364,	11.364,	11.370,	11.370,	11.370,
X	11.570,	13.040,	0.807,	0.705,	0.490,	0.133,	0.081,
X	0.060,	0.130,	0.162,	0.260,	0.387,	0.463,	0.499,
X	0.668,	0.939,	1.180,	2.240,	0.500,	0.200,	0.100,
X	0. ,	0. ,	0.400,	0.200,	0.100,	0. ,	0. ,
X	0.230,	0.230,	0.020,	0. ,	0. ,	0.123,	0. ,
X	0. ,	0. ,	0. ,	0.071,	0. ,	0. ,	0. ,
X	0. ,	0.050,	0. ,	0. ,	0. ,	0. ,	0.120,
X	0. ,	0. ,	0. ,	0. ,	0.151,	0. ,	0. ,
X	0. ,	0. ,	0.233,	0. ,	0. ,	0. ,	0. ,
X	0.332,	0. ,	0. ,	0. ,	0. ,	0.365,	0. ,
X	0. ,	0. ,	0. ,	0.329,	0. ,	0. ,	0. ,
X	0.358,	0. ,	0. ,	0.429,	0. ,	0.270/	
810 REVISED RCH							
DATA T 9/							
X	1.563,	1.920,	2.280,	2.640,	4.320,	5.700,	10.520,
X	31.470,	72.470,	142.500,	205.500,	313.500,	537.500,	873.500,
X	1573.500,	3833.500,	0.753,	0.610,	1.060,	0.950,	1.810,
X	2.590,	7.420,	28.380,	69.380,	139.560,	202.610,	310.560,
X	534.610,	870.730,	570.490,	830.530,	0.670,	0. ,	0. ,
X	0. ,	0. ,	0.420,	0. ,	0. ,	0. ,	0. ,
X	0.880,	0. ,	0. ,	0. ,	0. ,	0.500,	0. ,
X	0. ,	0. ,	0. ,	0.380,	0. ,	0. ,	0. ,
X	0. ,	0.360,	0. ,	0. ,	0. ,	0. ,	0.370,
X	0. ,	0. ,	0. ,	0. ,	0.380,	0. ,	0. ,
X	0. ,	0. ,	0.380,	0. ,	0. ,	0. ,	0. ,
X	0.530,	0. ,	0. ,	0. ,	0. ,	0.580,	0. ,
X	0. ,	0. ,	0. ,	0.530,	0. ,	0. ,	0. ,
X	0.580,	0. ,	0. ,	0.700,	0. ,	0.460/	
TH232 REVISED RCH							
DATA T10/							
X	4.000,	4.220,	4.320,	5.040,	7.870,	13.480,	14.500,
X	14.500,	30.000,	30.000,	52.700,	12.500,	12.900,	13.500,
X	14.700,	19.600,	2.360,	2.190,	1.529,	0.710,	0.290,
X	0.544,	0.670,	1.964,	18.060,	18.085,	40.794,	0.585,
X	0.994,	1.613,	2.774,	7.600,	0.330,	0.460,	0.790,
X	0.530,	0.070,	0.350,	0.960,	0.640,	0.090,	0. ,
X	0.800,	0.550,	0.100,	0. ,	0. ,	0.500,	0.080,
X	0. ,	0. ,	0. ,	0.080,	0. ,	0. ,	0. ,
X	0. ,	0.064,	0. ,	0. ,	0. ,	0. ,	0.070,
X	0. ,	0. ,	0. ,	0. ,	0.064,	0. ,	0. ,
X	0. ,	0. ,	0.060,	0. ,	0. ,	0. ,	0. ,
X	0.085,	0. ,	0. ,	0. ,	0. ,	0.094,	0. ,
X	0. ,	0. ,	0. ,	0.085,	0. ,	0. ,	0. ,
X	0.094,	0. ,	0. ,	0.113,	0. ,	0.074/	
U238 REVISED RCH OXC							
DATA T11/							
X	4.000,	4.400,	4.500,	5.250,	8.200,	12.000,	14.000,
X	14.150,	13.070,	15.250,	12.570,	15.380,	9.400,	9.550,
X	10.000,	11.440,	2.816,	2.575,	1.594,	0.720,	0.240,
X	0.550,	0.760,	1.210,	2.120,	6.310,	3.630,	6.440,
X	0.460,	0.610,	1.050,	2.440,	0.330,	0.460,	0.790,
X	0.530,	0.070,	0.350,	0.960,	0.640,	0.090,	0. ,
X	0.800,	0.550,	0.100,	0. ,	0. ,	0.500,	0.080,
X	0. ,	0. ,	0. ,	0.080,	0. ,	0. ,	0. ,
X	0. ,	0.100,	0. ,	0. ,	0. ,	0. ,	0.060,

X	0. ,	0. ,	0. ,	0. ,	0.060,	0. ,	0. ,	1
X	0. ,	0. ,	0.050,	0. ,	0. ,	0. ,	0. ,	2
X	0.060,	0. ,	0. ,	0. ,	0. ,	0.060,	0. ,	3
X	0. ,	0. ,	0. ,	0.060,	0. ,	0. ,	0. ,	4
X	0.060,	0. ,	0. ,	0.060,	0. ,	0.050/		5
U238 REVISED RCH B								
DATA T12/								
X	4.070,	4.400,	4.500,	5.250,	8.200,	12.000,	14.000,	9
X	14.000,	12.500,	14.500,	11.800,	14.000,	9.400,	9.550,	10
X	10.000,	11.440,	2.816,	2.575,	1.594,	0.720,	0.240,	11
X	0.550,	0.760,	1.060,	1.550,	5.560,	2.860,	5.060,	12
X	0.460,	0.610,	1.050,	2.440,	0.330,	0.460,	0.790,	13
X	0.530,	0.070,	0.350,	0.960,	0.640,	0.090,	0. ,	14
X	0.800,	0.550,	0.100,	0. ,	0. ,	0.500,	0.080,	15
X	0. ,	0. ,	0. ,	0.080,	0. ,	0. ,	0. ,	16
X	0. ,	0.100,	0. ,	0. ,	0. ,	0. ,	0.060,	17
X	0. ,	0. ,	0. ,	0. ,	0.060,	0. ,	0. ,	18
X	0. ,	0. ,	0.050,	0. ,	0. ,	0. ,	0. ,	19
X	0.060,	0. ,	0. ,	0. ,	0. ,	0.060,	0. ,	20
X	0. ,	0. ,	0. ,	0.060,	0. ,	0. ,	0. ,	21
X	0.060,	0. ,	0. ,	0.060,	0. ,	0.050/		22
PU239 REVISED RCH OXC								
DATA T13/								
X	4.250,	4.500,	4.800,	5.700,	8.400,	13.200,	13.500,	26
X	16.700,	35.000,	42.500,	71.200,	30.700,	33.000,	183.200,	27
X	1259.000,	990.000,	3.200,	3.080,	2.710,	2.150,	1.890,	28
X	2.350,	3.540,	6.740,	25.040,	32.560,	71.270,	20.760,	29
X	23.070,	173.280,	1249.060,	980.000,	0.200,	0.270,	0.450,	30
X	0.310,	0.040,	0.180,	0.500,	0.350,	0.050,	0. ,	31
X	0.450,	0.300,	0.060,	0. ,	0. ,	0.290,	0.050,	32
X	0. ,	0. ,	0. ,	0.050,	0. ,	0. ,	0. ,	33
X	0. ,	0.050,	0. ,	0. ,	0. ,	0. ,	0.040,	34
X	0. ,	0. ,	0. ,	0. ,	0.040,	0. ,	0. ,	35
X	0. ,	0. ,	0.040,	0. ,	0. ,	0. ,	0. ,	36
X	0.060,	0. ,	0. ,	0. ,	0. ,	0.070,	0. ,	37
X	0. ,	0. ,	0. ,	0.060,	0. ,	0. ,	0. ,	38
X	0.070,	0. ,	0. ,	0.080,	0. ,	0.060/		39
PU240 REVISED RCH OXC								
DATA T14/								
X	4.250,	4.500,	4.650,	5.450,	8.200,	11.450,	11.700,	43
X	13.000,	22.000,	36.600,	32.100,	39.000,	1220.000,	496.000,	44
X	151.000,	263.030,	3.220,	2.880,	2.410,	1.380,	0.330,	45
X	0.500,	0.750,	2.050,	11.050,	25.680,	20.190,	24.100,	46
X	717.260,	458.550,	140.070,	252.030,	0.240,	0.330,	0.550,	47
X	0.400,	0.050,	0.220,	0.580,	0.400,	0.060,	0. ,	48
X	0.500,	0.350,	0.060,	0. ,	0. ,	0.400,	0.060,	49
X	0. ,	0. ,	0. ,	0.050,	0. ,	0. ,	0. ,	50
X	0. ,	0.050,	0. ,	0. ,	0. ,	0. ,	0.050,	51
X	0. ,	0. ,	0. ,	0. ,	0.050,	0. ,	0. ,	52
X	0. ,	0. ,	0.050,	0. ,	0. ,	0. ,	0. ,	53
X	0.080,	0. ,	0. ,	0. ,	0. ,	0.090,	0. ,	54
X	0. ,	0. ,	0. ,	0.100,	0. ,	0. ,	0. ,	55
X	3.860,	0. ,	0. ,	0.350,	0. ,	0.070/		56
ZR REVISED RCH								
DATA T15/								
X	2.600,	3.100,	3.840,	6.210,	7.210,	7.910,	7.644,	60
X	6.552,	6.353,	6.155,	6.155,	6.155,	6.155,	6.155,	61
X	6.155,	6.156,	1.103,	0.705,	0.495,	0.232,	0.150,	62
X	0.122,	0.111,	0.099,	0.096,	0.127,	0.137,	0.126,	63

101

X	0.144,	0.182,	0.161,	0.159,	0.600,	0.300,	0.200,	1
X	0.	0.	0.400,	0.200,	0.100,	0.	0.	2
X	0.300,	0.150,	0.040,	0.	0.	0.220,	0.	3
X	0.	0.	0.	0.135,	0.	0.	0.	4
X	0.	0.107,	0.	0.	0.	0.	0.096,	5
X	0.	0.	0.	0.	0.084,	0.	0.	6
X	0.	0.	0.081,	0.	0.	0.	0.	7
X	0.112,	0.	0.	0.	0.	0.122,	0.	8
X	0.	0.	0.	0.111,	0.	0.	0.	9
X	0.122,	0.	0.	0.146,	0.	0.096/		10

DATA T16/

X	2.400,	2.900,	3.900,	6.200,	8.200,	7.900,	7.100,	14
X	8.050,	9.160,	19.360,	5.560,	5.660,	5.960,	6.460,	15
X	6.660,	7.560,	1.010,	0.770,	0.440,	0.270,	0.220,	16
X	0.270,	0.480,	1.580,	2.980,	13.500,	0.150,	0.180,	17
X	0.260,	0.390,	0.990,	0.500,	0.500,	0.300,	0.200,	18
X	0.	0.	0.400,	0.250,	0.100,	0.	0.	19
X	0.300,	0.100,	0.	0.	0.	0.200,	0.	20
X	0.	0.	0.	0.130,	0.	0.	0.	21
X	0.	0.100,	0.	0.	0.	0.	0.080,	22
X	0.	0.	0.	0.	0.080,	0.	0.	23
X	0.	0.	0.080,	0.	0.	0.	0.	24
X	0.100,	0.	0.	0.	0.	0.100,	0.	25
X	0.	0.	0.	0.100,	0.	0.	0.	26
X	0.110,	0.	0.	0.140,	0.	0.090/		27

K REVISED RCH

DATA T17/

X	2.110,	2.220,	1.690,	1.840,	1.970,	1.867,	1.425,	31
X	1.523,	1.700,	1.868,	1.966,	2.110,	2.260,	2.360,	32
X	2.660,	4.000,	0.315,	0.149,	0.195,	0.116,	0.072,	33
X	0.054,	0.045,	0.053,	0.067,	0.012,	0.157,	0.205,	34
X	0.306,	0.472,	0.685,	1.750,	0.315,	0.	0.	35
X	0.	0.	0.149,	0.	0.	0.	0.	36
X	0.195,	0.	0.	0.	0.	0.116,	0.	37
X	0.	0.	0.	0.072,	0.	0.	0.	38
X	0.	0.053,	0.	0.	0.	0.	0.042,	39
X	0.	0.	0.	0.	0.045,	0.	0.	40
X	0.	0.	0.050,	0.	0.	0.	0.	41
X	0.078,	0.	0.	0.	0.	0.089,	0.	42
X	0.	0.	0.	0.085,	0.	0.	0.	43
X	0.096,	0.	0.	0.112,	0.	0.075/		44

NI REVISED RCH

DATA T18/

X	2.290,	2.340,	2.530,	2.450,	3.380,	5.170,	14.580,	48
X	15.680,	16.680,	16.730,	17.300,	17.310,	17.510,	18.200,	49
X	19.000,	21.210,	1.070,	0.700,	0.310,	0.110,	0.090,	50
X	0.120,	0.670,	0.360,	0.380,	0.570,	0.710,	0.790,	51
X	1.090,	1.580,	2.030,	4.600,	0.500,	0.200,	0.100,	52
X	0.	0.	0.300,	0.200,	0.100,	0.	0.	53
X	0.250,	0.050,	0.	0.	0.	0.100,	0.	54
X	0.	0.	0.	0.080,	0.	0.	0.	55
X	0.	0.100,	0.	0.	0.	0.	0.290,	56
X	0.	0.	0.	0.	0.320,	0.	0.	57
X	0.	0.	0.330,	0.	0.	0.	0.	58
X	0.470,	0.	0.	0.	0.	0.530,	0.	59
X	0.	0.	0.	0.480,	0.	0.	0.	60
X	0.530,	0.	0.	0.640,	0.	0.430/		61

