

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

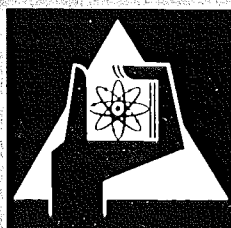
März 1971

KFK 1381

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

Das eindimensionale Transportprogramm DTK

C. Günther, W. Kinnebrock



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

März 1971

KFK 1381

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

Das eindimensionale Transportprogramm DTK

C. Günther

W. Kinnebrock



Gesellschaft für Kernforschung m.b.H., Karlsruhe

Zusammenfassung

DTK ist ein Fortran IV-Programm zur Lösung der eindimensionalen Neutronen-transportgleichung mit der Methode der Diskreten Ordinaten im Multigruppenmodell. Es wurde aus dem Code DTF-IV entwickelt. DTK unterscheidet sich von DTF-IV vorallem in folgenden Punkten:

- 1) Die Eingabe wurde vereinfacht.
- 2) DTK hat "interfaces" mit dem Programmsystem NUSYS.
- 3) Eine neue nichtkonstante Quellschätzung wurde eingebaut.
- 4) Eine Vielzahl von Bucklings stehen dem Benutzer zur Verfügung.
- 5) Der Benutzer kann in Plattengeometrie den einfallenden winkel- und energiesabhängigen Fluß am rechten Rand vorgeben.
- 6) Für die äußeren Iterationen wurde eine Tschebyscheff Extrapolation eingebaut.
- 7) Flüsse können von Fall zu Fall weitergegeben werden. Falls bei diesen S_N -Ordnung und Anzahl der Intervalle nicht übereinstimmen, nimmt das Programm eine Interpolation vor.

Der Bericht stellt eine Anleitung zur Benutzung des Codes dar. Er enthält neben der Eingabebeschriftung auch eine Programmliste und ein Eingabebeispiel.

Abstract

DTK is a FORTRAN-IV program for solving the multigroup onedimensional (plane, sphere, cylinder) transport equation by the methods of discrete ordinates. It has been developed from the code DTF-IV. DTK differs from the DTF-IV code essentially in the following features:

- 1) The input has been simplified.
- 2) DTK has "interfaces" with the program system NUSYS.
- 3) A new nonconstant initial source guess has been built in.
- 4) A variety of buckling options has been incorporated.
- 5) In plane geometry the user may prescribe the incoming energy- and space-dependent flux at the right boundary.
- 6) Tchebychew extrapolation for the outer iteration has been built in.
- 7) It is possible to pass fluxes from one case to the other. If S_N -order and the number of intervals do not agree, DTK is able to interpolate.

The report describes how to use the code and includes a code listing and a sample input.

Inhaltsverzeichnis

0. Einführung
1. Physikalische Möglichkeiten des Codes
2. Technische Möglichkeiten des Codes
3. Aufruf
4. Bemerkungen zur Eingabe
5. Eingabebeschreibung
6. Bemerkungen und Erläuterungen zur Ausgabe des Codes

Anhang 1: Beispiel einer vollständigen Eingabe für DTK
samt Jobkontrollkarten

Anhang 2: Beispiel einer NUSYS-Eingabe zur Erzeugung von
Wirkungsquerschnitten und deren Bereitstellung
auf einem Zwischenspeicher

Anhang 3: Programmliste

0. Einführung

DTK ist ein eindimensionales S_N -Programm zur Lösung der Transportgleichung in ebener, sphärischer und zylindrischer Geometrie. Dieser Code ist eine Weiterentwicklung des in Los Alamos entstandenen Programmes DTF-IV /⁻1_7. Die wesentlichen Änderungen gegenüber DTF-IV sind folgende:

- 1) Die Eingabe wurde vereinfacht.
- 2) DTK hat "Interfaches" mit dem Programmsystem "NUSYS". Man kann Querschnittsblöcke von NUSYS übernehmen und Flußblöcke an NUSYS weitergeben. Dies erlaubt, die Rechenergebnisse von DTK allen eindimensionalen Auswerteprogrammen innerhalb NUSYS zuzuführen.
- 3) Es wurde eine automatische nichtkonstante Anfangsquellschätzung eingebaut.
- 4) Es wurde ermöglicht, eine Vielzahl von Bucklingarten zu berücksichtigen.
- 5) In Plattengeometrie ist es möglich, am rechten Rand den winkelabhängigen einfallenden Fluß vorzugeben.
- 6) DTK benutzt Tschebyscheff Extrapolation für die äußeren Iterationen.
- 7) Es besteht die Möglichkeit, Skalarflüsse bei gleicher Gruppen- und Zonenzahl von Fall zu Fall weiterzugeben. S_N -Ordnung und Anzahl der Intervalle können verschieden sein, da DTK in diesem Fall interpolieren kann.
Dies gestattet es,
 - a) bei größeren Fällen (Vielgruppenfälle, Vielpunktfälle) einen oder mehrere Vorfälle mit kleinerer S_N -Ordnung, weniger Ortspunkten und eventuell geringerer Genauigkeit durchzuführen.
 - b) bei umfangreichen Parameterstudien als Flußschätzung das reservierte Ergebnis eines Modellfalles zu verwenden.

Der vorliegende Bericht soll keine Programmbeschreibung sein. Er soll angeben, welche Möglichkeiten das Programm DTK bietet und wie Eingaben für DTK zu erstellen sind. Die beiden Anhänge sollen bei diesem Bemühen behilflich sein. Die Funktion einer Programmbeschreibung erfüllt wenigstens zum Teil der Report /⁻1_7.

Der Bericht setzt voraus, daß es sich um eine Programmversion handelt, die auf dem System OS der IBM 360 betrieben wird.

Zum folgenden Text ist zu bemerken:

Die im folgenden Text aufgeführten Variablennamen stimmen überein mit der Bezeichnung dieser Größen in der nachfolgenden Eingabebeschriftung (Teil 5). In α -Zeichen gefaßte Namen sind feste Bezeichnungen aus dem Programmsystem NUSYS.

Die Abkürzung "JCL" bedeutet - "JOB CONTROL LANGUAGE im OS der IBM/360/65" - .

1. Physikalische Möglichkeiten des Codes

Mit DTK lassen sich außer der Flußverteilung bei gegebener äußerer Quelle oder Fluß am Rand und k_{eff} die folgenden vier impliziten "Eigenwerte" berechnen:

- 1) der Zeiteigenwert (α -Iteration)
- 2) der kritische Radius eines Systems (Radieniteration)
- 3) die kritische Dicke einer oder mehrerer Zonen eines Systems (Delta-Iteration)
- 4) die kritische Konzentration eines Systems, einer Mischung oder eines Mischungselements.

Die Auswahl einer dieser Möglichkeiten geschieht über die Eingabeparameter IEVT und IPVT. Für IEVT = 0 oder 1, -äußere Quelle oder k_{eff} -Rechnung - können IPVT und auch PV beliebig sein, da sie vom Programm nicht benutzt werden.

Für IEVT = 2,3,4,5 und IPVT = 0 wird der "Eigenwert" so errechnet, daß das System kritisch wird ($k_{\text{eff}} = 1$). Ist für dieselben IEVT der Wert IPVT = 1 und PV = α , so wird der Eigenwert so bestimmt, daß $k_{\text{eff}} = \alpha$ ist.

Für IPVT = 2 und PV = β wird der Eigenwert $EV = \lambda$ im Falle IPVT = 2 so bestimmt, daß das System mit dem Zeiteigenwert ($\lambda + \beta$) kritisch wird, im Falle IEVT = 3,4,5, IPVT = 2 und PV = β wird EV so bestimmt, daß das System bei vorgegebener Zeitabsorption β für EV kritisch wird.

Für die Delta-Iteration (IEVT = 4) sind Zonenparameter $B(J)$, $J=1,2,\dots,IZM$, einzugeben, die folgende Bedeutung haben: Sind $R(J)$, $J=1,2,\dots,IZM$ die Zonenradien, so wird der Eigenwert $EV = \delta$ so bestimmt, daß das System mit den modifizierten Zonenradien

$$\bar{R}(J+1) = \bar{R}(J) + (1+\delta B(J+1)) \cdot (R(J+1)-R(J))$$

$$J=0,1,2,\dots,IZM-1, \bar{R}(0)=R(0)=0$$

entweder kritisch für $IPVT = 0$ wird

oder $k_{eff} = \alpha$ ergibt für $IPVT = 1$ und $PV = \alpha$

oder kritisch mit Zeitabsorption β für $IPVT = 2$ und $PV = \beta$ wird.

Hier ist z.B. die Möglichkeit gegeben, die eine Zone (es sei die 1. Zone) so zu modifizieren, daß das System kritisch wird (in dem man $B(1) = 1$ und die übrigen $B(J) = 0$ setzt).

Für die Herleitung dieser aufgeführten Ergebnisse bietet das Programm DTK noch folgende physikalischen Möglichkeiten:

- a) Man kann intern aus eingegebenen Mischungen weitere Mischungen herstellen. Dies ist vor allem dann von Bedeutung, wenn irgendeine Form von Materialiteration im Sinne des obigen Punktes 4 durchgeführt werden soll. Die Steuerung geschieht über die Größe MS (siehe K1) und die Eingabegrößen von K13. Das Eingabeschema von K13 lautet:

$$(I, K(I), L(I), A(I), I=1, MS)$$

für jede der MS Mischungsspezifikationen eine Karte (im Format 3I6,E12.5). Wenn σ_m ($m=1,\dots,MTP$) ein Querschnittstyp ist, gilt für alle Querschnittstypen

falls	$K(I) = 0$	$:\sigma_n = A(I) \cdot \sigma_n$	$n=L(I);$
falls	$K(I) = J, A(I) = 0$	$:\sigma_n = \sigma_n + A(I) \cdot \sigma_j$	$n=L(I), j=J;$
falls	$K(I) = 0, A(I) = 0$	$:\sigma_n = \sigma_n \cdot EV$	$n=L(I);$

Das unter K13 angegebene Schema soll noch an einem Beispiel erläutert werden:

Es sei σ_k die Menge der Querschnitte der k. Mischung und es seien für ein System mit drei Zonen die Querschnitte $\sigma_1, \sigma_2, \dots, \sigma_5$ eingelesen. A_3, A_4 und A_5 seien Gewichte und γ ein Konzentrationswert. Die Mischungen sollen wie folgt auf die einzelnen Zonen verteilt sein:

Zone 1: σ_1

Zone 2: $\gamma\sigma_2 + A_3\sigma_3 + A_4\sigma_4$

Zone 3: $A_5\sigma_5$

Für dieses Beispiel läßt sich die Eingabe in K13 (mit MS = 6) folgendermaßen durchführen:

I	K(I)	n(I)	A(I)	dh:
1	0	6	0.0	$\sigma_6 := 0$
2	2	6	1.0	$\sigma_6 := \sigma_6 + \sigma_2$
3	1	6	0.0	$\sigma_6 := \sigma_6 \cdot \gamma$
4	3	6	A_3	$\sigma_6 := \sigma_6 + A_3\sigma_3$
5	4	6	A_4	$\sigma_6 := \sigma_6 + A_4\sigma_4$
6	0	5	A_5	$\sigma_5 := A_5 \cdot \sigma_5$

Im Falle IEVT = 3 muß mindestens eine Mischungsspezifikation vorhanden sein!

- b) Man kann die Auflösung der Winkelvariablen verschieden weit durchführen. Dabei muß beachtet werden, daß die diskreten Richtungen bei Zylindergeometrie von zwei Variablen abhängen. In diesem Falle sind diese als auf der Oberfläche einer Viertelskugel gegeben zu betrachten, wenn man sie als durch Einheitsvektoren in einer Einheitskugel vorgegeben annimmt. Zu jeder diskreten Richtung (Ordinate) gehört noch ein Gewicht. Das rührt daher, daß man aus dem Winkelströmen $\psi(x, \mu_m)$ u.a. den Skalarfluß $\phi(x)$ gemäß

$$\phi(x) = \sum_{m=1}^M w_m \psi(x, \mu_m)$$

berechnet. Die in DTK benutzten diskreten Richtungen und zugehörigen Gewichte beschafft sich das Programm selbst. Es sind derzeit die in [2], Band II, p. 136-138, gelisteten Größen. Es handelt sich dabei für S_N -Ordnung ISN bei Platten- und Kugelgeometrie um ISN+1 diskrete Richtungen, bei Zylindergeometrie um $\frac{ISN(ISN+4)}{4}$.

- c) Man kann Bucklings verschiedenster Art benutzen. Steuerung über MBK. Im Falle MBK = 1 wird ein schwach gruppen- und zonenabhängiges Buckling benutzt. Man verwendet die Höhe DY (und bei Plattengeometrie auch die Tiefe DZ), um als Bucklingkorrektor den zusätzlichen Absorptionsterm

$$SRI_g^1 = \frac{1}{3\sigma_{tr}^{gl}} \left\{ \frac{\pi}{2Z_0} + \frac{\sigma_{tr}^{gl}}{DY} \right\}^2$$

zu erhalten. Es ist g der Gruppenindex, l der Mischungsindex, Z_0 die Milnesche Extrapolationslänge und σ_{tr}^{gl} immer der NUSYS-Typ $\alpha STR\alpha$, falls es sich wie hier um die genäherte Berechnung von DB^2 in getrennter Form für die nicht berücksichtigten Ortsrichtungen handelt, wobei D immer $1/(3\sigma_{tr}^{gl})$ gesetzt wird. Hat man MBK = -1 oder -2, so hat man ein stark gruppenabhängiges bzw. ein gruppen- und zonenabhängiges Buckling der Gestalt

$$SRI_g^1 = \frac{1}{3\sigma_{tr}^{gl}} * A_{lg}$$

wobei für MBK = -1 die Größen A_{lg} nur von g abhängen, für MBK = -2 dagegen von l und g. Hat man MBK = -11, bzw. = -12, so ist A_{lg} wie zuvor, nur wird nicht durch $3\sigma_{tr}^{gl}$ dividiert. Dies entspricht einer zusammengefaßten Approximation von (DB^2) .

- d) Man kann anisotrope Streuung berücksichtigen. Der Code enthält die Möglichkeit, anisotrop beliebigen Grades zu rechnen. Dies ist möglich, wenn Streumomente genügend hoher Ordnung verfügbar sind, die es gestatten, eine Kugelfunktionsentwicklung des winkelabhängigen Streuquerschnitts zu verwenden.

Zur Zeit ist das Programm 00446 in NUSYS im Stande, das erste Streumoment unter dem Namen $\alpha SME1\alpha$ im Block $\alpha SIGMN\alpha$ zu erzeugen.

Abgesehen von den Eingabeänderungen ist noch folgendes zu beachten:

Im isotropen Transportfall benutzt der Code als

- σ_{tot}^g die Größe σ_{tr}^g ($\equiv \alpha\text{STR}\alpha$ oder $\equiv \alpha\text{STRTR}\alpha$)
- $\sigma_s^{g \rightarrow g}$ die Größe $\sigma_{\text{tr}}^g - \sigma_{\text{rem}}^g$ ($\sigma_{\text{rem}}^g \equiv \alpha\text{SREM}\alpha$)
- $\sigma_{s_{h \neq g}}^{g \rightarrow h}$ die Größe $\sigma_s^{g \rightarrow h}$ ($\equiv \alpha\text{SMT}\phi\text{T}\alpha$)

Im anisotropen Falle werden, wenn wir

$$\sigma_{s1}^g = \int_h \sigma_{s1}^{g \rightarrow h} \quad \text{nennen,}$$

wobei $\sigma_{s1}^{g \rightarrow h}$ die ersten Streumomente sind,

$$\begin{aligned} \sigma_{\text{tot}}^g &= \sigma_{\text{tr}}^g + \sigma_{s1}^g \quad \text{und} \\ \sigma_s^{g \rightarrow g} &= \sigma_{\text{tr}}^g - \sigma_{\text{rem}}^g + \sigma_{s1}^g \end{aligned}$$

gesetzt. Neben dem als σ_{tot}^g benutzten Wert wird vom Programm auch immer noch die in NUSYS $\alpha\text{STR}\alpha$ genannte Größe zur Berechnung der Bucklingkorrekturen im Speicher gehalten. Will man anisotrope Streuung höherer Ordnung rechnen, so müssen in einem $\alpha\text{SIGMN}\alpha$ -Block, aus NUSYS kommend, auch die Momente 2., 3., ... Ordnung unter den Typennamen $\alpha\text{SME}2\alpha$, $\alpha\text{SME}3\alpha$, ... vorhanden sein.

Die Programmeingabe ändert sich für den anisotropen Fall in folgender Weise:

- a) Auf der Eingabekarte K1 muß $\text{ISCT} \neq 0$ gesetzt werden.
- b) Die Eingabegrößen MT und MTP auf K2 sind verschieden. MTP ist die Anzahl der zu speichernden (nicht unbedingt benutzten) Mischungen, MT die Anzahl der zu reservierenden Tabellen; jede Mischung mit anisotropem Anteil verlangt $\text{ISCT}+1$ Tabellen, daher $\text{MT} > \text{MTP}$.
- c) Wenn $\text{ISCT} > 0$ ist, müssen die Zahlen $1, 2, \dots, \text{MTP}$ als Materialkennzahlen mit positivem oder negativem Vorzeichen als Karte

K5 eingelesen werden. Hat Mischung I, $1 \leq I \leq MTP$, einen anisotropen Anteil, so hat I in der obigen einzugebenden Folge negatives Vorzeichen, sonst positives.

- d) Die Eingabekarte K14 ist folgendermaßen zu interpretieren:
Gelesen werden die Materialzahlen $M(I)$, $I=1$, IZM (IZM=Anzahl der Zonen). $|M(I)|$ ist der Tabellenplatz, an dem das nullte Moment der Mischung gespeichert wird. Hat diese Mischung einen anisotropen Anteil, so muß $M(I)$ negativ sein. Dann wird das erste Moment dieser Mischung in der $(|M(I)|+1)$. Tabelle gespeichert.

Bemerkung:

Wenn ISCT=0 ist, ist i.a., wenn nicht IEVT=3 ist, $MT=MTP$. Auch im isotropen Fall, ISCT=0, kann $IZM > MTP$ oder $IZM < MTP$ sein. Der erste Fall kann eintreten, wenn man beispielsweise Mischung 1 in Zone 1 und 3, Mischung 2 in Zone 2 plaziert. Umgekehrt kann man, wenn $MTP=10$, $IZM=1$, wahlweise eine der zehn gespeicherten Mischungen in die eine vorgegebene Zone bringen.

Jedem vorhandenen Moment einer jeden Mischung wird eine Tabelle reserviert und zwar in folgender Reihenfolge:

Nulltes Moment Mischung 1
Erstes Moment Mischung 1, falls vorhanden
Zweites Moment Mischung 1, falls vorhanden
:
:
Nulltes Moment Mischung 2
Erstes Moment Mischung 2, falls vorhanden
usf.

Im Sinne dieser Anordnung versteht sich, welche Bedeutung Materialkennzahlen und Materialzahlen haben. Die Materialkennzahl gibt an, ob hinter der Tabelle für das nullte Moment einer Mischung noch weitere Tabellen bereit gehalten werden sollen.

Die Materialkennzahl für eine Zone (!) gibt den Tabellenplatz

des nullten Moments der Mischung in dieser Zone an. Ist diese Zahl negativ, so werden die dahinter stehenden Tabellen als höhere Streumomente für diese Mischung benutzt.

Ein Beispiel soll dies noch verdeutlichen. Es sei

ISCT = 1
IZM = 3 Zonenzahl
MT = 14 Gesamtzahl der Tabellen
MTP = 8 Anzahl der Mischungen

Aus den Zahlenwerten geht hervor, daß 6 der 8 gespeicherten Mischungen einen anisotropen Anteil haben. Dies seien die Mischungen 3 bis 8. Es soll Mischung 1 in Zone 1, Mischung 3 in Zone 2, Mischung 8 in Zone 3.

Die Materialkennzahlen sind dann, in formatloser Form geschrieben:

1 2 -3 -4 -5 -6 -7 -8.

Die Materialzahlen sind

1 -3 -13.

Noch eine Bemerkung zur Querschnittsbeschaffung. Bei der Beschaffung der Querschnitte muß in 446 in NUSYS zusätzlich der Typ α DIFK ϕ α erzeugt werden, wenn man anisotrop rechnen will. Dieser Typ muß dann auch in 2290 mit in den neuen α SIGMN α -Block geschrieben werden.

- e) Neben der Möglichkeit, eine punkt- und gruppenabhängige isotrope Fremdquelle vorzugeben, besteht auch die, in Plattengeometrie am rechten Rand den winkelabhängigen Fluß für die einfallenden Richtungen ($\mu < 0$) pro Gruppe anzugeben. Benutzt man diese Option, muß IEVT=0 sein, IBR=1 und IQM=1, -1 oder 2. IQM=1 oder -1 haben die frühere Bedeutung, IQM=2 bedeutet, daß neben dem vorgegebenen Rand-Winkelfluß keine Quelle vorgegeben wird. In diesem Falle dürfen nicht alle vorgegebenen Rand-Winkel Flüsse =0 sein. Setzt man EV nicht 1., so wird die Spaltquelle mit dem Faktor $\frac{1}{EV}$ vervielfacht.

Die Eingabe des Winkelflusses am Rande geschieht zunächst für die erste Gruppe, alle Richtungen mit $\mu < 0$. In diesem Falle gibt es $(\frac{ISN}{2} + 1)$ Richtungen pro Gruppe, die eingelesen werden müssen.

2. Technische Möglichkeiten

Zu den technischen Möglichkeiten, die das Programm bietet, sind folgende Bemerkungen zu machen.

- a) Es ist möglich, beliebig viele verschiedene Fälle hintereinander zu rechnen. Steuerung über K28 oder K29. Zwischen ihnen braucht kein Zusammenhang zu bestehen.
- b) Wie schon in der Einführung erwähnt, besteht die Möglichkeit, Flüsse weiterzugeben. Das Programm DTK ist im Stande, am Ende eines Falles, d.h. nach Beendigung einer Transportrechnung, den Fluß und den Eigenwert zusammen mit einigen identifizierenden Größen auf einen Zwischenspeicher zu schreiben, um ihn für

- a) einen weiteren Fall ("Folgefall") im selben Jobstep,
- b) einen weiteren Fall in einem anderen Jobstep im selben Job,
- c) einen anderen Fall in einem anderen Job

als Flußschätzung zu verwenden. Je nach Verwendungsmöglichkeit muß der Benutzer bei der Deklaration dieser Zwischenspeicher auf dem JCL-Level und dem Problemprogrammlevel die geeigneten Parameter setzen.

Dies sieht z.B. in der Job-Control-Language im Falle a) so aus, wenn der Zwischenspeicher die Unit-Nr. 25 haben soll:

```
//G.FT25FO01 DD UNIT=SYSDA, SPACE=(TRK,(4)).
```

Im Falle c), der Fluß soll auf das dem Benutzer zur Verfügung stehende Band DVO200 geschrieben werden:

```
//G.FT25FO01 DD UNIT=TAPE9, VOL=SER=DVO200,  
// DSN=FLUX, DISP=(,KEEP)
```


Bei der Eingabe zur Wiederverwendung ändert sich der DISP-Parameter auf DISP=(OLD,DELETE) oder DISP=OLD.

Die Weitergabesteuerung wird in der Programmbeschreibung über die Größen ITPF, ITP1, IMØV, ITP2 (K3) angegeben.

Zur Benutzung dieser Möglichkeit sind noch folgende Bemerkungen notwendig:

Will man als Flußschätzung den skalaren Fluß von einem früheren Fall übernehmen, so müssen beim zu rechnenden und beim früheren Fall nur die Gruppenzahl, die Anzahl der Zonen und die rechten Zonenrandpunkte übereinstimmen. Haben die beiden Fälle verschiedene Ortspunktzahlen, so nimmt das Programm eine Interpolation vor.

