

**KERNFORSCHUNGSZENTRUM
KARLSRUHE**

Juli 1973

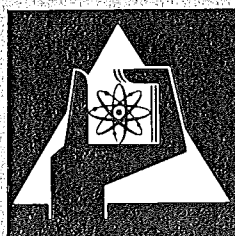
KFK 1826

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

SNOW

Ein zweidimensionales S_N -Programm zur Lösung der
Neutronentransportgleichung in Platten- und Zylindergeometrie

C. Günther, W. Kinnebrock



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

KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1826

Institut für Neutronenphysik und Reaktortechnik

Projekt Schneller Brüter

SNOW

Ein zweidimensionales S_N -Programm zur Lösung

der Neutronentransportgleichung in Platten-

und Zylindergeometrie

von

C. Günther

W. Kinnebrock

Gesellschaft für Kernforschung mbH., Karlsruhe

Zusammenfassung

SNOW ist ein FORTRAN IV-Programm zur Lösung der zweidimensionalen Neutronentransportgleichung mit der Methode der Diskreten Ordinaten im Multigruppenmodell (SN-Methode). Der Code rechnet die Geometrien (x,y)-Platte und (r,z) bzw. (r,θ)-Zylinder, die Streuquelle kann sowohl mit isotropen als auch mit anisotropen Querschnitten berechnet werden. Eine Vielzahl weiterer Optionen liegt vor, insbesondere wurden mehrere Verfahren zur Konvergenzbeschleunigung eingebaut. In der vorliegenden Version des Codes kann keine Aufwärtsstreuung behandelt werden.

SNOW, a Two-Dimensional SN-Program for Solving the Neutron-Transport-Equation in Rectangular and Cylinder Geometry

Abstract

SNOW is a FORTRAN IV program for solving the two dimensional multi-group transport equation by the method of discrete ordinates (SN-method). The geometries of (x,y)-plane and (r,z), (r,θ)-cylinder, isotropic and anisotropic scattering and a lot of other special options are permitted. Three methods of convergence acceleration are incorporated.

Inhalt

Einleitung

I. Spezifische Eigenschaften von SNOW

1. Differenzengleichungen
2. Geometrien
3. Richtungen und Gewichte
4. Iterationen und Konvergenzbeschleunigungen
5. Fremdquelle
6. Randbedingungen
7. Buckling
8. Anisotropie

II. Eingabe (Input)

1. Eingabebeschreibung
2. Eingabe der Querschnitte

III. Ausgabe (Output)

1. Die Ausgabeliste
2. Auswertungen
3. Bilanzen
4. Graphische Auswertung
5. Speicherung von Flüssen

IV. Implementierung

1. Subroutinen-Beschreibung
2. Erhöhung des Speicherbedarfs

Literaturangabe

Anhang I: Sample Problem

Anhang II: Programmliste von SNOW

Anhang III: Anlageabhängige Bestandteile des Codes

Einleitung

SNOW ist ein in Karlsruhe entwickeltes zweidimensionales SN-Programm zur Lösung der Neutronen-Transportgleichung. Die Diskretisierung der Boltzmann-Gleichung erfolgte mit Hilfe des "Diamond Difference Scheme" von B. Carlson und K. Lathrop /3/, die numerische Integration der winkelabhängigen Flüsse wurde mit Hilfe des Verfahrens von B. Carlson und C.E. Lee /1/ durchgeführt. Wegen der hohen Dimension der Matrizen war die Einführung von effektiven Konvergenzbeschleunigungen für die äußeren und inneren Iterationen unumgänglich. Neben "Rebalancing" und Tschebyscheff-Beschleunigungen wurde eine weitere Option eingebaut, die ohne großen zusätzlichen Aufwand die Berechnung des Problems mit verschiedenen Schrittweiten und verschiedenen SN-Ordnungen erlaubt und bei günstiger Kombination der Schrittweiten etwa 50 % der Rechenzeit einspart (Grobgittertechnik, /4/). Je nach den gewünschten Ergebnissen kann der Benutzer eine "flußorientierte" oder eine auf den Eigenwert k_{eff} ausgerichtete Genauigkeitsabfrage vergeben. Weitere Optionen sind: Berechnung inhomogener Probleme (Fremdquelle), Verwendung von Bucklings, Anisotropie, adjungierte Rechnung, verschiedene Randbedingungen. Mögliche Geometrien sind: Platte (x,y); Zylinder /(r,z) und (r,θ)/.

Der Bericht stellt eine Anleitung zur Benutzung des Codes dar. Er enthält neben der Eingabebeschreibung auch eine Programmliste (Anhang 2) und ein Eingabebeispiel (Anhang 1). Obwohl der vorliegende Bericht keine Programmbeschreibung sein soll, wurde versucht, die Funktion einzelner Unterprogramme in grober Form zu beschreiben (IV.1).

I. Spezifische Eigenschaften von SNOW

1. Differenzgleichungen

Die linke Seite der zu lösenden kontinuierlichen Boltzmann-Gleichung

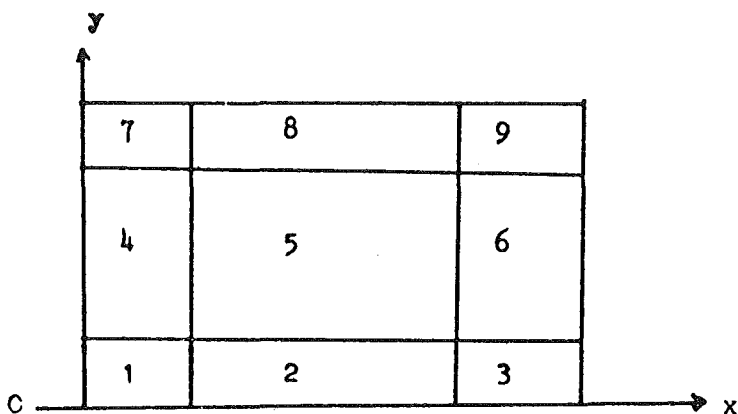
$$\Omega \nabla \phi + \sigma_t \phi = S \quad (\phi = \phi(w, \Omega))$$

läßt sich mit Hilfe der in /3/ beschriebenen Differenzgleichungen ("Diamond Difference Scheme") diskretisieren. Diese Differenzgleichungen besitzen neben der Konsistenz mit der Boltzmann-Gleichung noch die folgenden Eigenschaften:

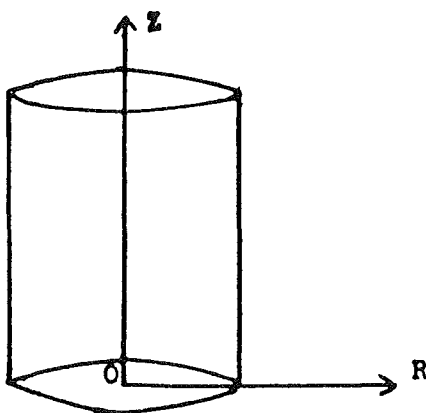
- a) Die Eindeutigkeit der Lösungen ist bei Vorgabe von Randbedingungen gewährleistet /5/
- b) Bei Plattengeometrie (x,y) konnte die Konvergenz der Lösungen gegen die des kontinuierlichen Problems bewiesen werden /5/.
- c) Es können negative winkelabhängige Flüsse auftreten. In diesem Fall werden vom Code Korrekturen vorgenommen ("Negative-flux-fix-up").

2. Geometrien

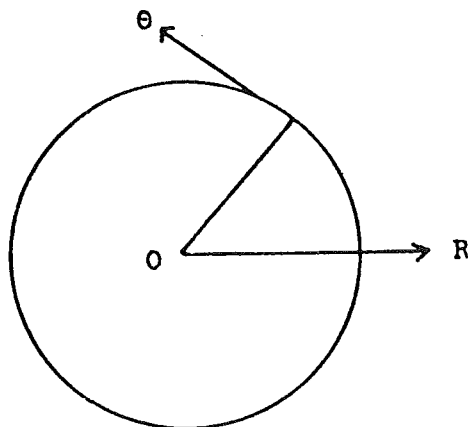
Die Koordinatenrichtungen sind in der folgenden Abbildung für alle Geometrien veranschaulicht. Die Zonengrenzen verlaufen achsenparallel (vgl. Abb.). In dem ersten Bild (xy-Geometrie) sind als Beispiel neuen Zonen eingezeichnet. Die Eingabe der Mischungszahlen (vgl. K11) verläuft in der Reihenfolge der eingezeichneten Zahlen



(x,y)-Geometrie



(R,Z)-Geometrie



(R,θ)-Geometrie

3. Richtungen und Gewichte

SNOW benutzt für die Diskretisierung der winkelvariablen Richtungen, die mit Hilfe der in /1/ beschriebenen Methoden berechnet wurden. Bei einer SN-Rechnung mit der Quadraturordnung ISN werden für die Vollkugel insgesamt

$$ISN_{\#}(ISN+4)$$

verschiedene Richtungen angenommen. Jeder Richtung ist ein Gewicht zugeordnet, welches in die numerische Integration der winkelabhängigen Flüsse bei der Berechnung des Skalarflusses eingeht. Es wird darauf hingewiesen, daß für niedrige ISN unter Umständen Unebenheiten der Flüsse auftreten können. Letzteres ist auf den bei zweidimensionalen SN-Programmen zuweilen auftretenden "Ray-Effekt" zurückzuführen.

Falls $\Omega_m = (\mu_m, \eta_m, \pm \sqrt{1 - \mu_m^2 - \eta_m^2})$ ein diskreter Richtungsvektor bedeutet, so lautet die für die numerische Integration der winkelabhängigen Flüsse benutzte Integrationsformel:

$$(1) \quad \oint \phi(w, \Omega) d\Omega \approx \sum_m w_m \phi(w, \Omega_m)$$

Da ϕ nur von zwei Ortsvariablen abhängig ist, kann der Fluß als symmetrisch bezüglich einer Ebene angenommen werden. Die Integrationsformel (1) benutzt daher nur Richtungen Ω_m einer Halbkugel. Man kann sich diese Halbkugel für (x,y) und (r, θ)-Geometrie liegend, für (r,z)-Geometrie stehend vorstellen.