DATA T19/

X	1.520,	1.790,	2.140,	2.270,	3.160,	4.080,	4.970,	3
X	6.870,	11.470,	19.870,	32.470,	55.470,	95.470,	154.470,	4
X	276.470,	672.770,	0.710,	0.480,	0.920,	0.580,	0.650,	5
X	0.970,	1.870,	3.780,	8.380,	16.930,	29.580,	52.530,	6
X	92.580,	151.700,	273.460,	669.300,	0.670,	0.	0.	7
X	0.	0.	0.420,	0.	0.	0.	0.	8
X	0.880,	0.	0.	0.	0.	0.500,	0.	9
X	0.	0.	0.	0.380,	0.	0.	0.	10
X	0.	0.360,	0.	0.	0.	0.	0.370,	11
X	0.	0.	0.	0.	0.380,	0.	0.	12
X	0.	0.	0.380,	0.	0.	0.	0.	13
X	0.530,	0.	0.	0.	0.	0.580,	0.	14
X	0.	0.	0.	0.530,	0.	0.	0.	15
X	0.580,	0.	0.	0.700,	0.	0.460/		16

AL REVISED RCH

DATA T20/

X	1.806,	2.022,	2.140,	2.721,	2.832,	1.435,	1.463,	20
X	1.365,	1.366,	1.366,	1.366,	1.385,	1.410,	1.460,	21
X	1.460,	1.560,	0.706,	0.360,	0.530,	0.251,	0.142,	22
X	0.075,	0.065,	0.061,	0.063,	0.090,	0.102,	0.101,	23
X	0.124,	0.165,	0.154,	0.200,	0.560,	0.100,	0.030,	24
X	0.	0.	0.230,	0.110,	0.020,	0.001,	0.	25
X	0.380,	0.140,	0.010,	0.	0.	0.250,	0.	26
X	0.	0.	0.	0.140,	0.	0.	0.	27
X	0.	0.070,	0.	0.	0.	0.	0.063,	28
X	0.	0.	0.	0.	0.060,	0.	0.	29
X	0.	0.	0.060,	0.	0.	0.	0.	30
X	0.084,	0.	0.	0.	0.	0.092,	0.	31
X	0.	0.	0.	0.084,	0.	0.	0.	32
X	0.094,	0.	0.	0.115,	0.	0.074/		33

U233 REVISED RCH

DATA T21/

X	4.250,	4.500,	4.600,	5.300,	8.100,	12.500,	16.500,	37
X	23.000,	44.000,	63.000,	148.000,	133.000,	422.000,	160.000,	38
X	235.000,	525.000,	3.060,	2.970,	2.780,	2.390,	2.440,	39
X	3.960,	6.890,	12.060,	32.060,	51.090,	136.090,	121.090,	40
X	410.090,	148.110,	223.080,	513.000,	0.200,	0.270,	0.450,	41
X	0.310,	0.040,	0.180,	0.500,	0.350,	0.050,	0.	42
X	0.450,	0.300,	0.060,	0.	0.	0.290,	0.050,	43
X	0.	0.	0.	0.050,	0.	0.	0.	44
X	0.	0.040,	0.	0.	0.	0.	0.050,	45
X	0.	0.	0.	0.	0.060,	0.	0.	46
X	0.	0.	0.060,	0.	0.	0.	0.	47
X	0.090,	0.	0.	0.	0.	0.090,	0.	48
X	0.	0.	0.	0.090,	0.	0.	0.	49
X	0.090,	0.	0.	0.110,	0.	0.080/		50

U238 REVISED RCH

DATA T22/

X	4.070,	4.400,	4.500,	5.250,	8.200,	12.000,	14.000,	54
X	14.500,	14.000,	17.000,	14.000,	18.000,	9.400,	9.550,	55
X	10.000,	11.440,	2.816,	2.575,	1.594,	0.720,	0.240,	56
X	0.550,	0.760,	1.560,	3.050,	8.060,	5.060,	9.060,	57
X	0.460,	0.610,	1.050,	2.440,	0.330,	0.460,	0.790,	58
X	0.530,	0.070,	0.350,	0.960,	0.640,	0.090,	0.	59
X	0.800,	0.550,	0.100,	0.	0.	0.500,	0.080,	60
X	0.	0.	0.	0.080,	0.	0.	0.	61
X	0.	0.100,	0.	0.	0.	0.	0.060,	62
X	0.	0.	0.	0.	0.060,	0.	0.	63

X	0.	0.	0.050	0.	0.	0.	0.	1
X	0.060	0.	0.	0.	0.	0.060	0.	2
X	0.	0.	0.	0.060	0.	0.	0.	3
X	0.060	0.	0.	0.060	0.	0.050/		4

PU239 REVISED RCH C

DATA T23/								7
X	4.250,	4.500,	4.800,	5.700,	8.400,	13.200,	13.500,	8
X	16.700,	34.500,	46.500,	88.000,	32.700,	33.000,	196.000,	9
X	1390.000,	990.000,	3.200,	3.080,	2.710,	2.150,	1.890,	10
X	2.350,	3.540,	6.740,	24.540,	36.560,	78.070,	22.760,	11
X	23.070,	186.080,	1380.060,	980.000,	0.200,	0.270,	0.450,	12
X	0.310,	0.040,	0.180,	0.500,	0.350,	0.050,	0.	13
X	0.450,	0.300,	0.060,	0.	0.	0.290,	0.050,	14
X	0.	0.	0.	0.050,	0.	0.	0.	15
X	0.	0.050,	0.	0.	0.	0.	0.040,	16
X	0.	0.	0.	0.	0.040,	0.	0.	17
X	0.	0.	0.040,	0.	0.	0.	0.	18
X	0.060,	0.	0.	0.	0.	0.070,	0.	19
X	0.	0.	0.	0.060,	0.	0.	0.	20
X	0.070,	0.	0.	0.080,	0.	0.060/		21

PU240 REVISED RCHC

DATA T24/								23
X	4.250,	4.500,	4.650,	5.450,	8.200,	11.450,	11.700,	25
X	13.000,	22.000,	38.000,	32.500,	39.000,	1309.400,	518.200,	26
X	151.000,	263.030,	3.220,	2.880,	2.410,	1.380,	0.330,	27
X	0.500,	0.750,	2.050,	11.050,	27.080,	20.590,	24.100,	28
X	805.260,	480.550,	140.070,	252.030,	0.240,	0.330,	0.550,	29
X	0.400,	0.050,	0.220,	0.580,	0.400,	0.060,	0.	30
X	0.500,	0.350,	0.060,	0.	0.	0.400,	0.060,	31
X	0.	0.	0.	0.050,	0.	0.	0.	32
X	0.	0.050,	0.	0.	0.	0.	0.050,	33
X	0.	0.	0.	0.	0.050,	0.	0.	34
X	0.	0.	0.050,	0.	0.	0.	0.	35
X	0.080,	0.	0.	0.	0.	0.090,	0.	36
X	0.	0.	0.	0.100,	0.	0.	0.	37
X	3.860,	0.	0.	0.350,	0.	0.070/		38

LI6 REVISED RCH

DATA T25/								41
X	1.030,	1.000,	1.120,	1.700,	4.470,	1.650,	2.200,	42
X	5.100,	10.500,	20.700,	35.400,	60.900,	109.900,	171.900,	43
X	309.900,	837.900,	0.596,	0.562,	0.850,	0.998,	2.561,	44
X	1.002,	1.555,	4.458,	9.858,	20.123,	35.845,	60.323,	45
X	109.345,	171.393,	309.294,	837.100,	0.486,	0.	0.	46
X	0.	0.	0.298,	0.034,	0.	0.	0.	47
X	0.590,	0.	0.	0.	0.	0.498,	0.	48
X	0.	0.	0.	0.611,	0.	0.	0.	49
X	0.	0.152,	0.	0.	0.	0.	0.155,	50
X	0.	0.	0.	0.	0.158,	0.	0.	51
X	0.	0.	0.158,	0.	0.	0.	0.	52
X	0.223,	0.	0.	0.	0.	0.245,	0.	53
X	0.	0.	0.	0.223,	0.	0.	0.	54
X	0.245,	0.	0.	0.293,	0.	0.194/		55

LI7 REVISED RCH

DATA T26/								58
X	1.100,	1.140,	1.240,	1.110,	2.990,	0.940,	0.970,	59
X	0.970,	0.970,	0.968,	0.968,	0.968,	0.969,	0.969,	60
X	0.969,	0.967,	0.670,	0.470,	0.830,	0.418,	0.620,	61
X	0.150,	0.160,	0.160,	0.160,	0.232,	0.254,	0.233,	62
X	0.257,	0.308,	0.211,	0.029,	0.670,	0.	0.	63

X	0.	0.	0.450,	0.020,	0.	0.	0.	1
X	0.830,	0.	0.	0.	0.	0.416,	0.002,	2
X	0.	0.	0.	0.620,	0.	0.	0.	3
X	0.	0.150,	0.	0.	0.	0.	0.160,	4
X	0.	0.	0.	0.	0.160,	0.	0.	5
X	0.	0.	0.160,	0.	0.	0.	0.	6
X	0.231,	0.	0.	0.	0.	0.253,	0.	7
X	0.	0.	0.	0.231,	0.	0.	0.	8
X	0.253,	0.	0.	0.302,	0.	0.199/		9

BE REVISED RCH

DATA T27/								12
X	1.291,	1.475,	2.380,	3.310,	3.940,	5.180,	5.280,	13
X	5.370,	5.420,	5.460,	5.460,	5.460,	5.460,	5.460,	14
X	5.460,	5.569,	0.859,	0.541,	1.207,	0.913,	0.634,	15
X	0.655,	0.680,	0.710,	0.710,	1.010,	1.110,	1.010,	16
X	1.111,	1.332,	0.884,	0.009,	0.818,	0.350,	0.	17
X	0.	0.	0.509,	0.120,	0.	0.	0.	18
X	1.207,	0.	0.	0.	0.	0.913,	0.	19
X	0.	0.	0.	0.634,	0.	0.	0.	20
X	0.	0.655,	0.	0.	0.	0.	0.680,	21
X	0.	0.	0.	0.	0.710,	0.	0.	22
X	0.	0.	0.710,	0.	0.	0.	0.	23
X	1.010,	0.	0.	0.	0.	1.110,	0.	24
X	0.	0.	0.	1.010,	0.	0.	0.	25
X	1.110,	0.	0.	1.330,	0.	0.980/		26

F REVISED RCH

DATA T28/								29
X	1.400,	1.900,	2.090,	3.200,	4.610,	3.780,	3.570,	30
X	3.570,	3.470,	3.280,	2.990,	3.180,	3.281,	3.282,	31
X	3.474,	3.468,	0.520,	0.620,	0.680,	1.060,	0.657,	32
X	0.250,	0.210,	0.220,	0.220,	0.290,	0.280,	0.280,	33
X	0.311,	0.372,	0.264,	0.008,	0.420,	0.	0.	34
X	0.	0.	0.320,	0.210,	0.090,	0.	0.	35
X	0.680,	0.	0.	0.	0.	1.060,	0.	36
X	0.	0.	0.	0.650,	0.007,	0.	0.	37
X	0.	0.250,	0.	0.	0.	0.	0.210,	38
X	0.	0.	0.	0.	0.220,	0.	0.	39
X	0.	0.	0.220,	0.	0.	0.	0.	40
X	0.290,	0.	0.	0.	0.	0.280,	0.	41
X	0.	0.	0.	0.280,	0.	0.	0.	42
X	0.310,	0.	0.	0.370,	0.	0.260/		43

CL REVISED RCH

DATA T29/								44
X	1.650,	1.660,	1.520,	1.660,	2.020,	3.400,	1.770,	47
X	1.912,	4.280,	9.830,	13.540,	17.110,	19.220,	22.470,	48
X	27.800,	45.700,	0.233,	0.103,	0.152,	0.074,	0.028,	49
X	0.109,	0.057,	0.066,	0.233,	0.833,	1.625,	2.804,	50
X	4.611,	7.738,	12.748,	30.000,	0.173,	0.	0.	51
X	0.	0.	0.103,	0.	0.	0.	0.	52
X	0.152,	0.	0.	0.	0.	0.074,	0.	53
X	0.	0.	0.	0.028,	0.	0.	0.	54
X	0.	0.109,	0.	0.	0.	0.	0.057,	55
X	0.	0.	0.	0.	0.064,	0.	0.	56
X	0.	0.	0.183,	0.	0.	0.	0.	57
X	0.443,	0.	0.	0.	0.	0.645,	0.	58
X	0.	0.	0.	0.704,	0.	0.	0.	59
X	0.791,	0.	0.	0.968,	0.	0.648/		60