Die hier beschriebene Möglichkeit gestattet es,

- a) bei größeren Fällen einen oder mehrere Vorläufe mit kleinerer S_N -Ordnung, weniger Ortspunkten und geringerer Genauigkeit durchzuführen,
- b) bei umfangreichen Parameterstudien als Flußschätzung das Ergebnis eines Modellfalles zu verwenden.

c) Anschlußmöglichkeiten an NUSYS-Auswertephasen

Es besteht die Möglichkeit, am Ende eines Falles den regulären oder adjungierten (Skalar-)Fluß so auf einem Zwischenspeicher mit der Unit-Nr. IAUSW zu schreiben, daß er vom Programm 451 in einem späteren NUSYS-Jobstep oder -Job auf LILLI gebracht werden kann. Dies gestattet es, mit dem aus der Transportrechnung stammenden eindimensionalen Fluß in alle derzeitig verfügbaren eindimensionalen NUSYS-Auswertephasen zu gehen.

Auf Problemmprogrammlevel muß, um den Fluß gerettet zu erhalten, nur die Größe IAUSW (K3) in der Eingabe >0 angegeben werden.

In den JCL-Karten muß eine DD-Karte für die Einheit IAUSW mit DISP=(,PASS) oder (,KEEP) liegen.

War im gerechneten Fall ITH=0 (reguläre Rechnung), so wird der Fluß mit dem Label α FLUX1 α gespeichert, war ITH=1 (adjungierte Rechnung), mit dem Label α ADFL1 α .

Bemerkung: Benötigt man sowohl den regulären als auch den adjungierten Fluß, so muß man einmal regulär und einmal adjungiert rechnen; dabei ist jeweils auf zwei verschiedene Speichereinheiten der Skalarfluß abzulegen und später in zwei 451-Phasen in NUSYS mit dem geeigneten Label auf LILI zu bringen.

Es folgt die Eingabe für einen NUSYS-Jobstep, der einen von DTK erzeugten gespeicherten (regulären) Fluß auf LILI liest. Der DTK-Jobstep hieß Q, IAUSW war 30, der Eingabe des NUSYS-Jobsteps war neben den üblichen JCL-Karten die DD-Karte

```
//G.FT27FOO1 DD DSN=H.Q.G.FT30FOO1, DISP=OLD
```

beigegeben.

```
α00397α  
451 0 0 0 0  
αENDEα  
α00451α  
0 0 0 0  
27 1 αFLUX1α 'READ'  
0  
αNUFINα
```

Entsprechend besteht die Möglichkeit, Wirkungsquerschnitte nicht über die Karteneingabe dem Programm zur Verfügung zu stellen, sondern in einem NUSYS-Step erzeugte Wirkungsquerschnitte über einen Zwischenspeicher heranzuführen.

Ist die Größe ID (=Identifikationsnummer) in K1 positiv, so wird angenommen, daß die Querschnitte im Eingabekartenpaket liegen. Ist ID negativ, so werden die Querschnitte von einem Zwischenspeicher übernommen.

Die Job-Control-Language-Eingabe für den DTK-Aufruf muß ein DD Statement für die Einheit IINP enthalten, die sich auf jene Einheit bezieht, die 451 beschrieben hat.

Beispiel:

Der erste Job-Step (sein Name sei PROG gewesen) enthält das DD Statement

```
//G.FTO9FOO1 DD DSN =&BAND9, DISP = (NEW, PASS),...  
//          UNIT=SYSDA, SPACE=(TRK,(100)),
```

d.h. 451 schreibt auf Einheit 9. Wenn dann IINP = 13 sein soll, muß der zweite Job-Step das DD Statement

```
//G.FT13FOO1 DD DSN & * PROG.G.FTO9FOO1, DISP = (OLD, DELETE)  
enthalten.
```

Es folgt in Anhang 2 die vollständige Eingabe für einen Job, der die Wirkungsquerschnitte in einen NUSYS-Lauf auf Zwischenspeicher schreibt.

- d) Um Rechenzeit zu sparen, wurde für die äußeren Iterationen die Möglichkeit geschaffen, Tschebyscheff-Extrapolation anwenden zu können. Sie ist nur bei inhomogenen und k_{eff} -Rechnungen (IEVT=0 oder 1) anwendbar. Wenn wir bei der folgenden Betrachtung das inhomogene Problem zugrunde legen, läßt sich, kurz gefaßt, folgendes sagen:

Ist x_N der Zustandsvektor der N. Iteration, wenn man ohne Beschleunigung das System $x = Ax + b$ iterativ durch $x_{N+1} = Ax_N + b$ löst, so wird eine neue Vektorfolge $\{y_N\}$, $N=1,2,\dots$ eingeführt, für die auch $\lim_{N \rightarrow \infty} y_N = \lim_{N \rightarrow \infty} x_N$, und für die, wenn bestimmte optimale Parameter bekannt sind,

$$y_{N+1} = \alpha_N A y_N + \beta_N y_{N-1}, \quad N=1,2,\dots$$

In α_N und β_N geht der Spektralradius von A, $\rho(A)$ ein. Da dieser nicht bekannt ist, muß er am Anfang der Rechnung angenähert berechnet werden. Rechnet man zu Anfang nach $x_{N+1} = Ax_N + b$, so gilt mit bestimmten Vektor- und Matrizennormen

$$\|x_{N+1} - x_N\| \leq \|A\| \|x_N - x_{N-1}\|$$

Es ist bekannt, daß man asymptotisch für $N \rightarrow \infty$ statt $\|A\|$ auch $\rho(A)$ setzen kann. Es liegt daher nahe, die Folge

$$\rho_i = \frac{\|x_{i+2} - x_{i+1}\|}{\|x_{i+1} - x_i\|}, \quad i=1,2,\dots,$$

als Folge von Schätzungen anzusehen, die gegen $\rho(A)$ konvergiert und aus der man einen angenäherten Wert für $\rho(A)$ herleiten kann. Falls man $\rho(A)$ genau oder angenähert kennt, kann man es als Eingabegröße RHØ in Karte K7 eingeben; in diesem Falle wird der oben angeführte Schätzvorgang nicht durchgeführt.

Wendet man diese Beschleunigung an (ITSCH=1), so benötigt das Programm dazu zur Zeit fest die Einheiten 20 und 21. Für diese Einheiten müssen JCL-Karten angegeben werden.

Der Rechenzeitgewinn hängt von mehreren Ursachen ab. Die eine ist zweifelsohne die, wie genau $\rho(A)$ geschätzt wurde. Weiter gilt auch die für viele Beschleunigungsverfahren zutreffende Tatsache, daß der Gewinn um so größer ist, je näher $\rho(A)$ bei 1. liegt.

3. Aufruf des Programmes

```
// EXEC FHG,LIB=NUSYS,NAME=DTK,REGION,G=300K,  
// TIME.G=...  
.....  
//G.SYSIN DD*
```

4. Bemerkungen zur Eingabe

Zunächst zu den Maßnahmen, die man auf Betriebssystemlevel zu treffen hat. Falls die Querschnitte nicht über die Karteneingabe eingegeben werden, muß eine DD-Karte für die Einheit IINP (vgl. K3) vorhanden sein.

Desgleichen müssen u.U. auf JCL-Niveau DD-Karten für folgende Zwischenspeicher deklariert werden:

- a) für Einheit 6, falls deren SPACE-Parameter überschrieben werden soll.
- b) für weitere Querschnittfiles, wenn in nacheinanderfolgenden Fällen verschiedene Querschnittseingabefiles benützt werden.

- c) für Einheiten 20 und 21, wenn Tschebyscheff-Extrapolation benutzt wird (vgl. 1,D). Diese Einheiten sind fest einprogrammiert.
- d) für die Einheiten ITP1 und/oder ITP2, falls Flüsse von Fall zu Fall weitergegeben werden (vgl. 1,E).
- e) für die Einheit(en) IAUSW, falls Flüsse zur Auswertung an NUSYS weitergegeben werden (vgl. 1,F).

Die Eingabegrößen, für die gegebenenfalls eine DD-Karte benötigt wird, sind in der Eingabebeschriftung durch einen geklammerten Stern gekennzeichnet.

Von den Zwischenspeichern können manche gleich sein, einige müssen verschieden sein.

- a) ITP1 und ITP2 müssen pro Fall verschieden sein.
- b) Falls IINP nicht mehr gelesen wird, kann es anderweitig verwendet werden als ITP2 oder IAUSW. IINP kann daher auch 20 oder 21 sein.
- c) 20 oder 21 können auch IAUSW sein.

Der Master-Array hat die Länge von 35 000 Worten und ist vom Typ REAL. Die von diesem Array in Anspruch genommene Anzahl von Worten wird am Anfang der Ausgabe ausgedruckt in der Form

FLÖATING PÖINT STÖRAGE = I1111.

Daher muß I1111 \leq 35 000 sein. Da jedoch die Anschlußprogramme zur Querschnittsbeschaffung und gegebenenfalls Flußweitergabe u.a. im Bereich zwischen I1111 und 35 000 arbeiten, darf auch 35 000 - I1111 nicht zu klein sein.

Falls die Arraylänge von 35 000 Worten, die REGION.G = 300K erlaubt, aus irgendeinem Grund nicht ausreicht, läßt sich die Länge des Hauptarray auf folgende System- und Assembler-unabhängige Weise vergrößern. Man führt einen CLG-Lauf durch nach folgendem Schema: (hier für 450 K)

```
// EXEC FHCLG,REGION.G=450K /',TIME.G=..._7
//C.SYSIN DD #
C
      SUBROUTINE DUMMY
C
C      THIS ROUTINE DEFINES THE SIZE OF THE
```

```
C      MASTER ARRAY FØR 450K
C
      CØMMØN LAENGE, A(72000)
      CØMMØN/FIX/LEN, IA(1000)
      LAENGE=72000
      LEN=1000
      RETURN
      END
```

//L.SYSIN DD UNIT = (wie zuvor)

Zur Eingabe auf Problemprogrammlevel ist noch zu vermerken:

- a) Die Namen der Eingabegrößen entsprechen den FØRTRAN-Typenkonventionen.
- b) Die Eingabe ist formatgebunden. Die Eingabedaten werden, abgesehen von den weiter unten aufgeführten Ausnahmen, als REAL- oder INTEGER-Datensätze eingelesen, REAL-Datensätze mit dem FØRTRAN-Format 6E12.5, INTEGER-Sätze mit 12I6. Dies gestattet, daß der Benutzer zumindest Spalte 73 bis 80 für eigene Zwecke zur Verfügung hat. Ausnahmen im Sinne dieser Regelung sind drei Datensätze:
 - 1) Der Informationstext auf Kart KO, der mit Format 20A4 gelesen und wieder ausgeschrieben wird.
 - 2) Die jeweilige Schlußkarte eines Falles, von der die ersten fünf Spalten numerisch (I5) gelesen und geprüft werden, ob sie mit 99999 übereinstimmen.
 - 3) Die Karte K13, bei der die Eingabedatensätze vom gemischten Typus sind! (3I6,E12.5).
- c) Falls die Querschnitte über Karten eingegeben werden, werden sie in einem Satz gelesen mit der FØRTRAN-Statementfolge

```
DØ 1 IMT=1,MT
```

```
1 READ(.....)((CS(I,J,IMT),I=1,IHM),J=1,IGM)
```

Es ist MT die Anzahl der Mischungstabellen,

IGM die Gruppenzahl

IHM die Tabellenlänge.

Falls keine Aufwärtsstreuung vorhanden ist, ist $IHM=IGM+5$, wird maximal IUP Gruppen hochgestreut, so hat IHM den Wert $IGM+IUP+6$. Die Querschnitte werden für jedes Material und jede Gruppe (Index) in folgender

Reihenfolge eingelesen.

Spalte 1:	σ_{tr}^g (flußgewichtet	Spalte IHS = IUP+7 :	$\sigma_s^{g \rightarrow g}$
Spalte 2:	σ_f^g	IHS+1 :	$\sigma_s^{(g-1) \rightarrow g}$
Spalte 3:	$\sigma_{abs}^g = s_{cap}^g + s_f^g$.	.
Spalte 4:	$v\sigma_f^g$.	.
Spalte 5:	σ_{tr}^g	.	.
Spalte 6:	σ_{up}^g	Falls keine Aufwärtsstreuung vorhanden, entfallen Spalte 6 bis Spalte IHS-1	
Spalte 7:	$\sigma_s^{g+IUP \rightarrow g}$		
	⋮		

5. Eingabe

K0		Informationstext (20A4)
K1	ID	Fallidentifikationszahl, wenn ≤ 0 Querschnittseingabe von Band IINP, sonst über K26
	ITH	0 { regulär 1 { adjungiert
	ISCT	0 { isotrop N<0 { anisotrop N. Ordnung (im Augenblick $N \leq 1$)
	ISN	SN-Ordnung für { Platte, Kugel 2,4,6,8,12,16 } erlaubt { Zylinder 2,4,6,8 }
	IGE	Geometrie: 1 Platte 2 Zylinder 3 Kugel

- IBL Linke Randbedingung
- 0 Vakuum
 - 1 Reflektierend
 - 2 Periodisch
- IBR Rechte Randbedingung
- 0 } wie bei IBL
 - 1 }
 - 2 }
- 1 Einfallender Winkelfluß am rechten Rand vorgegeben, bei Plattengeometrie
Zugelassen sind die Kombinationen:
- | IBL | IBR |
|-----|-----|
| 0 | 0 |
| 1 | 0 |
| 1 | 1 |
| 2 | 2 |
| 0 | -1 |
- IZM Anzahl der Zonen (die Zonen werden von links durchnummeriert)
- IEVT Eigenwert:
- 0 Fremdquelle oder (und) am rechten Rand einfallender Winkelfluß vorgegeben.
 - 1 k_{eff} -Rechnung
 - 2 Alpha-Rechnung
 - 3 Konzentrations-Suche
 - 4 Delta-Rechnung
 - 5 Radieniteration
- IGM Anzahl der Gruppen
- MS Anzahl der Mischungsspezifikationen
- MBK Bucklingkennzahl:
- 0 kein Buckling
 - +1 gruppenunabhängiges Buckling
 - 1 gruppenabhängiges Buckling
 - 2 gruppen- und zonenabhängiges Buckling

-11 wie -1: es wird nicht durch $3 \cdot \sigma_{tr}$
dividiert

-12 wie -2: es wird nicht durch $3 \cdot \sigma_{tr}$
dividiert

K2	MT	Gesamtzahl der "Tabellen" (vgl. Abschnitt A)
	MTP	Anzahl der zu speichernden (nicht unbedingt benutzten) Mischungen.
	MUP	Aufwärtsstreuung 0 nein 1 ja
	IQM	Fremdquelle, einzugeben 0 nein 1 ja, punkt- und gruppenweise -1 ja, zonen- und gruppenweise 2 nein, wenn IEVT=0, IBR=1.
	IM	Anzahl der Intervalle
	IPVT	Parametrischer Eigenwerttyp (vgl. Abschnitt 1)
	ICM	Maximalzahl äußerer Iterationen
	IIM	Maximalzahl innerer Iterationen
	IIL	Maximalzahl innerer Iterationen zu Beginn der Rechnung
	ID1	{ 1 Ausdruck der Winkel Flüsse 0 kein Ausdruck der Winkel Flüsse
	ID2	{ 1 Ausdruck der Neutronenbilanzen, gruppen- und zonenweise 0 kein Ausdruck der Neutronenbilanzen
	NFF	{ 1 kein Ausdruck der negativen Fluß- korrektur 0 Ausdruck der negativen Flußkorrektur
K3	IINF ^(M)	Unit-Nr. des Querschnittsfiles, falls ID <u>≤</u> 0, sonst beliebig.

ITSCH	$\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right.$	$\left\{ \begin{array}{l} \text{Tschebyscheff-Extrapolation wird nicht} \\ \text{durchgeföhrt} \\ \text{Tschebyscheff-Extrapolation wird durch-} \\ \text{geföhrt} \end{array} \right.$
ITPF	$\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right.$	$\left\{ \begin{array}{l} \text{Quellschätzung wird vom Programm durch-} \\ \text{geföhrt} \\ \text{Skalarflußschätzung von einem früheren} \\ \text{fall }^{(*)}$
ITP1 ^(**)		Unit-Nr. der Einheit, auf der als Schätzung verwendete Fluß steht, falls ITPF=1, sonst ohne Bedeutung
IMØV	$\left\{ \begin{array}{l} 0 \\ 1 \end{array} \right.$	$\left\{ \begin{array}{l} \text{ohne Wirkung} \\ \text{Der Skalarfluß wird am Ende des Falles} \\ \text{auf einen Zwischenspeicher geschrieben} \end{array} \right.$
ITP2 ^(**)		Unit-Nr. der Einheit, auf die der Skalarfluß geschrieben wird, falls IMØV=1, sonst ohne Bedeutung
IAUSW ^(**)	$\begin{array}{l} \leq 0 \\ > 0 \end{array}$	$\begin{array}{l} \text{ohne Bedeutung} \\ \text{Unit-Nr. der Einheit, auf die der} \\ \text{Skalarfluß zwecks späterer Auswertung} \\ \text{geschrieben werden soll. (vgl. 2c)} \end{array}$
S4		wenn ISCT>0 K5, sonst K6
K5	(±)I, I=1, MTP	Materialkennzahlen (vgl. 1,d)
K6	EV	a) für IEVT \geq 2: Schätzung für den gesuchten Parameter b) IEVT = 1: bedeutungslos c) IEVT = 0 und IBR.NE.2: bedeutungslos d) IEVT = 0 und IBR.EQ.2: Spaltquelle wird mit 1./EV multipliziert.
EPS		Genauigkeitsfaktor für k_{eff}
BF		Bucklingfaktor, wenn MBK=1, sonst bedeutungslos

	DY	Bucklinghöhe (in cm) für Platte und Zylinder, wenn $MBK > 0$, sonst ohne Bedeutung
	DZ	Bucklingbreite (in cm) für Plattengeometrie, wenn $MBK > 0$, sonst ohne Bedeutung
	PV	parametrischer Eigenwert a) für $IPVT=0$ ohne Bedeutung b) für $IPVT=1$ das gewünschte k_{eff} zum gesuchten Parameter c) für $IPVT=2$ wird das System mit dem Eigenwert EV so bestimmt, daß es α) für $IEVT=2$ mit der zusätzlichen Absorption PV (d.h. in jeder Gruppe PV/v_g) kritisch wird. β) für $IEVT=2$ mit der zusätzlichen Absorption ($EV+PV$) kritisch wird.
K7	EPSA	≤ 0 . ohne Bedeutung > 0 . Genauigkeitsabfrage für den Fluß
	RH \emptyset	≤ 0 . ohne Bedeutung > 0 . geschätzter oder bekannter Konvergenzfaktor für Tschebyscheff-Extrapolation, falls $ITSCH=1$
S8		falls $MUP > 0$ K9 sonst K10
K9	IJS	Die größte Differenz ($i-j$), für die der Streuquerschnitt σ_{i+j} mit $i > j$ für irgendeine der einzugebenden Tabellen nicht verschwindet.
K10	$R(I), I=1, IZM$	$R(I)$ = rechter Randpunkt der I. Zone
K11	$N(I), I=1, IZM$	$N(I)$ = Anzahl der (als gleich groß vorausgesetzten) Intervalle in der I. Zone ($\sum N(I) = IM!$)
S12		falls $MS > 0$ K13, sonst K14

- K13 (I,K(I),L(I), für jede der MS M schungsspezifikationen
A(I)),I=1,MS eine Karte. Wenn σ_m (m=1,...,MTP) ein Quer-
(3I6,E12.5) schnitt (auch Moment höherer Ordnung) ist,
gilt:
$$K(I) = 0 \quad \sigma_n = A(I) \cdot \sigma_n, \quad n=L(I)$$

falls $K(I) = J, A(I) \neq 0: \sigma_n = \sigma_n + A(I) \cdot \sigma_j, \quad j=J$
$$K(I) = 0, A(I) \neq 0: \sigma_n = \sigma_n \cdot EV$$

(vgl. dazu Abschnitt 1a)
- K14 M(I),I=,IZM M(I), Nummer der Mischung in der I. Zone. Sind
für diese Mischung höhere Momente vorhanden
(ISCT>0), so ist M(I) negativ und |M(I)| ist
der Platz des nullten Moments. (vgl. Abschnitt
1d)
- S15 falls IEVT=4 K16, sonst S17
- K16 B(I),I=1,IZM Zonenparameter (vgl. Abschnitt 1)
- S17 falls IQM<0 K18, dann S20
falls IQM=0 S20
falls IQM>0 K19, dann S20
- K18 A(I),I=1,IZM:IGM Fremdquelle pro Zone und Gruppe,
zunächst für Gruppe 1 alle Zonen, usw.
- K19 A(I),I=1,IM:IGM Fremdquelle pro Radius (Intervall) und Gruppe,
zunächst für Gruppe 1 alle Zonen, usw.
- S20 falls ID>0 K21, dann K22 und S23,
andernfalls sofort S23
- K21 (XKI(I),I=1,IGM)Spaltspektrum
- K22 (VE(I),I=1,IGM) mittlere Gruppengeschwindigkeit
- S23 falls MBK = 0, S26
= 1, S26
= -1, K24
= -2, K25
= -11, K24
= -12, K25
}, dann S26

- K24 BK(I), I=1, IGM gruppenabhängiges Buckling; es wird in jeder Gruppe I die Bucklingkorrektur
- $$B_I = BK(I) \cdot C_I$$
- benutzt. Wenn MBK=-1, ist $C_I = \frac{1}{3\sigma_{tr}}$.
Wenn MBK=11, ist $C_I = 1$.
 σ_{tr} ist hier in jedem Fall der NUSYS-Querschnittstyp $\alpha STR \alpha$.
- K25 ((BK(I,J), J=1, IZM), I=1, IGM gruppen- und zonenabhängiges Buckling; es wird in jeder Gruppe I und Zone J die Korrektur
- $$B_{IJ} = BK(I,J) \cdot C_{IJ}$$
- benutzt. Wenn MBK=-2, ist $C_{IJ} = \frac{1}{3\sigma_{tr}}$.
Falls MBK=-12, ist $C_{IJ} = 1$.
- S26 falls IBR=-1 (und IGE=1) K27 und S28,
sonst sofort S28
- K27 ((A(I,J), I=1, M), J=1, IGM) Es ist $M = \frac{ISN}{2} + 1$. A(I,J) ist der Winkelfluß am rechten Rand in der J. Gruppe für die M einwärts gerichteten Richtungen. Sie werden nach aufsteigenden Richtungscosinus geordnet. Die erste Richtung ist demnach $\mu = -1$.
- S28 falls ID>0 K29 und S30, andernfalls S30
- K29 Eingabe der Wirkungsquerschnitte (vgl. 4c)
- S30 falls keine weiteren Fälle kommen, K32,
sonst K31 und wieder K0.
- K31 linksbündig (I5) fünf numerische Zeichen
‡ 99999
- K32 99999 Konstante

6. Ausführungen zur Ausgabe

Die folgenden Bemerkungen haben den Zweck, Erläuterungen zur Ausgabe zu machen. Für einige Größen werden die Berechnungsformeln angegeben.

a) Zu den Querschnitten

Für jede Tabelle (im isotropen Falle heißt das: für jede Mischung) und für jede Gruppe g werden die Querschnitte in folgender Reihenfolge ausgedruckt (-dabei wird noch angenommen, daß keine Aufwärtsstreuung vorhanden ist):

<u>Spalte</u>	<u>Name</u>	<u>Kurzbezeichnung</u>	<u>NUSYS-Bezeichnung</u>
1	stromgewichteter Transportquerschnitt	σ_{tr}^g	$\alpha STR\alpha$
2	Spaltquerschnitt	σ_f^g	$\alpha SFISS\alpha$
3	Absorptionsquerschnitt	σ_{abs}^g	$\alpha SFISS\alpha + \alpha SCAPT\alpha$
4		$(v\sigma_f)^g$	$\alpha NUSF\alpha$
5	totaler Wirkungsquerschnitt	σ_{tot}^g	$\alpha STR\alpha$ oder $\alpha STRTR\alpha$
6	"selfscatter"-Querschnitt	$\sigma_s^{g \rightarrow g}$	(vgl. Abschnitt 1,A)
7	Streuquerschnitt	$\sigma_s^{(g-1) \rightarrow g}$	SMTØT
8			
	.	.	
	.	.	
	.	.	