Im folgenden sind die in SNOW einprogrammierten Richtungen (μ, η) und Gewichte (w_m) gelistet.

a.) S_2 -Gewichte und Richtungen

Gewichte	η	μ
0.00000000	-0.57735027	-0.81649658
0.25000000	-0.57735027	-0.57735027
0.25000000	-0.57735027	-0.57735027
0.00000000	0.57735027	0.81649658
0.25000000	0.57735027	-0.57735027
0.25000000	0.57735027	0.57735027

b.) S_4 -Gewichte und Richtungen

Gewichte	η	μ
0.00000000	-0.33333333	-0.94280903
0.08333333	-0.33333333	-0.88191710
0.08333333	-0.33333333	-0.33333333
0.08333333	-0.33333333	0.33333333
0.08333333	-0.33333333	0.88191710
0.00000000	-0.88191710	-0.47140451
0.08333333	-0.88191710	-0.33333333
0.08333333	-0.88191710	0.33333333
0.00000000	0.33333333	-0.94280903
0.08333333	0.33333333	-0.88191710
0.08333333	0.33333333	-0.33333333
0.08333333	0.33333333	0.33333333
0.08333333	0.33333333	0.88191710
0.00000000	0.88191710	-0.47140451
0.08333333	0.88191710	-0.33333333
0.08333333	0.88191710	0.33333333

c.) S_6 -Gewichte und Richtungen

Gewichte	η	μ
0.00000000	-0.25819889	-0.96609180
0.04021531	-0.25819889	-0.93094934
0.04311802	-0.25819889	-0.68313005
0.04021531	-0.25819889	-0.25819889
0.04021531	-0.25819889	0.25819889
0.04311802	-0.25819889	0.68313005
0.04021531	-0.25819889	0.93094934
0.00000000	-0.68313005	-0.73029674
0.04311802	-0.68313005	-0.68313005
0.04311802	-0.68313005	-0.25819889
0.04311802	-0.68313005	0.25819889
0.04311802	-0.68313005	0.68313005
0.00000000	-0.93094934	-0.36514839
0.04021532	-0.93094934	-0.25819889
0.04021532	-0.93094934	0.25819889
0.00000000	0.25819889	-0.96609180
0.04021531	0.25819889	-0.93094934
0.04311802	0.25819889	-0.68313005
0.04021531	0.25819889	-0.25819889
0.04021531	0.25819889	0.25819889
0.04311802	0.25819889	0.68313005
0.04021531	0.25819889	0.93094934
0.00000000	0.68313005	-0.73029674
0.04311802	0.68313005	-0.68313005
0.04311802	0.68313005	-0.25819889
0.04311802	0.68313005	0.25819889
0.04311802	0.68313005	0.68313005
0.00000000	0.93094934	-0.36514839
0.04021532	0.93094934	-0.25819889
0.04021532	0.93094934	0.25819889

d.) S_8 -Gewichte und Richtungen

Gewichte	η	μ
o.o0000000	-o.21821789	-o.97590007
o.o2665019	-o.21821789	-o.95118973
o.o2529318	-o.21821789	-o.78679579
o.o2529318	-o.21821789	-o.57735027
o.o2665019	-o.21821789	-o.21821789
o.o2665019	-o.21821789	o.21821789
o.o2529318	-o.21821789	o.57735027
o.o2529318	-o.21821789	o.78679579
o.o2665019	-o.21821789	o.95118973
o.o0000000	-o.57735027	-o.81649658
o.o2529318	-o.57735027	-o.78679579
o.o1829035	-o.57735027	-o.57735027
o.o2529318	-o.57735027	-o.21821789
o.o2529318	-o.57735027	o.21821789
o.o1829035	-o.57735027	o.57735027
o.o2529318	-o.57735027	o.78679579
o.o0000000	-o.78679579	-o.61721340
o.o2529318	-o.78679579	-o.57735027
o.o2529318	-o.78679579	-o.21821789
o.o2520318	-o.78679579	o.21821789
o.o2529318	-o.78679579	o.57735027
o.o0000000	-o.95118973	-o.30860670
o.o2665019	-o.95118973	-o.21821789
o.o2665019	-o.95118973	o.21821789
o.o0000000	o.21821789	-o.97590007
o.o2665019	o.21821789	-o.95118973
o.o2529318	o.21821789	-o.78679579
o.o2529318	o.21821789	-o.57735027
o.o2665019	o.21821789	-o.21821789
o.o2665019	o.21821789	o.21821789
o.o2529318	o.21821789	o.57735027
o.o2529318	o.21821789	o.78679579
o.o2665019	o.21821789	o.95118973
o.o0000000	o.57735027	-o.81649658
o.o2529318	o.57735027	-o.78679579
o.o1829035	o.57735027	-o.57735027
o.o2529318	o.57735027	-o.21821789
o.o2529318	o.57735027	o.21821789
o.o1829035	o.57735027	o.57735027
o.o2529318	o.57735027	o.78679579
o.o0000000	o.78679579	-o.61721340
o.o2529318	o.78679579	-o.57735027
o.o2529318	o.78679579	-o.21821789
o.o2529318	o.78679579	o.21821789
o.o2529318	o.78679579	o.57735027
o.o0000000	o.95118973	-o.30860670
o.o2665019	o.95118973	-o.21821789
o.o2665019	o.95118973	o.21821789

4. Iterationen und Konvergenzbeschleunigungen

4.1 Äußere Iterationen

Die Gesamtheit der Differenzgleichungen bildet ein Gleichungssystem der Gestalt

$$(2) \quad Tu = \frac{1}{k} Fu + Su \quad (k = k_{\text{eff}})$$

wobei T, F, S Matrizen und u der Fluß in Vektorform bedeuten (T = Leckage, F = Spaltung, S = Streuung). Die Lösung erfolgt durch die Iterationen

$$(3) \quad Tu^{p+1} = \frac{1}{k_p} Fu^p + Su^{pM} \quad (k_p = ||Fu^p||)$$

Die Iterationen werden abgebrochen, wenn für $\lambda_p = k_p/k_{p-1}$ die folgende Ungleichung erfüllt ist:

$$|1-\lambda_p| \cdot 0,6 + |1-\lambda_{p-1}| \cdot 0,3 + |1-\lambda_{p-2}| \cdot 0,2 < \text{EPS}$$

wobei EPS vom Programm benutzer eingegeben wird.

4.2 Innere Iterationen

Die Lösung von

$$(T - S)u^{p+1} = \frac{1}{k_p} Fu^p$$

läßt sich auf Teilaktivitäten reduzieren, die für jede Gruppe in der Lösung eines Teil-Gleichungssystems bestehen. Diese Systeme werden durch "innere Iterationen" gelöst. Abbruch erfolgt, wenn für die Flüsse ϕ_{ij} der betreffenden Gruppe gilt: (q = Iterationsindex)

$$N \cdot \max_{i,j} |\phi_{i,j}^{(q)} - \phi_{i,j}^{(q-1)}| \leq 10 \cdot \epsilon \cdot \sum_{i,j} |\phi_{i,j}^{(p)}| \quad (N=\text{Anzahl der Raumpunkte})$$

Dabei wird i.A. $\epsilon = 0,1 \cdot \text{EPS}$ (vgl. Kap. 4.1) gesetzt. Falls vom Benutzer die Kenngröße EPSA gesetzt wurde, wählt das Programm $\epsilon = \text{EPSA}$.

4.3 Konvergenzbeschleunigungen

Die folgenden Konvergenzbeschleunigungen stehen SNOW zur Verfügung:

- 1) Rebalancing (vgl. /2/) (innere Iterationen)
- 2) Tschebyscheff-Beschleunigung (innere und äußere Iterationen)
- 3) Grobgittermethode (vgl. /4/).

Zum letzten Verfahren ist folgendes zu bemerken:

Mit Hilfe der in (vgl. Eingabebeschreibung) Größen IB in K2, ITP3 und ITP4 in K3 und den Größen von K26 ist es möglich, vor den eigentlichen SN-Lauf eine Folge von Vorläufen mit weniger Ortspunkten und einer niedrigeren Quadraturordnung vorzuschalten. Jeder folgende Lauf übernimmt dann als Anfangsflußschätzung den im Vorlauf berechneten und anschließend linear interpolierten Fluß. Durch günstige Strategien erhält man bis zu 60 % Rechenzeitgewinn. Versuchsweise wurden 120 verschiedene k_{eff} -Rechnungen durchgeführt mit verschiedenen Vorlauf-Strategien (vgl. /4/). Als im statistischen Sinne optimal (60 %) erwies sich die folgende:

Alle Vorläufe werden mit der gleichen Quadraturordnung und mit der gleichen Fehlerschranke (EPS) gerechnet. Im ersten Vorlauf sollten in jeder Zone mindestens 2x2 Stützstellen liegen. Wenn die eigentliche Rechnung mit $p \times q$ Punkten gerechnet wird, sollte der letzte Vorlauf mit $(\frac{1}{2} p) \times (\frac{1}{2} q)$ Punkten, der vorletzte und $(\frac{1}{4} p) \times (\frac{1}{4} q)$, der vorvorletzte mit $(\frac{1}{3} p) \times (\frac{1}{3} q)$ usw. rechnen. Der Rechenzeitgewinn liegt zwischen 50 und 60 %.

Zur Weitergabe der Flüsse mit oder ohne Vorläufe werden die Einheiten ITP3 und ITP4 benutzt. Am Anfang eines Falles wird der Fluß von ITP4 gelesen, falls ITP4 $\neq 0$. Eine Ausnahme (siehe K2). Am Ende eines Falles

wird auf ITP3 geschrieben, falls ITP3 \neq 0. Wird mit IB \neq 0 gerechnet, so werden für jeden neuen Fall ITP3 und ITP4 vertauscht.

5. Fremdquelle

SNOW erlaubt auch die Berechnung des inhomogenen Problems:

$$T_u = F_u + S_u + r$$

wobei r eine einzugebende "Fremdquelle" ist. Sie kann sowohl punkt- und gruppenabhängig als auch zonen- und gruppenabhängig vorgegeben werden.