NB REVISED RCH

DATA T30/								61
								62
								63

X	2.400,	2.900,	3.900,	6.200,	8.200,	8.000,	7.200,	1
X	8.600,	8.800,	6.400,	6.000,	6.000,	6.100,	6.100,	2
X	6.400,	7.000,	1.010,	0.770,	0.440,	0.250,	0.210,	3
X	0.500,	1.280,	2.680,	0.500,	0.500,	0.120,	0.140,	4
X	0.170,	0.240,	0.460,	1.020,	0.500,	0.300,	0.200,	5
X	0.	0.	0.400,	0.250,	0.100,	0.	0.	6
X	0.300,	0.100,	0.	0.	0.	0.200,	0.	7
X	0.	0.	0.	0.130,	0.	0.	0.	8
X	0.	0.100,	0.	0.	0.	0.	0.080,	9
X	0.	0.	0.	0.	0.080,	0.	0.	10
X	0.	0.	0.080,	0.	0.	0.	0.	11
X	0.100,	0.	0.	0.	0.	0.100,	0.	12
X	0.	0.	0.	0.100,	0.	0.	0.	13
X	0.110,	0.	0.	0.140,	0.	0.090/		14
								15
								16
								17
DATA T31 /								18
X	0.600,	0.970,	1.330,	1.830,	3.000,	5.200,	6.330,	18
X	6.666,	6.669,	6.672,	6.676,	6.817,	6.830,	6.847,	19
X	6.920,	7.090,	1.440,	1.931,	3.236,	3.768,	4.868,	20
X	7.296,	9.021,	9.635,	9.611,	11.652,	12.125,	11.864,	21
X	12.380,	13.386,	11.150,	0.290,	0.769,	0.239,	0.239,	22
X	0.144,	0.049,	0.690,	0.690,	0.415,	0.116,	0.020,	23
X	1.796,	1.076,	0.300,	0.052,	0.012,	2.827,	0.781,	24
X	0.132,	0.022,	0.005,	4.041,	0.684,	0.117,	0.022,	25
X	0.004,	6.022,	1.045,	0.187,	0.030,	0.012,	7.372,	26
X	1.349,	0.209,	0.061,	0.030,	7.880,	1.220,	0.360,	27
X	0.122,	0.052,	6.719,	1.920,	0.680,	0.192,	0.096,	28
X	7.757,	2.719,	0.780,	0.232,	0.156,	8.494,	2.418,	29
X	0.719,	0.360,	0.120,	7.906,	2.364,	1.182,	0.387,	30
X	7.409,	3.684,	1.242,	9.982,	3.334,	11.020/		31
								32
								33
								34
DATA T32 /								35
X	0.610,	0.950,	1.320,	1.820,	2.930,	5.200,	6.330,	35
X	6.666,	6.669,	6.672,	6.676,	6.817,	6.830,	6.847,	36
X	6.920,	7.090,	1.386,	2.047,	3.181,	3.624,	4.080,	37
X	7.296,	9.021,	9.635,	9.611,	11.652,	12.125,	11.864,	38
X	12.380,	13.386,	11.150,	0.290,	0.739,	0.231,	0.231,	39
X	0.139,	0.046,	0.731,	0.731,	0.439,	0.121,	0.025,	40
X	1.767,	1.060,	0.294,	0.049,	0.011,	2.718,	0.752,	41
X	0.127,	0.022,	0.005,	3.387,	0.571,	0.100,	0.018,	42
X	0.004,	6.022,	1.045,	0.187,	0.030,	0.012,	7.372,	43
X	1.349,	0.209,	0.061,	0.030,	7.880,	1.220,	0.360,	44
X	0.122,	0.052,	6.719,	1.920,	0.680,	0.192,	0.096,	45
X	7.757,	2.719,	0.780,	0.232,	0.156,	8.494,	2.418,	46
X	0.719,	0.360,	0.120,	7.906,	2.364,	1.182,	0.387,	47
X	7.409,	3.684,	1.242,	9.982,	3.334,	11.020/		48
								49
								50
								51
DATA T33 /								52
X	1.077,	1.623,	2.033,	2.473,	2.382,	2.270,	2.270,	52
X	2.270,	2.270,	2.270,	2.270,	2.270,	2.270,	2.267,	53
X	2.267,	2.267,	0.984,	1.503,	2.106,	2.154,	1.687,	54
X	1.367,	1.390,	1.417,	1.411,	1.798,	1.897,	1.798,	55
X	1.897,	2.081,	1.646,	0.000,	0.540,	0.200,	0.270,	56
X	0.011,	0.	0.535,	0.633,	0.335,	0.	0.	57
X	1.142,	0.964,	0.	0.	0.	1.903,	0.251,	58
X	0.	0.	0.	1.660,	0.027,	0.	0.	59
X	0.	1.336,	0.031,	0.	0.	0.	1.353,	60
X	0.037,	0.	0.	0.	1.380,	0.037,	0.	61
X	0.	0.	1.234,	0.177,	0.	0.	0.	62
X	1.482,	0.316,	0.	0.	0.	1.625,	0.272,	63

X	0.	0.	0.	1.482,	0.309,	0.007,	0.	1
X	1.625,	0.272,	0.	1.877,	0.204,	1.646/		2
								3
								4
DATA T34 /								5
X	1.077,	1.623,	2.033,	2.321,	2.405,	2.270,	2.270,	6
X	2.270,	2.270,	2.270,	2.270,	2.270,	2.270,	2.267,	7
X	2.267,	2.267,	0.984,	1.503,	2.106,	2.038,	1.348,	8
X	1.367,	1.390,	1.417,	1.411,	1.798,	1.897,	1.798,	9
X	1.897,	2.081,	1.646,	0.000,	0.540,	0.200,	0.270,	10
X	0.011,	0.	0.535,	0.633,	0.335,	0.	0.	11
X	1.142,	0.964,	0.	0.	0.	1.826,	0.212,	12
X	0.	0.	0.	1.336,	0.012,	0.	0.	13
X	0.	1.336,	0.031,	0.	0.	0.	1.353,	14
X	0.037,	0.	0.	0.	1.380,	0.037,	0.	15
X	0.	0.	1.234,	0.177,	0.	0.	0.	16
X	1.482,	0.316,	0.	0.	0.	1.625,	0.272,	17
X	0.	0.	0.	1.482,	0.309,	0.007,	0.	18
X	1.625,	0.272,	0.	1.877,	0.204,	1.646/		19
								20
								21
DATA T35 /								22
X	0.570,	1.480,	1.690,	2.170,	3.294,	5.436,	7.359,	23
X	8.092,	8.493,	9.330,	9.590,	9.640,	9.740,	9.880,	24
X	10.160,	11.190,	0.580,	0.470,	0.490,	0.390,	0.172,	25
X	0.442,	0.614,	0.688,	0.729,	1.150,	1.310,	1.260,	26
X	1.460,	1.850,	1.630,	1.670,	0.330,	0.	0.	27
X	0.	0.	0.360,	0.	0.	0.	0.	28
X	0.450,	0.	0.	0.	0.	0.350,	0.	29
X	0.	0.	0.	0.170,	0.	0.	0.	30
X	0.	0.440,	0.	0.	0.	0.	0.610,	31
X	0.	0.	0.	0.	0.680,	0.	0.	32
X	0.	0.	0.710,	0.	0.	0.	0.	33
X	1.110,	0.	0.	0.	0.	1.240,	0.	34
X	0.	0.	0.	1.140,	0.	0.	0.	35
X	1.240,	0.	0.	1.490,	0.	0.990/		36
								37
								38
DATA T36 /								39
X	4.250,	4.500,	4.650,	5.200,	7.900,	12.400,	15.100,	40
X	21.100,	37.200,	68.000,	100.000,	94.000,	49.000,	90.000,	41
X	234.000,	621.000,	3.050,	2.730,	2.350,	1.780,	1.740,	42
X	3.200,	5.550,	11.150,	27.250,	58.050,	90.050,	84.050,	43
X	39.050,	80.050,	224.040,	611.000,	0.270,	0.370,	0.650,	44
X	0.440,	0.060,	0.240,	0.670,	0.450,	0.070,	0.	45
X	0.550,	0.400,	0.070,	0.	0.	0.350,	0.080,	46
X	0.	0.	0.	0.080,	0.	0.	0.	47
X	0.	0.050,	0.	0.	0.	0.	0.050,	48
X	0.	0.	0.	0.	0.050,	0.	0.	49
X	0.050,	0.	0.	0.	0.	0.050,	0.	50
X	0.	0.	0.	0.050,	0.	0.	0.	51
X	0.	0.	0.	0.050,	0.	0.	0.	52
X	0.050,	0.	0.	0.050,	0.	0.040/		53
								54
								55
DATA T37 /								56
X	4.070,	4.400,	4.500,	5.250,	8.200,	12.000,	14.000,	57
X	15.000,	22.000,	59.000,	65.000,	119.000,	9.400,	9.550,	58
X	10.000,	11.440,	2.816,	2.575,	1.594,	0.720,	0.240,	59
X	0.550,	0.760,	2.060,	11.050,	50.060,	56.060,	110.060,	60
X	0.460,	0.610,	1.050,	2.440,	0.330,	0.460,	0.790,	61
X	0.530,	0.070,	0.350,	0.960,	0.640,	0.090,	0.	62
X	0.800,	0.550,	0.100,	0.	0.	0.500,	0.080,	63

```

X 0. , 0. , 0. , 0.080, 0. , 0. , 0. , 1
X 0. , 0.100, 0. , 0. , 0. , 0. , 0.060, 2
X 0. , 0. , 0. , 0. , 0.060, 0. , 0. , 3
X 0. , 0. , 0.050, 0. , 0. , 0. , 0. , 4
X 0.060, 0. , 0. , 0. , 0. , 0.060, 0. , 5
X 0. , 0. , 0. , 0.060, 0. , 0. , 0. , 6
X 0.060, 0. , 0. , 0.060, 0. , 0.050/ 7

```

PU239 REVISED RCH A 9

```

DATA T38/
X 4.250, 4.500, 4.800, 5.700, 8.400, 13.200, 13.500, 11
X 16.700, 35.100, 93.000, 143.000, 53.000, 33.000, 234.000, 12
X 2023.000, 990.000, 3.200, 3.080, 2.710, 2.150, 1.890, 13
X 2.350, 3.540, 6.740, 25.140, 83.060, 133.070, 43.060, 14
X 23.070, 224.080, 2013.060, 980.000, 0.200, 0.270, 0.450, 15
X 0.310, 0.040, 0.180, 0.500, 0.350, 0.050, 0. , 16
X 0.450, 0.300, 0.060, 0. , 0. , 0.290, 0.050, 17
X 0. , 0. , 0. , 0.050, 0. , 0. , 0. , 18
X 0. , 0.050, 0. , 0. , 0. , 0. , 0.040, 19
X 0. , 0. , 0. , 0. , 0.040, 0. , 0. , 20
X 0. , 0. , 0.040, 0. , 0. , 0. , 0. , 21
X 0.060, 0. , 0. , 0. , 0. , 0.070, 0. , 22
X 0. , 0. , 0. , 0.060, 0. , 0. , 0. , 23
X 0.070, 0. , 0. , 0.080, 0. , 0.060/ 24

```

PU240 REVISED RCH A 25

```

DATA T39/
X 4.250, 4.500, 4.650, 5.450, 8.200, 11.450, 11.700, 28
X 13.000, 22.000, 71.000, 43.000, 39.000, 7149.400, 936.200, 29
X 151.000, 263.030, 3.220, 2.880, 2.410, 1.380, 0.330, 30
X 0.500, 0.750, 2.050, 11.050, 60.080, 31.090, 24.100, 31
X 6645.260, 898.550, 140.070, 252.030, 0.240, 0.330, 0.550, 32
X 0.400, 0.050, 0.220, 0.580, 0.400, 0.060, 0. , 33
X 0.500, 0.350, 0.060, 0. , 0. , 0.400, 0.060, 34
X 0. , 0. , 0. , 0.050, 0. , 0. , 0. , 35
X 0. , 0.050, 0. , 0. , 0. , 0. , 0.050, 36
X 0. , 0. , 0. , 0. , 0.050, 0. , 0. , 37
X 0. , 0. , 0.050, 0. , 0. , 0. , 0. , 38
X 0.080, 0. , 0. , 0. , 0. , 0.090, 0. , 39
X 0. , 0. , 0. , 0.100, 0. , 0. , 0. , 40
X 3.860, 0. , 0. , 0.350, 0. , 0.070/ 41

```

END

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43

```

*****
*
* A N H A N G I *
*
*****

```

FORTRAN-LISTEN DES PROGRAMMSYSTEMS ATTOW-K
(NEU ERSTELLTE PROGRAMME)

----- INHALTSVERZEICHNIS -----

	SEITE
*** MAINPROGRAM ***	
MAINPROGRAM NUSQU	I2
*** SUBROUTINES ***	
SUBROUTINE ISOKFK	I5
SUBROUTINE KFKINR	I7
SUBROUTINE KFKNUS	I6
SUBROUTINE REMCGU	I4
*** FUNCTIONS ***	
FUNCTION VAL	I6
*** BLOCK DATAS ***	
BLGCK DATA COMMON /RECRSE/	I10

```

PROGRAM: NUSQU TO MAKE CROSS-SECTIONS CALCULATED BY NUSYS
AVAILABLE FOR ATTOh
A G.BOENISCH
D *** NOVEMBER 1972 ***

DIMENSION S(5),T(50000),NT(50000),SM(132)
EQUIVALENCE (T(1),NT(1))
DATA DIFKO /'DIFK'/,SREM /'SREM'/,SMTOT /'SMTOT'/
3 READ(5,1C40) KCND,NSCAT,NTP1
1040 FORMAT (4I10)
REWIND NTP1
IF(KOND.EQ.0) GO TO 50
READ(NTP1)
READ(NTP1)
50 READ(NTP1) L1,S1,S2,L2
READ(NTP1) L3,IGRZA,MIZA,(T(I),I=4,L2)
REWIND NTP1
WRITE (NTP1) IGRZA,MIZA,NSCAT,NTP1
KZ2=NT(4)+1
KZ3=(L2-3)/KZ2
DO 80 N1=1,MIZA

===== DIFKO FOR EACH MIXTURE OF MATERIALS =====

KZ1=5
DO 1 I1=1,KZ3
IF (T(KZ1).NE.DIFKO) GO TO 1
IF (NT(KZ1+5).EQ.N1) GO TO 2
1 KZ1=KZ1+KZ2
WRITE (6,1010) N1
1010 FORMAT ('0',10X,'***** DATA BLOCK DIFKO FOR MIXTURE',I3,' NOT FOUN
1D ON CROSS-SECTION TAPE *****')
GO TO 5
2 K1=KZ1+6
DO 100 L=1,IGRZA
SM(L)=T(K1)
100 K1=K1+1
WRITE (NTP1) (SM(I),I=1,IGRZA)

===== SREM FOR EACH MIXTURE OF MATERIALS =====

9 KZ1=5
DO 4 I2=1,KZ3
IF (T(KZ1).NE.SREM) GO TO 4
IF (NT(KZ1+5).EQ.N1) GO TO 5
4 KZ1=KZ1+KZ2
WRITE (6,1015) N1
1015 FORMAT('0',10X,'***** DATA BLOCK SREM FOR MIXTURE',I3,' NOT FOUND
1CN CROSS-SECTION TAPE *****')
GO TO 10
5 J1=KZ1+6
DO 102 L=1,IGRZA
SM(L)=T(J1)
102 J1=J1+1
WRITE (NTP1) (SM(I),I=1,IGRZA)

===== SMTOT-MATRIX FOR EACH MIXTURE OF MATERIALS =====

10 NNN=-1
IZ=C
KTMAX=IGRZA-1
LIMAX=KTMAX