Im Falle anisotroper Streuung wird in Spalte 5 noch eine Korrektur durchgeführt (vgl. Abschnitt 1,d).

b) Zu den wahlweisen Ausgaben:

Falls es erwünscht ist, kann man über die Eingabegrößen ID1, ID2 und NFF ausgeben:

- a) die Winkelflüsse
- b) die zonen- und gruppenabhängigen Teilchenbilanzen
- c) Information über Korrektur negativer Flüsse.

Zu a): Hat man von der Möglichkeit Gebrauch gemacht, den Winkelfluß am rechten Rand vorzugeben (IBR=-1), so werden die vorgegebenen Winkelflußwerte hier mit ausgedruckt.

Zu b): Die verwendeten Formeln werden in Abschnitt E angegeben.

c) Ortskoordinaten:

Die theoretische Formulierung der S_N -Differenzengleichung hat zur Folge, daß die meisten physikalischen Größen in der Mitte der Koordinatenintervalle berechnet werden. Da jedoch einige Größen auch an Intervallrändern von Interesse sind, so z.B. der Winkelfluß zur Berechnung der Zonen- oder Gesamtleckage, werden diese Größen für die Ortsintervallgrenzen ausgegeben.

d) Monitorausdruck:

Der Monitorausdruck gibt u.a. pro äußerer Iteration an:

- a) die bisher benötigte Zeit des "Falles"
- b) die Gesamtzahl der bisher durchgeführten äußeren und inneren Iterationen
- c) die Größe "EIGENVALUE". Sie ist im Falle

IEVT=0 die Größe, durch die die Spaltquelle dividiert wird

IEVT=1 das derzeitige k_{eff}

IEVT>2 der derzeitige Wert des gesuchten Parameters.

d) die Größe "EIGENVALUE SLOPE":

Sie gibt die vom Programm zuletzt geschätzte oder berechnete Veränderung von "EIGENVALUE" bei IEVT>2 an:

e) die Größe "LAMBDA":

für IEVT=0,1 steht hier das Verhältnis von neuer zu alter (Spalt)Quelle. Der Abstand dieser Größe von 1. ist in diesen beiden Fällen ein Maß für die Güte der bisher erfolgten Konvergenz. Wenn IEVT=1, ist zusätzlich

$$k_{\text{eff}}^{(u)} = \text{LAMBDA} * k_{\text{eff}}^{(n-1)}, \quad u=1,2,\dots$$

für IEVT>2 steht in der LAMBDA-Spalte für festen Parameter "EIGENVALUE" die Folge der dazu gehörigen k_{eff} .

f) die Größe "RED.FACTØR" ist ein vom Programm ermittelter Schätzwert für die in Abschnitt 2,d ρ genannte Variable. Sobald Tschebyscheff-Extrapolation begonnen wurde (= das Programm druckt dazu einen Vermerk aus -), hat diese Größe keine Bedeutung mehr.

- g) Es folgt noch ein Ausdruck, der zu jeder Gruppe ausdrückt
- α) Nummer der Energiegruppe
 - β) Anzahl der inneren Iterationen in dieser Gruppe während der letzten äußeren Iteration
 - γ) der zu der letzten dieser inneren Iterationen gehörige Rebalancingfaktor.

e) Die Bilanzenausdrücke:

In jedem Falle werden die gruppenweisen Teilchenbilanzen ausgegeben. Falls ID2=1, werden (vgl. b) auch zonen- und gruppenabhängige Teilchenbilanzen ausgedrückt. Die in diesen Tabellen genannten Größen werden nach folgenden nachstehenden Formeln berechnet.

Es sei definiert:

g	der jeweilige Gruppenindex ($1 \leq g \leq IGM$)
i	der Ortskoordinatenindex ($1 \leq i \leq IM$)
m	der laufende Index der diskreten Richtungen ($1 \leq m \leq ISN$) bei Zylindergeometrie: ($1 \leq m \leq ISN \cdot (ISN+4) / 4$)
x_i	Intervallmittelpunkt
\bar{x}_i	Intervallrandpunkt, \bar{x}_O = Linker Systemrandpunkt, \bar{x}_M = rechter Systemrandpunkt
σ^g	Querschnittstyp der Gruppe g
μ_m	diskrete Richtung
w_m	zur Richtung μ_m gehörendes Gewicht
$\phi_g(x_i)$	Skalarfluß der Gruppe g im Intervall i
$\psi_g(x_i, \mu_m)$ oder $\psi_g(\bar{x}_i, \mu_m)$	Winkelfluß in Gruppe g im Innern oder am Rande von Intervall i
$Q_g(x_i)$	Fremdquelle im Intervall i
$V(x_i)$	Volumenelement des i . Intervalls
$A(\bar{x}_i)$	Flächenelement am Rande des i . Intervalls
X_g	Spaltspektrum

v_g mittlere Gruppengeschwindigkeit
 α Absorptionskonstante beim Ansatz $\phi_g(x,t) = e^{-\alpha t} \phi_g(x)$
 BK(...) gruppenabhängige oder gruppen- und zonenabhängige
 Bucklingkorrektur

Zunächst ist für alle Intervalle und Gruppen

$$\phi_g(x_i) = \sum_{m=1}^{ISN} w_m \psi_g(x_i, \mu_m)$$

Wird der Neutronenstrom $S_g(x_i)$ ausgedrückt, so wird er berechnet nach

$$S_g(x_i) = \sum_{m=1}^{ISN} w_m \mu_m \psi_g(x_i, \mu_m)$$

Dann ist (isotrop für eine Gruppe g und das ganze System formuliert)

"SOURCE" $= \sum_i Q_g(x_i) V(x_i)$

"FISSION SOURCE" $= \sum_i V(x_i) \chi_g \left\{ \sum_{h=1}^{IGM} (v\sigma_f)^h \cdot \phi_h(x_i) \right\} \cdot \frac{1}{EV}$

"INSCATTER" $= \sum_i V(x_i) \left\{ \sum_{h=g} \sigma_{s,o}^{h+g}(x_i) \phi_h(x_i) \right\}$

"SELFSCATTER" $= \sum_i V(x_i) \sigma_s^{g+g}(x_i) \phi_g(x_i)$

"OUTSCATTER" $= \sum_i V(x_i) \left\{ \sum_{h=g} \sigma_{s,o}^g(x_i) \phi_g(x_i) \right\}$

"NET LEAKAGE" $= -A(\bar{x}_0) \sum_m w_m \mu_m \psi_g(\bar{x}_0, \mu_m) + A(\bar{x}_M) \sum_m w_m \mu_m \psi_g(\bar{x}_M, \mu_m)$

"ABSORPTION" $= \sum_i \sigma_{abs}^g(x_i) V(x_i) \phi_g(x_i)$

"NEUTRON BALANCE" $= 1 - \frac{\text{"OUTSCATTER"} + \text{"NET LEAKAGE"} + \text{"ABSORPTION"}}{\text{"SOURCE"} + \text{"FISSION SOURCE"} + \text{"INSCATTER"}}$

"RIGHT LEAKAGE" $= A(\bar{x}_M) \cdot \sum_m \mu_m \psi_g(\bar{x}_M, \mu_m) \cdot w_m$

"FISSION NEUTRONS" $= \sum_i V(x_i) (v\sigma_f)^g(x_i) \phi_g(x_i)$
 (Herkunft)

<u>"NEUTRON DENSITY"</u> (falls IEVT=2)	$= \frac{1}{v_g} \sum_i V(x_i) \phi_g(x_i)$
<u>"NEUTRON FLUX"</u>	$= \sum_i V(x_i) \phi_g(x_i)$
<u>"RIGHT FLUX"</u>	$= \sum_m w_m \psi_g(\bar{x}_M, \mu_m)$
<u>"RIGHT CURRENT"</u>	$= \sum_m w_m \mu_m \psi_g(\bar{x}_M, \mu_m)$
<u>"RIGHT FLOW"</u>	$= \sum_{m, \mu_m > 0} w_m \mu_m \psi_g(\bar{x}_M, \mu_m)$
<u>"BUCKLING LEAKAGE"</u>	$= \sum_i BK(\dots) V(x_i) \phi_g(x_i)$
<u>"TIME ABSORPTION"</u>	$= \sum_i \frac{\alpha}{v_g} V(x_i) \phi_g(x_i)$

Literatur

[17

K.D. Lathrop, DTF-IV, a FORTRAN-IV-Program for Solving the Multigroup Transport Equation with Anisotropic Scattering, LA-3373, 1965

[27

B.G. Carlson, W.J. Worlton, W. Guber and M. Shapiro, DTF Users Manual UNC Phys/Math. 3321, 1963

Anhang 1

Beispiel einer Karteneingabe für DTK

```
// EXEC FHG,LIB=NUSYS,NAME=DTK,REGION.G=300K
//G.FT06FO01 DD SPACE=(TRK,(100))
//G.FT14FO01 DD UNIT=SYSDA,SPACE=(TRK,(10))
//G.FT15FO01 DD UNIT=SYSDA,SPACE=(TRK,(10))
//G.FT16FO01 DD UNIT=SYSDA,DSN=&SCHARMER,DISP=(OLD,PASS)
//G.SYS IN DD *
```

*****	THIEM	KNK II	*****									
3191	0 6	1	0	0	26	1	13	16	27	50		
7	7 0	0	11	0	1	21	22	23	24			
1	0 0	0						1				
0.000006	1.							2.E-6	0.004			
1	1 1	1	1	1	1	1	1	1	1	1	1	1
1												
1	1 1	1	1	1	1	1	1	1	1	1	1	1
1	1 1	1	1									
6.0283	6.8155	18.0322		21.5525	23.1633		25.9074					
27.2620	29.7081	40.3211		53.2309	66.3427		80.48					
101.5												
7.5359	38.2307	50.2378		92.657	112.0833		120.862					
126.2199	141.3158	161.3953		182.5085	192.6418		222.1953					
232.1415	242.2748	284.6883		288.1231								
6	6 6	6	6	6	6	6	6	6	6	6	6	6
6	7 7	7	7	7	7	7	7	7	7	7	7	7
8	8 6	6	6	6	6	6	6	6	6	6	6	6
6	26 27	5	8	5	14	14	14	14	14	14	14	14
14	14 26	27	4	8	4	14	14	14	14	14	14	14
14	14 14	26	27	4	8	4	13	14	14	14	14	14
13	13 14	14	26	27	4	8	4	13	16	8		
16	13 13	14	14	26	27	4	8	4	13	16	8	
8	16 13	13	14	25	26	27	2	8	9	11		
16	8 16	11	11	24	25	26	27	8	8	10		
12	16 8	16	12	21	24	25	26	27	8	8		
10	12 16	20	16	12	21	24	25	26	27	8		
8	10 12	16	19	16	12	21	24	25	26	27		
2	8 9	11	16	18	16	11	11	24	25	26		
27	2 8	9	11	16	18	16	11	11	23	25		
26	27 1	8	1	1	15	18	15	1	1	22		
22	26 27	1	8	1	1	15	17	15	1	1		
22	22 26	27										

99999

Anhang 2

Beispiel einer Karteneingabe für einen NUSYS-Step,
der Wirkungsquerschnitte auf die Einheit 16 schreibt

```
// EXEC FHG,LIB=NUSYS,NAME=ANUSY,REGION.G=300K,PARM.G=150000,  
//          TIME.G=(,20)  
//G.FT08F001 DD UNIT=SYSDA,SPACE=(TRK,(10))  
//G.FT04F001 DD DSN=GRUCCD,UNIT=2314,VOL=SER=NUSYSO,DISP=(SHR)  
//G.FT16F001 DD UNIT=SYSDA,DSN=&SCHARMER,SPACE=(TRK,(10)),DISP=(,PASS)  
//G.FT12F001 DD UNIT=SYSDA,SPACE=(TRK,(5))  
//G.SYSIN DD *  
@00397@  
446 0 0 0  
@ENDE@  
@00446@  
352 0 0 1 446  
@SIGMA@ 446  
0 0 0 446  
@KOMPO@  
@26-GR.MOXTOT001@ 0 26 2 1 11 446  
@STRTR@ 446  
@NI590U2350U2380M0960PU390PU400PU410PU4200 160AL270XY990@ 446  
2 2 .007695 3 0.039855 446  
2 2 0.0003366 3 .04775 446  
@WQFIN@  
@00352@  
2290 0 0 2  
@SIGMASPEKT@  
@ST352@  
26 2 0 1 0 1 0  
@SPEKT@  
.055 .31 .797 2.3 3.3 3.91 2.63 2.58 1.51 .421 .353 .1 .135 8.87-2  
5.82-2  
3.88-2 2.77-2 1.75-2 1.07-2 5.76-3 2.29-3 1.47-3 6.34-4 2.22-4 4.95-5  
8.12-6  
.105 .608 1.36 2.4 2.26 2.55 2.21 1.5 .833 .324 .15 .041 .0218 9.82-3  
3.96-3  
1.7-3 9.43-4 2.96-4 3.1-4 8.02-5 5.78-5 4.35-5 6.43-6 6.35-7 2.01-8  
4.8-9  
@CDKNT@  
2 16 26 0  
@ENDE@  
@02290@  
451 1 2 0 0  
2 1 2  
@SIGMA@ 9 @CHIA@ @NUSFA@ @SCAPT@ @SFISS@ @SMTOTA@ @SREMA@ @STR@ @1/V@  
@STRTR@  
@00451@  
0 0 0 0  
16 1 @SIGMNA@ @WRITA@  
1  
@NUFIN@
```

Anhang 3

Programmliste

C-----
 C-----
 C

```

COMMON LAENGE,A(200)
COMMON /FIX/LEN,IA(200)
EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
CALL FSPIE
CALL DUMMY
3 DO 111 I=1,LAENGE
111 A(I)=0.
DO 11 I=1,LEN
11 IA(I)=0
KRD=5
KWT=6
WRITE(KWT,26)
26 FORMAT(1H1,39X,16H DTK-VERSION VOM //          90H
1      10.02.1971  AUF DVO102
2
CALL FREESP(I)
6 FORMAT(//'0 FREIER KERNSPEICHER ZU BEGINN : ',I3,' K BYTES')
WRITE(KWT,6)I
7 FORMAT(//'0 FREIER KERNSPEICHER AM ENDE   : ',I3,' K BYTES')
DO 27 I=1,20
27 WRITE(KWT,28)
28 FORMAT(1X,131H
1
2
RLOG=1.E-10
CALL DTF
READ(KRD,1) I
1 FORMAT(I5)
IF(I-99999)3,4,4
4 CALL FREESP(I)
WRITE(KWT,7)I
STOP
END

```

C-----
 C-----
 C

```

SUBROUTINE DTF
COMMON LAENGE,A(200)
COMMON /FIX/LEN,IA(200)
COMMON /TSCH/ITSCH,IANF,RHO,ITM,INORM,ROM
DIMENSION DI(96)

```

```

EQUIVALENCE(ID,IA(1)),(ITH,IA(2)),(ISCT,IA(3)),(ISN,IA(4)),
1(IGE,IA(5)),(IBL,IA(6)),(IBR,IA(7)),(IZM,IA(8)),(IM,IA(9)),
2(IFN,IA(10)),(IEVT,IA(11)),(IGM,IA(12)),(IHT,IA(13)),(IHS,IA(14)),
3(IHM,IA(15)),(MS,IA(16)),(MCR,IA(17)),(MTP,IA(18)),(MT,IA(19)),
4(IPVT,IA(20)),(IQM,IA(21)),(IIM,IA(22)),(ID1,IA(23)),(ID2,IA(24)),
5(ID3,IA(25)),(ID4,IA(26)),(ICM,IA(27)),(NFF,IA(28)),(IC,IA(29)),
6(IA(31),KS852),(IA(32),LC),(IA(33),ICC),(IA(34),NFN),(IA(35),NM),
7(IA(36),IIG),(IA(37),IIC),(IA(38),KS851),(IA(39),ICVT),(IA(40),IHG
8),(IA(41),MM),(IA(42),MG),(IA(43),IG),(IA(44),IP),(IA(45),MI),
9(IA(46),IZP),(IA(47),IGP),(IA(48),ML)
EQUIVALENCE(IA(51),JR),(IA(52),JKI),(IA(53),JVE),(IA(54),JW),
1(IA(55),JD),(IA(56),JWD),(IA(57),JF),(IA(58),JN),(IA(59),JMD),
2(IA(60),JRM),(IA(61),JQ),(IA(62),JC),(IA(63),JL),(IA(64),JQG),
3(IA(65),JKE),(IA(66),JRA),(IA(67),JA),(IA(68),JV),(IA(69),JAV),

```

```

4(IA(70),JDA),(IA(71),JDB),(IA(72),JDC),(IA(73),JDS),
5(IA(75),JX),(IA(76),JB),(IA(77),JFG),(IA(78),JTB),(IA(79),JSR),
6(IA(80),KRD),(IA(81),JSN),(IA(82),JCT),(IA(83),JCS),(IA(84),JCH),
7(IA(85),JSA),(IA(86),JNN),(IA(87),JJJ),(IA(88),JNO),(IA(89),JNL),
8(IA(90),JRL),(IA(91),JAF),(IA(92),JST),(IA(93),JNA),(IA(94),JAT),
9(IA(95),JNI),(IA(96),KWT),(IA(97),JNE),(IA(98),JNR),(IA(99),JOR)
EQUIVALENCE(IA(100),JSC),(IA(101),JAG),(IA(102),JNG),
1(IA(103),JFN),(IA(104),JDG),(IA(105),JSD),(IA(106),JNB),
2(IA(107),JCA),(IA(108),KMA),(IA(109),KMZ),(IA(110),KMB),
3(IA(111),KMC),(IA(112),KMT),(IA(113),KM3),(IA(114),KMR),
4(IA(115),KM4),(IA(116),JT),(IA(117),JSM),(IA(118),JSZ),
5(IA(119),JT3),(IA(120),JT5),(A(1),EV),(A(2),EVM),
6(A(3),EPS),(A(4),EPSA),(A(5),BF),(A(6),DY),(A(7),DZ),
7(A(8),XNF),(A(9),PV),(A(10),XEPS),(A(11),XLAL),(A(12),XLAH),
8(A(13),XNPM),(A(14),STT),(A(15),FTP),(A(16),XLA),
9(IIL,IA(30)),(A(32),EQ),(IA(191),JJJR),(A(28),RLOG)
EQUIVALENCE(IA(192),JBKL),(IA(193),MBK)

```

C

```

EQUIVALENCE(A(27),XLAR)
EQUIVALENCE(IA(194),IINP),(IA(195),JFIN)
EQUIVALENCE(IFISS,IA(199))
EQUIVALENCE(IQUER,IA(198))

```

C

C

```

2 FORMAT(20A4)
722 FORMAT(1H1,20A4)
3947 CONTINUE
READ(KRD,2)(A(I),I=1,20)
WRITE(KWT,722)(A(I),I=1,20)
DO 11 I=1,20
11 A(I)=0.
3 FORMAT(12I6)
READ(KRD,3)((IA(I),I=1,8),IA(11),IA(12),IA(16),MBK)
1)
READ(KRD,3) MT,MTP,MUP,IQM,IM,IPVT,ICM,IIM,IIL,ID1,ID2,NFF
ITSCH=0
READ(KRD,3) IINP,ITSCH,IFISS,IFLUX,IPUT,ITPUT,IAUSW,INORM,IQUER
IF(ITPUT.GT.0)REWIND ITPUT
4 FORMAT(6E12.5)
COMMON/ZLIES2/IAL(40)
DO 4201 I=1,MTP
4201 IAL(I)=I
IF(ISCT.GT.0) READ(KRD,3)(IAL(I),I=1,MTP)
RHO=0.
READ(KRD,4)(A(1),A(3),A(5),A(6),A(7),A(9))
1,EPSA,RHO

```

C

```

IHT=5
IHS=6
IHM=IHT+IGM
EVM=EV/10.
XNF=1.
IF(IEVT.EQ.0)XNF=0.
IF(IBR.EQ.(-1))XNF=0.
XEPS=10.*EPS
XNPM=1.
XLAL=0.01
XLAH=0.5
IFN=0

```



```

IF(MUP) 5,5,6
6 READ(KRD,3)(IHT,IHS,IHM)
5 CONTINUE

```

C

```

GNULL=0.
TIME=ZEIT(GNULL)

```

C

```

COMPUTE INITIAL VALUES
A(18)=1.420892
A(21)=1.0
A(22)=6.283185
A(23)=12.56637
A(24)=1.0
A(25)=.5
A(26)=.3333333
IHG=IHM*IGM
MM=ISN+1
IF(IGE-2)699,698,699
698 MM=ISN*(ISN+4)/4
699 NM=1
1676 FORMAT(1X,8HRECHNUNG)
MG=MM*IGM
IG=IM*IGM
ITM=IG
IP=IM+1
MI=MM*IP
IZP=IZM+1
IGP=IGM+1
ML=MCR+MTP
IF(ISCT-1)52,697,697
697 NM=ISCT
IF(IGE-2)52,694,52
694 NM=NM*(NM+4)/4
C COMPUTE STORAGE REQUIREMENTS AND STORAGE INDICES
52 JR=51
JKI=JR+IP
JVE=JKI+IGP
JW=JVE+IGP
JD=JW+MM
JWD=JD+MM
JF=JWD+MM
JN=JF+IM
JMD=JN+IG
JRM=JMD+MS
JQ=JRM
IF(IEVT-4) 693,695,693
695 JQ=JRM+IZM
693 JC=JQ
IF(IQM)692,51,692
692 JC=JQ+IG
51 JL=JC+IHM*IGM*MT
JQG=JL+MM*NM
JKE=JQG+IGP
JRA=JKE+IGP
JA=JRA+IP
JV=JA+IP
JAV=JV+IP
JDA=JAV+IP
JDB=JDA+MI
JDC=JDB+MI

```

```
JDS=JDC+MI
JX=JDS+MI
JB=JX+IG*NM
JFG=JB
IF (IBR.EQ.0)GO TO 690
691 JFG=JB+MG
690 JTB=JFG+IGP
JSR=JTB+IGM
JSN=JSR+IM
JCT=JSN+IGP
JCS=JCT+IM
JCH=JCS+IM
JSA=JCH+IM*ISCT
JNN=JSA+IM*MM
JJJ=JNN+IM
JNO=JJJ+IM*NM
JNL=JNO+MM
JRL=JNL+IGP
JAF=JRL+IGP
JST=JAF+MI
JNA=JST+IM
JAT=JNA+IM
JNI=JAT+IM*MM
JNE=JNI+MM
JNR=JNE+IM
JOR=JNR+IM
JSC=JOR+MM
JAG=JSC+IGP
JNG=JAG+IGP
JFN=JNG+IGP
JDG=JFN+IGP
JSD=JDG+IGP
JNB=JSD+IGP
JCA=JNB+IGP
KMA=200
KMZ=KMA+IM
KMB=KMZ+IZM
KMC=KMB+MS
KMT=KMC+MS
KMR=KMT+MTP
KM3=KMR+MM
KM4=KM3+ID3
KIDIM=KM4+ID3
JTOTI=KIDIM+IGM
JT=JCA+IM
IT=NM+17
JSM=JT+IZP*IT
JSZ=JSM+IT
JT3=JSZ+IT
JT5=JT3+ID3*IZP
JJJR=JT5+ID3*IM
JBKL=JJJR+MI
IMBUCK=0
IF (MBK+1)1750,1751,1752
1750 IMBUCK=IGM*IZM
IF (MBK.EQ.(-11))GO TO 1751
GO TO 1752
1751 IMBUCK=IGM
1752 JFIN=JBKL+IMBUCK
```

```

KFUNK=JFIN
KBUCK=KFUNK+IGM
KTIMEA=KBUCK+IGP
JFIN=KTIMEA+IGP
JNBG=JNB+IGM
C STARTING INDICES
DO 54 I=1,70
KA=I+50
KB=I+120
54 IA(KB)=IA(KA)-1
JTOTAL=JFIN+JTOTI
WRITE(KWT,55) JTOTI,JFIN,JTOTAL
55 FORMAT(17H0INTEGER STORAGE ,I5/24H FLOATING POINT STORAGE=,I5/
120H TOTAL DATA STORAGE=,I6,6HWORDS //)
IF(JTOTI-LEN ) 743,743,744
743 IF(JFIN-LAENGE)745,745,744
744 WRITE(KWT,53)
STOP
745 CONTINUE
DO 4200 I=1,IGM
A(KFUNK+I-1)=0.
4200 IA(KIDIM+I-1)=0
53 FORMAT(29H0***STORAGE EXCEEDS COMMON*** //)
CALL WEIGHT(MM,IGE,KWT,D1)
C .....
C RADII
C .....
L1=IZM-1
IX=JKI+L1
IY=KMZ+L1
READ(KRD,721)(A(I),I=JKI,IX)
READ(KRD,1722)(IA(I),I=KMZ,IY)
DO 4700 I=1,IZM
4700 A(JFIN+I-1)=A(JKI+I-1)
C
IF(IQM.EQ.0) GOTO 1753
KAA=LEN-IZM+1
KAE=LEN
DO 1723 I=1,IZM
K=I+LEN-IZM
J=KMZ+I-1
1723 IA(K)=IA(J)
721 FORMAT(6E12.5)
1722 FORMAT(12I6)
1753 CONTINUE
ISUM=0
DO 723 K=1,IZM
I=JKI+K-1
J=KMZ+K-1
W=A(I)-A(I-1)
IF(W)738,738,724
724 IF(IA(J))739,739,723
723 CONTINUE
IX=51
IY=199
DO 726 K=1,IZM
K2=JKI+K-1
K1=K2-1
K3=KMZ+K-1

```

```

M1=IA(K3)
ISUM=ISUM+IA(K3)
WX=M1
H=(A(K2)-A(K1))/WX
DO 727 J=1,M1
IX=IX+1
IY=IY+1
K4=IX-1
A(IX)=A(K4)+H
727 IA(IY)=K
726 CONTINUE
IF(ISUM.NE.IM) GO TO 739
DO 730 I=1,IZM
K1=JKI+I-1
K2=KMZ+I-1
A(K1)=0.0
730 IA(K2)=0
C
C .....
C WEIGHTS AND DIRECTIONS
C .....
DO 674 I=1,MM
IX=JW+I-1
IY=JD+I-1
K1=MM+I
A(IY)=D1(K1)
674 A(IX)=D1(I)
C
C .....
C MIXTURES
C .....
IF(MS)400,400,401
401 DO 402 J=1,MS
K2=KMC+J-1
K3=KMB+J-1
K4=JMD+J-1
READ(KRD,408)(K1,IA(K2),IA(K3),A(K4))
IF(IA(K3))460,404,460
460 IF(K1-J)404,402,404
402 CONTINUE
408 FORMAT(3I6,E12.5)
C
C .....
C MATERIAL NUMBERS
C .....
400 K1=KMB-1
READ(KRD,405)(IA(I),I=KMZ,K1)
405 FORMAT(12I6)
DO 413 I=KMZ,K1
MZZ=IABS(IA(I))
IF(MZZ*(MT+1-MZZ).LE.0)GO TO 412
IF(ISCT.LE.0)GO TO 413
DO 1414 J=KMZ,K1
IF(IA(J).GT.0)GO TO 1414
IF((IABS(IA(J))-ISCT).EQ.MZZ)WRITE(KWT,1415)
1414 CONTINUE
1415 FORMAT(////20X,'THERE SEEMS TO BE AN ERROR IN THE MATERIAL NUMBERS
1 ')
413 CONTINUE
C
C .....
C RADIAL MODIFIERS
C .....