6. Randbedingungen

In der Plattengeometrie (x,y) sind die Parameter für die Identifikation der Randbedingungen (vgl. Seite 13, K4) folgendermaßen zu setzen: IBL für x=0; IBR für x=R; IBU für y=0; IBO für y=R. In der (r,z)-Zylindergeometrie steht IBU für z=0; IBO für z=R und IBR für r=R. IBL ist stets 1. Schließlich ist in der (r,θ)-Zylindergeometrie IBL=3 und IBU=IBO=2 zu setzen, während IBR die Randbedingung bei r=R charakterisiert (i.A.=0).

7. Buckling

Unter bestimmten physikalischen Voraussetzungen läßt sich für ein dreidimensionales Plattengeometriesystem (x-y-z), wenn

- 1) die physikalischen Daten, d.h. die Querschnitte, von der z-Koordinate nicht abhängen,
- 2) der Winkelfluß $\psi_g(x,y,z,\Omega)$ an den z-Rändern $z = +\frac{H}{2}$ und $z = -\frac{H}{2}$ die Vakuumrandbedingung erfüllt,

ein Ansatz für den Winkelfluß $\psi_g(x,y,z,\Omega)$ in der Gruppe g in der Form

$$\psi_g(x,y,z,\Omega) = \bar{\psi}_g(x,y,\Omega) \cdot \cos Bz$$

machen. Integriert man die winkelabhängige Neutronentransportgleichung, so sieht man, daß dieser Ansatz zu einer Gleichung führt, bei der die Leckage in z-Richtung durch einen Term $DB^2 \cdot \phi$, in dem D die Diffusionskonstante ist, beschrieben wird, der wie eine zusätzliche Absorption behandelt werden kann, $\phi = \int \bar{\psi}_g(x,y,\Omega) d\Omega$.

Es gibt verschiedene Arten, die Leckage in z-Richtung mittels Buckling mehr oder minder gut zu erfassen. Man hat die Möglichkeit,

- 1) sowohl DB^2 als ganzes oder auch D und B^2 getrennt zu approximieren. Im zweiten Falle ist D immer $1/3\sigma_{tr}$, σ_{tr} hängt dann immer von der Gruppe und der Mischung ab,
- 2) DB^2 oder B^2 von der Energiegruppe oder von der Zone und der Energiegruppe abhängen zu lassen.

Falls man nur die Ausdehnung H in z-Richtung kennt, verwendet man als DB^2 den Wert

$$\left[\frac{1}{3\sigma_{tr}} \right] \left[\frac{\pi}{H+2z_0 \cdot \lambda_{tr}^g} \right]^2,$$

d.h. als B die Zahl $\frac{\pi}{H+2z_0 \cdot \lambda_{tr}^g}$, wobei $(H+2z_0 \cdot \lambda_{tr}^g)$ die extrapolierte Höhe (z-Richtung) in der Gruppe g ist. Dieser Option entspricht in der Eingabe die Größe MBK=1 in K4.

Will man, sofern man mehr Information über die Leckage in z-Richtung hat, dem Programm dieselbe zugänglich machen, so kann man sie als gruppenabhängiges wie auch als gruppen- und zonenabhängiges B^2 (MBK=2 oder MBK=3) oder DB^2 (MBK=-2 oder MBK=-3) in K13 oder K14 eingeben.

Der Fall MBK=4 bei Zylindergeometrie liegt ähnlich. Man bestimmt in einer r-z-Rechnung die B gruppen- und r-abhängig, so daß sie in einer r- θ -Rechnung verwendet werden können zur Erfassung der Leckage in z-Richtung.

Die übrigen Fälle der Zylindergeometrie (r- θ -Systeme) entsprechen den Fällen in Plattengeometrie.

8. Anisotropie

Entwickelt man den winkelabhängigen Streuquerschnitt $\sigma_s(w, v', v, \Omega', \Omega)$ nach Legendre-Polynomen $P_j(\Omega'\Omega)$ ($\Omega'\Omega =$ Streuwinkel):

$$\sigma_s(w, v', v, \Omega', \Omega) = \sum_{j=0}^M \frac{2j+1}{4\pi} \sigma_j(w, v', v) \cdot P_j(\Omega'\Omega) \quad (M \geq ISN)$$

und setzt diese Rechenentwicklung in die Streuquelle der Boltzmann-Gleichung ein, erhält man eine anisotrope Darstellung der Streuquelle. SNOW ist bei Vorgabe der Streuquerschnitte σ_j in der Lage, in jeder beliebigen Zone bis zu beliebig hohem M anisotrope Quellen zu berechnen (Querschnitteingabe siehe Kap. II,2).

II. Eingabe der Parameter

1. Eingabebeschreibung

Im folgenden sind Konventionen beschrieben, unter denen die Eingabeparameter dem Code vorgegeben werden können. Die Parameter sind Worte von Sätzen mit fester Länge (80 bytes, Lochkartenformat). Ganze Zahlen (Integers) werden grundsätzlich im FORTRAN-Format I6, reelle Zahlen (Reals) im FORTRAN-Format E 12.5 eingegeben. Die Eingabe-Records (Karten) werden im folgenden durchlaufend mit K1, K2 etc. bezeichnet; S1, S2 ... bedeutet eine Verzweigung zu bestimmten "Karten". Über die Eingabe der Querschnitte siehe II.2.

Karte	Variable	Bedeutung
K1		beliebiger Informationstext (wird ausgedruckt)
K2	ID	beliebige Identifikationszahl (>0)
	ITH	0 -reguläre Rechnung 1 -adjungierte Rechnung
	ISN	Quadraturordnung (2,4,6,8)(vgl.Kap.3.1)
	IGE	Geometrie: 0 - Platte 1 - Zylinder (R,Z) 2 - Zylinder (R,θ)
	IB	Wenn IB=n, werden vor dem SN-Lauf n verschiedene Vorläufe mit den in K26 spezifizierten Größen gerechnet. (IB<0: Im ersten Vorlauf wird der Fluß von Band ITP4 gelesen). Bei einer Rechnung mit Fremdquelle ist IB=0 zu setzen. Wenn keine Vorläufe gewünscht: IB=0 (vgl. I, 4.3)
	IB2	normal 0. Für IB2=1 werden auch die Daten der Vorläufe ausgedruckt.
	IGM	Anzahl der Gruppen
	IEVT	1-k _{eff} -Rechnung 0- Fremdquelle

Karte	Variable	Bedeutung
	IXM	Anzahl der Zonen in x bzw., R-Richtung
	IYM	Anzahl der Zonen in y bzw., z bzw. θ -Richtung
	MT	Anzahl der Mischungen
	ICM	Maximalzahl äußerer Iterationen
K3	IIM	Maximalzahl innerer Iterationen pro äußere Iteration
	IIL	(=IIM)
	KTR	-1 -Alle Flüsse werden ausgedruckt 0 -Keine Flüsse werden ausgedruckt 1 -Einige Flüsse werden ausgedruckt (werden in K20 spezifiziert)
	KDUM	0 - Querschnitte von Bandeinheit IINP 1 - Querschnitte über Karteneingabe
	IINP	0 falls KDUM=1, sonst vgl. KDUM ($\neq 5,6,9,10,11,12,13,14$)
	IREBAL	0
	ITSCH	0 -keine Tschebyscheff-Beschleunigung der äußeren Iteration 1 -Tschebyscheff-Beschleunigung
	ITP1	$\left\{ \begin{array}{l} \text{zu deklarierende Bandnummern, falls} \\ \text{ITSCH=1, sonst ohne Bedeutung, ITP1 u. ITP2} \\ (\neq 5,6,9,10,11,12,13,14) \end{array} \right.$
	ITP2	
	ITP3	Bandnummer ($\neq 5,6,9,10,11,12,13,14$)
	ITP4	Bandnummer ($\neq 5,6,9,10,11,12,13,14$) Bedeutung in I.4.3 beschrieben
K4	IBL	linke Randbedingung 0-Vakuum
	IBR	rechte Randbedingung 1-Reflektor
	IBU	untere Randbedingung 2-Periodizität
	IBO	obere Randbedingung (vgl. Kap.I.6)
	IQUELL	0 - keine Fremdquelle 1 - Fremdquelle, gruppen- und punktweise -1 - Fremdquelle, gruppen- und zonenweise (bei IQUELL=1 ist IB=0 zu setzen (K2))
	MBK	0 - kein Buckling 1 - Buckling-Korrektur und $B^2 = \frac{\sigma_{tr}^g}{3} \cdot \left\{ \frac{\pi}{BF \cdot \sigma_{tr}^g + 1,42} \right\}^2$ (BF aus K6)

Karte	Variable	Bedeutung
		2 - Gruppenabhängiges Buckling
		3 - Gruppen- und zonenabhängiges Buckling
		4 - Gruppen- und ortsabhängiges Buckling
		zu 2,3,4: Die eingegebenen Buckling- werte werden in jeder Gruppe durch $3 \cdot \sigma_{tr}^g$ dividiert (vgl. Kap. I.7.). Falls $MBK = -2$ bzw. $= -3$ wie $MBK = 2$, bzw. $= 3$, nur wird nicht durch $3 \cdot \sigma_{tr}^g$ dividiert.
	JBKTP	0 falls $MBK \neq 4$, sonst Bändeinheit, von der Buckling und Flußschätzung gelesen werden. JBKTP muß in einem früheren Fall JOUT gewesen sein (vgl. Kap. I.7)
	JOUT	Auf Einheit JOUT werden Buckling und Fluß geschrieben (vgl. Kap. I.7)
	IZT	0
	INNTS	0 - keine Tschebyscheff-Beschleunigung der inneren Iteration
		1 - Tschebyscheff-Beschleunigung
	IQUER	Ausdruck der Querschnitte: 0 nein 1 ja
	KAUSW	Auswertekennzahl letzte Ziffer: Anzahl der Bilanzenwertung für Quasizonenmodelle vorletzte Ziffer: 0 - keine Zonenbilanzen 1 - Zonenbilanzen drittletzte Ziffer: 0 - keine Mischungsbilanzen 1 - Mischungsbilanzen
K5		(Falls keine Auswertung gewünscht, ist K5 eine Leerkarte)
	ID1	
	ID2	
	ID3	siehe Kap. III.2
	ID4	
	ID5	
	NFF	normal 0 falls 1, wird ein anderes "Negativ-flux- fix-up" benutzt. Es wird nicht empfohlen, NFF=1 zu wählen.