```

```

1 IF (NSCAT.GT.KTMAX) NSCAT=KTMAX
2 KT=NSCAT
3 KZ1=5
4 DC 7 I3=1,KZ3
5 IF (T(KZ1).NE.SMTOT) GO TO 7
6 IF (NT(KZ1+2).EQ.2.AND.NT(KZ1+5).EQ.N1) GO TO 8
7 KZ1=KZ1+KZ2
8 WRITE (6,1030) N1
9 1030 FORMAT('0',10X,'***** DATA BLOCK SMTOT FOR MIXTURE',I3,' NOT FOUND
10 1D ON CROSS-SECTION TAPE *****')
11 GO TO 99
12 8 M2=KZ1+6
13 DO 70 JA=1,LIMAX
14 NNN=NNN+1
15 M1=M2+NNN*MIZA*KZ2+NNN
16 DO 60 JB=1,KT
17 IZ=IZ+1
18 SM(IZ)=T(M1)
19 60 M1=M1+MIZA*KZ2
20 KTMAX=KTMAX-1
21 IF (KT.LE.KTMAX) GO TO 70
22 KT=KT-1
23 70 CONTINUE
24 INZ=IZ+1
25 DO 13 I=INZ,132
26 13 SM(I)=S9S9.
27 WRITE (NTP1) (SM(I),I=1,130)
28 80 CONTINUE
29 =====
30 REWIND NTP1
31 99 STOP
32 END
33
-110-

```

```

SUBROUTINE REMCQU
COMMON /NAME1/ ERG(19),RAD,CCRST,CCRTP,SHEST,SHEND,SHEBOT,RM(21),
1 SHETOP,ZM(21),RC(20),ZC(20),THETA(20),NG,KR,KZ,KT,
2 JR,JZ,MR,MZ,MTYPE(400)
COMMON /NAME2/ SIGREM(25,18),PHIR(50,2),PHIZ(50,2),ANU,FS(18),NMAT
1 ,IHPR,IHPZ
COMMON /RECRSE/ REMQU(750)
EQUIVALENCE (WD,IWD)

NMAT=0
IF (NG.NE.18) GO TO 100
DO 2 I=1,2F
DO 2 K=1,18
2 SIGREM(I,K)=0.

5 IWD=ITDEC(4)
CALL INDEX(J1,IWD,4HM*I8)
GO TO (10,12,110), J1
10 IMAT=INDEC(0)
IF (IMAT.LE.0) GO TO 99
NMAT=MAXC(IMAT,NMAT)
GO TO 5
12 IINT=INDEC(0)
TEIDI=FLDEC(0)
IZ=(IINT-1)*18
DO 14 I=1,NG
14 SIGREM(IMAT,I) = SIGREM(IMAT,I)+TEIDI*REMCQU(IZ+I)
GO TO 5

99 WRITE (6,1060)
1060 FORMAT ('0'/5X,'===== CALCULATED REMVAL CRCS-SECTIONS =====')
DO 20 I=1,25
IF (SIGREM(I,1).LE.0.) GO TO 20
IF (I.GT.9) GO TO 18
WRITE (6,1000) I,(SIGREM(I,K),K=1,NG)
1000 FORMAT ('0',5X,'M',I1,2X,9(1PE11.3)/10X,9(1PE11.3))
GO TO 20
18 WRITE (6,1010) I,(SIGREM(I,K),K=1,NG)
1010 FORMAT ('0',5X,'M',I2,1X,9(1PE11.3)/10X,9(1PE11.3))
20 CONTINUE

WRITE (6,1080)
1080 FORMAT ('0'/)
RETURN
100 WRITE (6,1020)
1020 FORMAT ('0'/40X,'***** ERROR MESSAGE *****/20X,'SUBROUTINE REMO
1QU NEEDS INPUT-CARD: ENERGY BANDS 18(1)0 ')
GO TO 999
110 IF (NMAT.GT.0) GO TO 130
WRITE (6,1040)
1040 FORMAT ('0'/40X,'***** ERROR MESSAGE *****/20X,'SUBROUTINE REMO
1QU EXPECTED M OR I AND DOES NOT FUNCTION ONE OF THEM ')
GO TO 999
130 WRITE (6,1100)
1100 FORMAT ('0'/40X,'***** ERROR MESSAGE *****/20X,'END THE INPUT F
1OR CALCULATING THE REMOVAL CRCS-SECTIONS WITH: MO ')
999 STOP
END

```

```

1 SLBROUTINE ISCKFK (ID)
2 COMMON/DALINK/IAST,IDST,ISCA,IMAT,ACC,PCC,NGLIB,NSCAT,NISO,IB,
3 1 MATSTT,NEWGP(20),S(16,4) ,MTAPE
4 COMMON/PHI/PHI(1700)
5 COMMON /NUM/ NZ,NR,NROW,NCCL,NUMIJ,NTYPE,NG,NGAUSS,NSLOR,NMAT
6 COMMON /NUM/ NBACK
7 COMMON ACCUR(4),WD,CMEGA,DIFF,PRCDIF ,LHALF(2),MHALF(2),NHALF(2),
8 1 G,NIT,IDIST
9 DIMENSION D(1),A(1),SCA(1)
10 INTEGER ADD
11 INTEGER G
12 EQUIVALENCE (PHI(1),A(1),D(1),SCA(1))
13 EQUIVALENCE (WD,IWD)
14
15 IMIXT=0
16 IMAT=-5
17 GO TO 15
18 10 IMAT=-1
19 15 A=VAL(2)
20 IF (ID-1) 20,20,30
21 20 CALL KFKINR (NSCAT,NG,IMAT)
22 GO TO 18
23 30 CALL KFKNUS (NSCAT,NG,IMAT)
24
25 18 IMIXT=IMIXT+1
26 ISUF=IMAT+IDST
27 DO 1 IG=1,NG
28 D(ISUF)=VAL(1)
29 1 ISUF=ISUF+NMAT
30 ISUF=IMAT+IAST
31 DO 2 IG=1,NG
32 A(ISUF)=VAL(1)
33 2 ISUF=ISUF+NMAT
34 ISUF=IMAT+ISCA
35 IF (NG.EQ.1) RETURN
36 NGL=NG-1
37 DO 3 G=1,NGL
38
39 SCATTER IS OUT OF GROUP G
40
41 ISUF=IMAT+ISCA
42 ITOP=MINC(NG,NBACK+G)
43 IBOT=G+1
44 IF (G.LE.NBACK) ISUF=ISUF+NMAT*((G*(G+1))/2 -1)
45 IF (G.GT.NBACK) ISUF=ISUF+NMAT*(NBACK*(G-NBACK)+ (NBACK*(NBACK+1) )/
46 1 2-1)
47 ADD=MINO(G,NBACK-1)
48 DO 4 IG=IBOT,ITOP
49 SCA(ISUF)=VAL(1)
50 ISUF=ISUF+ADD*NMAT
51 IF (ADD.LT.NBACK-1) ADD=ADD+1
52 4 CONTINUE
53 3 CONTINUE
54
55 IF (IMIXT.LT.NMAT) GO TO 10
56 RETURN
57 END
58

```

111

```

FUNCTION VAL(ID)
COMMON /CRSECT/ CRSEC(182)

GO TO (20,10), ID
10 J=0
VAL=0.
RETURN
20 J=J+1
VAL=CRSEC(J)
RETURN
END

```

```

SUBROUTINE KFKNUS (NSCAT,NG,IMAT)
COMMON /CRSECT/ CRSEC(182)

IF (IMAT.GT.-4) GO TO 20
MAT=0
READ (15) NNG,MIZA,NSCATT,NTP1
IF (NNG.EQ.NG) GO TO 1
WRITE (6,1000) NNG,NG
1000 FORMAT ('0',5X,'***** ERROR-MESSAGE *****/5X,
1'THE NUMBER OF ENERGY-GROUPS WAS DEFINED IN THE NUSYS-STEP: NG=',
2I3,' AND IN THE ATTOw-STEP: NG=',I3/' THEY MUST BE IDENTIC
3AL')
GO TO 595

1 IF (NSCATT.EQ.NSCAT) GO TO 5
IF (NSCATT.LT.6.AND.NSCAT.LT.6) GO TO 5
WRITE (6,1010) NSCATT,NSCAT
1010 FORMAT ('0',5X,'***** ERROR-MESSAGE *****/5X,
1'THE NUMBER OF SCATTERING GROUPS WAS DEFINED IN THE NUSYS-STEP: N
2SCAT=',I3,' AND IN THE ATTOw-STEP: NSCAT=',I3/5X,'THEY MUST BE
3IDENTICAL AND NCT GREATER THAN 5')
GO TO 595

5 WRITE (6,1050)
1050 FORMAT ('0',5X,'***** CALCULATED CROSS-SECTIONS FROM NUSYS ****
1**')
20 MAT=MAT+1
IF (MAT.GT.MIZA) GO TO 30
IMAT=MAT
IF (IMAT.GT.9) GO TO 22
WRITE (6,2000) IMAT
2000 FORMAT ('0',8X,'FCR MIXTURE',I2)
GO TO 24
22 WRITE (6,2010) IMAT
2010 FORMAT ('0',8X,'FOR MIXTURE',I3)
24 N1=1
N2=NG
READ (NTP1) (CRSEC(I),I=N1,N2)
WRITE (6,2020) (CRSEC(I),I=N1,N2)
2020 FORMAT (E8,1P1CE12.4)
N1=N1+NG
N2=N2+NG
READ (NTP1) (CRSEC(I),I=N1,N2)
WRITE (6,2020) (CRSEC(I),I=N1,N2)
N1=N2+1
N2=N1+125
READ (NTP1) (CRSEC(I),I=N1,N2)

```

```

1 NZ=-1
2 DO 26 I=N1,N2
3 IF (CRSEC(I).EQ.5999.) GO TO 28
4 26 NZ=NZ+1
5 28 N2=N1+NZ
6 WRITE (6,2020) (CRSEC(I),I=N1,N2)
7 IF (MAT.EQ.MIZA) WRITE (6,2030)
8 2030 FORMAT ('0'//)
9 GO TO 59
10 30 WRITE (6,1020)
11 1020 FORMAT ('0',5X,'***** ERROR-MESSAGE *****/5X,
1'NMAT IN THE ATTOw-STEP IS GREATER THAN THE NUMBER OF MIXTURES FOR
1 WICH CROSS-SECTIONS WERE CALCULATED BY NUSYS')
14 GO TO 595
15 59 RETURN
16 595 STOP
17 END

```

```

SUBROUTINE KFKINR(NSCAT,NG,IMATER)
COMMON /CRSECT/ CRSEC(182)
DIMENSION D(26,3C),SR(26),ST(130)
DIMENSION S(5,26),NGD(26),IINT(30),TEIDI(30),KWDNEW(26),SRNEW(26)
DIMENSION SIGMA(26,26),STNEW(26,26)
EQUIVALENCE (WD,IWD)
REAL KWDNEW

IMAT=IMATER
DO 8 I=1,26
KWDNEW(I)=0.
SRNEW(I)=0.
DO 7 M=1,26
SIGMA(I,M)=0.
STNEW(I,M)=0.
7 CONTINUE
8 CCNTINUE
IF (IMAT.GT.-4) GO TO 18
IF (NSCAT.NE.5) GO TO 170
NGD(1)=0
SETS=C.
NGDI=0
IMAT=-1

===== EINLESEN DER EINGABE-WERTE EINER MISCHUNG =====
2 IWD=ITDEC(4)
CALL INDEX (J1,WD,9HI*#*CG*SE)
GO TO (100,5,15,10,19),J1
5 IF (NG.LT.26.AND.SETS.LT.1.) GO TO 110
IF (NG.LT.26.AND.NGD(1).LE.0) GO TO 140
SETS=2.
DO 6 I=1,26
NGD(I)=1
6 S(1,I)=1.
GO TO 20
10 IS=INDEC(0)
SETS=1.
DO 12 I=1,26
12 S(IS,I)=FLDEC(0)
IS=-1