```

```

IF(IEVT-4) 409,406,409
406 K1=JQ-1
READ(KRD,721)(A(I),I=JRM,K1)
409 CONTINUE
C .....
C DISTRIBUTED SOURCE
C .....
714 IF(IQM) 714,710,711
K1=IGM*IZM+JC-1
READ(KRD,721)(A(I),I=JC,K1)
K3=0
DO 496 IGT=1,IGM
DO 496 J=1,IZM
IX=JC+IZM*(IGT-1)+J-1
B=A(IX)
K2=KAA+J-1
K1=IA(K2)
DO 496 I=1,K1
IY=JQ+K3
A(IY)=B
496 K3=K3+1
DO 1724 I=KAA,KAJ
1724 IA(I)=0
K1=JC+IGM*IZM-1
DO 715 I=JC,K1
715 A(I)=0.
GOTO 710
711 K1=JC-1
IF(IQM.EQ.2) GO TO 1780
READ(KRD,721)(A(I),I=JQ,K1)
403 CONTINUE
GO TO 710
1780 DO 1781 I=JQ,K1
1781 A(I)=0.
710 CONTINUE
C .....
C FISSION FRACTIONS AND VELOCITIES
C .....
7 IF(IA(1).LE.0)GO TO 8
K1=JVE-2
READ(KRD,721)(A(I),I=JKI,K1)
K1=JW-2
READ(KRD,721)(A(I),I=JVE,K1)
8 CONTINUE
C .....
C BUCKLINGS
C .....
5555 IF (MBK) 5555,5556,5556
K1=KFUNK-1
READ(KRD,721)(A(I),I=JBKL,K1)
5556 CONTINUE
C .....
C FISSION INPUT
C .....
JFINIS=JFIN+2*(IM+IZM)
I1=JFIN+IZM
DO 1915 I=1,IM
1915 A(I1+I-1)=0.5*(A(JR+I-1)+A(JR+I))
C

```

C

```

IF (IPUT.LE.0.OR.ITPUT.LE.0)GO TO 1916
WRITE (ITPUT) ID, IGM, IM, IZM
I1=JFIN
I2=I1+IZM-1
WRITE (ITPUT) (A(I),I=I1,I2)
I1=I2+1
I2=I1+IM-1
WRITE (ITPUT) (A(I),I=I1,I2)
1916 CONTINUE
IF (IFISS)4400,4300,4500
4500 CALL PROVD0(A(JFIN),A(JN),IGM,IM,IZM,IFISS,IFLUX,A(JFINIS),
1 A(JFIN+IZM),A(JFIN+IM+IZM),A(JFIN+2*IM+IZM))
IF (IFISS.NE.0) GO TO 4400
4300 CONTINUE
IF (IBR.EQ.(-1)) GO TO 4713
B=1.2
K3=JKI-1
VKR=A(K3)
IF (VKR)739,739,1127
1127 IF (VKR-1.)1128,1128,1129
1128 VKR=0.5
GOTO 1154
1129 VKR=1.-1./VKR
1154 CONTINUE
K3=2*IBL-IBR+2
GOTO(510,511,512,513),K3
512 VKR=0.0
E0=0.
E1=0.
E2=0.
GO TO 514
510 E0=2*IM
E1=1.0
E2=IM
GO TO 514
511 E0=IM
E1=1.0
E2=IM
GO TO 514
513 E0=2*IM
E1=IM+1
E2=2*IM
514 K1=JF+IM-1
DO 515 J=JF,K1
Z=J-JF
Z=Z+E1
W=2.0*Z-1.0-E0
W=W/(E0-1.)
W=VKR*W*W
W=B*(1.0-W)
515 A(J)=W
4713 CONTINUE
IF (IBR.NE.(-1))GO TO 4400
K1=JF+IM-1
DO 4401 I=JF,K1
4401 A(I)=1.E-7
4400 CONTINUE
IF (IBR.NE.(-1))GO TO 9200

```

```

IF(IGE.NE.1)GO TO 9201
ME=(MM+1)/2
READ(KRD,721)((A(JB+J-1+(I-1)*IGM),I=1,ME),J=1,IGM)
GO TO 9200
9201 WRITE(KWT,9202)
9202 FORMAT(/10X,' FOR IGE.NE.1 IS IBR=-1 NOT PERMITTED')
STOP
9200 CONTINUE
IF(ICM) 1988,1987,1987
1988 ICM=-ICM
1987 CONTINUE
C
C
C TEST FOR SUITABILITY OF TSCHEBYCHEFF EXTRAPOLATION
IF(ITSCH.EQ.0) GO TO 4002
IF(IEVT.GT.1) GO TO 4002
WRITE(KWT,4001)
4001 FORMAT(45HOTSCHEBYSCHEFF EXTRAPOLATION WIRD VERSUCHT )
GO TO 4004
4002 WRITE(KWT,4003)
4003 FORMAT(50HOTSCHEBYSCHEFF EXTRAPOLATION WIRD NICHT VERSUCHT )
4004 CONTINUE
C
C
C OUTPUT
WRITE(KWT,1983)
1983 FORMAT(///18H CORRECT DTK INPUT///)
701 FORMAT(1H0,27H ID IDENTIFICATION NUMBER,60X,I8/35H ITH THEORY
1(O/1=REGULAR/ADJOINT),53X,I8/55H ISCT SCATTERING (O/N=ISOTROPIC/N
2TH ORDER ANISOTROPIC),33X,I8)
702 FORMAT(41H ISN QUADRATURE (SN,PN-1,DPN/2-1, ETC) , 47X,I8/45H
1IGE GEOMETRY (1/2/3=PLANE/CYLINDER/SPHERE),43X,I8/73H IBL/IBR LEF
2T/RIGHT BOUNDARY CONDITION (O/1/2=VACUUM/REFLECTIVE/PERIODIC),I7,
3I8)
703 FORMAT(22H IZM NUMBER OF ZONES,66X,I8/26H IM NUMBER OF INTERV
1ALS,62X,I8/78H IEVT EIGENVALUE TYPE (O/1/2/3/4/5=SOURCE/K/ALPHA/C
2ONCENTRATION/DELTA/RADIUS),10X,I8)
704 FORMAT(23H IGM NUMBER OF GROUPS,65X,I8/39H MS NUMBER OF MIXTU
1RE SPECIFICATIONS,49X,I8/ 64H MTP NUMBER OF MATERIAL CROSS SECTI
2ONS READ FROM CARDS OR DISK,24X,I8)
6705 FORMAT(31H MBK BUCKLING (O=NO BUCKLING),57X,I8/32H MUP UPSCATT
1ERING (O/1 NO/YES),56X,I8)
705 FORMAT(32H MT TOTAL NUMBER OF MATERIALS,56X,I8/54H IPVT PARAME
1TRIC EIGENVALUE TYPE (O/1/2=NONE/K/ALPHA),34X,I8/86H IQM DISTRIB
2UTED SOURCE INDICATOR (O/1/-1=NONE/PRO RAD.AND GROUP/PRO ZONE AND
3GROUP) ,2X,I8)
706 FORMAT(42H IIM INNER ITERATION MAXIMUM (PER GROUP),46X,I8/38H I
1D1 PRINT ANGULAR FLUX (O/1=NO/YES),50X,I8/48H ID2 PRINT BALANCE
2TABLES BY GROUP (O/1=NO/YES),40X,I8)
707 FORMAT(30H ICM OUTER ITERATION MAXIMUM,58X,I8/54H NFF DETAILED
1PRINT IF FLUX-FIX-UP USED (O/1=YES/NO),34X,I8/74H IIL INNER ITER
2ATION LIMIT USED UNTIL ONE MINUS LAMBDA IS WITHIN TEN*EPS,14X,I8//
3)
708 FORMAT(23H EV EIGENVALUE GUESS,55X,E18.7/28H EPS CONVERGENCE
1CRITERION,50X,E18.7/22H BF BUCKLING FACTOR,56X,E18.7/28H DY/DZ
2BUCKLING HEIGHT/DEPTH,36X,2E16.7)
709 FORMAT(28H PV PARAMETRIC EIGENVALUE,50X,E18.7/58H EPSA SPECIAL
1 CONVERGENCE CRITERION USED ONLY IF NON-ZERO,20X,E18.7/
2 66H RHO INPUT-RHO,IF TSCHEBYCHEV EXTRAPOLATION IS USED(ITSCH.NE

```

```

3.0)      ,12X,E18.7)
3710 FORMAT(47H IHT POSITION OF TOTAL CROSS SECTION IN TABLE,34X,I8/
154H IHS POSITION OF SELF-SCATTER CROSS SECTION IN TABLE,34X,I8)
3711 FORMAT( 33H IHM CROSS SECTION TABLE LENGTH,55X,I8//)
WRITE(KWT,701)(IA(I),I=1,3)
WRITE(KWT,702)(IA(I),I=4,7)
WRITE(KWT,703)(IA(8),IA(9),IA(11))
WRITE(KWT,704)(IA(12),IA(16),MTP)
WRITE(KWT,6705) MBK,MUP
WRITE(KWT,705)(IA(I),I=19,21)
WRITE(KWT,706)(IA(I),I=22,24)
WRITE(KWT,707)(IA(27),IA(28),IA(30))
4731 FORMAT(56H IINP UNIT NUMBER OF THE CROSS SECTION FILE IF USED
1,32X,I8/65H ITSCH TRIGGER FOR TSCHEBYSCHIEFF EXTRAPOLATION(0/1=NOT
2USED/USED),23X,I8/55H ITPF FLUX INPUT FROM PREVIOUS CALCULATION(0
3/1=NO/YES),33X,I8/44H ITP1 UNIT NR. OF THE OPTIONAL FLUX INPUT ,
544X,I8/48H IMOV FLUX OUTPUT ON UNIT NR. ITP2 (0/1=NO/YES) ,40X,I8
6/46H ITP2 UNIT NUMBER OF THE OPTIONAL FLUX OUTPUT,42X,I8 )
WRITE(KWT,4731) IINP,ITSCH,IFISS,IFLUX,IPUT,ITPUT
WRITE(KWT,4733) IAUSW,IQUER
4733 FORMAT(70H IAUSW UNIT NR. OF THE OPTIONAL FLUX OUTPUT FOR EVALUATI
IONS
,18X,I8/46H IQUER WRITE OUT CROSS SECTIONS(0/1=NO/
2YES) ,42X,I8//)
WRITE(KWT,708)(A(1),A(3),A(5),A(6),A(7))
WRITE(KWT,709) A(9),EPSA,RHO
IF(MUP) 712,712,713
713 WRITE(KWT,3710) IHT,IHS
WRITE(KWT,3711) IHM
712 CONTINUE
C
201 FORMAT(1H0,5HRADII,I6)
202 FORMAT(1H0,12HZONE NUMBERS,I6)
203 FORMAT(1H0,16HMATERIAL NUMBERS,I6)
204 FORMAT(1H0,11HMIX COMMAND,I6)
205 FORMAT(1H0,10HMIX NUMBER,I6)
206 FORMAT(1H0,11HMIX DENSITY,I6)
207 FORMAT(1H0,16HRADIAL MODIFIERS,I6)
208 FORMAT(1H0,12HDISTR SOURCE,I6)
209 FORMAT(1X,10E12.5)
210 FORMAT(1X,10I12)
WRITE(KWT,201) IP
K1=JKI-1
WRITE(KWT,209)(A(I),I=JR,K1)
WRITE(KWT,202) IM
K1=KMZ-1
WRITE(KWT,210)(IA(I),I=KMA,K1)
WRITE(KWT,203) IZM
K1=KMB-1
WRITE(KWT,210)(IA(I),I=KMZ,K1)
IF(MS) 220,221,220
220 WRITE(KWT,204) MS
K1=KMT-1
WRITE(KWT,210)(IA(I),I=KMC,K1)
WRITE(KWT,205) MS
K1=KMC-1
WRITE(KWT,210)(IA(I),I=KMB,K1)
WRITE(KWT,206) MS
K1=JRM-1
WRITE(KWT,209)(A(I),I=JMD,K1)

```



```

221 IF(IEVT-4)222,223,222
223 WRITE(KWT,207)IZM
      K1=JQ-1
      WRITE(KWT,209)(A(I),I=JRM,K1)
222 IF(IQM)224,225,224
224 WRITE(KWT,208)IG
      K1=JC-1
      WRITE(KWT,209)(A(I),I=JQ,K1)
225 CONTINUE
211 FORMAT(1H0,7HWEIGHTS,I6)
212 FORMAT(1H0,10HDIRECTIONS,I6)
      WRITE(KWT,211)MM
      K1=JD-1
      WRITE(KWT,209)(A(I),I=JW,K1)
      WRITE(KWT,212)MM
      K1=JWD-1
      WRITE(KWT,209)(A(I),I=JD,K1)
      IF(IFISS.GT.0)GO TO 4725
213 FORMAT(1H0,13HFISSION GUESS,I6)
      WRITE(KWT,213)IM
      K1=JN-1
      WRITE(KWT,209)(A(I),I=JF,K1)
      GO TO 4726
4725 CONTINUE
4728 FORMAT(1H0,2X,80H FLUX GUESS FROM PREVIOUS CALCULATION FOR ALL POI
      INTS OF EACH GROUP
      WRITE(KWT,4728)
      DO 4727 IGL=1,IGM
      WRITE(KWT,6431)IGL
4727 WRITE(KWT,209)(A(JN-1+I+(IGL-1)*IM),I=1,IM)
6431 FORMAT(1H0,'      FLUX-ESTIMATION IN GROUP ',I3)
4726 CONTINUE
      IF(MBK)7762,7761,7761
7762 I3=IMBUCK/IGM
      WRITE(KWT,7213)IGM,I3
7213 FORMAT(1H0,9HBUCKLINGS,I6,'*',I4)
      K1=KFUNK-1
      WRITE(KWT,209)(A(I),I=JBKL,K1)
7761 CONTINUE
C      DONE WITH INPUT
C      COMPUTE AND PRINT SN-CONSTANTS
908 CALL RECS(A(JC),IHM,IGM,MT)
909 J1=JKE
      J2=J1+MM
      J3=J2+MM
      J4=J3+MM
      KA=ISCT+1
      KB=KA+ISCT
      J5=J4+KB
      J6=J5+MM*KA*KA
      CALL SNCN(A(JW),A(JD),A(JWD),A(JL),IA(KMR),A(J1),A(J2),A(J3),
      IA(J4),A(J5),MM,NM,KA,KB)
C      ZERO TEMPORARY STORAGE
      DO 9090 K=J1,J6
9090 A(K)=0.0
C      TEST FOR ADJOINT PROBLEM
      IF(ITH.NE.1)GO TO 910
C      COMPUTE ADJOINT REVERSALS AND CROSS-SECTIONS
      CALL ADJREV(A(JKI),A(JVE),A(JQ),A(JN),IQM,IFN,IGM,IM,IC,

```

```

1A(JC),IHM,MT,ML,IHT,IHS)
C   COMPUTE INITIAL FUNCTIONS
910 CALL IFUNC(A(JKI),A(JKE),A(JVE),IGP,IQM,PV,IPVT,IEVT)
C   MIX AND PRINT CROSS-SECTIONS
   IF(ICC.EQ.0) GO TO 911
9100 IF(IEVT.NE.3) GO TO 9111
   911 CALL MIXCX(A(JC),IA(KMB),IA(KMC),A(JMD),IHM,IGM,MT,
   IMS,EV,IEVT,ICC)
9111 NFN=1
C   MODIFY RADII AND COMPUTE AREAS
   CALL RMAVGF(A(JRA),A(JA),A(JR),A(JV),A(JRM),A(JAV),IA(KMA),
   1A(JW),A(JD),A(JDA),A(JDB),A(JDC),A(JDS),IP,IM,IZM,MM)
C   CHECK FOR DISTRIBUTED SOURCE
912 IF(ICC.NE.0.OR.IQM.EQ.0) GO TO 914
C   DISTRIBUTED SOURCE
   CALL DSOUR(A(JQ),A(JQG),A(JV),IM,IGM,IGP,IP,XNF)
C   FISSION AND NORMALIZATIONS
914 CALL FISSN(A(JN),A(JC),A(JX),A(JB),IGM,IM,IHM,MT,NM,MM,A(JFG),IGP
   1,A(JF),A(JKE),IA(KMA),IA(KMZ),IZM,A(JV),A(JQG))
C   MONITOR PRINT
8300 IF(ICC.GT.0) GO TO 915
   WRITE(KWT,321)EPS,EPSA,ID
   321 FORMAT(1H0//18H ITERATION MONITOR,5X,4HEPS=,E16.8,5X,5HEPSA=,
   1E16.8,5X,11H PROBLEM ID,I8//11H TIME IN ,4X,
   2          SHOUTER,7X,5HINNER,7X,7HNEUTRON,7X,10HEIGENVALUE,5X
   3,10HEIGENVALUE,7X,6HLAMBDA,6X,11HRED. FAKTOR,5X,9H
   7          / 22H MINUTES ITERATIONS,2X,10HITERAT
   4IONS,5X,7HBALANCE,24X,5HSLOPE,38X,10H          /)
915 TI=ZEIT(TIME) /60.
   ROM=0.
   IF(ABS(1.-XLAR).GT.RLOG)ROM= (1.-XLA)/(1.-XLAR)
   WRITE(KWT,331)TI,ICC,LC,A(JNBG),EV,EQ,XLA,ROM
   331 FORMAT(1X,F9.3,1X,I8,4X,I8,4X,5E15.8)
4296 FORMAT(1H0)
   WRITE(KWT,4296)
   DO 4100 IGIM=1,IGM
4100 WRITE(KWT,4101)IGIM,IA(KIDIM+IGIM-1),A(KFUNK+IGIM-1)
4101 FORMAT(32X,I3,4X,I2,5X,E15.8)
   WRITE(KWT,4296)
   IF(ICC.LE.ICM)GOTO337
C FLUX DUMP
332 WRITE(KWT,333)
333 FORMAT(40H000 MANY OUTER ITERATIONS-FLUXES DUMPED)
   GOTO 917
C ALPHA OPTION
337 E1=0.
   IF(IEVT.EQ.2)E1=EV
   IF(IPVT.EQ.2)E1=E1+PV
   DO 916 I=1,IGM
   JTBI=JTB+I-1
   JVEI=JVE+I-1
   A(JTBI)=0.0
   IF(ABS(A(JVEI)).GT.RLOG) A(JTBI)=E1/A(JVEI)
916 CONTINUE
C BEGIN GROUP LOOP
   CALL OUTER(A(JC),A(JSA),A(JL),A(JN),A(JX),A(JCH),A(JQ),
   1A(JB),A(JAF),A(JJJ),A(JT),A(JSM),A(JSZ),IHM,IGM,MT,IM,
   2MM,NM,ISCT,IT,IP,IZP,IA(KMA),IA(KMZ),IZM,A(JF),A(JCS),A(JV),
   3A(JSR),A(JCT),A(JCA),A(JNN),A(JND),A(JST),A(JNR),A(JW),