Karte	Variable	Bedeutung
	JBKOUT	0 normal 1 Vertikale Bucklings 2 Horizontale Bucklings 3 Alle Bucklings } werden auf Unit Nr. KPCH ausgegeben.
	KPCH	siehe JBKOUT (normal 0)
	KPLOT	normal 0 falls ± 7 , werden Plots der Flüsse ausgegeben (siehe Karte 21 - 23) (KPLOT=Unit-Number)
	ISCT	0: isotrope Quellberechnung n: anisotrope Quellberechnung bis zur Ordnung n.
S1	falls ISCT=0 K7, sonst K6	
K6	MTP KTPUN1 KTPUN2	Anzahl der zu verwendenden Mischungen Unit-Nummern (dazu DD-statements definieren)
K7	EPS EV BF RHO EPSA TEPS	Fehlerschranke (vgl. Kap. I.4.1.) 1.0 normal 0. Nur definieren, falls MBK=1 (in K4) normal 0 falls $\neq 0$ und ITSCH=1 (K3) Schätzung für den Fehlerreduktionsfaktor Fehlerschranke, Falls EPSA=0, ohne Bedeutung (vgl. Kap. I.4.2) $0.006 > \text{TEPS} > 0.002$
K8	$N_1(I), I=1, IXM$	Anzahl der Zonenintervalle pro Zone in X(bzw.R)-Richtung, Summe der $N_1(I)=IM$.
K9	$N_2(I), I=1, IYM$	Anzahl der Zonenintervalle pro Zone in y (bzw. z, θ)-Richtung, Summe der $N_2(I)=JM$
K10	$RX(I), I=1, IXM$	Rechte Zonenradien (X,R-Richtung)

Karte	Variable	Bedeutung
K11	RY(I), I=1, IYM	Zonenradien in Y (bzw. Z, θ)-Richtung
K12	MZ(I, J), I=1, IXM, J=1, IYM	Mischungszahlen der Zonen, I läuft in x- (R)-Richtung und J in Y(Z, θ)-Richtung. (vgl. Kap. I.2)
S2	falls ISCT=0 in K5, nächste Karte K14, sonst K13	
K13	$\pm I$, I=1, MTP (das Vorzeichen ist +, falls für die Mischung i nur isotrope Querschnitte eingegeben werden, andernfalls: -1.	
S3	falls MBK \leq 1 oder MBK=4	S4
	falls MBK = 2, -2 K13	dann S4
	Falls MBK = 3, -3 K14	dann S4
K14	B(J), J=1, IGM	Bucklingfaktor für jede Gruppe J
K15	B(J, I), J=1, IGM, I=1, IXMxIYM	Bucklingfaktor pro Gruppe und Zone
S4	falls IQUELL=0	S5
	" " =1, K16	dann S5
	" " =2, K17	dann S5
	" " =-1, K18	dann S5
	" " =-2, K19	dann S5
K16	(((Q(I, J, K), I=1, IGM), Punkt- und gruppenabhängige Fremdquelle J=1, IM), K=1, JM)	
K17	Q1(I), I=1, IGM, ((Q2(J, K), J=1, IM), K=1, JM)	Quellspektrum Ortsabhängige Intensität. Es wird $Q(I, J, K) = Q1(I) \times Q2(J, K)$
K18	(((Q(I, J, K), I=1, IGM), Zonen- und gruppenabhängige Fremdquelle J=1, IXM), K=1, IYM)	
K19	Q1(I), I=1, IGM, ((Q2(J, K), J=1, IXM), K=1, IYM)	Quellspektrum Zonenabhängige Intensität. Es wird $Q(I, J, K) = Q1(I) \times Q2(J, K)$
S5	falls KTR=1 K20, sonst S6	
K20	N, (J(I), I=1, N)	N=Anzahl der Gruppen, für die Skalarflüsse ausgedrückt werden sollen, J(I) = Nummer einer Gruppe, für die Flüsse ausgedrückt werden soll.

Karte	Variable	Bedeutung
S6		falls $KPLOT \neq 0$ (s.K5), folgt S7, andernfalls S9
S7		falls $KPLOT=7$ K21 " $KPLOT=-7$ K22
K21	$N, (IG(I), I=1, N)$	N =Anzahl der Gruppen, die gezeichnet werden soll. $IG(I)$, die Gruppen, die gezeichnet werden sollen.
S8		Weiter zu K23
K22	$(\pm I, I=1, IGM)$	die Gruppennzahlen in aufsteigender Reihenfolge, mit + und - Zeichen, benachbarte Gruppen mit gleichem Vorzeichen werden zusammengefaßt.
K23	KNUMB MODUS IBILD	die Anzahl verschiedener Höhenlinien pro Zeichnung 1 - Äquidistante Höhenlinien 2 - Logarithmisch angeordnete Niveaus, gehäuft bei hohen Werten 3 - Logarithmisch angeordnete Niveaus, gehäuft bei niederen Werten 0 - Netzlinien werden gezeichnet 1 - Netzlinien werden nicht gezeichnet.
S9		falls $KDUM=-1$ K24, sonst S10
K24	$(X(J), J=1, IGM)$	Spaltspektrum
K25		Querschnittseingabe, vgl. 2,
S10		Falls $IB \neq 0$ für $I=1, 2, \dots, IB$ je eine Karte K26, andernfalls S11
K26	ISN' IM' JM' EPS'	$(\leq ISN)$, Quadraturordnung des Vorlaufes Nr.I $(\leq IM)$, Anzahl der Intervalle in X- oder R-Richtung bei Vorlauf Nr.I $(\leq JM)$, Anzahl der Intervalle in Y- oder Z- oder θ -Richtung bei Vorlauf Nr. I $(\neq 0)$, Fehlerschranke analog EPS für Vorlauf Nr.I
	(K26 hat das Format 3I6, E12.5)	

Karte	Variable	Bedeutung
S11		Falls KAUSW=0 oder die letzte Ziffer von KAUSW=0 S12 Falls die letzte Ziffer von KAUSW=K>0, folgt K-mal K27
K27	MU, ((MZT(I,J), I=1,IXM),J=1,IYM)	MU=Anzahl der vorhandenen Quasizonen Für jede der IXMxIYM Zonen wird ange- geben, zu welcher Quasizone K=MZT(I,J), (1≤K≤MU), sie gehört.
S12		Falls Karteneingabe für einen weiteren Fall folgt, K28, sonst K29
K28		Leerkarte, dann K1
K29		99999 (linksbündig)

2. Eingabe der Querschnitte

Die Querschnitte werden entweder über externe Einheiten mit Hilfe des Karlsruher Programmsystems NUSYS in das Programm eingegeben oder direkt über die Karteneingabe (siehe "Eingabebeschreibung"). Zur Erläuterung der letzten Möglichkeit gelten die folgenden Betrachtungen:

Der Querschnittsblock ist ein dreidimensionaler Array der Form $C(K,L,M)$. Dabei bedeutet L die Gruppenanzahl und M die Zahl der Mischung (Tabellen). K gibt die Anzahl der Typen an. Es entspricht:

	Querschnitt	NUSYS-Bezeichnung
K=1:	σ_{tr}	STR
K=2:	σ_f	SFISS
K=3:	σ_{abs}	SFISS+SCAPT
K=4:	$(\nu\sigma_f)$	NUSF
K=5:	σ_{tot}	STR oder STRTR
K=6:	$\sigma_{g \rightarrow g}$	
K=7:	$\sigma_{(g-1) \rightarrow g}$	SMTOT

Die Querschnitte werden in den Block eingegeben durch den FORTRAN-Befehl:

```
DO 1 I=1,M
  1 READ (5,5) ((C(k,1,I), k=1,K), L=1,L).
  5 FORMAT (6E125)
```

Entsprechend sind die Eingabe-Records zu setzen.

Im Falle der Anisotropie erhält die Mischungszahl P, die dem Material entspricht, das anisotrop gerechnet werden soll, ein negatives Vorzeichen. Der Teilblock

$$C(k,1,|P|) \quad (k=1, \dots, K, L=1, 2, \dots, L)$$

enthält dann die isotropen Querschnitte. Die ersten Momente befinden sich in der Tabelle $C(k,1,|P|+1)$ (wobei Nicht-Streuquerschnitte wie σ_{abs} , $\nu \cdot \sigma_g$ etc 0 zu setzen sind). Entsprechend befinden sich die anisotropen Querschnitte zweiter Ordnung auf $C(k,1,|P|+2)$ etc.

Beispiel: Bei zwei Mischungen und zwei Gruppen mit anisotroper Rechnung 1. Ordnung in der ersten Mischung hat der Querschnittsblock den folgenden Aufbau:

I \ J	1	2
1	σ_{tr}^1	σ_t^2
2	σ_f^1	σ_f^2
3	σ_{abs}^1	σ_{abs}^2
4	$(\nu\sigma_f)^1$	$(\nu\sigma_f)^2$
5	σ_{tot}^1	σ_{tot}^2
6	$\sigma_{1 \rightarrow 1}^{(0)}$	$\sigma_{2 \rightarrow 2}^{(0)}$
7	0	$\sigma_{1 \rightarrow 2}^{(0)}$

I \ J	1	2
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	$\sigma_{1 \rightarrow 1}^{(1)}$	$\sigma_{2 \rightarrow 2}^{(1)}$
7	0	$\sigma_{1 \rightarrow 2}^{(1)}$

I \ J	1	2
1	σ_{tr}^1	σ_{tr}^2
2	σ_f^1	σ_f^2
3	σ_{abs}^1	σ_{abs}^2
4	$(\nu\sigma_f)^1$	$(\nu\sigma_f)^2$
5	σ_{tot}^1	σ_{tot}^2
6	$\sigma_{1 \rightarrow 1}$	$\sigma_{2 \rightarrow 2}$
7	0	$\sigma_{1 \rightarrow 2}$

Die Materialzahlen (K...) lauten entsprechend: -1, 2. Es ist $MT=3$ und $MTP=2$

III. Ausgabe (Output)

1. Die Ausgabeliste

Nach der Reproduktion der Eingangsparameter folgt der "Iteration monitor", ein Protokoll über den Vorlauf der Iterationszyklen. Für jede äußere Iteration wird die Gesamtzahl der bis dahin ausgeführten inneren Iterationen, der Eigenwert (k_{eff}), der Wert $k_{\text{eff}}^{(n)}/k_{\text{eff}}^{(n-1)} = \text{LAMBDA}$ sowie der ermittelte Reduktionsfaktor ausgedruckt. Es folgen die Anzahl der inneren Iterationen pro Gruppe sowie Rebalancing-Faktoren, deren Abstand von 1 die Güte der in den inneren Iterationen errechneten Werte an gibt.