```

```

GO TO 2
15 IF (SETS.LT.1.) GO TO 110
DO 17 I=1,NG
NGD(I)=INDEC(0)
17 NGDI=NGDI+NGD(I)
IF (NGDI.NE.26) GO TO 120
18 IWD=ITDEC(4)
19 CALL INDEX (J1,W0,9HI**CG*SE)
GO TO (3C,20,130,35,130),J1
20 IF (IS.LT.0.AND.IMAT.NE.-1) GO TC 150
IMAT=INDEC(0)
IMATER=IMAT
NII=0
NI=C
GO TO 18
30 NII=NII+1
IINT(NII)=INDEC(0)
TEIDI(NII)=FLDEC(0)
GO TO 18
35 IS=INDEC(0)
IF (IS.NE.0.AND.NG.EQ.26) GO TC 160
IF (IS.LE.0) IS=1

===== BERECHNUNG DER WIRKUNGSQUERSCHNITTE =====

IE = 0
DO 39 I = 1,NG
IA = IE+1
IE = IE+NGD(I)
DD = 0
DO 37 K = IA,IE
37 DD = DD+S(IS,K)
DO 38 K = IA,IE
38 S(IS,K) = S(IS,K)/DD
39 CONTINUE

40 NI=NI+1
IE=0
IREAD=IINT(NI)-1
REWIND 15
IF (IREAD.EQ.0) GO TO 43
DO 42 I=1,IREAD
42 READ(15)
43 READ(15) (D(K,NI),K = 1,26),(SR(L),L = 1,26),(ST(N),N = 1,125)
DO 54 I = 1,NG
IA = IE+1
IE = IE+NGD(I)
DD = 0
DO 52 K = IA,IE
ME = IA-1
MA = K
DO 50 M = 1,NG
ME = ME+NGD(M)
46 MA = MA+1
IF(MA.GT.ME) GOTO 48
IF(MA.GT.K+5) GOTO 46
J = (K-1)*5+MA-K
SIGMA(K,M) = SIGMA(K,M)+ST(J)
GOTO 46
48 STNEW(I,M) = STNEW(I,M)+TEIDI(NI)*SIGMA(K,M)*S(IS,K)
50 MA = MA-1
52 DD = DD+S(IS,K)*SR(K)
54 SRNEW(I) = SRNEW(I)+DD*TEIDI(NI)

```

```

1
2 DO 56 I = 1,26
3 DO 56 K = 1,26
4 56 SIGMA(I,K) = 0
5 IF (NI.LT.NII) GO TC 40
6
7 IE = 0
8 DO 59 I = 1,NG
9 DI = 0
10 IA = IE+1
11 IE = IE+NGD(I)
12 DO 58 K = IA,IE
13 DD = 0
14 DO 57 NI = 1,NII
15 57 DD = DD+TEIDI(NI)/D(K,NI)
16 58 DI = DI+S(IS,K)/DD
17 59 KWDNEW(I) = DI
18
19 J=0
20 DO 60 I=1,NG
21 J=J+1
22 60 CRSEC(J) = KWDNEW(I)
23 DO 62 I=1,NG
24 J=J+1
25 SRNEW(I) = SRNEW(I)-STNEW(I,I)
26 62 CRSEC(J) = SRNEW(I)
27 IE=NG-1
28 DO 64 I=1,IE
29 NA=I+1
30 NE=I+5
31 IF (NE.GT.NG) NE=NG
32 DO 64 N=NA,NE
33 J=J+1
34 64 CRSEC(J)=STNEW(I,N)
35 IF (IMAT.GT.9) GO TO 210
36
37 WRITE (6,1000) IMAT
38 1000 FORMAT ('0',9X,'***** CALCULATED CRCS-SECTIONS FOR MIXTURE',I2,
39 1' *****')
40 GO TO 22C
41 210 WRITE (6,1010) IMAT
42 1010 FORMAT ('0',9X,'***** CALCULATED CRCS-SECTIONS FOR MIXTURE',I3,
43 1' *****')
44
45 220 I=2*NG+1
46 WRITE (6,1020) (CRSEC(M),M=1,NG)
47 WRITE (6,1020) (SRNEW(M),M=1,NG)
48 WRITE (6,1020) (CRSEC(M),M=I,J)
49 1020 FORMAT (8X,1P1CE12.4)
50 WRITE (6,1030)
51 1030 FORMAT ('C'/)
52
53 IS=-1
54 IMAT=-1
55 GO TO 995
56 100 WRITE (6,2000)
57 2000 FORMAT ('0',5X,'***** ERROR-MESSAGE *****'/5X,
58 1'DO NOT BEGIN INPUT FOR MIXTURES WITH I BUT WITH S OR M')
59 GO TO 99
60 110 WRITE (6,2010)
61 2010 FORMAT ('C',5X,'***** ERROR-MESSAGE *****'/5X,
62 1'DEFINE VALUES FOR WEIGHT-FACTORS')
63 GO TO 99

```

```

120 WRITE (6,2020) NGDI
2020 FORMAT ('0',5X,'***** ERRCR=MESSAGE *****'/5X,
1'SUM FOR COMBINED ENERGY=GRUPS=',I3,' .IT MUST BE 26')
GO TO 99
130 WRITE (6,2030) IWD
2030 FORMAT ('0',5X,'***** ERROR=MESSAGE *****'/5X,
1'PROGRAM EXPECTED I OR M AND FCUND: ',A4/5X,
2'MAYBE THAT NMAT IS NOT IDENTICAL WITH THE DEFINED NUMBER OF MIXTU
3RES')
GO TO 99
140 WRITE (6,2040)
2040 FORMAT ('0',5X,'***** ERRCR=MESSAGE *****'/5X,
1'DEFINE VALUES FOR COMBINED GRUPS')
GO TO 99
150 WRITE (6,2050)
2050 FORMAT ('0',5X,'***** ERROR=MESSAGE *****'/5X,
1'THE INPLT FOR EVERY MIXTURE MUST END WITH A S-VALUE')
GO TO 99
160 WRITE (6,2060)
2060 FORMAT ('0',5X,'***** ERRCR=MESSAGE *****'/5X,
1'NO COMBINING CF ENERGY=GROUPS, THEREFCR END THE INPUT OF EACH MIX
2TURE WITH S0')
GOTO 99
170 WRITE (6,2070) NSCAT
2070 FORMAT ('0',5X,'***** ERROR=MESSAGE *****'/5X,
1'DO NOT DEFINE NSCAT=',I2,'BUT NSCAT=5 FOR USING KFKINR CROSS-SECT
2ICN DATAS')
GO TO 99
999 RETURN
99 STOP
END

```

BLOCK DATA

```

REMOVAL-CROSS-SECTIONS FOR 18 ENERGY=GRUPS (1MEV TO 18MEV)
WITH AN ENERGYINTERVALL OF 1 MEV
(CORRECTED VALUES FROM THE DATA-LIBRARY FOR SABINE,MARCH 1966)

```

```

COMMON/RECRSE/ Q1(18),Q2(18),Q3(18),Q4(18),Q5(18),Q6(18),Q7(18),
1 Q8(18),Q9(18),Q10(18),Q11(18),Q12(18),Q13(18),
2 Q14(18),Q15(18),Q16(18),Q17(18),Q18(18),Q19(18),
3 Q20(18),Q21(18),Q22(18),Q23(18),Q24(18),Q25(18),
4 Q26(18),Q27(18),Q28(18),Q29(18),Q30(18),Q31(18),
5 Q32(18),Q33(18),Q34(18),Q35(18),Q36(18),Q37(18),
6 REMQU(84)

```

*** H ***

```

DATA Q 1 /
X 0.525CE 00, 0.6217E 00, 0.5551E 00, 0.7389E 00, 0.6414E 00,
X 0.8014E 00, 0.8159E 00, 0.8986E 00, 0.9762E 00, 0.1071E 01,
X 0.1115E 01, 0.1295E 01, 0.1452E 01, 0.1527E 01, 0.1687E 01,
X 0.2298E 01, 0.3150E 01, 0.2436E 01 /

```

*** C ***

```

DATA Q 2 /
X 0.1050E 01, 0.1277E 01, 0.1049E 01, 0.1261E 01, 0.1026E 01,
X 0.1202E 01, 0.1152E 01, 0.1133E 01, 0.1074E 01, 0.1168E 01,
X 0.1071E 01, 0.7859E 00, 0.6935E 00, 0.9089E 00, 0.1991E 01,
X 0.7501E 00, 0.1700E 01, 0.1327E 01 /

```

```

*** H2C ***
DATA Q 3 /
X 0.2085E 01, 0.2482E 01, 0.2168E 01, 0.2759E 01, 0.2309E 01,
X 0.2805E 01, 0.2784E 01, 0.2930E 01, 0.3026E 01, 0.3310E 01,
X C.3310E 01, 0.3368E 01, 0.3558E 01, 0.3999E 01, 0.5308E 01,
X C.5335E 01, 0.8167E 01, 0.5981E 01 /

```

*** C ***

```

DATA Q 4 /
X C.865CE 00, 0.1040E 01, 0.8500E 00, 0.9951E 00, 0.7868E 00,
X 0.9016E 00, 0.8159E 00, 0.8010E 00, 0.7809E 00, 0.7788E 00,
X C.1022E 01, C.8009E 00, 0.8243E 00, 0.1018E 01, 0.1517E 01,
X C.1294E 01, C.1600E 01, 0.9745E 00 /

```

*** LI ***

```

DATA Q 5 /
X 0.3100E 00, 0.3783E 00, 0.3556E 00, 0.4975E 00, 0.4789E 00,
X 0.6411E 00, 0.6911E 00, 0.7717E 00, 0.8395E 00, 0.9150E 00,
X C.5831E 00, 0.1048E 01, 0.1180E 01, 0.1345E 01, 0.1393E 01,
X 0.1479E 01, 0.1367E 01, 0.8132E 00 /

```

*** BE ***

```

DATA Q 6 /
X C.585CE 00, 0.1123E 01, C.9454E 00, 0.1158E 01, 0.9236E 00,
X C.1082E 01, 0.1037E 01, 0.1055E 01, 0.1054E 01, 0.1051E 01,
X 0.1071E 01, 0.1123E 01, 0.1162E 01, 0.1254E 01, 0.1469E 01,
X 0.1585E 01, 0.1500E 01, 0.1153E 01 /

```

*** B ***

```

DATA Q 7 /
X 0.5750E 00, 0.1135E 01, 0.9627E 00, 0.1217E 01, 0.9835E 00,
X 0.1152E 01, 0.1104E 01, 0.1123E 01, 0.1132E 01, 0.1139E 01,
X 0.1149E 01, 0.1168E 01, 0.1180E 01, 0.1200E 01, 0.1251E 01,
X C.141CE 01, 0.1633E 01, 0.1240E 01 /

```

*** AL ***

```

DATA Q 8 /
X 0.1325E 01, C.1513E 01, 0.1266E 01, 0.1527E 01, 0.1240E 01,
X 0.1453E 01, 0.1392E 01, 0.1416E 01, 0.1415E 01, 0.1460E 01,
X 0.1460E 01, 0.1534E 01, 0.1670E 01, 0.1636E 01, 0.1545E 01,
X 0.1701E 01, 0.1533E 01, 0.8401E 00 /

```

*** FE ***

```

DATA Q 9 /
X 0.125CE 01, 0.1489E 01, 0.1275E 01, 0.1650E 01, 0.1437E 01,
X C.1723E 01, 0.1699E 01, 0.1778E 01, 0.1835E 01, 0.1869E 01,
X C.1947E 01, 0.2096E 01, 0.2106E 01, 0.2054E 01, 0.1991E 01,
X C.1581E 01, 0.1733E 01, 0.6720E 00 /

```

*** PB ***

```

DATA Q10 /
X C.3000E 01, 0.3333E 01, 0.2949E 01, 0.3744E 01, 0.2908E 01,
X 0.3406E 01, 0.3359E 01, 0.3419E 01, 0.3417E 01, 0.3407E 01,
X 0.3504E 01, 0.3668E 01, 0.3631E 01, 0.3472E 01, 0.3412E 01,
X C.2942E 01, 0.3367E 01, 0.1949E 01 /

```

*** D ***

```

DATA Q11 /
X 0.305CE 01, 0.3499E 01, 0.2888E 01, 0.3448E 01, 0.2891E 01,
X 0.3386E 01, 0.3244E 01, 0.3341E 01, 0.3426E 01, 0.3553E 01,
X C.3699E 01, 0.4042E 01, 0.4357E 01, 0.4636E 01, 0.4834E 01,
X C.4754E 01, 0.5033E 01, 0.3394E 01 /