```

```

4A(JRA),A(JD),A(JA),A(JWD),
5A(JSN),A(JSC),A(JNL),A(JAG),A(JSD),A(JRL),A(JNB),IGP,
6A(JFN),A(JNG),A(JDG),A(JQG),A(JFG),A(JKE),A(JKI),
7 A(KBUCK),A(KTIMEA),A(KFUNK),IA(KIDIM)
GOTO(917,8300,852),KS851
917 WRITE(KWT,537)
537 FORMAT(23H1FINAL ITERATION RESULT /23H -----//)
TI=ZEIT(TIME)/60.
WRITE(KWT,321)EPS,EPSA,ID
IF(ABS(1.-XLAR).GT.RLOG)ROM= (1.-XLA)/(1.-XLAR)
WRITE(KWT,331)TI,ICC,LC,A(JNBG),EV,EQ,XLA,ROM
WRITE(KWT,4296)
DO 4120 IGIM=1,IGM
4120 WRITE(KWT,4101)IGIM,IA(KIDIM+IGIM-1),A(KFUNK+IGIM-1)
WRITE(KWT,4296)
WRITE(KWT,338)TI
338 FORMAT(16HOEXECUTION TIME= ,F11.4,8H MINUTES )
C FINAL PRINT
IF(IPUT.GT.0.AND.ITPUT.GT.0)WRITE(KWT,4730)ITPUT
4730 FORMAT(////1X,60H*****
1***** /2H *,58X,1H*/9H * REMARK,51X,1H*
7 /2H *,58X,1H*/2H *,58X,
21H*/ 43H * THE SCALAR FLUX HAS BEEN WRITTEN ON UNIT,I3,14X,1H*,2X/
32H *,58X,1H*/61H *****
6***** )
IF(IPUT.GT.0.AND.ITPUT.GT.0)WRITE(ITPUT)
1 ((A(JN-1+(J-1)*IM+I ),I=1,IM),J=1,IGM)
IF(IPUT.GT.0.AND.ITPUT.GT.0)WRITE(ITPUT) EV
IF(IAUSW.NE.0)CALL AUSWER(A(JN ),IM,IGM,IAUSW)
CALL FINPR(A(JN),A(JX),A(JT3),A(JT5),A(JC),IGM,IM,NM,ID3,
1IYP,IHM,MT,A(JNB),IGP,IA(KMA),A(JRA),A(JAV),A(JV),A(JF),IP,IA(KM3)
2,IA(KM4),IA(KMZ),IZM,A(JMD),MS,IA(KMC),IA(KMB))
CALL REMUS
RETURN
C NEW PARAMETERS FOR IMPLIZIT EIGENVALUE-SEARCH
852 CALL NEWPAR
GOTO(917,8300,9100),KS852
C ERROR-OUTPUT
520 FORMAT(///32H INPUT OF ZONE RADII NOT CORRECT///)
521 FORMAT(///77H NO CORRESPONDENCE BETWEEN IM (=NUMBER OF RADII) AND
1NUMBER OF ZONE-INTERVALS//10X,7HIA(J) =,I5,2X,4HIM =,I5,6HISUM =,
2I5)
522 FORMAT(///42H INCORRECT INPUT OF MIXTURE-SPECIFICATIONS///)
738 WRITE(KWT,520)
STOP
739 WRITE(KWT,521) IA(J),IM,ISUM
STOP
404 WRITE(KWT,522)
STOP
412 WRITE(KWT,1415)
STOP
END
C-----
C-----
C
C PERFORMS OUTER ITERATION
SUBROUTINE OUTER(C,SA,CL,XN,XJ,CH,Q,B,XND,XJJ,T,
1SUMT,SUMZ,IHM,IGM,MT,IM,MM,NM,ISCT,IT,IP,IZ, MA,MZ,IZM,F,CS,V,
2SR,CT,CA,XNN,XNO,ST,XNR,W, RA,D,AV,WD,

```

```

3          SNG,SCG,XNL,AG,SDG,RL,XNB,IGP,FNG,XNG,XNDG,QG,
4FG,XKE,XKI,BUCK,TIMEAB,FUNK,IDIM)
  DIMENSION BUCK(IGP),TIMEAB(IGP),FUNK(IGM),IDIM(IGM)
  DIMENSION SNG(IGP),SCG(IGP),XNL(IGP),AG(IGP),SDG(IGP),RL(IGP),
1XNB(IGP),FNG(IGP),XNG(IGP),XNDG(IGP),QG(IGP),FG(IGP),XKE(IGP),
2XKI(IGP)
  DIMENSION C(IHM,IGM,MT),SA(IM,MM),CL(MM,NM),XN(IM,IGM),
1XJ(IGM,IM,NM),CH(IM,ISCT),Q(IM,IGM),B(IGM,MM),XND(IP,MM),
2XJJ(IM,NM),T(IZ,IT),SUMT(IT),SUMZ(IT),MA(IM),MZ(IZM),F(IM),CS(IM),
3V(IP),SR(IM),CT(IM),CA(IM),XNN(IM),XND(MM),          ST(IM),
4XNR(IM),W(MM),RA(IP),          D(MM),AV(IP),WD(MM)
  COMMON LAENGE,A(200)
  COMMON/FIX/LEN,IA(200)
  COMMON/TSCH/ITSCH,IANF,RHO,ITM,INORM,ROM
  EQUIVALENCE(EV,A(1)), (ICC,IA(33)),(XLA,A(16)), (IEVT,IA(11)),
1(EPS,A(3)),(E1,A(34)),(E2,A(35)),(IIL,IA(30)),(IIM,IA(22))
  EQUIVALENCE (ITH,IA(2)),(IG,IA(36)),(IHT,IA(13)),(IHS,IA(14)),
1(IQM,IA(21)),(BF,A(5)),(DY,A(6)),(DZ,A(7)),(IGE,IA(5)),
2(IBR,IA(7)),(IDT,IA(28)) ,(JAT,IA(94)),(JDA,IA(70)),(JDB,IA(71)),
3(JDC,IA(72)),(JDS,IA(73)),(ICVT,IA(39)),(ID1,IA(23)),(ID2,IA(24)),
4(JNA,IA(93)),(JNI,IA(95)),(KMR,IA(114)),(JNE,IA(97)),(JOR,IA(99)),
5(IIC,IA(37)),(R,A(14)),(JVEA,IA(123)),(JTBA,IA(148)),(XT,A(18)),
6(NFT,IA(50)),(XLAR,A(27)),(IA(191),JJJR)
  EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
  EQUIVALENCE(IA(195),JFIN)
  EQUIVALENCE(IA(192),JBKL),(IA(193),MBK)
3947 CONTINUE
  UPUS=1.
  IF(IBR.EQ.(-1))UPUS=1./EV
  IHB=IHT
  IF(IHT+1.NE.IHS)IHB=IHB+1
  IG=1
824 JTB=JTBA+IG
  JVE=JVEA+IG
  R=0.0
  TAB=A(JTB)
C ISOTROPIC SOURCE NOT INCLUDING SELF SCATTER
  IF(ITH.GT.0)IHA=IHT-1
  DO 23 I=1,IM
  L=MA(I)
  L=IABS(MZ(L))
  AS=0.0
  IF(IQM.NE.0)AS=Q(I,IG)
  IF(ITH.LE.0)GO TO 13
  AS=AS+F(I)*C(IHA,IG,L)*UPUS
  GO TO 14
13 AS=AS+F(I)*XKE(IG) *UPUS
14 IH=IHB+1
17 K=IG+IHS-IH
  IF(IH.EQ.IHS) GO TO 20
  IF(K.LE.0)GO TO 22
  AA=C(IH,IG,L)
19 AS=AS+AA*XN(I,K)
20 IH=IH+1
  IF(IH.LE.IHM) GO TO 17
22 CS(I)=V(I)*C(IHS,IG,L)
  AS=AS*V(I)
  R=R+AS
  SR(I)=AS

```

```

SRI=0.0
AZ=C(IHT-4,IG,L)
IF(MBK.LT.(-10))AZ=0.3333333
AT=C(IHT,IG,L)
AA=C(IHT-2,IG,L)
IF(MBK) 763,764,765
763 IF(MBK.EQ.(-2).OR.MBK.EQ.(-12))GO TO 766
NBKL=JBKL+IG-1
SRI=A(NBKL)/(3.*AZ)
GO TO 764
766 LJ=MA(I)
NBKL=JBKL+LJ-1+(IG-1)*IZM
SRI=A(NBKL)/(AZ*3.)
GOTO 764
765 CONTINUE
IF(DY.GT.RLOG)SRI=((BF/(DY*AZ+XT))**2)*AZ
IF(DZ.GT.RLOG)SRI=SRI+((BF/(DZ*AZ+XT))**2)*AZ
764 CONTINUE
CT(I)=V(I)*(SRI+AT+TAB)
23 CA(I)=V(I)*(SRI+AA+TAB)
SNG(IG)=R-QG(IG)-FG(IG)
C ANISOTROPIC SOURCE NOT INCLUDING SELF SCATTER
IF(ISCT.EQ.0) GO TO 999
DO 25 I=1,IM
DO 21 M=1,MM
21 SA(I,M)=0.0
25 CONTINUE
DO 37 I=1,IM
L=MA(I)
L=MZ(L)
IF(L.GE.0) GO TO 37
DO 36 N=1,ISCT
J=N-L
CH(I,N)=V(I)*C(IHS,IG,J)
IH=IHB+1
28 K=IG+IHS-IH
IF(IH.EQ.IHS) GO TO 35
AB=FLOAT(N)+.5
AA=AB*V(I)*C(IH,IG,J)
IF(ABS(AA).GE.RLOG) GOTO 29
IF(IH-IHS) 35,35,36
29 IF(IGE.EQ.2) GO TO 31
AC=AA*XJ(K,I,N)
DO 30 M=1,MM
30 SA(I,M)=SA(I,M)+AC*CL(M,N)
GO TO 35
31 IRS=((N+1)*(N+1))/4
IRT=IRS+(2*N+1)/4
DO 32 IR=IRS,IRT
AC=AA*XJ(K,I,IR)
DO 32 M=1,MM
32 SA(I,M)=SA(I,M)+AC*CL(M,IR)
35 IH=IH+1
IF(IH.LE.IHM) GO TO 28
36 CONTINUE
37 CONTINUE
C FLUX CONTROL
999 IIC=0
IF(ABS(R).LE.RLOG) GOTO 129

```

```

      DO 121 I=1,IM
      XNN(I)=XN(I,IG)
      DO 121 N=1,NM
121  XJJ(I,N)=XJ(IG,I,N)
      DO 128 M=1,MM
      XNO(M)=0.0
      IF(IABS(IBR).GT.0) XNO(M)=B(IG,M)
128  CONTINUE
      GO TO 832
129  XNL(IG)=0.0
      RL(IG)=0.0
      DO 130 I=1,IP
      DO 130 M=1,MM
130  XND(I,M)=0.0
      DO 131 I=1,IM
      ST(I)=0.0
131  XNN(I)=0.0
      DO 132 M=1,MM
132  XNO(M)=0.0
      GO TO 841
C     BEGIN INNER ITERATION
832  CALL INNER (A(JAT),SA,CH,XJJ,CL,XND,A(JDA),A(JDB),A(JDC),
      1A(JDS),IM,MM,ISCT,NM,IP,A(JNA),XNN,ST,CS,SR,W,D,A(JNI),XNO,
      2IA(KMR),A(JNE),CT,WD,AV,XNR,A(JOR),A(JJJR),FUNK,IDIM)
C     GROUP CODE
841  E1=0.0
      E2=0.0
      E3=0.0
      E5=0.0
      E6=0.0
      E100=0.
      E200=0.
      DO 202 I=1,IM
      L=MA(I)
      L=IABS(MZ(L))
      E1=E1+XNN(I)*CT(I)
      E2=E2+XNN(I)*CS(I)
      E3=E3+XNN(I)*V(I)*C(IHT-2,IG,L)
      SRI=0.
      AZ=C(IHT-4,IG,L)
      IF(MBK.LT.(-10))AZ=0.3333333
      IF(MBK)3763,3764,3765
3763 IF(MBK.EQ.(-2).OR.MBK.EQ.(-12))GO TO 3766
      NBKL=JBKL+IG-1
      GO TO 3767
3766 NBKL=JBKL+MA(I)-1+IZM*(IG-1)
3767 SRI=A(NBKL)/(3.*AZ)
      GO TO 3764
3765 IF(DY.GT.RLOG)SRI=((BF/(DY*AZ+XT))**2)*AZ
      IF(DZ.GT.RLOG)SRI=SRI+((BF/(DZ*AZ+XT))**2)*AZ
3764 E100=E100+XNN(I)*V(I)*SRI
      E200=E200+XNN(I)*V(I)*TAB
      E5=E5+XNN(I)*V(I)
      XNB(IG)=0.
      GTR=(QG(IG)+FG(IG)+SNG(IG))
202  E6=E6+XNN(I)*V(I)*C(IHT-1,IG,L)
      SCG(IG)=E2
      AG(IG)=E3
      FNG(IG)=E6

```

```

XNG(IG)=E5
BUCK(IG)=E100
TIMEAB(IG)=E200
IF(ABS(A(JVE)).GT.RLOG)XNDG(IG)=E5/A(JVE)
SDG(IG)=E1-E2-E3-E100-E200
IF(ABS(GTR)-RLOG) 560,560,561
561 XNB(IG)=1.-(XNL(IG)+E3+SDG(IG)+E100+E200)/GTR
560 CONTINUE
IF(IBR.EQ.0) GO TO 213
DO 212 M=1,MM
212 B(IG,M)=XNO(M)
213 DO 219 I=1,IM
219 XN(I,IG)=XNN(I)
IF(ISCT.EQ.0) GO TO 216
DO 218 M=1,MM
E1=W(M)
DO 218 N=1,NM
E2=E1*CL(M,N)
DO 218 I=1,IM
IF(M.EQ.1)XJJ(I,N)=0.0
218 XJJ(I,N)=XJJ(I,N)+E2*(XND(I,M)+XND(I+1,M))
DO 217 I=1,IM
DO 217 N=1,NM
217 XJ(IG,I,N)=XJJ(I,N)
216 IF(ICVT.GT.0) GO TO 230
215 IG=IG+1
IF(IG.LE.IGM) GO TO 824
C TOTAL GROUP SUMS,FISSION AND CONVERGENCE NUMBER CALCULATION
BUCK(IGP)=0.
TIMEAB(IGP)=0.
XNB(IGP)=0.0
DO 18 J=1,IGM
18 XNB(IGP)=XNB(IGP)+XNB(J)
IF(ICVT.EQ.0.AND.(ICC+1).NE.IA(27))GO TO 79
DO 26 J=1,IGM
SNG(IGP)=SNG(IGP)+SNG(J)
SCG(IGP)=SCG(IGP)+SCG(J)
XNL(IGP)=XNL(IGP)+XNL(J)
AG(IGP)=AG(IGP)+AG(J)
SDG(IGP)=SDG(IGP)+SDG(J)
RL(IGP)=RL(IGP)+RL(J)
FNG(IGP)=FNG(IGP)+FNG(J)
XNG(IGP)=XNG(IGP)+XNG(J)
BUCK(IGP)=BUCK(IGP)+BUCK(J)
TIMEAB(IGP)=TIMEAB(IGP)+TIMEAB(J)
26 XNDG(IGP)=XNDG(IGP)+XNDG(J)
WRITE(KWT,271)(I,QG(I),FG(I),SNG(I),SCG(I),SDG(I),XNL(I),I=1,IG)
WRITE(KWT,272)(I,AG(I),XNB(I),RL(I),FNG(I),XNDG(I),XNG(I),I=1,IG)
WRITE(KWT,3770)(I,BUCK(I),TIMEAB(I),I=1,IGP)
3770 FORMAT(///' BUCKL. LEAKAGE ', ' TIME ABSORPTION ' /
1(I4,E16.8,E20.8))
271 FORMAT(///13X,7H SOURCE,5X,15HFISSION SOURCE ,10X,
110HIN SCATTER,8X,12HSELF SCATTER,9X,11HOUT SCATTER,
29X,11HNET LEAKAGE/(I4,E16.8,5E20.8))
272 FORMAT(///9X,11H ABSORPTION,5X,15HNEUTRON BALANCE,
17X,13HRIGHT LEAKAGE,4X,16HFISSION NEUTRONS,5X,
215HNEUTRON DENSITY,8X,12HNEUTRON FLUX/(I4,E16.8,5E20.8))
79 ICC=ICC+1
XLAR=XLA

```

```

C      FISSION CALCULATION
      JFINIS=JFIN+10
      CALL      TSCHEB(NM,IGM,IM,MM,IGP,ICC,A(JFINIS),ITM)
      CALLFISSN(XN,C,XJ,B,IGM,IM,IHM,MT,NM,MM,FG,IGP,F,XKE,MA,MZ,IZM,V,
1QG)
C      UP-SCATTER SCALING
      IF(IHT+1.EQ.IHS) GO TO 83
      E3=E4
      E4=0.0
      DO 80  I=1,IM
      L=MA(I)
      L=IABS(MZ(L))
      DO 80  J=1,IGM
80  E4=E4+V(I)*XN(I,J)*C(IHT+1,J,L)
      IF(ICC.LE.1) GO TO 83
      E5=QG(IGP)+FG(IGP)
      E6=E5/(E5+E3-E4)
      IF(E6.LE.RLOG) GOTO 83
      IF(IBR.EQ.(-1)) E6=1.
      DO 81  I=1,IM
      F(I)=E6*F(I)
      DO 81  J=1,IGM
      XN(I,J)=E6*XN(I,J)
      DO 81  N=1,NM
81  XJ(J,I,N)=E6*XJ(J,I,N)
      FG(IGP)=E6*FG(IGP)
      DO 82  J=1,IGM
82  FG(J)=E6*FG(J)
      IF(IBR.LT.1) GO TO 85
      DO 84  J=1,IGM
      DO84 M=1,MM
84  B(J,M)=E6*B(J,M)
85  IF(IFVT.EQ.1)EV=XKI(IGP)/XKE(IGP)
      IF(ABS(E3-E4).GT.EPS*E5) GO TO 119
C      CONVERGENCE NUMBERS
83  CONTINUE
      E1=1.-XLA
      E2=ABS(E1)
      E9=(ABS(1.-XLAR)+ABS(1.-XLA))/1.4
      IF(E9.LE.10.*EPS)IIL=IIM
      IF(IEVT.EQ.0) GO TO 114
      IF(IEVT.GT.1) GO TO 113
      EV=XKI(IGP)/XKE(IGP)
      GO TO 114
113  IA(38)=3
      RETURN
114  IF(ICVT.EQ.0) GO TO 116
      IA(38)=1
      RETURN
116  IF(E9.LE.EPS) GO TO 120
119  IA(38)=2
      RETURN
120  ICVT=1
      GO TO 119
230  IF(NFT.LT.1) GO TO 231
      NFT=0
      WRITE(KWT,273) IG
273  FORMAT(34H0***NEG. FLUX FIX-UP USED IN GROUP,I3,3H***/)
231  IF(ID1.EQ.0) GO TO 300

```



```

WRITE(KWT,243) IG,MM
DO241 I=1,IP
241 WRITE(KWT,242)I,RA(I),(XND(I,M),M=1,MM)
242 FORMAT(7H RADIUS,I3,1H=,F16.8/(1P8E15.5))
243 FORMAT(37H1FLUX BY RADIUS AND DIRECTION, GROUP=,I3,
110X,18H DIRECTIONS (1 TO ,I3,22H) SEQUENTIALLY BY ROWS//)
300 IF(ID2.EQ.0) GO TO 215
C
BALANCE ECIT BY GROUP AND ZONE
WRITE(KWT,301) IG
301 FORMAT(1H0///18H BALANCE FOR GROUP,I3/)
DO303 I=1,IT
IF(IG.EQ.1)SUMT(I)=0.0
303 SUMZ(I)=0.0
DO 328 L=1,IZM
DO 307 I=1,IT
307 T(L,I)=0.0
DO 309 I=1,IM
IF(MA(I).NE.L) GO TO 309
U=V(I)
AN=XNN(I)
C
FIXED SOURCE
IF(IQM.EQ.0) GO TO 381
T(L,1)=T(L,1)+U*Q(I,IG)
381 LA=IABS(MZ(L))
C
FISSIONS
IF(ITH.GT.0) GO TO 386
T(L,2)=T(L,2)+U*F(I)*XKE(IG)
GO TO 389
386 T(L,2)=T(L,2)+U*F(I)*C(IHT-1,IG,LA)
389 T(L,3)=T(L,3)+ST(I)
C
SELF-SCATTER
T(L,4)=T(L,4)+CS(I)*AN
T(L,5)=T(L,5)+CT(I)*AN
C
ABSORPTION
T(L,6)=T(L,6)+AN*V(I)*C(IHT-2,IG,LA)
C
T(L,6)=T(L,6)+CA(I)*AN
SRI=0.
AZ=C(IHT-4,IG,LA)
IF(MBK)2763,2764,2765
2763 IF(MBK.EQ.(-2))GO TO 2766
NBKL=JBKL+IG-1
GO TO 2767
2766 NBKL=JBKL+MA(I)-1+IZM*(IG-1)
2767 SRI=A(NBKL)/(3.*AZ)
GO TO 2764
2765 IF(DY.GT.RLOG)SRI=((BF/(DY*AZ+XT))**2)*AZ
IF(DZ.GT.RLOG)SRI=SRI+((BF/(DZ*AZ+XT))**2)*AZ
2764 T(L,16)=T(L,16)+AN*V(I)*SRI
T(L,17)=T(L,17)+AN*V(I)*TAB
C
TOTAL FLUX AND FISSION DENSITY
T(L,9)=T(L,9)+AN*U
T(L,15)=T(L,15)+AN*U*C(IHT-1,IG,LA)
309 CONTINUE
LMAX=0
LMIN=IM+2
DO 310 I=1,IM
IF(MA(I).NE.L) GO TO 310
IF(I.GE.LMAX)LMAX=I
IF(LMIN.GT.I)LMIN=I

```

```

310 CONTINUE
DO 311 M=1,MM
C LEFT CURRENT
T(L,7)=T(L,7)+XND(LMIN,M)*WD(M)
E1=XND(LMAX+1,M)*W(M)
C RIGHT FLUX
T(L,11)=T(L,11)+E1
E=D(M)
IF(E.LE.RLOG) GO TO 305
C RIGHT FLOW
T(L,12)=T(L,12)+E*E1
C RIGHT CURRENTS
305 T(L,13)=T(L,13)+E*E1
IF(ISCT.EQ.0) GO TO 311
DO 906 N=1,NM
906 T(L,N+17)=T(L,N+17)+E1*CL(M,N)
311 CONTINUE
C RIGHT LEAKAGE
T(L,14)=T(L,13)*AV(LMAX+1)
C NET LEAKAGE
T(L,7)=T(L,14)-T(L,7)*AV(LMIN)
C IN SCATTER
T(L,3)=T(L,3)-T(L,4)-T(L,2)-T(L,1)
C CUT , SCATTER
T(L,5)=T(L,5)-T(L,4)-T(L,6)-T(L,16)-T(L,17)
C T(L,5)=T(L,5)-T(L,4)-T(L,6)
C NEUTRON BALANCE
GNENN=T(L,1)+T(L,2)+T(L,3)
T(L,10)=0.
C IF(GNENN.GT.1.E-13) T(L,10)=1. -(T(L,7)+T(L,6)+T(L,5))/GNENN
IF(GNENN.GT.1.E-13) T(L,10)=1. -(T(L,7)+T(L,6)+T(L,5)+T(L,16)
1 +T(L,17))/GNENN
C TOTAL DENSITY
T(L,8)=0.0
IF(ABS(A(JVE)).GT.RLOG) T(L,8)=T(L,9)/A(JVE)
DO 314 I=1,IT
SUMZ(I)=SUMZ(I)+T(L,I)
314 SUMT(I)=SUMT(I)+T(L,I)
328 CONTINUE
WRITE(KWT,329)(L,(T(L,I),I=1,8),L=1,IZM)
WRITE(KWT,330)(SUMZ(I),I=1,8)
WRITE(KWT,331)(L,(T(L,I),I=9,15),L=1,IZM)
WRITE(KWT,330)(SUMZ(I),I=9,15)
WRITE(KWT,350)(L,(T(L,I),I=16,17),L=1,IZM)
WRITE(KWT,330)(SUMZ(I),I=16,17)
350 FORMAT(///7H ZONE ,14HBUCKL.LEAKAGE ,15H TIME ABSORPT. //
1(I4,3X,2E14.6))
IF(ISCT.LE.1) GO TO 340
WRITE(KWT,332)
DO 334 L=1,IZM
C 334 WRITE(KWT,335)L,(T(L,I),I=16,IT)
334 WRITE(KWT,335)L,(T(L,I),I=18,IT)
C WRITE(KWT,330)(SUMZ(I),I=16,IT)
WRITE(KWT,330)(SUMZ(I),I=18,IT)
340 IF(IG.NE.IGM) GO TO 215
WRITE(KWT,333)
WRITE(KWT,329)L,(SUMT(I),I=1,8)
WRITE(KWT,331)L,(SUMT(I),I=9,15)
WRITE(KWT,350)L,(SUMT(I),I=16,17)