Hinter der Wiedergabe der Bilanzen findet sich unter "EIGENVALUE" der endgültig errechnete Wert für k_{eff} . Schließlich werden auf Wunsch die über alle Winkel integrierten Flüsse ausgedruckt.

Es besteht die Möglichkeit, Flüsse und Bucklings auf externe Einheiten zu schreiben.

2. Auswertung

Die durch die räumliche Diskretisierung definierten Raumvolumina seien V_{ij} . Es sei σ_g ein vom Gruppenindex g abhängender makroskopischer Querschnitt sowie $\phi_g(i,j)$ der (diskretisierte) Fluß. Dann können die folgenden Werte berechnet und ausgedruckt werden:

$$(1) \quad \sum_g \sigma_g \cdot \phi_g(i,j) \quad \text{für alle } i,j$$

für alle Querschnittstypen sowie

$$(2) \quad \sum_g \sigma_g \cdot \phi_g(i,j) \cdot V_{ij} \quad \text{für alle } i,j$$

Die Auswertung (1), (2) wird durch die Parameter ID1, ID2, ID3, ID4, ID5 in K5 gesteuert. Es bedeuten

0 - keine Auswertung
ID1= 1 - Auswertung (1) für $\sigma_g = \text{STR}$
2 - Auswertung (2)

ID2 - wie ID1, für $\sigma_g = \text{SFISS}$
ID3 - wie ID1, für $\sigma_g = \text{SCAPT+SFISS}$
ID4 - wie ID1, für $\sigma_g = \text{NUSF}$
ID5 - wie ID1, für $\sigma_g = \text{STRTR}$

3. Bilanzen

Die Bilanzen, die SNOW ausdrückt, sind von vierfacher Form:

- 1) Systembilanz, wird immer ausgedruckt
- 2) Mischbilanzen, werden wahlweise ausgedruckt (KAUSW=...1..)
- 3) Zonenbilanzen, werden wahlweise ausgedruckt (KAUSW=...1.)
- 4) Quasizonenbilanzen, werden wahlweise ausgedruckt (KAUSW=....n)

Falls KAUSW=0, wird nur die Systembilanz ausgegeben, andernfalls veranlassen die Ziffern von KAUSW weitere Bilanzenausdrucke. Diese sind i.a. selbsterläuternd. Folgende Bemerkungen sind dazu noch notwendig.

- 1) Bei der Systembilanz wird die Leckage aus den Winkelflüssen berechnet, bei den übrigen Bilanzen aus der Neutronenbilanz. Verlangt man die Zonenbilanzen, so erhält man zusätzlich die Leckage aus dem System (!) pro Zone, die am Rand liegt. Außerdem werden pro vertikales und horizontales Zonenband mittlerer Fluß, mittlere (systemverlassende) Leckage und das Verhältnis (Leckage/mittl. Fluß) pro Gruppe ausgedruckt. Diese Größen können u.U. bei eindimensionalen Rechnungen als Buckling (für DB^2) verwendet werden. Das Verhältnis "Leckage/mittl.Fluß" kann pro Gruppe und vertikale und/oder horizontale Zone auf Einheit KPCH geschrieben werden. Dieses Schreiben findet formalisiert mit dem Format 6E12.5 statt. Die Streuung dieser Möglichkeit geschieht über die Eingabegröße JBKOUT.

2) Unter Quasizonenbilanzen wird folgendes verstanden:

Man denkt sich jede der $IXM \times IYM$ Zonen einer Quasizone i $1 \leq i \leq m$ zugehörig. Über diese Quasizonen werden dann Bilanzen berechnet. Ist KAUSW=...n, so wird dies n mal getan. Man muß dazu jedesmal $(IXM \times IYM + 1)$ Zahlen einlesen, zuerst m, die Anzahl der Quasizonen und dann pro Zone in der üblichen Reihenfolge die Zugehörigkeit zur entsprechenden Quasizone. Man kann damit z.B. eindimensionale radiale und axiale Bilanzen simulieren.

4. Graphische Auswertung

Darstellungen von Höhenlinien des Neutronenflusses sind möglich. Diese werden in einem Rechteckgitter, mit (x,y) , (r,z) oder (r,θ) Koordinaten eingetragen.

Man kann solche Darstellungen für alle Gruppen (KPLOT=7) oder auch für Zusammenfassungen von Energiebereichen (=Gruppen) erhalten (KPLOT=-7). Die weiteren Parameter, die die Form dieser graphischen Ausgabe steuern, sind in der Eingabebeschreibung unter K23 angegeben. Sollen diese Darstellungen auf Papier gezeichnet werden, so ist eine Job-Control-Karte der Gestalt

```
|| G.FT07FOO1 DD SYSOUT=P
```

zu verwenden.

Es besteht auch die Möglichkeit, diese Darstellung über das System TCP auf einen Bildschirm zu bringen. Da dieses System in näherer Zukunft nicht mehr zur Verfügung stehen wird, soll diese Variante hier nicht beschrieben werden.

5. Speicherung von Flüssen

Die Flüsse und einige weitere informative Daten können temporär oder zur Weiterverwendung auf die Einheiten ITP1, ITP2, ITP3 oder ITP4 geschrieben oder von diesen auch wieder gelesen werden.

ITP1 und ITP2 werden bei der Tschebyscheff-Extrapolation für die äußeren Iterationen benötigt, wenn sich nicht alle zu speichernden Flüsse im Kernspeicher halten lassen. Sie sind nur zum internen Gebrauch gedacht.

Zur Flußweitergabe werden die Einheiten ITP3 und ITP4 benutzt, deren Funktion in Abschnitt I.4.3. beschrieben wird. Ist IB=0 gewesen, so steht nach Abschluß einer Rechnung, falls ITP3≠0 ist, eine Datenmenge auf Einheit ITP3, die u.a. die Skalarflüsse enthält und deren Struktur weiter unten beschrieben wird. Man kann diese Daten daher auch unabhängig von SNOW weiterverarbeiten.

Ist IB=0 gewesen, so steht am Ende der Richtung dieselbe Datenmenge auf ITP3 oder ITP4. Auf welcher Einheit diese Daten stehen, läßt sich der Druckausgabe entnehmen. Falls ITP3=ITP4 ist, dies ist zulässig und in vielen Fällen sinnvoll, erübrigt sich die Fragestellung. Die Datenmenge, die auch die Skalarflüsse enthält, wird in verschiedenen Sätzen (records) geschrieben. Das Unterprogramm, in dem dies geschieht, heißt TOPROV. Es enthält der 1.Satz 13 Integer-Zahlen, deren Namen sind ID, ICC, IM, JM, IGM, IXM, IYM, ISN, MM, IBL, IBR, IBU, IBO. Die Benutzung der Namen ID, IM, JM, IGM, IXM, IYM, ISN, IBL, IBR, IBU, IBO läßt sich der Eingabebeschreibung, Abschnitt II, entnehmen. Es bedeutet

ICC - Anzahl der äußeren Iterationen des abgespeicherten Falls
MM - $ISN \times (ISN+4)/2$.

es enthält weiter

Satz 2 - die Zonenradien in X-(R-)-Richtung
Satz 3 - " " " Y-(Z-O)-Richtung
Satz 4 - die Intervallradien in Y-(R-)-Richtung
Satz 5 - " " " Y-(Z-O)-Richtung
Satz 6 - zwei REAL x 4 Zahlen.

Dann folgen IGM Sätze, die mit den folgenden beiden FORTRAN-Anweisungen geschrieben werden:

```
DO 5 IG = 1, IGM  
5 WRITE (ITP3) ((XN(I,J,IG), I=1, IM), J=1, JM)
```

XN(I,J,IG) ist der skalare Neutronenfluß im 1.Punkt der X-(R-)-Richtung im J. Punkt der Y-(R-O-)-Richtung und in der IG.Energiegruppe.

Falls die Größe IBL, IBR, IBU, IBO nicht alle den Wert Null haben, enthält die beschriebene Datenmenge noch winkelabhängige Neutronenflüsse des Systemrandes. Eine genaue Beschreibung der Struktur dieser Daten soll hier nicht vorgenommen werden.

IV. Implementierung

1. Allgemeines

SNOW ist ein FORTRAN IV-Programm, bestehend aus 35 Subroutinen und ca. 6060 Statements. Es wurde auf der IBM 370/165 ausgetestet, zur Übersetzung wurde der H2-Compiler benutzt.

2. Kurze Beschreibung der Subroutinen

- Subroutine ADJG: Transponiert die Matrizen für eine adjungierte Rechnung.
- Subroutine ANISO: Berechnet Kugelfunktionen für anisotrope Quellberechnungen.
- Subroutine ANFISS: Berechnet die Anfangsquellschätzung.
- Subroutine BILAN1: Berechnet Bilanzen.
- Subroutine DUMMY: Fixiert den Gesamtspeicherbedarf.
- Subroutine FFISSION: Berechnet die Spaltquelle.
- Subroutine FOLGE: Setzt und berechnet notwendige Parameter für Vorläufe (Folgeläufe)
- Subroutine GECKO: Führt eine Eingabepfung nach Plausibilitätsüberlegungen durch.
- Subroutine GEO: Berechnet die Koeffizienten der Differenzgleichungen.
- Subroutine IFUNC: Hilfsroutine zur Initialisierung der Spaltspektren.
- Subroutine INNER: Führt die inneren Iterationen durch.
- Subroutine IPOLAN: Teilroutine zu PROV1
- Subroutine LEVLOC: Zeichnet Flußhöhenlinien.
- Subroutine UESZ: Übernimmt die Querschnitte von dem in Karlsruhe zur Verfügung stehenden Programmsystem NUSYS.
- Subroutine MALE: Zeichnet eine Skizze der zu berechnenden Reaktor-Anordnung.
- Subroutine MESSAG:
- Subroutine NEWS: Druckt am Jobanfang neue Mitteilungen aus für den Benutzer
- Subroutine ORDV: Erstellt Steuerparameter für den Output der Flüsse.
- Subroutine OUTER: Führt die äußeren Iterationen durch und berechnet die isotrope Streuquelle.