```

```

*** U235 ***
DATA Q12 /
X C.2775E 01, 0.3191E 01, 0.2741E 01, 0.3217E 01, 0.2566E 01,
X 0.3005E 01, 0.2880E 01, 0.2960E 01, 0.3036E 01, 0.3193E 01,
X 0.3387E 01, 0.3743E 01, 0.4030E 01, 0.4290E 01, 0.4360E 01,
X C.4120E 01, 0.4000E 01, 0.2890E 01 /

*** U238 ***
DATA Q13 /
X 0.1170E 01, 0.1111E 01, 0.8673E 00, 0.1207E 01, 0.8552E 00,
X 0.1052E 01, 0.1070E 01, 0.1133E 01, 0.1201E 01, 0.1309E 01,
X C.1460E 01, 0.1647E 01, 0.1837E 01, 0.2018E 01, 0.2237E 01,
X 0.2758E 01, 0.1883E 01, 0.9409E 00 /

*** SI ***
DATA Q14 /
X C.9750E 00, 0.1147E 01, 0.9714E 00, 0.1118E 01, 0.9664E 00,
X 0.1152E 01, 0.1094E 01, 0.1094E 01, 0.1084E 01, 0.1061E 01,
X 0.1090E 01, 0.1160E 01, 0.1344E 01, 0.1509E 01, 0.1640E 01,
X 0.2044E 01, 0.2300E 01, 0.1512E 01 /

*** P ***
DATA Q15 /
X 0.1100E 01, 0.1253E 01, 0.1075E 01, 0.1256E 01, 0.9065E 00,
X 0.1062E 01, 0.1065E 01, 0.1084E 01, 0.1074E 01, 0.1090E 01,
X 0.1178E 01, 0.1347E 01, 0.1507E 01, 0.1636E 01, 0.1877E 01,
X 0.1581E 01, 0.2000E 01, 0.1327E 01 /

*** S ***
DATA Q16 /
X C.8800E 00, 0.1099E 01, 0.9714E 00, 0.1207E 01, 0.9921E 00,
X 0.1182E 01, 0.1123E 01, 0.1114E 01, 0.1103E 01, 0.1139E 01,
X 0.1236E 01, 0.1422E 01, 0.1434E 01, 0.1727E 01, 0.1829E 01,
X 0.1664E 01, 0.1733E 01, 0.1025E 01 /

*** K ***
DATA Q17 /
X 0.1135E 01, 0.1300E 01, 0.1101E 01, 0.1340E 01, 0.1052E 01,
X 0.1222E 01, 0.1209E 01, 0.1289E 01, 0.1376E 01, 0.1519E 01,
X 0.1665E 01, 0.1804E 01, 0.1943E 01, 0.2145E 01, 0.2180E 01,
X 0.2145E 01, 0.1757E 01, 0.1018E 01 /

*** CA ***
DATA Q18 /
X C.1135E 01, 0.1300E 01, 0.1101E 01, 0.1340E 01, 0.1129E 01,
X 0.1332E 01, 0.1277E 01, 0.1328E 01, 0.1415E 01, 0.1528E 01,
X 0.1655E 01, 0.1849E 01, 0.2033E 01, 0.2145E 01, 0.2275E 01,
X 0.2324E 01, 0.1980E 01, 0.8737E 00 /

*** TI ***
DATA Q19 /
X 0.1050E 01, 0.1229E 01, 0.1101E 01, 0.1399E 01, 0.1146E 01,
X 0.1402E 01, 0.1401E 01, 0.1494E 01, 0.1581E 01, 0.1684E 01,
X C.1791E 01, 0.1946E 01, 0.2055E 01, 0.2200E 01, 0.2341E 01,
X 0.2324E 01, 0.2200E 01, 0.1186E 01 /

*** V ***
DATA Q20 /
X 0.1225E 01, 0.1442E 01, 0.1258E 01, 0.1502E 01, 0.1206E 01,
X 0.1453E 01, 0.1459E 01, 0.1563E 01, 0.1699E 01, 0.1791E 01,
X C.1908E 01, 0.2036E 01, 0.2052E 01, 0.2036E 01, 0.2085E 01,
X C.1849E 01, 0.1900E 01, 0.1226E 01 /

```

```

*** CR ***
DATA Q21 /
X C.1100E 01, 0.1300E 01, 0.1154E 01, 0.1404E 01, 0.1189E 01,
X 0.1473E 01, 0.1459E 01, 0.1563E 01, 0.1679E 01, 0.1791E 01,
X C.1518E 01, 0.2036E 01, 0.2106E 01, 0.2036E 01, 0.1991E 01,
X C.1823E 01, 0.1750E 01, 0.1226E 01 /

*** MN ***
DATA Q22 /
X 0.1125E 01, 0.1324E 01, 0.1197E 01, 0.1502E 01, 0.1257E 01,
X 0.1523E 01, 0.1507E 01, 0.1621E 01, 0.1728E 01, 0.1850E 01,
X C.1966E 01, 0.2036E 01, 0.2052E 01, 0.2072E 01, 0.2322E 01,
X C.2081E 01, 0.2067E 01, 0.1344E 01 /

*** CC ***
DATA Q23 /
X C.1200E 01, 0.1418E 01, 0.1249E 01, 0.1591E 01, 0.1291E 01,
X 0.1563E 01, 0.1574E 01, 0.1680E 01, 0.1786E 01, 0.1898E 01,
X C.2015E 01, 0.2171E 01, 0.2179E 01, 0.2181E 01, 0.2057E 01,
X 0.1812E 01, 0.2117E 01, 0.1714E 01 /

*** AI ***
DATA Q24 /
X 0.1200E 01, 0.1418E 01, 0.1258E 01, 0.1626E 01, 0.1343E 01,
X 0.1653E 01, 0.1670E 01, 0.1788E 01, 0.1884E 01, 0.1976E 01,
X C.2064E 01, 0.2171E 01, 0.2106E 01, 0.2072E 01, 0.2085E 01,
X 0.2023E 01, 0.2583E 01, 0.2251E 01 /

*** CU ***
DATA Q25 /
X 0.1900E 01, 0.2317E 01, 0.1960E 01, 0.2562E 01, 0.2053E 01,
X 0.2454E 01, 0.2448E 01, 0.2588E 01, 0.2684E 01, 0.2774E 01,
X C.2672E 01, 0.2954E 01, 0.2905E 01, 0.2818E 01, 0.2588E 01,
X 0.2324E 01, 0.2750E 01, 0.1714E 01 /

*** ZN ***
DATA Q26 /
X 0.1330E 01, 0.1560E 01, 0.1370E 01, 0.1773E 01, 0.1488E 01,
X 0.1813E 01, 0.1795E 01, 0.1885E 01, 0.1972E 01, 0.2015E 01,
X C.2054E 01, 0.2111E 01, 0.2070E 01, 0.2018E 01, 0.1972E 01,
X 0.1902E 01, 0.3300E 01, 0.0 /

*** NA ***
DATA Q27 /
X 0.9000E 00, 0.1052E 01, 0.8760E 00, 0.1084E 01, 0.8467E 00,
X 0.1012E 01, 0.9886E 00, 0.9963E 00, 0.1015E 01, 0.1071E 01,
X 0.1071E 01, 0.1048E 01, 0.1064E 01, 0.1300E 01, 0.1441E 01,
X C.2007E 01, 0.2333E 01, 0.2453E 01 /

*** MG ***
DATA Q28 /
X C.8550E 00, 0.1028E 01, 0.8760E 00, 0.1133E 01, 0.8296E 00,
X C.5617E 00, 0.8927E 00, 0.9670E 00, 0.1015E 01, 0.1032E 01,
X C.1012E 01, 0.9805E 00, 0.1100E 01, 0.1200E 01, 0.1137E 01,
X C.1495E 01, 0.2550E 01, 0.2201E 01 /

*** ZR ***
DATA Q29 /
X C.1750E 01, 0.2104E 01, 0.1839E 01, 0.2438E 01, 0.1967E 01,
X 0.2314E 01, 0.2227E 01, 0.2295E 01, 0.2353E 01, 0.2375E 01,
X C.2326E 01, 0.2193E 01, 0.2179E 01, 0.2272E 01, 0.2389E 01,
X C.2800E 01, 0.4867E 01, 0.0 /

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63


```

*** MO ***
DATA Q30 /
X 0.1850E 01, 0.2175E 01, 0.1778E 01, 0.2241E 01, 0.1924E 01,
X 0.2314E 01, 0.2256E 01, 0.2335E 01, 0.2372E 01, 0.2307E 01,
X 0.2266E 01, 0.2171E 01, 0.2088E 01, 0.2145E 01, 0.2237E 01,
X C.2800E 01, 0.3567E 01, 0.3360E 01 /

*** CD ***
DATA Q31 /
X C.2C00E 01, 0.2364E 01, 0.1995E 01, 0.2562E 01, 0.2121E 01,
X 0.2534E 01, 0.2409E 01, 0.2422E 01, 0.2372E 01, 0.2258E 01,
X 0.2258E 01, 0.2246E 01, 0.2251E 01, 0.2290E 01, 0.2379E 01,
X 0.3064E 01, 0.4667E 01, 0.3461E 01 /

*** IN ***
DATA Q32 /
X 0.2150E 01, 0.2530E 01, 0.2168E 01, 0.2759E 01, 0.2224E 01,
X 0.2625E 01, 0.2496E 01, 0.2559E 01, 0.2518E 01, 0.2434E 01,
X 0.2375E 01, 0.2320E 01, 0.2288E 01, 0.2290E 01, 0.2379E 01,
X 0.3064E 01, 0.4167E 01, 0.0 /

*** SN ***
DATA Q33 /
X 0.2175E 01, 0.2600E 01, 0.2212E 01, 0.2759E 01, 0.2266E 01,
X 0.2685E 01, 0.2582E 01, 0.2588E 01, 0.2499E 01, 0.2473E 01,
X 0.2395E 01, 0.2395E 01, 0.2360E 01, 0.2490E 01, 0.2654E 01,
X 0.3170E 01, 0.4333E 01, 0.3159E 01 /

*** BA ***
DATA Q34 /
X 0.2300E 01, 0.2648E 01, 0.2255E 01, 0.2833E 01, 0.2318E 01,
X 0.2705E 01, 0.2553E 01, 0.2530E 01, 0.2450E 01, 0.2774E 01,
X 0.2317E 01, 0.2305E 01, 0.2306E 01, 0.2363E 01, 0.2569E 01,
X 0.3275E 01, 0.4033E 01, 0.2285E 01 /

*** DY ***
DATA Q35 /
X 0.1750E 01, 0.2033E 01, 0.1735E 01, 0.2118E 01, 0.1728E 01,
X 0.2023E 01, 0.1901E 01, 0.1797E 01, 0.1357E 01, 0.1139E 01,
X C.1042E 01, 0.1048E 01, 0.1053E 01, 0.1151E 01, 0.1422E 01,
X 0.1743E 01, 0.3267E 01, 0.2386E 01 /

*** W ***
DATA Q36 /
X 0.2500E 01, 0.2908E 01, 0.2472E 01, 0.3054E 01, 0.2395E 01,
X 0.2785E 01, 0.2697E 01, 0.2794E 01, 0.2850E 01, 0.2901E 01,
X C.2920E 01, 0.3069E 01, 0.3159E 01, 0.3454E 01, 0.3791E 01,
X 0.4437E 01, 0.5333E 01, 0.3427E 01 /

*** AU ***
DATA Q37 /
X 0.2850E 01, 0.3310E 01, 0.2732E 01, 0.3325E 01, 0.2600E 01,
X 0.2995E 01, 0.2832E 01, 0.2882E 01, 0.3866E 01, 0.3008E 01,
X 0.3212E 01, 0.3466E 01, 0.3667E 01, 0.4108E 01, 0.4455E 01,
X 0.4437E 01, 0.4067E 01, 0.2890E 01 /

```

END

```

*****
*
* ANHANG K *
*
*****

```

FORTRAN-LISTEN DES PROGRAMMSYSTEMS ATTOW-K
(RISLEY BIBLIOTHEKSPROGRAMME)

----- INHALTSVERZEICHNIS -----

	SEITE
*** SUBROUTINES ***	
SUBROUTINE CHDEC	K8
SUBROUTINE CHTOWD	K11
SUBROUTINE CVFN	K9
SUBROUTINE DECTP	K12
SUBROUTINE INDEX	K3
SUBROUTINE MESSAGE	K3
SUBROUTINE NEXTCH	K9
SUBROUTINE QCHOUT	K9
SUBROUTINE QMOVE	K12
SUBROUTINE QSHIFT	K12
SUBROUTINE QSTCCH	K11
SUBROUTINE WORDEC	K8
*** SUBROUTINE-ENTRIES ***	
ENTRY ERRCR	K3
ENTRY OUTDEC	K12
ENTRY SELDEC	K12
ENTRY TERROR	K3
*** FUNCTIONS ***	
FUNCTION DBLDEC	K13
FUNCTION FLDEC	K7
FUNCTION INDEC	K7
FUNCTION ITDEC	K6
FUNCTION KCHAR	K11
FUNCTION NEWDEC	K7
FUNCTION NEXCH	K4
FUNCTION NOARG	K14
FUNCTION QCHAR	K11

*** FUNCTION-ENTRIES ***		SEITE
ENTRY BAKCNE	K 6	
ENTRY CCLDEC	K 6	
ENTRY ERRDEC	K 5	
ENTRY GCDEC	K 5	
ENTRY HEXDEC	K 15	
ENTRY KFETCH	K 14	
ENTRY KTYPE	K 14	
ENTRY LENARG	K 14	
ENTRY NERDEC	K 6	
ENTRY NEXDEC	K 6	
ENTRY PCFDEC	K 6	
ENTRY PCNDEC	K 6	
ENTRY QINDEC	K 7	
ENTRY QSTCRE	K 14	
ENTRY TESDEC	K 8	

ATTOW SPECIAL RISLEY LIBRARY ROUTINES

```

SUBROUTINE MESSAGE (/H/)
INTEGER H(2),W(28),X(3)
INTEGER IS /* */
LOGICAL TERR /.FALSE./
L=LENARG(H)
NS=1
M=1
DO 2 I=1,L
K=KCHAR (H,I)
IF (K.NE.IS) GO TO 3
NS=NS+1
KZ=KFETCH (NS,1)
CALL CWFN (KZ,X,N)
5 N=N-1
CALL QMOVE (X,2,h,M,N)
M=M+N
GO TO 4
3 CALL QSTOCH (W,M,K)
M=M+1
4 IF (M.GE.100) GO TO 6
2 CONTINUE
6 M=M-1
CALL QCHOUT (W,M,0)
CALL QCHOUT (W,0,0)
RETURN
===== ENTRY TERROR =====
ENTRY TERROR(RLTINE)
TERR=.TRUE.
RETURN
===== ENTRY ERROR =====
ENTRY ERROR (/H/)
WRITE (6,500)
500 FORMAT (//' *****'/' *****
1 ERROR - MESSAGE *****'/' *****
2*****')
L=LENARG(H)
IF (L.EQ.1) RETURN
NS=1
M=1
DO 8 I=1,L
K=KCHAR (H,I)
CALL QSTOCH (W,M,K)
M=M+1
8 CONTINUE
M=M-1
CALL QCHOUT (W,M,0)
CALL QCHOUT (W,0,0)
STOP
END

```

-117-