```

```

IF(ISCT.LE.1) GO TO 215
WRITE(KWT,332)
WRITE(KWT,335)L,(SUMT(I),I=18,IT)
C WRITE(KWT,335)L,(SUMT(I),I=16,IT)
GO TO 215
329 FORMAT(7H ZONE ,14H FIXED SOURCE ,14HFISSION SOURCE,
114H IN SCATTER ,14H SELF SCATTER ,14H OUT SCATTER ,
214H ABSORPTION ,14H NET LEAKAGE ,14H TOTAL DENSITY//
3(I4,3X,1P8E14.6))
330 FORMAT(7H0 TOTAL,1P8E14.6//)
331 FORMAT(7H ZONE ,14H TOTAL FLUX ,14H NEUTRON BAL ,
114H RIGHT FLUX ,14H RIGHT FLOW ,14H RIGHT CURRENT,
214H RIGHT LEAKAGE,14H FISSIION DENS //(I4,3X,1P7E14.6))
332 FORMAT(40HORIGHT CURRENTS - ANISOTROPIC SCATTERING,
122H ISCT GREATER THAN ONE//7H ZONE )
333 FORMAT(1H0//29H TOTAL BALANCE FOR ALL GROUPS//)
335 FORMAT(I4,3X,1P8E14.6)
END
C -----
C -----
C
C PERFORMS INNER ITERATION
SUBROUTINE INNER (SAT,SA,CH,XJJ,CL,XND,DA,DB,DC,DS,IM,MM,ISCT,NM,
1IP,XNA,XNN,ST,CS,SR,W,D,XNI,XNO,MR,XNE,CT,WD,AV,XNR,XNOR,XJJR,
2 FUNK, IDIM)
DIMENSION FUNK(1),IDIM(1)
DIMENSION SAT(IM,MM),SA(IM,MM),CH(IM,ISCT),XJJ(IM,NM),
1CL(MM,NM),XND(IP,MM),DA(IP,MM),DB(IP,MM),DC(IP,MM),XJJR(IP,MM),
2DS(IP,MM),XNA(IM),XNN(IM),ST(IM),CS(IM),SR(IM),W(MM),D(MM),XNI(MM)
3,XNO(MM),MR(MM),XNE(MM),CT(IM),WD(MM),AV(IP),
4XNR(IM),XNOR(MM)
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
COMMON/TSCH/ITSCH,IANF,RHO,ITM,INORM,ROM
EQUIVALENCE(IG,IA(36)),(IGE,IA(5)),(LC,IA(32)),(IBR,IA(7)),
1(IBL,IA(6)),(EPG,A(17)),(IIC,IA(37)),(NFF,IA(28)),(NFT,IA(50)),
2(IIL,IA(30)),(IQM,IA(21)),(EPSA,A(4)),(ICVT,IA(39)),
3(JNLA,IA(159)),(JRLA,IA(160)),(R,A(14))
EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
IDIFF=LC
JNL=JNLA+IG
JRL=JRLA+IG
C COMPUTE SOURCE FOR GROUP BY INCLUDING SELF SCATTER
832 DO 8 I=1,IM
XNA(I)=XNN(I)
ST(I)=XNN(I)*CS(I)+SR(I)
XNN(I)=0.0
IF(ISCT.LT.1) GO TO 8
DO 1 M=1,MM
1 SAT(I,M)=SA(I,M)
DO 7 N=1,ISCT
E1=(FLOAT(N)+.5)*CH(I,N)
IF(IGE.EQ.2) GO TO 3
E2=E1*XJJ(I,N)
DO 2 M=1,MM
2 SAT(I,M)=SAT(I,M)+E2*CL(M,N)
GO TO 7
3 IRS=((N+1)*(N+1))/4

```

```

      IRT=IRS+(2*N+1)/4
      DO 4 IR=IRS,IRT
      E2=E1*XJJ(I,IR)
      DO 4 M=1,MM
4     SAT(I,M)=SAT(I,M)+E2*CL(M,IR)
      7 CONTINUE
      8 CONTINUE
C     COMPUTE ANGULAR FLUX AND NEUTRON SUMS
      ITRIG=0
C     COMPUTE ANGULAR FLUX BOUNDARY CONDITIONS
842   LC=LC+1
      DO 425 M=1,MM
      IF(D(M).GE.RLOG) GO TO 114
      IT=IP
      IS=IP+1
      IF(IBR.EQ.1) GO TO 106
      XNI(M)=XNO(M)
      GO TO 108
106   K=MR(M)
      XNO(M)=XNO(K)
      XNI(M)=XNO(K)
108   IF(W(M).LE.RLOG) GO TO 109
      GO TO 113
109   DO 112 I=1,IM
112   XNE(I)=0.0
113   XNMM=XNI(M)
      GO TO 843
114   IT=0
      IS=0
      IF(IBL-1)118,115,120
115   K=MR(M)
      XNO(M)=XNI(K)
      XNI(M)=XNI(K)
      GO TO 121
118   XNO(M)=0.0
      XNI(M)=0.0
      GO TO 121
120   XNI(M)=XNO(M)
121   XNMM=XNO(M)
C     COMPUTE INWARD OR OUTWARD NEUTRON FLOW
843   DO 420 J=1,IM
      I=IABS(IT-J)
      K=IABS(IS-J)
      XND(K,M)=XNMM
      E1=ST(I)
      IF(ISCT.GT.0)E1=E1+SAT(I,M)
      E2=(XNMM*DA(I,M)+XNE(I)*DC(I,M)+E1)/(DB(I,M)+CT(I))
      E3=XNE(I)
      E4=XNMM
      XNE(I)=E2+E2-XNE(I)
      XNMM=E2+E2-XNMM
      IF(ABS(W(M)).LT.RLOG) XNE(I)=E2
      IF(XNE(I).GE.RLOG.AND.XNMM.GE.RLOG) GO TO 420
C     NEGATIVE FLUX FIXUP
      IF(XNE(I).GE.RLOG) GO TO 415
      IF(XNMM.GE.RLOG.OR.XNMM.GE.XNE(I)) GO TO 416
415   XNMM=0.0
      E2=((DA(I,M)-DS(I,M))*E4+DC(I,M)*E3+E1)/(CT(I)+DB(I,M)-2.*DS(I,M))
      XNE(I)=E2+E2-E3

```

```

IF(ABS(W(M)).LT.RLOG) XNE(I)=E2
IF(XNE(I).GE.RLOG) GO TO 418
GO TO 417
416 XNE(I)=0.0
E2=(DA(I,M)*E4+(DC(I,M)-.5*DB(I,M)+DS(I,M))*E3+E1)/
1(CT(I)+2.*DS(I,M))
XNMM=E2+E2-E4
IF(XNMM.GE.RLOG) GO TO 418
417 IF(ABS(CT(I)).LT.RLOG) GO TO 418
XNMM=0.0
XNE(I)=0.0
E2=((DA(I,M)-DS(I,M))*E4+(DC(I,M)-.5*DB(I,M)+DS(I,M))*E3+E1)/CT(I)
418 IF(ICVT.LT.1) GO TO 420
NFT=1
IF(NFF.GT.0) GO TO 420
WRITE(KWT,419)IG,M,I,XNE(I),XNMM
419 FORMAT(27H NEGATIVE FLUX FIXUP,GROUP=,I4,12H, DIRECTION=,I3,
113H,SPACE POINT=,I5,10X,8H XNE(I)=,E14.5,6H XNMM=,E14.5)
420 XNN(I)=XNN(I)+W(M)*E2
I=IABS(IS-IP)
XND(I,M)=XNMM
IF(D(M).GE.RLOG) GO TO 421
XNI(M)=XNMM
GO TO 425
421 XNO(M)=XNMM
425 CONTINUE
C COMPUTE NEUTRON SUMS
IF(IBR.NE.2) GO TO 507
DO 506 M=1,MM
IF(D(M).LE.RLOG) XNO(M)=XNI(M)
506 CONTINUE
507 E1=0.0
E2=0.0
DO 512 M=1,MM
E1=E1+WD(M)*XNI(M)
512 E2=E2+WD(M)*XNO(M)
A(JRL)=E2*AV(IP)
A(JNL)=A(JRL)-E1*AV(1)
IF(ITRIG.GT.0) GO TO 847
IF(IBR.LT.1.OR.A(JNL).LE.EPG) GO TO 833
C COMPUTE BOUNDARY SOURCE
834 DO 606 I=1,IM
XNR(I)=XNN(I)
XNN(I)=0.0
ST(I)=0.0
DO 606 M=1,MM
606 SAT(I,M)=0.0
DO 607 I=1,IP
DO 607 M=1,MM
607 XJJR(I,M)=XND(I,M)
DO 612 M=1,MM
612 XNOR(M)=XNO(M)
XNLB=A(JNL)
RLB=A(JRL)
ITRIG=1
GO TO 842
C COMBINE FLUX COMPONENTS
847 E1=-XNLB/A(JNL)
A(JNL)=0.0

```

```

      A(JRL)=E1*A(JRL)+RLB
      DO 707 I=1,IM
707  XNN(I)=E1*XNN(I)+XNR(I)
      DO 712 M=1,MM
712  XNO(M)=E1*XNO(M)+XNR(M)
      DO 714 I=1,IP
      DO 714 M=1,MM
714  XND(I,M)=E1*XND(I,M)+XJJR(I,M)
C     PERFORM INNER ITERATION
833  IIC=IIC+1
      E1=0.0
      E2=0.0
      E4=0.0
      E3=0.0
      DO 808 I=1,IM
      TM=XNN(I)-XNA(I)
      E1=E1+TM*CS(I)
      E2=E2+ABS(TM)*CS(I)
      E3=E3+ABS(TM)*(CT(I)-CS(I))
      UIS=RLOG
      IF(XNA(I).GT.1.E-10) UIS=TM/XNA(I)
      UIS=ABS(UIS)
      IF(EPSA.GT.RLOG) E4=AMAX1(E4,UIS)
808  CONTINUE
      E1=R/(R-E1)
      IF(IBR.EQ.(-1))E1=1.
      A(JNL)=A(JNL)*E1
      A(JRL)=A(JRL)*E1
      DO 815 I=1,IM
815  XNN(I)=XNN(I)*E1
      DO 819 M=1,MM
819  XNO(M)=XNO(M)*E1
      IF(IIC.GT.IIL) GOTO 826
      IF((E2-EPG).GT.RLOG.OR.(E3-EPG).GT.RLOG) GO TO 831
      IF(ABS(EPSA).LE.RLOG) GOTO 820
      IF((E4-EPSA).GT.RLOG) GO TO 831
820  IF(ITRIG.EQ.0) GO TO 826
C     COMPUTE CURRENTS
831  DO 901 M=1,MM
      E2=W(M)*E1
      DO 901 N=1,NM
      E3=E2*CL(M,N)
      DO 901 I=1,IM
      IF(M.EQ.1)XJJ(I,N)=0.0
901  XJJ(I,N)=XJJ(I,N)+E3*(XND(I,M)+XND(I+1,M))
      GO TO 832
C     FLUX CONVERGED
826  DO 830 I=1,IP
      DO 830 M=1,MM
830  XND(I,M)=E1*XND(I,M)
      FUNK(IG)=E1
      IDIM(IG)=LC-IDIFF
      RETURN
      END
-----
C-----
C-----
C
C     COMPUTE AND PRINT DISCRETE ORDINALES CONSTANTS
      SUBROUTINE SNCN (W,D,WD,CL,MR,AB,Z,CP,AF,CT,MM,NM,ISC,IFM)

```

```

DIMENSION W(MM),D(MM),WD(MM),CL(MM,NM),MR(MM),AF(IFM),
1AB(MM),Z(MM),CP(MM),CT(MM,ISC,ISC)
COMMON LAENGE,AX(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE (IA(80),KRD),(IA(96),KWT),(AX(28),RLOG)
IS=ISC-1
C FORM MATED DIRECTIONS AND TEST INPUT DATA FOR ERRORS
E1=0.0
M=MM
5 B=W(M)
C=D(M)
A=B*C
WD(M)=A
E1=E1+A+B
IF(A.GT.RLOG) GO TO 18
IF((0.0-A).GT.RLOG) GO TO 12
8 IF(ABS(C).LT.RLOG) GO TO 66
MR(M)=MR(M+1)
18 M=M-1
IF(M.GT.0) GO TO 5
IF(.0001-ABS(1.-E1).GE.RLOG) GO TO 21
WRITE(KWT,70)
GO TO 67
12 K=M
13 K=K+1
IF(K.GT.MM) GO TO 68
IF(.0001-ABS(D(K)+C).LT.RLOG) GO TO 13
MR(K)=M
MR(M)=K
GO TO 18
C CHECK FOR ANISOTROPIC SCATTERING
21 IF(IS.EQ.0) GO TO 65
C CHECK GEOMETRY TYPE
IF(NM.GT.IS) GO TO 44
C SPHERES AND SLABS (IF LINEAR SCATTER,CYLINDER ALSO)
DO 41 M=1,MM
41 CL(M,1)=D(M)
IF(IS.EQ.1) GO TO 65
DO 42 M=1,MM
42 CL(M,2)=.5*(3.*D(M)**2-1.)
IF(IS.EQ.2) GO TO 39
DO 43 N=3,IS
A=N
B=1.-1./A
C=B+1.
DO 43 M=1,MM
43 CL(M,N)=C*CL(M,N-1)*D(M)-B*CL(M,N-2)
39 WRITE(KWT,72)
GO TO 62
C CYLINDERS
C GENERATE XI FUNCTIONS AND ANGLE PHI
44 DO 47 M=1,MM
A=D(M)
IF(W(M).GT.RLOG) GO TO 45
Z(M)=SQRT(1.-A**2)
AB(M)=ABS(A)
GO TO 46
45 Z(M)=Z(M-1)
AB(M)=AB(M-1)

```

```

46 B=ATAN(SQRT(1.-Z(M)**2-A**2)/A)
   IF(A.LT.RLOG) B=B+3.14159
   IF(ABS(A).LT.RLOG) B=1.57079
47 CP(M)=B
   DO 49 M=1,MM
C  GENERATE COEFFICIENTS FOR GENERAL SCATTER
   CT(M,1,1)=1.0
   CT(M,2,1)=Z(M)
   DO 49 N=2,IS
   A=N
   B=1.-1./A
   C=B+1.
49 CT(M,N+1,1)=C*CT(M,N,1)*Z(M)-B*CT(M,N-1,1)
   DO 53 M=1,MM
   DO 53 J=2,ISC
   DO 53 N=1,ISC
   A=2*J-3
   IF(N-J)53,51,52
51 CT(M,N,J)=AB(M)*CT(M,N-1,J-1)*A
52 IF(N.EQ.ISC) GO TO 53
   A=N+J-2
   B=N-J+1
   C=2*N-1
   CT(M,N+1,J)=(C*Z(M)*CT(M,N,J)-A*CT(M,N-1,J))/B
53 CONTINUE
   AF(1)=1.
   DO 50 I=2,IFM
50 AF(I)=FLOAT(I-1)*AF(I-1)
   DO 501 J=2,ISC
   A=J-1
   DO 501 N=J,ISC
   K=N+J-1
   KA=N-J+1
   B=SQRT(2.*AF(KA)/AF(K))
   DO 501 M=1,MM
501 CT(M,N,J)=B*CT(M,N,J)*COS(A*CP(M))
C  STORE FUNCTIONS IN CL
   IR=1
   DO 54 N=2,ISC
   DO 54 J=1,N
   K=(N+J)/2
   KA=(N+J+1)/2
   IF(K.NE.KA) GO TO 54
   DO 55 M=1,MM
55 CL(M,IR)=CT(M,N,J)
   IR=IR+1
54 CONTINUE
   WRITE(KWT,75)
   DO 61 J=1,ISC
   JJ=J-1
   NA=1
   NB=9
59 NC=MINO(NB,ISC)
   WRITE(KWT,76)(N,JJ,N=NA,NC)
   DO 60 M=1,MM
60 WRITE(KWT,74)M,(CT(M,N,J),N=NA,NC)
   NA=NA+9
   NB=NB+9
   IF(NA.LE.ISC) GO TO 59

```



```

61 CONTINUE
   GO TO 65
62 NA=1
   NB=9
63 NC=MINO(NB,IS)
   WRITE(KWT,73)(N,N=NA,NC)
   DO 64 M=1,MM
64 WRITE(KWT,74)M,(CL(M,N),N=NA,NC)
   NA=NA+9
   NB=NB+9
   IF(NA.LE.IS) GO TO 63
65 WRITE(KWT,77)(M,MR(M),W(M),D(M),WD(M),M=1,MM)
   RETURN
66 WRITE(KWT,69)
67 STOP
68 WRITE(KWT,71)
   GO TO 67
69 FORMAT(28HOINCORRECT DIRECTION COSINES)
70 FORMAT(18HOINCORRECT WEIGHTS)
71 FORMAT(26HORAY INDEX EXCEEDS MAXIMUM)
72 FORMAT(37HOANISOTROPIC SCATTER- SLAB OR SPHERE ,
  127HLEGENDRE POLYNOMIALS PN(MU))
73 FORMAT(17HODIRECTION NUMBER,9(3X,2HN=I3,3X))
74 FORMAT(7X,I5,5X,1P9E11.4)
75 FORMAT(41HOANISOTROPIC SCATTER-CYLINDER, ASSOCIATED,
  138H LEGENDRE POLYNOMIALS P(N,R)COS(R*PHI)//)
76 FORMAT(17HODIRECTION NUMBER,9(1X,2HN=I2,3H R=I2,1X))
77 FORMAT(14HODIRECTION NO.,25H REFLECTED DIRECTION NO.,
  15X,6HWEIGHT,7X,20H DIRECTION COSINE ,20H WEIGHT X DIRECTION //
  2(5X,I4,13X,I4,14X,1PE14.7,6X,E14.7,6X,E14.7))
   END

```

C-----

C-----

C

```

C COMPUTE INITIAL FUNCTIONS
  SUBROUTINE IFUNC(XKI,XKE,VE,IGP,IQM,PV,IPVT,IEVT)
  DIMENSION XKI(IGP),XKE(IGP),VE(IGP)
  COMMON LAENGE,A(1)
  COMMON/FIX/LEN,IA(1)
  EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
  IGM=IGP-1
  IF(IQM.EQ.0.AND.IEVT.LE.0) GO TO 23
  IF(IPVT.NE.1)GOTO17
  IF(ABS(PV).LT.RLOG) GO TO 25
  DO 16 I=1,IGM
16  XKI(I)=XKI(I)/PV
17  T=0.
   DO22 I=1,IGM
   XKE(I)=XKI(I)
   T=T+XKE(I)
   IF(IEVT.NE.2) GO TO 27
   IF(ABS(VE(I)).LT.RLOG) GO TO 26
27 CONTINUE
22 CONTINUE
   XKE(IGP)=T
   XKI(IGP)=T
   WRITE(KWT,1)(XKI(I),XKE(I),VE(I),I,I=1,IGP)
   RETURN
23 WRITE(KWT,2)

```

```

24 STOP
25 WRITE(KWT,3)
   GOTO24
26 WRITE(KWT,4)I
   GO TO 24
1  FORMAT(1H0/8X,3HCHI,12X,6HCHI/PV,7X,12H VELOCITIES ,
17H  GROUP//(3E16.7,I5))
2  FORMAT(44HONO DISTRIBUTED SOURCE SPECIFIED FOR SOURCE ,
112HTYPE PROBLEM //)
3  FORMAT(17HOPV=0 WHEN IPVT=1 //)
4  FORMAT(22HOZERO VELOCITY, GROUP=,I4//)
   END

```

```

C-----
C-----
C

```

C FISSION CALCULATION AND NORMALIZATIONS

```

SUBROUTINE FISSN(XN,C,XJ,B,IGM,IM,IHM,MT,NM,MM,FG,IGP,F,XKE,MA,MZ
1,IZM,V,QG)
  DIMENSION XN(IM,IGM),C(IHM,IGM,MT),XJ(IGM,IM,NM),B(IGM,MM),
1FG(IGP),F(IM),XKE(IGP),MA(IM),MZ(IZM),V(IM),QG(IGP)
  COMMON LAENGE,A(1)
  COMMON/FIX/LEN,IA(1)
  COMMON/TSCH/ITSCH,IANF,RHO,ITM,INORM,ROM
  EQUIVALENCE(IHT,IA(13)),(FTP,A(15)),(IA(2),ITH),(IA(34),NFN),
1(XLA,A(16)),(IEVT,IA(11)),(XNF,A(8)),(IBR,IA(7)),
2(IA(10),IFN),(A(17),EPG),(EPS,A(3))
  EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
  EQUIVALENCE(IFISS,IA(199)),(ICC,IA(33)),(EV,A(1))
  EQUIVALENCE(ICVT,IA(39))
  IF(ICVT.NE.1.OR .INORM.NE.1)GO TO 400
  E1=0.
  E2=0.
  DO 428 I=1,IM
    L=MA(I)
    L=IABS(MZ(L))
    DO 428 J=1,IGM
      UW=V(I)*XN(I,J)
      E1=E1+UW*C(IHT-3,J,L)
428 E2=E2+UW*C(IHT-1,J,L)
    XNUE=E2/E1
    WRITE(KWT,402)XNUE
402 FORMAT(////////10X,20H MITTLERES NUE      = ,E15.7  )
    XNF=XNUE*3.125E+10
400 CONTINUE
    IF(ICC.EQ.0.AND.IFISS.GT.0)GO TO 20
21 CONTINUE
20 CONTINUE
    IH=IHT-1
    FTP=FG(IGP)
    IF(IFN.GT.0)GOTO110
    DO 107 J=1,IM
    DO 107 I=1,IGM
107 XN(J,I)=0.0
    IFN=1
    GO TO 129
110 DO 113 I=1,IM
113 F(I)=0.0
    IF(ITH.LE.0)GO TO 123
    DO 118 I=1,IM

```

```

DO 118 J=1,IGM
118 F(I)=F(I)+XKE(J)*XN(I,J)
DO 119 J=1,IGM
FG(J)=0.0
DO 119 I=1,IM
L=MA(I)
L=IABS(MZ(L))
119 FG(J)=FG(J)+V(I)*F(I)*C(IH,J,L)
GO TO 220
123 DO 128 I=1,IM
L=MA(I)
L=IABS(MZ(L))
DO 128 J=1,IGM
128 F(I)=F(I)+XN(I,J)*C(IH,J,L)
129 E1=0.0
DO 215 I=1,IM
215 E1=E1+V(I)*F(I)
DO 219 J=1,IGM
219 FG(J)=E1*XKE(J)
220 BB=0.0
DO 224 J=1,IGM
224 BB=BB+FG(J)
FG(IGP)=BB
IF(ICC.EQ.0.AND.IFISS.GT.0)GO TO 10
E1=FTP+QG(IGP)
IF(NFN.EQ.1) GOTO248
IF(E1.LE.RLOG) GO TO 331
XLA=(FG(IGP)+QG(IGP))/E1
GO TO 11
10 XLA=EV
11 CONTINUE
IF(IEVT.EQ.0)GOTO248
IF(FG(IGP).LE.RLOG) GO TO 332
IF(IEVT.NE.1) GO TO 248
BB=1./XLA
DO 241 J=1,IGP
FG(J)=BB*FG(J)
241 XKE(J)=BB*XKE(J)
IF(ITH.LE.0) GOTO248
DO 247 I=1,IM
247 F(I)=BB*F(I)
248 NFN=0
IF(IEVT.NE.0) GOTO 308
304 EPG=(QG(IGP)+FG(IGP))*4.*EPS/FLOAT(IGM+3)
RETURN
308 IF(XNF.LE.RLOG) GO TO 304
E1=XNF/FG(IGP)
FG(IGP)=XNF
DO 316 J=1,IGM
FG(J)=E1*FG(J)
DO 316 I=1,IM
XN(I,J)=E1*XN(I,J)
DO 316 N=1,NM
316 XJ(J,I,N)=XJ(J,I,N)*E1
DO 325 I=1,IM
325 F(I)=E1*F(I)
IF(IBR.LT.1) GO TO 304
DO 330 J=1,IGM
DO 330 M=1,MM

```

```

330 B(J,M)=B(J,M)*E1
    GO TO 304
331 WRITE(KWT,1)
    GO TO 333
332 WRITE(KWT,2)
    1 FORMAT(1X,39HND DISTRIBUTED SOURCE OR FISSION SOURCE//)
    2 FORMAT(1X,17HND FISSION SOURCE//)
333 STOP
    END