Subroutine PROU1: a) Liest mit TOPROU auf Einheit ITP3 geschriebene Flüsse, jetzt unter dem Namen ITP4.
b) Interpoliert bei nichtübereinstimmender Ortspunktzahl und nichtübereinstimmender S_N -Ordnung die Skalarflüsse und die Winkelflüsse am Systemrand.

Subroutine PUT: Hilfsroutine zum Auffüllen von Arrays.

Subroutine REMUS: -Dummy-Subroutine. Sie dient als offenes Programmende. Sie kann bei einem CLG-Lauf durch eine beliebige Routine mit gleichem Namen ersetzt werden.

Subroutine READER: Hilfsroutine

Subroutine RICH2, RICH4, RICH6, RICH8: Die Routinen erstellen Richtungen und Gewichte für eine S2, S4, S6, S8-Rechnung.

Subroutine SECRET: Druckt alle Parameter im Falle eines anormalen Abbruches einer SN-Rechnung.

Subroutine SNOW: Die Routine steuert den Gesamt Ablauf einer SN-Rechnung.

Subroutine TEMP1: Führt die Tschebyscheff- Extrapolation für die inneren Iteration durch und kontrolliert deren Ablauf.

Subroutine TOMBO: Hilfsroutine

Subroutine TOPROV: Schreibt die Flüsse auf mit ITP3, so daß sie mit der Routine PROU1 wieder gelesen werden können.

Subroutine TSCHEB: Führt die Tschebyscheff- Extrapolation für die äußeren Iterationen durch.

Subroutine WOORG: Hilfsprogramm für LIES2.

Subroutine WRIT: Hilfsprogramm für den Output.

Subroutine WRITER: Hilfsroutine für TSCHEB.

2. Erhöhung des Speicherbedarfs

Falls der Kernspeicherplatz, der durch die Dimensionierung der Hauptarrays A(I); IA(I) vorbestimmt ist, nicht ausreicht, läßt sich eine Vergrößerung des Speicherplatzes leicht dadurch erreichen, daß die erwähnten Arrays in ihrer Dimension (normal:A(27000), IA(1000)) in der aus sieben Statements bestehenden Hilfsroutine DUMMY entsprechend vergrößert werden.

Literatur

- /1/ B. Carlson, C.E. Lee
Mechanical quadrature and the transport equation
LA-2573 (1961)
- /2/ K. Lathrop
DTF-IV, a FORTRAN IV-Program for solving the multigroup transport
equation with anisotropic scattering
LA-3373 (1965)
- /3/ B. Carlson, K. Lathrop
Numerical solution of the Boltzmann-transport equation
J.of Comp.phys. 2, 173-197 (1967)
- /4/ W. Kinnebrock
Strategien zur Beschleunigung zweidimensionaler SN-Verfahren
mit Hilfe der Grobgittertechnik
KFK 1361 (1971)
- /5/ N.K.Madsen
Convergence of Singular Difference Approximations for the
Discrete Ordinate Equations in x-y-Geometry
Math.of Comp. 26, 45, 1972.

Anhang I

Sample Problem

Anhang II

Programmliste

Anhang III

Anlageabhängige Teile

SNOW enthält einige Programmteile, die bei einer Übertragung des Codes auf eine andere Anlage nicht funktionieren werden. Es handelt sich dabei fast ausschließlich um Teile, die Unterprogramme aufrufen, die nur auf der Karlsruher Anlage verfügbar sind.

1. Die Möglichkeit der Wirkungsquerschnittseingabe mit dem Parameter KDUM=0 ist nur gegeben, wenn gleichzeitig sogenannte SIGMN-Blöcke aus dem Programmsystem NUSYS verfügbar sind. Das wird im Normalfall nur der Fall sein, wenn dieses System ebenfalls verfügbar sein wird.
2. Die Möglichkeiten zur graphischen Ausgabe benutzen eine spezielle in Karlsruhe viel benutzte Routine PLOTA. Bei externer Benutzung sollte daher KPLOT=0 gewählt werden.
3. Um alle anlageabhängigen Referenzen auch bei externer Benutzung auflösen zu können, wird empfohlen, folgendes Kartenpaket bei der Compilation hinzuzufügen.

SUBROUTINE PLOTA	FUNCTION ZEIT (X)
ENTRY FSPIE	RETURN
ENTRY FREESP	END

Anhang I: Sample Problem

***** SNEAK 3A1 - SNOW-RECHNUNG 0000070*****

PEAL CORE STORAGE 10955
INTEGER CORE STORAGE 338
NUMBER OF INTERNAL TAPES USED 0

IDENTIFICATION 101 (ID)
THEORY (0=REGULAR,1=ADJOINT) 0 (ITH)
SN-ORDER 4 (ISN)
GEOMETRY (0=(X,Y),1=(R,Z),2=(R,THETA)) 0 (IGE)
NUMBER OF GROUPS 4 (IGM)
NUMBER OF ZONES (X/R-DIRECTION) 3 (IZM)
NUMBER OF ZONES (Y/Z/THETA-DIRECTION) 1 (IYM)
NUMBER OF MATERIALS 3 (MT)
NUMBER OF INTERVALS (X/P-DIRECTION) 65 (IM)
NUMBER OF INTERVALS (Y/Z/THETA-DIRECTION) 3 (JM)
MAXIMUM NUMBER OF OUTER ITERATIONS 80 (ICM)
MAXIMUM NUMBER OF INNER ITERATIONS 80 (IIM)
INITIAL-MAXIMUM NUMBER OF INNER ITERATIONS 80 (IIL)
REBALANCING USED (1=NO,0=YES) 1 (IREBAL)
TSCHEBYSCHIEFF-ACCELERATION USED (0=NO,1=YES) 0 (ITSCH)
CROSS SECTION INPUT BY TAPE (1=NO/0=YES) 1 (KDUM)
CROSS SECTION TAPE NUMBER 0 (INP)
PRINT CROSS SECTION TABLES (0=NO,1=YES) 1 (IQUER)
FIRST FILE FOR OUTER TCHEBYCHEW EXTRAP. 0 (ITP1)
SECOND FILE,SAME PURPOSE,DIFFERENT FROM ITP1 0 (ITP2)
FINAL FLUX WRITTEN ON UNIT (IF=0 NOT USED) 0 (ITP3)
AFTER EACH OUTER ITERATION
INITIAL FLUX READ FROM UNIT (0=NOT USED) 0 (ITP4)
EVALUATION TRIGGER,BALANCES 0 (KAUSW)
NUMBER OF INTERPOLATING STEPS 0 (IBI)
TCHEBYCHEW ACCELERATION IN INNER(0=NO) 0 (INNTS)
DISTRIBUTED SOURCE (0=NO,1=BY GROUP AND
SPACE,-1=BY GROUP AND ZONE 0 (IQUELL)
BUCKLING(0=NO,1=YES,2=BY GROUP,3=BY GROJP
AND ZONE,4=BY GROUP AND SPACE 0 (MBK)
PROBLEM TYPE 1 (IEVT)
DETAILED NEG. FLUX FIXUP HANDLING(0=NO,1=YES) 0 (NFF)
BUCKLING OUTPUT ON (PCH 0 (JBKOUT)
UNIT TO GET BUCKLINGS

CONVERGENCE CRITERION 0.10000E-03 (EPS)
BUCKLING HEIGHT 0.0 (DZ)
FLUX CONVERGENCE CRITERION 0.0 (EPSA)

RADII (X/R-DIRECTION) 66

```

0.0            0.74074F 00 0.14815E 01 0.22222E 01 0.29630E 01 0.37037E 01 0.44444E 01 0.51852E 01 0.59259E 01 0.66667E 01
0.74074F 01 0.81481E 01 0.88889E 01 0.96296E 01 0.10370E 02 0.11111E 02 0.11852E 02 0.12593E 02 0.13333E 02 0.14074E 02
0.14815E 02 0.15556E 02 0.16296E 02 0.17037E 02 0.17778E 02 0.18518E 02 0.19259E 02 0.20000E 02 0.21000E 02 0.22000E 02
0.23000E 02 0.24000E 02 0.25000E 02 0.26000E 02 0.27000E 02 0.28000E 02 0.29000E 02 0.30000E 02 0.31000E 02 0.32000E 02
0.33000E 02 0.34000E 02 0.35000E 02 0.36000E 02 0.37000E 02 0.38000E 02 0.39000E 02 0.40000E 02 0.40833E 02 0.41667E 02
0.42500E 02 0.43333E 02 0.44167E 02 0.45000E 02 0.45833E 02 0.46667E 02 0.47500E 02 0.48333E 02 0.49167E 02 0.50000E 02
0.50833E 02 0.51667E 02 0.52500E 02 0.53333E 02 0.54167E 02 0.55000E 02
    
```

RADII (Y/Z/THETA-DIRECTION) 4

```

0.0            0.33333E 00 0.66667E 00 0.10000E 01
    
```

THE 65 INTERVALS IN (X/R)-DIRECTION ARE LOCATED IN THE ZONES WITH NUMBER

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	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3