```

SUBROUTINE INDEX (J,h,/H/,NN)
DIMENSION H(1),NN(1),W(1),IX(8)
DATA IT,IB,IS,IE / 4H ,4H* ,4H* /
N=4
IF (NOARG(0).NE.4) GO TO 11
N=NN(1)
IF (N.GT.8) N=8

```

```

11 L=LENARG(H)
   I=0
   J=0
   M=C
   DO 2 K=1,N
     IX(K)=KCHAR(h,K)
     IF (IX(K).NE.IB) M=K
2 CONTINUE
6 J=J+1
   DO 3 K=1,N
     I=I+1
     IT=KCHAR(H,I)
     IF (I.GT.L) IT=IE
     IF (IT.EQ.IS.OR.IT.EQ.IE) GC TC 4
     IF (IT.NE.IX(K)) GO TO 5
3 CONTINUE
   GO TO 8
5 I=I+1
   IT=KCHAR(H,I)
   IF (I.GT.L) IT=IE
7 IF (IT.EQ.IS) GC TO 6
   IF (IT.NE.IE) GC TO 5
   J=J+1
   GO TO 8
4 IF (K.GT.M) GO TO 8
   GO TO 7
8 RETURN
   END

```

```

-----
FUNCTION NEXCH (ICC,II)
REAL*8 DDATUM,HZEIT
INTEGER BSS(33),IDIG(10),ILET(26),WD(33),FORM(4)
LOGICAL PSAVE,GODX,PSWT,PDONE,SWT
DATA IDIG / 4H0 ,4H1 ,4H2 ,4H3 ,4H4 ,4H5 ,4H6 ,
1 4H7 ,4H8 ,4H9 /
DATA ILET / 4HA ,4HB ,4HC ,4HD ,4HE ,4HF ,4HG
1 ,4HH ,4HI ,4HJ ,4HK ,4HL ,4HM ,4HN ,4HO ,4HP
2 ,4HQ ,4HR ,4HS ,4HT ,4HU ,4HV ,4HW ,4HX ,4HY
3 ,4HZ /
DATA IB,IEND,IPL,IM / 4H ,4HEND ,4H+ ,4H- /
DATA ISTAR /4H* /
DATA PSWT,PDONE / .TRUE.,.FALSE. /
DATA PASTIM / -1.0 /
DATA NCOL,NCH,NPERM,NERR / 1,-1,0,0 /
DATA FORM /'( ,X1H=','/IHO',') '/
DATA WD /' INP','UT E','RRCR',', ' /

SWT=.FALSE.
25 IF (NCOL.LE.NCH+1) GO TO 2
CALL DECTP (BSS,NCH)
NCOL=1
PDONE=.FALSE.
2 IF (PSWT.AND..NOT.PDONE) CALL CUTDEC (BSS,NCH)
IF (PSWT.AND..NOT.PDONE) WRITE (6,1000)
1000 FORMAT ('+',88X,'===== INPUT-CARD =====')
PDONE=.TRUE.
ICH=KCHAR (BSS,NCOL)
IF (NCOL.GT.NCH) ICH=IEND
NCOL=NCOL+1

```

```

1 NTP=1
2 DO 13 I=1,10
3 II=I
4 IF (ICH.EQ.IDIG(I)) GO TO 10
5 13 CONTINUE
6 NTP=3
7 DO 14 I=1,26
8 II=I
9 IF (ICH.EQ.ILET(II)) GO TO 10
10 14 CONTINUE
11 NTP=2
12 IF (ICH.EQ.IB.CR.ICH.EQ.IENC) GO TO 10
13 NTP=4
14 IF (ICH.EQ.IPL.CR.ICH.EQ.IP) GC TC 10
15 NTP=5
16 IF (SWT) GO TO 16
17 ICC=ICH
18 NEXCH=NTP
19 RETURN
20
===== ENTRY GODEC =====
21 ENTRY GODEC (/TITLE/)
22 17 PSAVE=PSWT
23 PSWT=.FALSE.
24 IF (PASTIM.GE.0.0) GO TO 11
25 PASTIM=ZEIT(0)
26 CALL DATUM (DDATUM,HZEIT)
27 WRITE (6,21) DDATUM,HZEIT
28 21 FORMAT (IHL,70X,'DATE: ',A8,5X,'TIME: ',A8///)
29 GO TO 15
30 11 TT=ZEIT(PASTIM)
31 XT=TT/60.
32 WRITE (6,20) XT
33 20 FORMAT (/100X,'TIME =',F7.3,' MINUTES'///)
34 5 PASTIM=ZEIT(PASTIM)
35 15 SWT=.TRUE.
36 GO TO 25
37 16 IF (NTP.EQ.2) GO TO 25
38 IF (ICH.EQ.ISTAR.AND.NCOL.EQ.2) GC TC 7
39 NCM=NCOL-1
40 IF (ICH.NE.ILET(5)) WRITE (6,23) NCM
41 23 FORMAT (17HODATA FROM COLUMN,I4,9H IGNCRD )
42 4 CALL DECTP (BSS,NCH)
43 IF (KCHAR(BSS,1).NE.ISTAR) GO TO 4
44 7 CALL OUTDEC (BSS,NCH)
45 WRITE (6,1010)
46 1010 FORMAT ('+',88X,'===== TITLE =====')
47 CALL DECTP (BSS,NCH)
48 IF (KCHAR(BSS,1).EQ.ISTAR) GO TO 7
49 NCOL=1
50 PDONE=.FALSE.
51 PSWT=PSAVE
52 WRITE (6,22)
53 22 FORMAT (' ')
54 RETURN
55
===== ENTRY ERRDEC =====
56 ENTRY ERRDEC (/HH/)
57 I=LENARG (HH)
58 NERR=NERR+1
59 IF (NERR.LE.NPERM) GC TO 33
60 CALL ERRCR (2H E)
61 33 CALL QMOVE (HH,1,WD,16,I)
62
63

```

```

CALL QCHOUT (WD,I+15,0)
CALL QCHOUT (BSS,NCH,2)
CALL QMOVE (3H ,1,FORM,2,3)
L=4
N=NCOL-1
34 NDIG=MOD(N,10)
CALL QSTCCH (FORM,L,IDIG(NDIG+1))
N=N/10
L=L-1
IF (N.NE.0) GO TC 34
WRITE (6,FORM)
STOP
===== ENTRY BAKONE ===
ENTRY BAKONE (/ZZZ/)
NCOL=NCOL-1
RETURN
===== ENTRY PONDEC ===
ENTRY PONDEC (/ZZZ/)
PSWT=.TRUE.
RETURN
===== ENTRY POFDEC ===
ENTRY POFDEC (/ZZZ/)
PSWT=.FALSE.
RETURN
===== ENTRY COLDEC ===
ENTRY COLDEC(NN)
NCOL=IABS(NN)
IF (NN.GT.0) GO TO 18
POONE=.FALSE.
CALL DECTP(BSS,NCH)
18 RETURN
===== ENTRY NEXDEC ===
ENTRY NEXDEC (/ZZZ/)
NEXCH=NCCL
ZZZ=ZZZ
RETURN
===== ENTRY NERDEC ===
ENTRY NERDEC (MP)
NEXCH=NERR
NERR=0
NPERM=MP
IF (NPERM.EQ.0) NPERM=50
RETURN
END

```

```

FUNCTION ITDEC (M)
DATA IE / 4HEND /
IF (M.EQ.1) GO TO 2
IF (NEWDEC(0).GE.0) CALL ERRDEC (
1 31HFIRST CHARACTER NOT ALPHABETIC& )
CALL WORDEC (K,4,0)
GO TO 3
2 CALL NEXTCH (K,I)
IF (K.EQ.IE) GC TO 2
3 ITDEC=K
RETURN
END

```

```

1 FUNCTION INDEC(MODEL)
2 DCUBLE PRECISICN Z,ZZ
3 X=QINDEC(MODEL,Z)
4 CALL QMOVE (X,1,I,1,4)
5 IF (TESDEC(0) 2,2,3)
6 ZZ=Z
7 IF (DABS(ZZ).LE.2147483647.000) GC TC 4
8 CALL ERRDEC (19HEXCESSIVE INTEGER+8)
9 Z=0.C
10 4 I=Z+DSIGN(0.500,Z)
11 2 INDEC=I
12 RETURN
13 END

```

```

19 FUNCTION NEWDEC (M)
20 DATA IE,IPPOINT,IPLUS,IDASH,IB / 4HEND ,4H. ,4H+ ,4H- ,
21 1 4H /
22 2 CALL NEXTCH (K,I)
23 IF (K.EQ.IB.OR.K.EQ.IE) GO TO 2
24 CALL BAKCNE (0)
25 IF (K.EQ.IPCINT) I=1
26 IF (M.EQ.0.AND.(K.EQ.IPLUS.CR.K.EQ.IDASH)) I=1
27 NEWDEC=I
28 RETURN
29 END

```

```

35 FUNCTION FLDEC(MODEL)
36 DCUBLE PRECISICN WORD,DBLDEC
37 LOGICAL FLSWT
38 EQUIVALENCE (FLAG,IFLAG)
39 DATA X,Y,Z / 4H ,4H ,4H /
40 FLSWT=.TRUE.
41 GO TO 10
===== ENTRY QINDEC ===
42 ENTRY QINDEC (MODEL,Q)
43 DCUBLE PRECISICN Q(1)
44 FLSWT=.FALSE.
45 1C MODE=MODEL
46 IF (MODE.GT.1) GO TC 2
47 7 IFLAG=1
48 WORD=DBLDEC(MODE)
49 IF (.NOT.FLSWT) Q(1)=WORD
50 FLDEC=WORD
51 RETURN
52 2 IF (MODE.EQ.6) GO TC 8
53 M=0
54 IF (MODE.GT.3.AND.MCDE.NE.7) M=1
55 I=NEWDEC(M)
56 IF (I) 5,4,3
57 3 IF (MODE.GE.3) GC TC 111
58 CALL ERRDEC (21HNUMBER NCT EXPECTED+8)
59 111 MODE=1
60 GC TO 7
61 4 IFLAG=C
62 6 CALL CHDEC (X,1,4H ,M)
63

```

```

FLDEC=X
RETURN
5 IFLAG=-1
  IF (MODE.EQ.5) GO TO 6
  M=0
  IF (MODE.GE.7) M=1
  CALL WORDEC (Y,4,M)
  FLDEC=Y
  RETURN
6 CALL NEXTCH (Z,IFLAG)
  FLDEC=Z
  RETURN
===== ENTRY TESDEC =====
ENTRY TE SDEC (/ZZZ/)
ZZZZ=ZZZ
FLDEC=FLAG
RETURN
END

```

```

SUBROUTINE WORDEC (A,N,MM)
DIMENSION A(1),MM(1)
DATA IE,IB / 4HEND ,4H /
I=0
IT=NEWDEC(1)
M=0
IF (NOARG(0).GT.2) M=MM(1)
5 CALL NEXTCH (IX,IZ)
  IF (IX.EQ.IE) GO TO 2
  IF (I.EQ.0.AND.IZ.EQ.0) GO TO 4
  IF (IZ.EQ.0.OR.(M.EQ.0.AND.IZ.NE.IT)) GO TO 3
  IF (I.GE.N) GO TO 5
  I=I+1
  CALL QSTOCH (A,I,IX)
  GO TO 5
3 CALL BAKCNE (0)
2 IF (I.GE.N) GO TO 6
7 I=I+1
  CALL QSTOCH (A,I,IB)
  GO TO 2
4 CALL ERRDEC (31HFIRST CHARACTER NCT ALPHANERIC )
  GO TO 7
6 RETURN
END

```

```

SUBROUTINE CHDEC (A,N,TT,J)
DIMENSION A(1),TT(1),J(1)
LOGICAL LAST
DATA IE,IB / 4HEND ,4H /
NARG=NOARG(0)
LAST=.FALSE.
IT=IE
IF (NARG.GT.2) IT=KCHAR(TT,1)
IF (NARG.EQ.4) J(1)=N
DO 3 I=1,N
  IF (LAST) GO TO 3

```

```

1 5 CALL NEXTCH (M,K)
2   IF (M.EQ.IE) GC TO 4
3   IF (M.NE.IT) GC TO 3
4   CALL BAKCNE (0)
5   GO TO 8
6 4 IF (I.EQ.1) GO TO 5
7 8 IF (NARG.EQ.4) J(1)=I-1
8   LAST=.TRUE.
9   M=IB
10 3 CALL QSTOCH (A,I,M)
11   RETURN
12  END

```

```

SUBROUTINE NEXTCH (I,IFLAG)
INTEGER IND(5)
DATA IND / 1,0,-1,0,0 /
NTYP=NEXCH(I,N)
IFLAG=IND(NTYP)
RETURN
END

```

```

SUBROUTINE QCHCUT (A,K,K)
DIMENSION A(1),IC(3)
DATA IB,IC / 4H ,4H0 ,4H1 ,4H /
M=K+1
IF (M.LT.1.OR.M.GT.3) M=3
J=N/4
J4=J*4
L=N-J4
Lh=IB
IF (L.EQ.0) GO TO 3
DO 2 I=1,L
  JJ=J4+I
2 CALL QSTCCH (Lh,I,QCHAR(A,JJ))
3 IF (J.EQ.0) GO TO 4
  WRITE (6,10) IC(M),(A(I),I=1,J),LM
10 FORMAT (A1,33A4)
  RETURN
4 WRITE (6,10) IC(M),Lh
  RETURN
END

```