```

C-----

C-----

C

C MODIFY RADII AND COMPUTE AREAS, VOLUMES, AND GEOMETRIC FUNCTIONS AND

C PRINT AREAS AND VOLUMES

```

SUBROUTINE RMAVGF(RA,AA,R,V,RM,RAV,MA,W,D,DA,DB,DC,DS,
1IP,IM,IZM,MM)
DIMENSION RA(IP),R(IP),RM(IZM),MA(IM),AA(IP),V(IP),
1RAV(IP),W(MM),D(MM),DA(IP,MM),DB(IP,MM),DC(IP,MM),DS(IP,MM)
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE (IGE,IA(5)),(ICC,IA(33)),(IEVT,IA(11)),(EV,A(1))
EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
IF(ICC.NE.0)GOTO8
DO 7 I=1,IM
RA(I+1)=R(I+1)
IF(ABS(RA(I+1)).LT.RLOG) GO TO 24
IF((R(I+1)-R(I)).LE.RLOG) GO TO 25
7 CONTINUE
RA(1)=0.0
8 IF(IEVT-4)201,10,20
10 DO 18 I=1,IM
K=MA(I)
B=1.+EV*RM(K)
RA(I+1)=RA(I)+(R(I+1)-R(I))*B
IF(B.LT.RLOG) GO TO 26
18 CONTINUE
GO TO 201
20 DO 22 J=1,IP
22 RA(J)=EV*R(J)/R(IP)
GO TO 201
24 WRITE(KWT,28)
GO TO 27
25 WRITE(KWT,29) I
GO TO 27
26 WRITE(KWT,30)
27 STOP
28 FORMAT(31HORADIUS LESS THAN OR EQUAL ZERO)
29 FORMAT(31HOR(I+1) LESS THAN OR EQUAL R(I),2HI=I3)
30 FORMAT(39HNEGATIVE FINAL RADIUS FOR DELTA OPTION)
201 IF(ICC.NE.0.AND.IEVT.LE.3)GOTO500
IF(IGF.NE.1)GOTO100
DO 101 J=1,IP
101 AA(J)=1.0
GO TO 102
100 AF=A(IGE+20)
IB=IGE-1
DO 14 J=1,IP
14 AA(J)=AF*RA(J)**IB
102 VF=A(IGE+23)

```

```

17 DO 21 I=1,IM
   RAV(I)=0.5*(RA(I+1)+RA(I))
21 V(I)=VF*(AA(I+1)*RA(I+1)-AA(I)*RA(I))
   IF(ICC.GT.0)GOTO23
   WRITE(KWT,1)(RAV(I),RA(I),V(I),AA(I),I,I=1,IP)
   1 FORMAT(1H0/6X,10HAVG RADIUS,10X,6HRADIUS,10X,6HVOLUME,
112X,4HAREA//(4E16.7,I5))
23 DO 12 M=1,MM
   AB=D(M)
   AC=ABS(AB)
   AD=W(M)
   AE=AB*AD
   AF=W(M-1)
   AG=D(M-1)*AF
   DO12 I=1,IM
   BB=AA(I+1)
   BC=AA(I)
   DA(I,M)=AC*(BB+BC)
   IF(AB.LE.RLOG) GO TO 83
   DS(I,M)=BB*AC
   GO TO 84
83 DS(I,M)=BC*AC
84 IF(ABS(AD).GT.RLOG) GO TO 11
   DC(I,M)=0.0
   GO TO 12
11 DC(I,M)=((BC-BB)*(AE+AG)+DC(I,M-1)*AF)/AD
12 DB(I,M)=DA(I,M)+DC(I,M)
500 RETURN
   END

```

C-----

C-----

C

C COMPUTES DISTRIBUTED SOURCE

```

SUBROUTINE DSOUR(Q,QG,V,IM,IGM,IGP,IP,XNF)
DIMENSION Q(IM,IGM),QG(IGP),V(IP)
COMMON LAENGE,AX(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE (IA(80),KRD),(IA(96),KWT),(AX(28),RLOG)
EQUIVALENCE(IBR,IA(7))
B=0.0
DO3J=1,IGM
A=0.0
DO2I=1,IM
2 A=A+Q(I,J)*V(I)
  QG(J)=A
3 B=B+A
  IF(B.LE.RLOG.AND.IBR.NE.(-1)) GO TO 11
  QG(IGP)=B
  IF(XNF.LE.RLOG) GO TO 77
  E=XNF/B
  DO7J=1,IGM
  QG(J)=QG(J)*E
  DO 7 I=1,IM
7 Q(I,J)=Q(I,J)*E
  QG(IGP)=XNF
77 DO8J=1,IGM
8 WRITE(KWT,9)J,(Q(I,J),I=1,IM)
9 FORMAT(19H0DISTRIBUTED SOURCE,5X,6H GROUP,I3/(1P10E12.5))
WRITE(KWT,10)(QG(I),I=1,IGP)

```

```

10 FORMAT(19H0GROUP TOTAL SOURCE/(1P10E12.5))
   RETURN
11 WRITE(KWT,12)
12 FORMAT(36H0ZERO OR NEGATIVE DISTRIBUTED SOURCE//)
   STOP
   END

```

```

C-----
C-----
C
C
C MIX CROSS SECTIONS AND PRINT
  SUBROUTINE MIXCX(C,MB,MC,XMD,IHM,IGM,MT,MS,EV,IEVT,ICC)
  DIMENSION C(IHM,IGM,MT),MB(MS),MC(MS),XMD(MS)
  COMMON LAENGE,AX(1)
  COMMON/FIX/LEN,IA(1)
  EQUIVALENCE(IQUER,IA(198))
  EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(AX(28),RLOG)
  IF(MS.EQ.0) GOTO20
  DO 18 M=1,MS
  K=MB(M)
  L=MC(M)
  A=XMD(M)
  DO 18 I=1,IHM
  DO 18 J=1,IGM
  IF(L)16,15,16
15 C(I,J,K)=C(I,J,K)*A
  GO TO 18
16 IF(ABS(A).LE.RLOG) GO TO 17
19 C(I,J,K)=C(I,J,K)+A*C(I,J,L)
  GOTO18
17 C(I,J,K)=C(I,J,K)*EV
18 CONTINUE
  IF(IEVT.EQ.3.AND.ICC.NE.0) GO TO 53
  WRITE(KWT,201)(MB(I),MC(I),XMD(I),I,I=1,MS)
20 IF(IQUER.EQ.0)GO TO 53
  DO 52 I=1,MT
  WRITE(KWT,101) I
  KA=1
  KB=8
51 KC=MINO(KB,IGM)
  WRITE(KWT,102)(K,K=KA,KC)
  WRITE(KWT,103)
  DO 510 J=1,IHM
510 WRITE(KWT,104)J,(C(J,K,I),K=KA,KC)
  KA=KA+8
  KB=KB+8
  IF(KA.LE.IGM) GO TO 51
52 CONTINUE
53 RETURN
101 FORMAT(1H0///26H CROSS SECTION OF MATERIAL,I3/)
102 FORMAT(1H0,5X,8(2X,5HGROUP,I3,3X))
103 FORMAT(1H )
104 FORMAT(I4,1P8E13.5)
201 FORMAT(1H0/16H MIXTURE NUMBER ,16H MIX COMMAND ,
  124H MATERIAL ATOMIC DENSITY/(4X,I8,8X,I8,8X,E16.8,8X,I8))
  END

```

```

C-----
C-----
C-----

```

C COMPUTES ADJOINT REVERSALS AND CROSS SECTIONS

```

SUBROUTINE ADJREV(XKI,VE,Q,XN,IQM,IFN,IGM,IM,LC,
  1C,IHM,MT,ML,IHT,IHS)
  DIMENSION XKI(IGM),VE(IGM),Q(IM,IGM),XN(IM,IGM),
  1C(IHM,IGM,MT)
  IG=1
  K=IGM
  4 IF(K.LE.IG) GOTO10
  T=XKI(IG)
  TA=VE(IG)
  XKI(IG)=XKI(K)
  VE(IG)=VE(K)
  XKI(K)=T
  VE(K)=TA
  IG=IG+1
  K=K-1
  GO TO 4
  10 IF(IQM.EQ.0) GO TO 9
  DO 8 I=1,IM
  IG=1
  K=IGM
  6 IF(K.LE.IG) GOTO8
  T=Q(I,IG)
  Q(I,IG)=Q(I,K)
  Q(I,K)=T
  K=K-1
  IG=IG+1
  GO TO 6
  8 CONTINUE
  9 IF(LC.GE.IFN) GO TO 12
  DO 13 I=1,IM
  IG=1
  K=IGM
  11 IF(K.LE.IG)GOTO13
  T=XN(I,IG)
  XN(I,IG)=XN(I,K)
  XN(I,K)=T
  K=K-1
  IG=IG+1
  GO TO 11
  13 CONTINUE
  12 DO17I=1,ML
  DO 17 JJ=1,IHM
  J=IHM+1-JJ
  IG=1
  K=IGM
  IF(J.LE.IHT)GOTO14
  IA=J-IHS
  IF(IA.LE.0)GOTO15
  IG=IA+1
  GO TO 14
  15 IA=IA+IGM
  IF(IA.LE.0)GOTO17
  K=IA
  14 IF(K.LE.IG)GOTO17
  T=C(J,K,I)
  C(J,K,I)=C(J,IG,I)
  C(J,IG,I)=T
  IG=IG+1

```

```

K=K-1
GO TO 14
17 CONTINUE
RETURN
END

```

```

C-----
C-----
C

```

C READS CROSS SECTIONS

```

SUBROUTINE RECS(CS, IHM, IGM, MT)
DIMENSION CS(IHM, IGM, MT), LD(72)
COMMON LAENGE, A(1)
COMMON/FIX/LEN, IA(1)
DIMENSION IU(1)
EQUIVALENCE(A(1), IU(1))
EQUIVALENCE(KD, IA(112)), (MCR, IA(17)), (JA, IA(52)), (JB, IA(53)),
1(IA(18), MTP), (IA(48), ML), (IA(13), IHT), (IA(14), IHS), (A(3), EPS)
EQUIVALENCE(IA(80), KRD), (IA(96), KWT), (A(28), RLOG)
IF(IA(1).LE.0) GO TO 12
DO1 IMT=1, MCR
READ(KRD, 3)((CS(I, J, IMT), I=1, IHM), J=1, IGM)
3 FORMAT(6E12.5)
1 CONTINUE
GO TO 11
12 CONTINUE
JFIN=IA(195)
IBEG=JFIN+10
IRIS=LAENGE-10-JFIN
CALL LIES2(CS, IHM, IGM, MT, A(1), IU(1), IRIS)
11 CONTINUE
IHA=IHT
IF(IHS.GT.IHT+1) IHA=IHT+1
DO 205 I=1, MT
IF(CS(IHT, 1, I).LE.RLOG) GO TO 205
DO 204 J=1, IGM
B=CS(IHT, J, I)-CS(IHT-2, J, I)
JC=IHS-J+1
JD=1
201 IF(JC.GT.IHA) GOTO202
JC=JC+1
JD=JD+1
GO TO201
202 B=B-CS(JC, JD, I)
JC=JC+1
JD=JD+1
IF(JD.LE.IGM.AND.JC.LE.IHM) GO TO 202
B=ABS(B)/CS(IHT, J, I)
IF(B.LE.EPS) GOTO204
WRITE(KWT, 203) I, J, B
203 FORMAT(12HOIN MATERIAL, I3, 6H, GROUP, I3, 40H, (1, 0-CALC TOT X-SECT/INP
1UTTOT X-SECT)= , E13.6)
204 CONTINUE
205 CONTINUE
111 RETURN
END

```

```

C-----
C-----
C

```

C COMPUTES NEW PARAMETERS FOR IMPLICIT EIGENVALUE SEARCH


```

SUBROUTINE NEWPAR
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE(A(16),XLA),(A(11),XLAL),(A(12),XLAH),(A(13),XNPM),
1(A(10),XEPS),(A(3),EPS),(A(27),XLAR),(A(31),EQP),
X(A(32),EQ),
2(A(1),EV),(A(33),EVP),(A(34),E1),(A(35),E2),(IA(39),ICVT),
3(A(2),EVM),(IA(31),KS852),(IA(11),IEVT),(A(36),EVPP),
4(A(37),XLAPP),(A(30),XLAP),(IA(49),ICNT)
EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
E3=ABS(XLA-XLAR)
IF(ABS(XLAPP).LE.RLOG) GO TO 2
IF(E3.GT.XEPS)GOTO16
D=EVPP-EVP
E=EVPP-EV
F=EVP-EV
DEN=D*E*F
EQA=((XLAPP-1.)*F*EVP*EV-(XLAP-1.)*E*EV*EVPP+(XLA-1.)*D*EVPP*EVP)/
1DEN
EQB=-((XLAPP*F*(EVP+EV)-XLAP*E*(EV+EVPP)+XLA*(EVPP+EVP)*D)/DEN
EQC=(XLAPP*F-XLAP*E+XLA*D)/DEN
R=EQB**2-4.*EQA*EQC
IF(R.LE.RLOG) GOTO 3
IF(E2.LE.XLAL)GOTO8
EQ=1./((EQB+2.*EV*EQC)
XLAPP=XLAP
XLAP=XLA
EVPP=EVP
EVP=EV
EV1=(-EQB+SQRT(R))/(2.*EQC)
EV2=(-EQB-SQRT(R))/(2.*EQC)
IF(ABS(EV1-EV).GT.ABS(EV2-EV))GOTO1
EV=EV1
GOTO6
1 EV=EV2
GOTO6
2 IF(ABS(XLAP).LE.RLOG) GO TO 7
IF(E3.GT.XEPS)GOTO16
3 EQ=(EVP-EV)/(XLAP-XLA)
IF(ICNT.NE.0)GOTO4
IF(E2.LE.XLAL) GOTO8
IF(E2.LE.XLAH) GOTO4
E1=SIGN(XLAH,E1)
4 XLAPP=XLAP
XLAP=XLA
EVPP=EVP
EVP=EV
5 EV=EV+XNPM*EQ*E1
6 IF(((XLAPP-1.)/(XLAP-1.)).GT.RLOG) GO TO 12
B=AMAX1(EVP,EVPP)
C=AMIN1(EVP,EVPP)
IF(EV.GT.B.OR.EV.LT.C)EV=(EVP+EVPP)*.5
GO TO 12
7 IF(ABS(EQ).LE.RLOG) GO TO 9
IF(ICVT.NE.0)GOTO15
IF(E2.GT.EPS) GOTO5
ICVT=1
GOTO14
8 ICNT=1

```

```

XLAP=0.0
XLAPP=0.0
GOTO5
9 IF(E3.GT.EPS)GOTO16
  XLAP=XLA
  EVP=EV
  IF(E1.GT.RLOG) GO TO 11
  EV=EV+EVM
  GOTO12
11 EV=EV-EVM
12 CONTINUE
13 IF(IEVT.EQ.2) GOTO14
  KS852=3
  RETURN
14 KS852=2
  RETURN
15 KS852=1
  RETURN
16 GOTO 14
  END

```

C-----
C-----
C
C

FINAL PRINT

```

SUBROUTINE FINPR(XN,XJ,T3,T5,C,IGM,IM,NM,ID3,IZP,IHM,MT,XNB,IGP,
1ME,RA,RAV,V,F,IP,KM3,KM4,MZ,IZM,XMD,MS,MF,MG)
  DIMENSION XN(IM,IGM),XJ(IGM,IM,NM),T3(ID3,IZP),
1T5(ID3,IM),C(IHM,IGM,MT),XNB(IGP),ME(IM),RA(IP),RAV(IP),V(IP),
2F(IM),KM3(ID3),KM4(ID3),MZ(IZM),XMD(MS),MF(MS),MG(MS)
  COMMON LAENGE,A(1)
  COMMON/FIX/LEN,IA(1)
  EQUIVALENCE(IA(1),ID),(IA(33),ICC),(IA(32),LC),(JN,IA(58)),
1(A(3),EPS),(EV,A(1)),(XLA,A(16)),(IG,IA(43)),(IA(3),ISCT),
2(ID4,IA(26)),(EV,A(1)),(EQ,A(32))
  EQUIVALENCE(IA(80),KRD),(IA(96),KWT),(A(28),RLOG)
  WRITE(KWT,2)(I,ME(I),RA(I),RAV(I),F(I),V(I),I =1,IP )
2 FORMAT(1H1,5X,4HZONE,5X,11HFINAL RADII,3X,
1 13HAVERAGE RADII,8X,8HFISSIONS,10X,6HVOLUME/
2(2I5,4E16.7))
  WRITE(KWT,3)
3 FORMAT(30H1FLUX BY GROUP AND SPACE POINT/)
  MA=1
  MB=7
99 MC=MINO(MB,IGM)
  WRITE(KWT,4)(I,I=MA,MC)
  WRITE(KWT,5)
  DO 100 I=1,IM
100 WRITE(KWT,6)I,ME(I),RAV(I),(XN(I,J),J=MA,MC)
  MA=MA+7
  MB=MB+7
  IF(MA.LE.IGM) GOTO99
4 FORMAT(1H0,4X,17H ZONE  AVG RADIUS,7(3X,5HGROUP,I3,3X))
5 FORMAT(1H )
6 FORMAT(2I5,1PE12.5,1P7E14.5)
  IF(ISCT.LT.1)GOTO105
  DO 103 N=1,NM
  WRITE(KWT,7)N
  MA=1
  MB=7

```

```

101 MC=MINO(MB,IGM)
    WRITE(KWT,4)(I,I=MA,MC)
    WRITE(KWT,5)
    DO 102 I=1,IM
102 WRITE(KWT,6)I,ME(I),RAV(I),(XJ(J,I,N),J=MA,MC)
    MA=MA+7
    MB=MB+7
    IF(MA.LE.IGM) GOTO101
103 CONTINUE
    7 FORMAT(15HICURRENT NUMBER,I3,25H BY GROUP AND SPACE POINT/)
C DUMP FLUXES
105 CONTINUE
500 RETURN
    END

```

```

C
C
C
-----
SUBROUTINE WEIGHT(MM,IGE,KWT,D1)
DIMENSION B(135),C(93),NK(38),BH(228),D1(96)
DATA B/ 0.0,0.5,0.5,-1.0,-0.57735,0.57735, 0.0, 0.1666667,
12*0.333333, 0.1666667, -1.0, -0.881917,-0.333333,0.333333,
20.881917, 0.0, 0.0, 0.0, 0.0804306, 0.1724721 ,2*0.2470973,
30.1724711, 0.0804306, -1.0 , -0.930949, -0.68313, -0.258199,
40.2581989, 0.68313 , 0.9309493,4*0.0, 0.0, 0.0533004, 0.1011727,
50.1377534, 0.2077735, 0.2077735, 0.1377534, 0.1011727, 0.0533004,
6-1.0, -0.9511900, -0.786796, -0.57735, -0.218218, 0.218218,
70.5773503, 0.7867958, 0.9511897, 0.0, 0.0318843, 0.0568999,
80.0627447, 0.0756516, 0.1067751, 2*0.1660444, 0.1067751,0.0756516,
90.0627447, 0.0568999, 0.0318843, -1.0, -0.969223, -0.870388,
A-0.758787, -0.627646, -0.460566, -0.174078, 0.174078,
B0.4605662, 0.6276456, 0.7587869,0.8703883, 0.9692234, 4*0.0,
C 0.0, 0.0227558, 0.0397746, 0.0416887, 0.0459606, 0.0526346,
D 0.0640174, 0.0905041, 0.1426643, 0.1426643, 0.0905041,
E0.0640174, 0.0526346, 0.0459606, 0.0416887, 0.0397746,
F0.0227558, -1.0, -0.977525, -0.906765, -0.829993, -0.745356,
G -0.649786, -0.537484, -0.394405, -0.149071, 0.149071,
H 0.394405, 0.5374839, 0.6497843, 0.745356, 0.8299933,0.9067647,
J 0.9775252, 2*0.0, 0.0, 0.5, 0.5,-1.0, -0.57735, 0.57735,
K 0.0, 2*0.1666667, 0.0,2*0.1666667, 2*0.1666667,-0.471405/
DATA C/
L-0.3333333, 0.3333333, -0.942809, -0.881917, -0.3333333,
M 0.3333333, 0.8819171, 2*0.0, 0.0, 2* 0.0804306, 0.0,
N 4*0.0862361, 0.0, 0.0804306, 0.086236, 0.0804306, 0.0804306,
O 0.086236, 0.0804306, -0.365148, -0.258199, 0.2581989,
P -0.730297,-0.68313, -0.258199, 0.258199 ,0.6831301,-0.966092,
Q -0.930949, -0.68313,-0.258199, 0.258199, 0.6831301,0.9309493,
R 0.0, 2*0.0533004, 0.0, 4*0.0505864, 0.0, 0.0505864, 0.0365807,
S 0.0505864, 0.0505864, 0.0365807, 0.0505864, 0.0, 0.0533004,
T0.0505864,0.0505864, 2* 0.0533004, 2* 0.0505864, 0.0533004,
U-0.308607,-0.218218, 0.218218,-0.617213, -0.57735,
V-0.218218, 0.218218,0.5773503, -0.816497,-0.786796,-0.57735,
W-0.218218 ,0.218218,0.5773503, 0.7867958,-0.9759,-0.95119,
X-0.786796,-0.57735, -0.218218,0.218218, 0.5773503, 0.7867958,
Y 0.9511897 ,6*0.0/
DATA NK/ 3, 5, 5, 7, 7, 7, 9, 9, 9, 5*13, 6*17, 6, 3*16, 5*30,
1 8*48,999/
DO 135 I=1,135
135 BH(I)=B(I)
DO 235 I=136,228

```

```
K=I-135
235 BH(I)=C(K)
    M1=0
    L1=2*MM
    IF(IGE-2)667,668,667
667 K4=MM
    GOTO 669
668 K4=L1
669 K1=1
    K2=6
    J0=0
    JP=0
661 DO 50 I=K1,K2
    JP=JP+1
    50 D1(I)=BH(JP)
    J0=J0+1
    K3=NK(J0)
    IF(K3-999)665,666,665
665 IF(K3-K4)661,662,661
662 M1=M1+1
    K1=K1+6
    K2=K2+6
    JKIL=6*M1
    IF(JKIL-L1)661,666,666
666 IF(M1)670,671,670
671 WRITE(KWT,673)
673 FORMAT(1X/////1X,43HINCORRECT NUMBER OF WEIGHTS AND DIRECTIONS//)
    STOP
670 RETURN
    END
```

```

C-----
C-----
CC
SUBROUTINE WRIT
WRITE(6,1)
1 FORMAT(//20X,'**',65(' '),'**')
RETURN
END
C-----
C-----
C
SUBROUTINE DUMMY
C
C THIS ROUTINE DEFINES THE SIZE OF THE MASTER-ARRAY
C
COMMON /FIX/LEN,IA(1000)
LEN=1000
COMMON LAENGE,A(39000)
LAENGE=39000
RETURN
END
C-----
C-----
C
SUBROUTINE REMUS
RETURN
END
C-----
C-----
C
SUBROUTINE LIES2(C,LK1,LK2,LK3,U,IU,IRIS)
C
C
C
C LIES2 READS THE GROUP CONSTANTS-FILE USING WQORG,WQVEKT AND WQSKAL
C AND PREPARES THE CROSS SECTIONS FOR DTK.
C ORGANIZATION, THE GROUP CONSTANT-FILE IS ASSUMED TO BE PRESENT AT
C TAPE "IINP".WQORG IS FILLING THE ARRAY U,RESP.IU.
C
C
C
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
DIMENSION U(IRIS),IU(IRIS)
DIMENSION C(LK1,LK2,LK3)
C
REAL*8 NTYP(2)/' ','/' ,DTKTYP(15)/'CHI ',' '
11/V ' ,SMTOT ' ,SFISS ' ,NUSF ' ,STR ' ,SREM ' ,
2 SCAPT ' ,STRTR ' ,SME1 ' ,SME2 ' ,SME3 ' ,SME4 ' ,
3 SME5 ' ,SME6 ' /
EQUIVALENCE(JFIN,IA(195))
NEW APRAY
C
C
C
GENERAL EQUIVALENCE STATEMENTS
C
EQUIVALENCE(IGM,IA(12)),(MTP,IA(18)),(IHM,IA(15)),(IHS,IA(14)),
1(IHT,IA(13)),(JA,IA(52)),(JB,IA(53)),(MCR,IA(17)),(MT,IA(19)),
2(KWT,IA(96)),(KRD,IA(80))