THE 3 INTERVALS IN (Y/Z/THETA)-DIRECTION ARE LOCATED IN THE ZONES WITH NUMBER

```

1            1            1
    
```

BOUNDARY CONDITIONS

```

X=0: REFLEXION
X=R: VACUUM
Y=0: PERIODIC
Y=R: PERIODIC
    
```

DIRECTIONS REFL.DIRECTIONS WEIGHTS

MUE	ETA	MUE	ETA		
-0.94281E 00	-0.33333E 00	5	9	0.0	1
-0.88192E 00	-0.33333E 00	5	10	0.83333E-01	2
-0.33333E 00	-0.33333E 00	4	11	0.83333E-01	3
0.33333E 00	-0.33333E 00	3	12	0.83333E-01	4
0.88192E 00	-0.33333E 00	2	13	0.83333E-01	5
-0.47140E 00	-0.88192E 00	8	14	0.0	6
-0.33333E 00	-0.88192E 00	8	15	0.83333E-01	7
0.33333E 00	-0.88192E 00	7	16	0.83333E-01	8
-0.94281E 00	0.33333E 00	13	1	0.0	9
-0.88192E 00	0.33333E 00	13	2	0.83333E-01	10
-0.33333E 00	0.33333E 00	12	3	0.83333E-01	11
0.33333E 00	0.33333E 00	11	4	0.83333E-01	12
0.88192E 00	0.33333E 00	10	5	0.83333E-01	13
-0.47140E 00	0.88192E 00	16	6	0.0	14
-0.33333E 00	0.88192E 00	16	7	0.83333E-01	15
0.33333E 00	0.88192E 00	15	8	0.83333E-01	16

FISSION FRAC 5

0.90500E 00 0.95000E-01 0.0 0.10000E 01

MATERIAL 1, GROUP 1, (1.0 - CALC TOT X-SECT/INPUT TOT X-SECT) = -0.287128E-03

MATERIAL 2, GROUP 1, (1.0 - CALC TOT X-SECT/INPUT TOT X-SECT) = 0.550648E-02

MATERIAL 3, GROUP 1, (1.0 - CALC TOT X-SECT/INPUT TOT X-SECT) = 0.388803E-02

MATERIAL 3, GROUP 2, (1.0 - CALC TOT X-SECT/INPUT TOT X-SECT) = -0.244741E-01

CROSS SECTIONS

MATERIAL 1

GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP
0.17246E 00	0.28344E 00	0.39199E 00	0.49656E 00	
0.0	0.0	0.0	0.0	
0.50919E 02	0.90090E-02	0.45315E-01	0.16841E 00	
0.98967E 02	0.11115E-01	0.52403E-01	0.16795E 00	
0.17246E 00	0.28344E 00	0.39199E 00	0.49656E 00	
0.15018E 00	0.27276E 00	0.34588E 00	0.32817E 00	
0.0	0.17228E-01	0.16690E-02	0.80936E-03	
0.0	0.0	0.95905E-05	0.89721E-05	
0.0	0.0	0.0	0.0	

MATERIAL 2

GROUP 1	GROUP 2	GROUP 3	GROUP 4	GROUP
---------	---------	---------	---------	-------

***** ITERATION MONITOR *****

OUTER ITERATIONS	INNER ITERATIONS	EIGENVALUE	LAMBDA	REDUCT.FACTOR
0	0	1.00000000	0.0	0.10000000E 01
GROUP	INNER ITERATIONS	PER GROUP		
1	0	0.0	0	0.0
2	0	0.0	0	0.0
3	0	0.0	0	0.0
4	0	0.0	0	0.0
1	160	0.54896414	0.54896384E 00	0.45103616E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	80	0.10155239E 01	0	0.0
2	80	0.73076129E 00	0	0.0
3	0	0.0	0	0.0
4	0	0.0	0	0.0
2	447	1.00854015	0.18371696E 01	-0.18561029E 01
GROUP	INNER ITERATIONS	PER GROUP		
1	80	0.10007973E 01	0	0.0
2	80	0.11107149E 01	0	0.0
3	80	0.10070686E 01	0	0.0
4	47	0.10001068E 01	0	0.0
3	701	1.15504456	0.11452637E 01	0.17351758E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	57	0.10002470E 01	0	0.0
2	80	0.10397139E 01	0	0.0
3	80	0.10014315E 01	0	0.0
4	37	0.10001125E 01	0	0.0
4	914	1.20729923	0.10452404E 01	0.31143641E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	25	0.10002003E 01	0	0.0
2	80	0.10149403E 01	0	0.0
3	77	0.10005560E 01	0	0.0
4	31	0.10001049E 01	0	0.0
5	1097	1.22644901	0.10158615E 01	0.35060495E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	24	0.10001297E 01	0	0.0
2	80	0.10058174E 01	0	0.0
3	55	0.10006104E 01	0	0.0
4	24	0.10001240E 01	0	0.0

6	1260	1.23349571	0.10057459E 01	0.36225349E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	27	0.10000467E 01	0	0.0
2	80	0.10023336E 01	0	0.0
3	38	0.10005884E 01	0	0.0
4	18	0.10001326E 01	0	0.0
7	1345	1.23474121	0.10010099E 01	0.17576760E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	25	0.10000057E 01	0	0.0
2	30	0.10017033E 01	0	0.0
3	19	0.10004549E 01	0	0.0
4	11	0.10001068E 01	0	0.0
8	1383	1.23502636	0.10002308E 01	0.22051741E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	15	0.99999517E 00	0	0.0
2	10	0.10015392E 01	0	0.0
3	7	0.10004234E 01	0	0.0
4	6	0.10000887E 01	0	0.0
9	1402	1.23513126	0.10000849E 01	0.36776859E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	7	0.99999231E 00	0	0.0
2	4	0.10014763E 01	0	0.0
3	4	0.10003901E 01	0	0.0
4	4	0.10000772E 01	0	0.0
10	1409	1.23515129	0.10000162E 01	0.19101119E 00
GROUP	INNER ITERATIONS	PER GROUP		
1	4	0.99999100E 00	0	0.0
2	1	0.10014591E 01	0	0.0
3	1	0.10003796E 01	0	0.0
4	1	0.10000744E 01	0	0.0

NEUTRON BALANCE OF THE WHOLE SYSTEM: 1

GP.	EXTERN.SOURCE	FISSION SOURCE	INSCATTER	TOTAL PRODUCT.	TOTAL FLUX	FISSION PROD.	SELFSCATTER
1	0.0	0.7326970E 00	0.0	0.7326970E 00	0.3149583E 02	0.3125668E 00	0.4854129E 01
2	0.0	0.7691276E-01	0.5478097E 00	0.6247224E 00	0.6124174E 02	0.5884868E 00	0.1794461E 02
3	0.0	0.0	0.8932173E-01	0.8932173E-01	0.1959155E 01	0.9701574E-01	0.6849385E 00
4	0.0	0.0	0.1971111E-02	0.1971111E-02	0.1182033E-01	0.1943029E-02	0.3889271E-02
5	0.0	0.8096097E 00	0.6391025E 00	0.1448711E 01	0.9470854E 02	0.1000011E 01	0.2348755E 02

GP.	ABSORPTION	BUCKL.LEAKAGE	OUTSCATTER	TOTAL LEAKAGE	TOTAL LOSSES	CAPTURE DENSITY	FISSION DENSITY
1	0.1558189E 00	0.0	0.5489908E 00	0.1630848E-01	0.7202182E 00	0.1558189E 00	0.0
2	0.5657695E 00	0.0	0.8951318E-01	0.5625579E-01	0.7115384E 00	0.5657695E 00	0.0
3	0.8736736E-01	0.0	0.1498986E-02	0.3976682E-03	0.8926392E-01	0.8736736E-01	0.0
4	0.1970684E-02	0.0	0.0	0.1977304E-06	0.1970882E-02	0.1970684E-02	0.0
5	0.8109264E 00	0.0	0.6391029E 00	0.7296205E-01	0.1522991E 01	0.8109264E 00	0.0

GP.	NEUTRON BAL.	LEFT LEAKAGE	RIGHT LEAKAGE	LOWER LEAKAGE	UPPER LEAKAGE	HORIZ. LEAKAGE	VERTIC. LEAKAGE
1	0.1017326E 01	0.0	0.1630848E-01	0.0	0.0	0.1630848E-01	0.0
2	0.8779883E 00	0.0	0.5625579E-01	-0.4564448E-05	0.4564448E-05	0.5625579E-01	0.0
3	0.1000648E 01	0.0	0.3976682E-03	0.1901853E-06	-0.1901853E-06	0.3976682E-03	0.0
4	0.1000115E 01	0.0	0.1977304E-06	0.1485823E-08	-0.1485823E-08	0.1977304E-06	0.0
5	0.9512277E 00	0.0	0.7296205E-01	-0.4372777E-05	0.4372777E-05	0.7296205E-01	0.0

FLUX BY GROUP AND SPACE

(I=X/R-DIRECTION, J=Y/Z/THETA-DIRECTION)