```

SUBROUTINE CVFN (Z,V,I)
INTEGER V(2),W(2),IDIG(10),IB / 4H /,IS / 4H* /
EQUIVALENCE (X,IX)
DATA IDIG / 4H0 ,4H1 ,4H2 ,4H3 ,4H4 ,4H5 ,4H6 ,
1 4H7 ,4H8 ,4H9 /
DO 14 J=1,3
14 V(J)=IB
  X=Z

```

```

CALL QSTCCH (V,I,1H )
IF (X.NE.0.0) GO TO 9
I=2
CALL QSTCCH (V,2,1H0 )
GO TO 20
5 I=1
K=KTYPE(X)
GO TO (1C,8,12),K
12 CALL HEXDEC (X,K)
CALL QMOVE (W,I,V,3,8)
CALL QSTCCH (V,2,IS)
CALL QSTCCH (V,11,IS)
I=11
2C RETURN

10 X=IX
8 IF (X.GT.0.0) GO TO 22
I=I+1
CALL QSTCCH (V,I,1H- )
X=-X
22 J=5
9C FORMAT (2I15,3X 3A4)
2 IF (X.LT.1.0E5) GO TO 1
4 XX=X+0.5
IF (XX.GE.1.0E6) GO TO 5
IX=XX
L=8
LZ=0
3 N=IX/10
K=IX-10*N+1
CALL QSTCCH (W,L,IDIG(K))
IF (LZ.EQ.0.AND.K.NE.1) LZ=L
L=L-1
IX=N
IF (IX.GT.0) GO TO 3

11 IE=0
IF (J.LE.5.AND.J.GT.-4) GO TO 7
IE=J
J=0
7 IP=J+1
J=0
6 IF (IP.GE.0.AND.J.NE.IP) GO TO 17
I=I+1
CALL QSTCCH (V,I,1H.)
18 IF (IP.GE.0) GO TO 17
IP=IP+1
I=I+1
CALL QSTCCH (V,I,1H0 )
GO TO 18
17 J=J+1
I=I+1
L=L+1
CALL QMOVE (W,L,V,I,1)
IF (L.LT.LZ.OR.J.LT.IP) GO TO 6
13 IF (IE.EQ.0) GO TO 20
IF (I.LT.9.OR.IE.GT.-10) I=I+1
CALL QSTCCH (V,I,1HE )
X=IE
GO TO 8
1 IF (X.LT.0.5) GO TO 21
X=X*10.0
J=J-1
GO TO 2

```

```

1 21 X=X*1.0E8 1
2 J=J-8 2
3 GO TO 2 3
4 5 X=X/10.0 4
5 J=J+1 5
6 GO TO 4 6
7 END 7
-----
13 SUBROUTINE CHTCWD (A,N) 13
14 INTEGER A(1) 14
15 DATA IB / 4H / 15
16 K=(N+3)/4 16
17 K4=4*(K-1) 17
18 L=N-K4 18
19 I4=IB 19
20 2 DO 3 J=1,L 20
21 M=K4+J 21
22 3 CALL QSTCCH (I4,J,QCHAR(A,M)) 22
23 A(K)=I4 23
24 RETURN 24
25 END 25
-----
31 SUBROUTINE QSTCCH (A,N,X) 31
32 LOGICAL*1 A(1),X 32
33 A(N)=X 33
34 RETURN 34
35 END 35
-----
41 FUNCTION QCHAR (A,N) 41
42 REAL W / 4H / 42
43 LOGICAL*1 A(2),X 43
44 X=A(N) 44
45 CALL QSTCCH (W,I,X) 45
46 QCHAR=W 46
47 RETURN 47
48 END 48
-----
54 FUNCTION KCHAR (A,N) 54
55 INTEGER W /4H / 55
56 LOGICAL*1 A(2),X 56
57 X=A(N) 57
58 CALL QSTCCH (W,I,X) 58
59 KCHAR=W 59
60 RETURN 60
61 END 61

```

```

SUBROUTINE QMOVE (A,I,B,J,N)
DIMENSION A(1),B(1)
JJ=J
II=I
DO 2 K=1,N
CALL QSTOCH (B,JJ,QCHAR(A,II))
II=II+1
2 JJ=JJ+1
RETURN
END

```

1
2
3
4
5
6
7
8
9
10

```

SUBROUTINE QSHIFT (A,II,N,K)
DIMENSION A(1)
J=1
M=ISIGN(J,K)
I=II
IF (K) 4,5,3
3 I=I+N-1
4 DO 2 L=1,N
J=I+K
CALL QSTOCH (A,J,QCHAR(A,I))
2 I=I-M
5 RETURN
END

```

16
17
18
19
20
21
22
23
24
25
26
27
28

```

SUBROUTINE DECTP (A,KC)
DIMENSION A(33)
INTEGER IB(34)
DATA NN, KK / 5,72 /
DATA IB / 4H /
2 L=(KK+3)/4
READ (NN,1,END=99) (A(I),I=1,L)
1 FORMAT (33A4)
4 KC=KK
RETURN
99 STOP

```

34
35
36
37
38
39
40
41
42
43
44

C===== ENTRY SELDEC =====

```

ENTRY SELDEC (N,K)
NN=N
KK=K
RETURN

```

45
46
47
48
49

C===== ENTRY OUTDEC =====

```

ENTRY OUTDEC (I/B,MM)
CALL QMOVE (B,1,IB,5,MM)
CALL QCHOUT (IB,MM+4,2)
RETURN
END

```

50
51
52
53
54
55

```

DOUBLE PRECISION FUNCTION DELDEC (MCCE)
DOUBLE PRECISION WORD,DI
LOGICAL POINT,MINUS,MINX,DIGITS
INTEGER IEX / 'E ' /
DATA IPT,IM,IB,IEND / 4H. ,4H- ,4H ,4HEND /
DATA MEX /75/

```

1
2
3
4
5
6
7

```

DIGITS=.FALSE.
PCINT=.FALSE.
MINUS=.FALSE.
WORD=0.000
NTEN=0
NEXP=0

```

8
9
10
11
12

```

200 NTYP=NEXCH(ICH,I)
GO TO (210,200,204,205,206),NTYP
204 CALL ERRDEC (16HNUMBER EXPECTED& )
GO TO 200
206 IF (ICH.NE.IPT) GO TO 204
208 POINT=.TRUE.
GO TO 202
205 IF (ICH.EQ.IM) MINUS=.TRUE.
202 NTYP=NEXCH(ICH,I)
GO TO (210,213,212,213,214),NTYP
214 IF (ICH.NE.IPT) GO TO 213
IF (.NOT.POINT) GO TO 208
CALL ERRDEC (19HTWO DECIMAL PCINTS& )
GO TO 213
210 IF (POINT) NTEN=NTEN-1
DIGITS=.TRUE.
DI=I-1
WCRD=WORD*10.000+DI
GO TO 202

```

13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

```

212 IF (ICH.NE.IEX) GO TO 213
IF (.NOT.DIGITS) CALL ERRDEC (15HDIGIT EXPECTED& )
DIGITS=.FALSE.
MINX=.FALSE.
215 NTYP=NEXCH(ICH,I)
GO TO (216,217,218,219,218),NTYP
216 DIGITS=.TRUE.
NEXP=10*NEXP+I-1
240 NTYP=NEXCH(ICH,I)
IF (NTYP.EQ.1) GO TO 216
GO TO 218
219 IF (ICH.EQ.IM) MINX=.TRUE.
GO TO 240

```

29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45

```

217 IF (ICH.EQ.IB) GO TO 215
218 IF (NEXP.GT.MEX) CALL ERRDEC (19HEXCESSIVE EXPONENT& )
IF (MINX) NEXP=-NEXP
213 IF (MINUS) WORD=-WORD
IF (.NOT.DIGITS) CALL ERRDEC (15HDIGIT EXPECTED& )
IF (MODE.EQ.0.AND.NTYP.NE.2) CALL ERRDEC (15HBLANK EXPECTED& )
IF (ICH.NE.IEND) CALL BAKCNE (0)
NTEN=NTEN+NEXP
IF (NTEN.NE.0) WORD=WORD*10.000**NTEN
DBLDEC=WCRD
RETURN
END

```

46
47
48
49
50
51
52
53
54
55
56
57

```

NOARG  START 1
        ENTRY NOARG$ 2
        ENTRY LENARG 3
        ENTRY KTYPE 4
        ENTRY QFETCH 5
        ENTRY KFETCH 6
        ENTRY QSTORE 7
        ENTRY HEXDEC 8
        USING *,15 9
        L 0,NOARG$ 10
        BR 14 11
NOARG$ DC F'0' 12
*      FUNCTION LENARG (A) 13
        USING *,15 14
LENARG L 1,0(1) 15
        SR 0,0 16
        ST 14,RET 17
LLL    CLI 0(1),X'50' 18
        BC 8,RETU 19
        AH 0,=H'1' 20
        AH 1,=H'1' 21
        B LLL 22
RETU   L 14,RET 23
        BR 14 24
RET    DC F'0' 25
*      FUNCTION KTYPE(X) INTEGER=1 FPN=2 OTHER=3 26
        USING *,15 27
KTYPE ST 1,24(13) 28
        L 1,0(1) 29
        LA 0,1(0) 30
        TM 0(1),X'FF' 31
        BC 9,OUT 32
        LA 0,2(0) 33
        TM 1(1),X'F0' 34
        BC 5,OUT 35
        LA 0,3(0) 36
        L 1,24(13) 37
        BR 14 38
*      X=QFETCH(I,N) N-TH ITEM OF I-TH ARGUMENT 39
QFETCH DS OH 40
KFETCH DS OH 41
        USING *,15 42
        STM 14,4,12(13) 43
        LM 2,3,0(1) 44
        L 1,4(13) R13 ON ENTRY TO CALLING ROUTINE 45
        L 1,24(1) R1 CN ENTRY TO CALLING ROUTINE 46
        LA 4,4(0) 47
        L 2,0(2) I 48
        L 3,0(3) N 49
        SLL 2,2 50
        SR 2,4 R2=4(I-1) 51
        SLL 3,2 52
        SR 3,4 R3=4(N-1) 53
        L 1,0(2,1) R1=ADD. OF AI(1) 54
        LE 0,0(3,1) FRO=AI(N)=X 55
        L 0,0(3,1) RO=AI(N)=K 56
        LM 1,4,24(13) 57
        BR 14 58
*      CALL QSTORE (X,I,N) 59
QSTORE STM 14,5,12(13) 60
        LM 2,4,0(1) 61
        L 1,4(13) 62
        L 1,24(1) 63

```

```

L 2,0(2) R2=X 1
L 3,0(3) 2
L 4,0(4) 3
BCTR 3,0 4
BCTR 4,0 5
SLL 3,2 6
SLL 4,2 7
L 1,0(3,1) R1=ADD. OF AI(1) 8
ST 2,0(4,1) 9
LM 1,5,24(13) 10
BR 14 11
*      4-BYTE WORD TO 8-BYTE HEXADECEIMAL 12
        USING *,15 13
HEXDEC STM 14,5,12(13) 14
        LM 3,4,0(1) 15
        L 2,0(3) 16
        LA 5,7 17
        LA 1,1 18
LOOP   SRDL 2,4 19
        SRL 3,28 20
        IC 0,CH(3) 21
        STC 0,0(5,4) 22
        SR 5,1 23
        BC 10,LOCP 24
        LM 2,5,28(13) 25
        BR 14 26
CH     DC C'0123456789ABCDEF' 27
        END 28

```


L. Programm zum Lesen der gespeicherten Resultate

A T T O W -FLUESSE VON EINHEIT 10 LESEN

DIMENSION PHI(17000),MHALF(2),RMESH(120),ZMESH(120)

READ(5,100) NG
READ(5,100) NROW,NCOL

MHALF(1) = NCOL*(NROW/2)
MHALF(2) = NCOL*NROW-MHALF(1)
WRITE(6,104)

REWIND 10

DO 5 N = 1,NG

IND = MHALF(1)

READ(10) (PHI(I),I = 1,IND)

IST = IND+1

IND = IND+MHALF(2)

READ(10) (PHI(I),I = IST,IND)

IST = 1

IND = MINO(11,NCOL)

1 WRITE(6,102) N,IST,IND,(I,I = IST,IND)

WRITE(6,108)

DO 2 J = 1,NROW

IJ = NROW-J+1

NST = (J-1)*NCOL+IST

NND = NST-IST+IND

2 WRITE(6,110) IJ,(PHI(M),M = NST,NND)

IF(IND.EQ.NCOL) GOTO 5

IST = IND+1

IND = MINO(IND+11,NCOL)

GOTO 1

5 CONTINUE

READ(10) NCOL,(RMESH(I),I = 1,NCOL)

READ(10) NROW,(ZMESH(I),I = 1,NROW)

WRITE(6,112)

WRITE(6,114) (J,RMESH(J),J = 1,NCOL)

WRITE(6,116)

WRITE(6,114) (J,ZMESH(J),J = 1,NROW)

100 FORMAT(2I6)

102 FORMAT('1FLUXES IN GROUP',I3,' FOR COLUMNS',I3,' TO',I3//
1' ROW/COL',I6,2X,10(I8,2X))

104 FORMAT(1H1,20X,'***** A T T O W -FLUXES *****'///)

108 FORMAT('0')

110 FORMAT(' ',I8,4X,1P11E10.3)

112 FORMAT(///5X,'HORIZONTAL MESH COL R OR X'//)

114 FORMAT(8(I3,F10.2))

116 FORMAT(//5X,'VERTICAL MESH ROW Z OR Y'//)

STOP

END