```

```

C
C   NEW EQUIVALENCE
C
C   EQUIVALENCE(IEVT,IA(11))
C   EQUIVALENCE(IINP,IA(194))
C
C
C   DO 8 I=1,IRIS
C     8 U(I)=0.
C   9999 CONTINUE
C     EQUIVALENCE(IA(3),ISCT)
C     COMMON/ZLIES2/IAL(40)
C     DIMENSION IPLATZ(40)
C     IF(MT.GT.40)GO TO 1007
C
C
C     IPLATZ(1)=1
C     DO 10 I=2,MTP
C       IMEHR=ISCT+1
C       IF(IAL(I-1).GT.0)IMEHR=1
C       IPLATZ(I)=IPLATZ(I-1)+IMEHR
C   10 CONTINUE
C     MTCONT=IPLATZ(MTP)
C     IF(IAL(MTP).LE.0)MTCONT=MTCONT+ISCT
C     IF(MTCONT.NE.MT) GO TO 1001
C
C
C     IGEL=IRIS
C     CALL WQORG(IGEL,U,IU,IINP,&1000)
C     IF(IU(3).NE.MTP)GO TO 1001
C     IF(IU(4).NE.IGM) GO TO 1002
C     IF((IU(6)+IU(8)).LT.7) GO TO 1003
C     J3=JB
C     J5=JA
C
C
C
C     DO 1 J=1,IGM
C
C     KENN=1
C     ITZW=1
C
C     CHI
C
C     NTYP(1)=DTKTYP(1)
C     CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
C     A(J5)=U(LPCR+1)
C     J5=J5+1
C
C
C     1/V
C
C     IF(IEVT.NE.2) GO TO 40
C     NTYP(1)=DTKTYP(2)
C     CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
C     A(J3)=1./U(LPCR+1)
C     J3=J3+1
C   40 CONTINUE
C
C
C     SMTOT

```

```

C
  N TYP(1)=DTKTYP(3)
  KENN=2
  CALL WQVEKT(KENN,NTYP,ITZW,J,LPCR,&1004,&1005,NG1,NG2)
  DO 7 M=NG1,NG2
  DO 6 N=1,MTP
  NA=IPLATZ(N)
  C(IHS-J+M,M,NA)=U(LPCR+N+MTP*(M-NG1))
6 CONTINUE
7 CONTINUE

C
C
C
  KENN=1
  DO 2 I=4,6
  N TYP(1)=DTKTYP(I)
  CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
  DO 3 N=1,MTP
  NA=IPLATZ(N)
  UW=U(LPCR+N)
  C(IHT+I-6,J,NA)=UW
  IF(I.EQ.4)C(IHT-3,J,NA)=UW
  IF(I.EQ.6)C(IHT-4,J,NA)=UW
3 CONTINUE
2 CONTINUE

C
C
  KENN=1
  N TYP(1)=DTKTYP(9)
  CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1090)
  DO 1091 N=1,MTP
  NA=IPLATZ(N)
1091 C(IHT,J,NA)=U(LPCR+N)
  IF(J.GT.1)GO TO 1090
  WRITE(KWT,1092)
1090 CONTINUE
1092 FORMAT(1H0,' **** ALS TOTALER WIRKUNGSQUERSCHNITT WIRD DER NUSYS-T
  1YP &STRTR& STATT &STR& BENUETZT ****')

C
C
C
  SELFSCATTERING

  N TYP(1)=DTKTYP(7)
  CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
  DO 4 N=1,MTP
  NA=IPLATZ(N)
  C(IHS,J,NA)=C(IHT,J,NA)-U(LPCR+N)
4 CONTINUE

C
C
C
C
C
  THIS PASSAGE MUST BE MODIFIED FOR UPSCATTERING
  C(IHS,J,N)=C(IHT,J,N)-U(LPCR+N)

C
  N TYP(1)=DTKTYP(8)
  CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
  DO 5 N=1,MTP
  NA=IPLATZ(N)
  C(IHT-2,J,NA)=C(IHT-2,J,NA)+U(LPCR+N)
5 CONTINUE
C

```

```

C
C
IF(ISCT.LE.0) GO TO 1
IF(MTCONT.EQ.MTP)GO TO 1
C
IF(ISCT.GT.6)GO TO 1019
DO 28 IK=1,ISCT
C
KENN=2
NTYP(1)=DTKTYP(9+IK)
CALL WQVEKT(KENN,NTYP,ITZW,J,LPCR,&1004,&1005,NG1,NG2)
DO 17 M=NG1,NG2
DO 16 N=1,MTP
IF(IAL(N).GE.0) GO TO 16
NA=IPLATZ(N)+IK
NB=IPLATZ(N)
UIK=U(LPCR+N+MTP*(M-NG1))
C(IHS-J+M,M,NA)=UIK
16 CONTINUE
17 CONTINUE
28 CONTINUE
C
C
DO 29 N=1,MTP
IF(IAL(N).GE.0)GO TO 29
S=0.
C
DO 203 IK=1,ISCT
C
NA=IPLATZ(N)
J1=IHS
J2=J
202 S=S+C(J1,J2,NA+IK)
J1=J1+1
J2=J2+1
IF(J1.LE.IHM.AND.J2.LE.IGM) GO TO 202
203 CONTINUE
C(IHS,J,NA)=C(IHS,J,NA)+S
C(IHT,J,NA)=C(IHT,J,NA)+S
29 CONTINUE
C
C
1 CONTINUE
C
C
*****RETURNS AND ERROR MESSAGES*****
WRITE(KWT,1010)
1010 FORMAT(1H0,33HCROSS SECTIONS CORRECTLY PREPARED )
DO 9 I=1,IRIS
9 U(I)=0.
RETURN
C
C
1000 WRITE(KWT,1011)
1011 FORMAT(1H0,28HINCORRECT END IN SUBR. LIES2 )
IUABS=IABS(IU(2))
WRITE(KWT,1012) IUABS
1012 FORMAT(1H0,28HARRAY NAMED U IS TOO SMALL ,1H(,14,7HSAETZE))
GO TO 1030

```



```

1001 WRITE(KWT,1011)
      WRITE(KWT,1013) IU(3),MTP,MTCONT,MT,ISCT,(IAL(JAQ),JAQ=1,MTP)
1013 FORMAT(1H0,96HTHE NUMBER OF INPUT MATERIALS DOES NOT AGREE WITH TH
      1E NUMBER OF MATERIALS OF THE SIGMA-BLOCK      /90X,8HIU(3) =,I2/
      290X,8H MTP =,I2          /// 90X,8HMTCONT =,I2/
      390X,8H MT =,I2/          90X,8H ISCT =,I2///
      42X,18HMATERIALKENNZAHLEN/(3X,20I4))
      GO TO 1030
1002 WRITE(KWT,1011)
      WRITE(KWT,1014) IU(4),IGM
1014 FORMAT(1H0,32HTHE NUMBERS OF GROUPS DONT AGREE      ,2I6)
      GO TO 1030
C
C
C
C
C
C
1003 WRITE(KWT,1011)
      WRITE(KWT,1015) IU(6),IU(8)
1015 FORMAT(1H0,27HTHERE ARE TOO LITTLE TYPES      ,I6,1H/,I6)
      GO TO 1030
1004 WRITE(KWT,1011)
      WRITE(KWT,1016)
1016 FORMAT(1H0,36H ERROR 1 IN WQ, INDICES DONT AGREE      )
      GO TO 1030
1005 WRITE(KWT,1011)
      WRITE(KWT,1017) NTYP(1)
1017 FORMAT(1H0,31HERROR 2 IN WQ, SPECIFIED TYPE      ,A8,9HNOT FOUND)
1030 STOP
1007 WRITE(KWT,1011)
      WRITE(KWT,1018)
      GO TO 1030
1019 WRITE(KWT,1011)
      WRITE(KWT,1020)
1020 FORMAT(///30X,' ISCT IST ZU GROSS')
      GO TO 1030
1018 FORMAT(1H0,40HMORE THAN 40 MATERIALS ARE NOT ALLOWED      )
      FND
-----
C
C
C
C
      QUERSCHNITTS - ORGANISATION
      SUBROUTINE WQORG(LDIM,S,LS,NFL,*)
C
C      COMMON KN(90)
C
C      DIMENSION S(1),LS(1),LX(250),NTYP(1)
C
C      EQUIVALENCE (KN(3),ICR),(KN(32),NG)
C
C      REWIND NFL
C
C      ADJUSTED STATEMENT
C
C      READ(NFL)
C

```

```

READ(NFL,ERR=3633,END=3633)K,Z
IF(K.GE.LDIM) RETURN 1
REWIND NFL
READ(NFL)
READ(NFL)K,(S(I+1),I=1,K)
LS(1)=K
BACKSPACE NFL
NMI=LS(3)
NGP=LS(4)
LPGA1=LS(5)
NSKV=LS(8)+1
IF(LS(2))10,88,9
9  NDAT=LS(2)
  IF(NDAT.GT.LDIM) RETURN 1
  READ (NFL) (LS(I),I=1,NDAT)
  LGR=0
  LDIM=NDAT
99 RETURN
88 RETURN 1
C
10 NDAT=-LS(2)
  IF(NDAT.LE.LDIM) GO TO 19
  MX=0
  DO 13 I=1,NGP
  IF(I.EQ.NGP) GO TO 11
  ND=LS(LPGA1+1+I)-LS(LPGA1+I)
  GO TO 12
11 ND=LS(2)-LS(LPGA1+I)
12 LX(I)=ND
13 MX=MAXO(MX,ND)
  LDA=LS(1)+1+MX
  NAD1=LS(1)+1
  DO 14 I=1,NGP
  K=NGP+1-I
  LDA=LDA+LX(K)
  IF(LDA.GT.LDIM) GO TO 15
14 CONTINUE
  GO TO 19
C
15 IF(I.EQ.1) RETURN 1
  LGR=K+1
  N1=K
  DO 16 I=1,N1
16 READ(NFL)N,Z
17 NWER=LS(1)+1+MX
  DO 18 I=LGR,NGP
  READ(NFL)N,(S(NWER+1+J),J=1,N)
  LS(NWER+1)=N
  LS(LPGA1+1+N1)=NWER
  DO 118 L=1,NSKV
118 LS(NWER+2+L)=LS(NWER+2+L)+NWER
18 NWER=NWER+N+1
  REWIND NFL
  READ(NFL)
  IG=0
  LGR=LGR-1
  IF(LGR.EQ.0) LDIM=NDAT
  GO TO 99
19 LGR=1

```

```

MX=0
N1=0
GO TO 17

```

C
C
C

```

ENTRY DASEEK(S,LS,KENN,NTYP,LPCR,IA,NG1,NG2)
LDISK=0
GO TO 20
ENTRY WQSKAL(KENN,NTYP,IA,NR,LPCR,*,*)
ENTRY WQVEKT(KENN,NTYP,IA,NR,LPCR,*,*,NG1,NG2)
NG=NR
LDISK=1
20 IF(IA.NE.0) GOTO 2226
MKK=2
GOTO 227
2226 MKK=4
227 IF(LGR.EQ.0) GO TO 25
IF(NG.GT.LGR) GO TO 25
IF(NG.EQ.IG) GO TO 25
IF(IG.NE.0) GO TO 22
READ(NFL)N,Z
22 READ(NFL)N,(S(NAD1+1+I)),I=1,N)
IF(NG.LT.IG) RETURN 1
LS(NAD1+1)=N
LS(LPGA1+NG)=NAD1
DO 223 L=1,NSKV
223 LS(NAD1+2+L)=LS(NAD1+2+L)+NAD1
IG=NG
IF(NG.EQ.LGR) GO TO 23
GO TO 25
23 IG=0
REWIND NFL
READ(NFL)

```

C
C
C

SUCHEN DER TYPEN UND DATEN

```

25 JTY=0
LPGBL=LS(LPGA1+NG)
JG=LS(LPGBL+2)
IF(LDISK.NE.0) GO TO 225
IF(JG.EQ.NG) GO TO 226
ICR=1
GO TO 98
225 IF(JG.NE.NG) RETURN 1
226 GO TO (26,36,36),KENN

```

C
C
C

SKALARTYPEN

```

26 ISKA=LS(6)
LPSK=LS(7)
ISKA=ISKA*MKK
DO 28 I=1,ISKA,MKK
JTY=JTY+1
IF(NTYP(1).NE.LS(LPSK+I)) GO TO 28
IF(NTYP(2).NE.LS(LPSK+I+1)) GO TO 28
IF(IA)27,29,27
27 IF(NTYP(3).NE.LS(LPSK+I+2)) GO TO 28
IF(NTYP(4).NE.LS(LPSK+I+3)) GO TO 28

```

```

GO TO 29
28 CONTINUE
IF(LDISK.NF.0) RETURN 2
ICR=-1
GO TO 98
29 LPCR=LS(LPGBL+3)+(JTY-1)*NMI
98 RETURN

```

C
C
C

VEKTORTYPEN

```

36 IVEK=LS(8)
LPV=LS(9)
IVEK=IVEK*MKK
DO 38 I=1, IVEK, MKK
JTY=JTY+1
IF(NTYP(1).NE.LS(LPV+I)) GO TO 38
IF(NTYP(2).NE.LS(LPV+I+1)) GO TO 38
IF(IA)37,39,37
37 IF(NTYP(3).NE.LS(LPV+I+2)) GO TO 38
IF(NTYP(4).NE.LS(LPV+I+3)) GO TO 38
GO TO 39
38 CONTINUE
IF(LDISK.NE.0) RETURN 2
ICR=-1
GO TO 98
39 LPCR=LS(LPGBL+3+JTY)+3
NG1=LS(LPCR-1)
NG2=LS(LPCR)
IF(KENN.NE.3) GO TO 98
LPCR=LPCR-3
GO TO 98
3633 CALL FSPIE
RETURN 1
END

```

C-----
C-----
C

```

SUBROUTINE AUSWER(XN,IM,IGM,IAUSW)
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE(KWT,IA(96)),(IA(2),ITH)
DIMENSION XN(IM,IGM)
REAL*8 C(2) / 'FLUX1 ', 'ADFL1 ', NTYP
1 FORMAT(///71HODER FLUSS WIRD ZWECKS SPAETERER AUSWERTUNG GESCHRIEB
IEN AUF EINHEIT NR. ,I3)
REWIND IAUSW
WRITE(KWT,1) IAUSW
NULL=0
NTYP=C(ITH+1)
NDAT=IM+1
NDA=(IM+2)*IGM
IEINS=-1
WRITE(IAUSW) NULL, NTYP, NDA
IF(ITH.GT.0) GO TO 3
WRITE(IAUSW)((NDAT, IEINS, (XN(J, IG), J=1, IM)), IG=1, IGM)
GO TO 4
3 WRITE(IAUSW)((NDAT, IEINS, (XN(J, (IGM-IG+1)), J=1, IM)), IG=1, IGM)
4 RETURN
END

```

```

C -----
C -----
C
SUBROUTINE READER(ITAPE,ILEN,BUF)
DIMENSION BUF(ILEN)
1 CONTINUE
READ(ITAPE) (BUF(IJ),IJ=1,ILEN)
RETURN
END

C -----
C -----
C
SUBROUTINE PROVOS(RNEU, XN,IGM,IM,IZM,IFISS,IFLUX,FLA,RXNEU,
1RXOLD,ROLD)
DIMENSION RXNEU(IM),RXOLD(IM),RNEU(IZM),ROLD(IZM)
DIMENSION FLA(IM,IGM)
DIMENSION XN(IM,IGM)
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE(IA(195),JFIN)
EQUIVALENCE(RLOG,A(28)),(KWT,IA(96))
1 CONTINUE
IF(IFLUX.GT.0) GO TO 4501
WRITE(KWT,4502)
4502 FORMAT(3X,90H THERE SEEMS NO UNIT NUMBER TO BE SPECIFIED FOR FLUX
1 INPUT
IFISS=0
GO TO 14

C
C
4501 REWIND IFLUX
READ(IFLUX,ERR=100,END=100)IDOLD,IGMOLD,IMOLD,IZMOLD
IF(IGMOLD.NE.IGM) GO TO 4503
IF(IZMOLD.NE.IZM) GO TO 4503
IF((LAENGE-JFIN).LE.(IM*IGM+2*(IM+IZM)))GO TO 104
READ(IFLUX)(ROLD(I),I=1,IZM)
DO 4301 I=1,IZM
GABS=ABS(ROLD(I)-RNEU(I))
IF(GABS.GT.0.1 ) GO TO 4503
4301 CONTINUE
READ(IFLUX)(RXOLD(IF),IF=1,IMOLD)
READ(IFLUX)((FLA(I,J),I=1,IMOLD),J=1,IGM)
READ(IFLUX)A(1)
4718 WRITE(KWT,4719) IDOLD,IFLUX
4719 FORMAT(4X,92H*****
1*****
1 /4X,1H*,90X,1H*/4X,1H*,4X,62
8HFLUX FROM PREVIOUS CALCULATION,PROBLEM WITH IDENTIFICATION NR.,I6
4,12H ON UNIT NR.,I3,3X,1H*/4X,92H*****
7***** )

C
C
RMAX=RXNEU(IM)
IANF=1
DO 10 I=1,IM
X=RXNEU(I)
DO 11 J=IANF,IMOLD
IF(X.GT.RXOLD(J)) GO TO 11
IX=J

```

```

      GO TO 12
11  CONTINUE
      IX=IMOLD+1
12  CONTINUE
      IANF=IX-1
      IF(IANF.EQ.0) IANF=1
C
C
      KNURR=(IX-1)*(IMOLD+1-IX)
      IF(KNURR.GT.0) GO TO 20
      IF(IX.EQ.1) GO TO 30
      IXB=IMOLD
      GO TO 40
30  IXB=2
      GO TO 40
20  IXB=IX
40  CONTINUE
      IXA=IXB-1
      DO 50 IG=1, IGM
      XN(I, IG)=FLA(IXA, IG)+(FLA(IXB, IG)-FLA(IXA, IG))*(X-RXOLD(IXA))/(RXOLD(IXB)-RXOLD(IXA))
50  CONTINUE
10  CONTINUE
      IA(10)=1
      GO TO 14
4503 WRITE(KWT,13)
      13  FORMAT(3X,99H EITHER IGM AND IGMOLD      OR      IZM AND IZMOLD      OR
      1THE RADDII DO NOT CORRESPOND
      IFISS=0
      14  RETURN
100  WRITE(KWT,101)
101  FORMAT(1H0,'      **** INPUT-ERROR-CONDITION ENCOUNTERED IN DTK-ROUTI
      1NE PROVOS ****')
      GO TO 200
104  WRITE(KWT,204)
204  FORMAT(1H0,'      **** A-ARRAY-END IS TOO SMALL TO PASS FLUX ****')
200  IFISS=0
      RETURN
      END

```

```

C-----
C-----
C

```

```

SUBROUTINE TSCHEB(NM, IGM, IM, MM, IGP, ICC, BUFF2, ITEL)
COMMON LAENGE, A(1)
COMMON/FIX/LEN, IA(1)
COMMON/TSCH/ITSCH, IANF, RHO, ITM, INORM, ROM
EQUIVALENCE (XLA, A(16)), (XLAR, A(27)), (KWT, IA(96))
EQUIVALENCE (JN, IA(58)), (JX, IA(75)), (JB, IA(76)), (JF, IA(57)),
1 (JFG, IA(77))

```

```

C
DIMENSION LENGTH(5), IBEGIN(5), BUFF2(ITEL)
EQUIVALENCE (IBR, IA(7))
INTEGER TAPE1, TAPE2, TAPOLD, TAPNEW
TAPE1=20
TAPE2=21

```

```

C
C      TWO TAPES ARE INTRODUCED ; WHICH CONTAIN THE PREVIOUS FLUXES A.S.O
C
11  CONTINUE

```

```

12 CONTINUE
   IF(ITSCH.EQ.0)RETURN
   REWIND 20
   REWIND 21
   NLEN=5
C
   LENGTH(1)=IM*IGM
   LENGTH(2)=IM*IGM*NM
   LENGTH(3)=IGP
   LENGTH(4)=IM
   LENGTH(5)=IGM*MM
C
   IBEGIN(1)=JN
   IBEGIN(2)=JX
   IBEGIN(3)=JFG
   IBEGIN(4)=JF
   IBEGIN(5)=JB
C
   IF(ICC-1)1,2,3
1  DO 180 I=1,5
   LENG=LENGTH(I)
   IF(LENG.GT.ITM)GO TO 81
180 CONTINUE
   GO TO 83
81 WRITE(KWT,82)LENG,ITM
82 FORMAT('0 IMPOSSIBLE TO PERFORM TSCHEBYSCHIEFF EXTRAPOLATION',2I6)
   ITSCH=0
83 RETURN
C
2  IBIS=1
   IBUT=-1
   RO=0.1
   ROALT=18.
60 IWR=20
   CALL WRITER(IWR,IBEGIN,LENGTH,NLEN)
   RETURN
C
3  IF(IBIS.NE.1) GO TO 7
   IBIS=2
61 IWR=21
   CALL WRITER(IWR,IBEGIN,LENGTH,NLEN)
   TAPOLD=20
   TAPNEW=21
   IWR=20
   RETURN
C
7  IBIS=IBIS+1
   IF(IBUT.GT.0) GO TO 51
   IF(RHO.NE.0..AND.ICC.GT.4)GO TO 127
   IF(RHO.NE.0..AND.ICC.LE.4)GO TO 63
   ROALT=RO
   RO=ROM
   IF(ABS(RO).GT.1.E-3)EPO=ABS((RO-ROALT)/RO)
   IF(EPO.GT.0.08.OR.ICC.LT.4) GO TO 63
   IF(RO.LE.0.) GO TO 63
C  RHO=(3.*RO+2.*ROALT+ROUR)/6.
   RHO=RO*1.05
   IF(RO.GT.0.92) RHO=RO

```

```

      IF(RHO*(1.-RHO).LE.1.E-4)GO TO 100
127 WRITE(KWT,126) ICC,RHO
      W=2.
126 FORMAT(1H0,21HTSCHEB. ITER. AB DER ,I2,23H. ITERATION MIT RHO =
      1 ,E15.8//)
      IBUT=1
51 W2=1.-W
72 CONTINUE
      IMEL=5
      IF( IBR.EQ.0)IMEL=4
      DO 78 IS=1,IMEL
      ILEN=LENGTH(IS)
      ICOM=IBEGIN(IS)-1
      CALL READER(TAPOLD,ILEN,BUFF2)
      DO 77 IND1=1,ILEN
      URB =W*A(ICOM+IND1)+W2*BUFF2(IND1)
      IF(URB.LE.0.)URB=BUFF2(IND1)*0.5
      A(ICOM+IND1)=URB
77 CONTINUE
78 CONTINUE
      REWIND TAPNEW
      REWIND TAPOLD
80 CONTINUE
      W=1./(1.-(RHO*RHO*W/4.))
63 CONTINUE
62 CALL WRITER(IWR,IBEGIN,LENGTH,NLEN)
      IF(IWR.EQ.20) GO TO 70
      IWR=20
      TAPNEW=21
      TAPOLD=20
      GO TO 71
70 IWR=21
      TAPNEW=20
      TAPOLD=21
71 RETURN
100 ITSCH=0
      WRITE(KWT,101)RHO
101 FORMAT('0',20X,'CODE UNABLE TO FIND REASONABLE RHO,FOUND RHO = ',
      1E13.5)
      RETURN
      END

```

C
C
C
C

```

SUBROUTINE WRITER(ITAPE,IBEGIN,LENGTH,NLEN)
DIMENSION IBEGIN(NLEN),LENGTH(NLEN)
COMMON LAENGE,A(1)
COMMON/FIX/LEN,IA(1)
EQUIVALENCE( IBR,IA(7))
1 CONTINUE
      IMEL=5
      IF( IBR.EQ.0)IMEL=4
      DO 60 IJ=1,IMEL
      IAB=IBEGIN(IJ)
      IZZ=IBEGIN(IJ)+LENGTH(IJ)-1
      WRITE(ITAPE)(A(J),J=IAB,IZZ)
60 CONTINUE
      REWIND ITAPE

```


RETURN
END