GROUP 1

J/I	0.3704	1.1111	1.8519	2.5926	3.3333	4.0741	4.8148	5.5555	6.2963	7.0370
0.1667	0.90791E 00	0.90753E 00	0.90580E 00	0.90569E 00	0.90423E 00	0.90240E 00	0.90020E 00	0.89765E 00	0.89473E 00	0.89145E 00
0.5000	0.90791E 00	0.90753E 00	0.90680E 00	0.90569E 00	0.90423E 00	0.90240E 00	0.90020E 00	0.89765E 00	0.89473E 00	0.89144E 00
0.8333	0.90791E 00	0.90753E 00	0.90680E 00	0.90570E 00	0.90423E 00	0.90240E 00	0.90020E 00	0.89765E 00	0.89473E 00	0.89145E 00
J/I	7.7778	8.5185	9.2592	10.0000	10.7407	11.4815	12.2222	12.9629	13.7037	14.4444
0.1667	0.88780E 00	0.88380E 00	0.87944E 00	0.87471E 00	0.86963E 00	0.86419E 00	0.85840E 00	0.85225E 00	0.84574E 00	0.83888E 00
0.5000	0.88780E 00	0.88380E 00	0.87943E 00	0.87471E 00	0.86963E 00	0.86419E 00	0.85840E 00	0.85225E 00	0.84574E 00	0.83888E 00
0.8333	0.88780E 00	0.88380E 00	0.87943E 00	0.87471E 00	0.86963E 00	0.86419E 00	0.85840E 00	0.85225E 00	0.84574E 00	0.83888E 00
J/I	15.1852	15.9259	16.6666	17.4074	18.1481	18.8888	19.6296	20.5000	21.5000	22.5000
0.1667	0.83166E 00	0.82409E 00	0.81617E 00	0.80789E 00	0.79925E 00	0.79025E 00	0.78089E 00	0.76940E 00	0.75565E 00	0.74130E 00
0.5000	0.83166E 00	0.82409E 00	0.81617E 00	0.80789E 00	0.79925E 00	0.79025E 00	0.78089E 00	0.76939E 00	0.75565E 00	0.74130E 00
0.8333	0.83166E 00	0.82409E 00	0.81617E 00	0.80789E 00	0.79925E 00	0.79025E 00	0.78089E 00	0.76940E 00	0.75565E 00	0.74130E 00
J/I	23.5000	24.5000	25.5000	26.5000	27.5000	28.5000	29.5000	30.5000	31.5000	32.5000
0.1667	0.72632E 00	0.71070E 00	0.69443E 00	0.67749E 00	0.65987E 00	0.64152E 00	0.62244E 00	0.60256E 00	0.58184E 00	0.56021E 00
0.5000	0.72632E 00	0.71070E 00	0.69443E 00	0.67749E 00	0.65987E 00	0.64153E 00	0.62244E 00	0.60256E 00	0.58184E 00	0.56021E 00
0.8333	0.72632E 00	0.71070E 00	0.69443E 00	0.67749E 00	0.65987E 00	0.64152E 00	0.62244E 00	0.60256E 00	0.58184E 00	0.56021E 00
J/I	33.5000	34.5000	35.5000	36.5000	37.5000	38.5000	39.5000	40.4166	41.2499	42.0833
0.1667	0.53757E 00	0.51381E 00	0.48877E 00	0.46224E 00	0.43391E 00	0.40340E 00	0.37015E 00	0.33373E 00	0.29826E 00	0.26748E 00
0.5000	0.53757E 00	0.51381E 00	0.48877E 00	0.46224E 00	0.43391E 00	0.40340E 00	0.37015E 00	0.33373E 00	0.29826E 00	0.26748E 00
0.8333	0.53757E 00	0.51381E 00	0.48877E 00	0.46224E 00	0.43391E 00	0.40340E 00	0.37015E 00	0.33373E 00	0.29826E 00	0.26748E 00
J/I	42.9166	43.7499	44.5833	45.4166	46.2499	47.0832	47.9166	48.7499	49.5832	50.4165
0.1667	0.24040E 00	0.21633E 00	0.19472E 00	0.17518E 00	0.15740E 00	0.14112E 00	0.12613E 00	0.11224E 00	0.99303E-01	0.87159E-01
0.5000	0.24041E 00	0.21633E 00	0.19472E 00	0.17518E 00	0.15740E 00	0.14112E 00	0.12613E 00	0.11224E 00	0.99303E-01	0.87160E-01
0.8333	0.24040E 00	0.21633E 00	0.19472E 00	0.17518E 00	0.15740E 00	0.14112E 00	0.12613E 00	0.11224E 00	0.99303E-01	0.87159E-01
J/I	51.2499	52.0832	52.9165	53.7499	54.5832					
0.1667	0.75662E-01	0.64655E-01	0.53966E-01	0.43390E-01	0.32669E-01					
0.5000	0.75662E-01	0.64655E-01	0.53966E-01	0.43390E-01	0.32670E-01					
0.8333	0.75662E-01	0.64655E-01	0.53966E-01	0.43390E-01	0.32669E-01					

GROUP 2

J/I	0.3704	1.1111	1.8519	2.5926	3.3333	4.0741	4.8148	5.5555	6.2963	7.0370
0.1667	0.15594E 01	0.15588E 01	0.15576E 01	0.15559E 01	0.15535E 01	0.15505E 01	0.15470E 01	0.15429E 01	0.15382E 01	0.15329E 01
0.5000	0.15594E 01	0.15588E 01	0.15576E 01	0.15559E 01	0.15535E 01	0.15505E 01	0.15470E 01	0.15429E 01	0.15382E 01	0.15329E 01
0.8333	0.15594E 01	0.15588E 01	0.15576E 01	0.15559E 01	0.15535E 01	0.15505E 01	0.15470E 01	0.15429E 01	0.15382E 01	0.15329E 01

J/I	7.7778	8.5185	9.2592	10.0000	10.7407	11.4815	12.2222	12.9629	13.7037	14.4444
0.1667	0.15270E 01	0.15206E 01	0.15137E 01	0.15061E 01	0.14980E 01	0.14894E 01	0.14803E 01	0.14706E 01	0.14604E 01	0.14497E 01
0.5000	0.15270E 01	0.15206E 01	0.15137E 01	0.15061E 01	0.14980E 01	0.14894E 01	0.14803E 01	0.14706E 01	0.14604E 01	0.14497E 01
0.8333	0.15270E 01	0.15206E 01	0.15137E 01	0.15061E 01	0.14980E 01	0.14894E 01	0.14803E 01	0.14706E 01	0.14604E 01	0.14497E 01

J/I	15.1852	15.9259	16.5666	17.4074	18.1481	18.8888	19.6296	20.5000	21.5000	22.5000
0.1667	0.14385E 01	0.14268E 01	0.14146E 01	0.14020E 01	0.13889E 01	0.13754E 01	0.13615E 01	0.13446E 01	0.13245E 01	0.13038E 01
0.5000	0.14385E 01	0.14268E 01	0.14146E 01	0.14020E 01	0.13889E 01	0.13754E 01	0.13615E 01	0.13446E 01	0.13245E 01	0.13038E 01
0.8333	0.14385E 01	0.14268E 01	0.14146E 01	0.14020E 01	0.13889E 01	0.13754E 01	0.13615E 01	0.13446E 01	0.13245E 01	0.13038E 01

J/I	23.5000	24.5000	25.5000	26.5000	27.5000	28.5000	29.5000	30.5000	31.5000	32.5000
0.1667	0.12825E 01	0.12606E 01	0.12383E 01	0.12155E 01	0.11923E 01	0.11689E 01	0.11453E 01	0.11216E 01	0.10980E 01	0.10745E 01
0.5000	0.12825E 01	0.12606E 01	0.12383E 01	0.12155E 01	0.11923E 01	0.11689E 01	0.11453E 01	0.11216E 01	0.10980E 01	0.10745E 01
0.8333	0.12825E 01	0.12606E 01	0.12383E 01	0.12155E 01	0.11923E 01	0.11689E 01	0.11453E 01	0.11216E 01	0.10980E 01	0.10745E 01

J/I	33.5000	34.5000	35.5000	36.5000	37.5000	38.5000	39.5000	40.4166	41.2499	42.0833
0.1667	0.10513E 01	0.10286E 01	0.10067E 01	0.98591E 00	0.96663E 00	0.94960E 00	0.93592E 00	0.92221E 00	0.90258E 00	0.87624E 00
0.5000	0.10513E 01	0.10286E 01	0.10067E 01	0.98591E 00	0.96663E 00	0.94960E 00	0.93592E 00	0.92221E 00	0.90258E 00	0.87624E 00
0.8333	0.10513E 01	0.10286E 01	0.10067E 01	0.98591E 00	0.96663E 00	0.94960E 00	0.93592E 00	0.92221E 00	0.90258E 00	0.87624E 00

J/I	42.9166	43.7499	44.5833	45.4166	46.2499	47.0832	47.9166	48.7499	49.5832	50.4165
0.1667	0.84481E 00	0.80930E 00	0.77035E 00	0.72845E 00	0.68395E 00	0.63715E 00	0.58833E 00	0.53768E 00	0.48539E 00	0.43159E 00
0.5000	0.84481E 00	0.80931E 00	0.77036E 00	0.72845E 00	0.68395E 00	0.63716E 00	0.58833E 00	0.53768E 00	0.48539E 00	0.43159E 00
0.8333	0.84481E 00	0.80930E 00	0.77035E 00	0.72845E 00	0.68395E 00	0.63715E 00	0.58833E 00	0.53768E 00	0.48539E 00	0.43159E 00

J/I	51.2499	52.0832	52.9165	53.7499	54.5832
0.1667	0.37631E 00	0.31947E 00	0.26070E 00	0.19907E 00	0.13261E 00
0.5000	0.37632E 00	0.31947E 00	0.26070E 00	0.19908E 00	0.13261E 00
0.8333	0.37631E 00	0.31947E 00	0.26070E 00	0.19907E 00	0.13261E 00

GROUP 3

J/I	0.3704	1.1111	1.8519	2.5926	3.3333	4.0741	4.8148	5.5555	6.2963	7.0370
0.1667	0.55925E-01	0.55904E-01	0.55862E-01	0.55800E-01	0.55716E-01	0.55612E-01	0.55486E-01	0.55341E-01	0.55175E-01	0.54988E-01
0.5000	0.55925E-01	0.55904E-01	0.55862E-01	0.55800E-01	0.55716E-01	0.55612E-01	0.55486E-01	0.55341E-01	0.55175E-01	0.54988E-01
0.8333	0.55925E-01	0.55904E-01	0.55862E-01	0.55800E-01	0.55716E-01	0.55612E-01	0.55486E-01	0.55341E-01	0.55175E-01	0.54988E-01

J/I	7.7778	8.5185	9.2592	10.0000	10.7407	11.4815	12.2222	12.9629	13.7037	14.4444
0.1667	0.54782E-01	0.54555E-01	0.54309E-01	0.54044E-01	0.53759E-01	0.53456E-01	0.53134E-01	0.52794E-01	0.52436E-01	0.52061E-01
0.5000	0.54782E-01	0.54555E-01	0.54309E-01	0.54044E-01	0.53759E-01	0.53456E-01	0.53134E-01	0.52794E-01	0.52436E-01	0.52061E-01
0.8333	0.54782E-01	0.54555E-01	0.54309E-01	0.54044E-01	0.53759E-01	0.53456E-01	0.53134E-01	0.52794E-01	0.52436E-01	0.52061E-01

J/I	15.1852	15.9259	16.6666	17.4074	18.1481	18.8888	19.6296	20.5000	21.5000	22.5000
0.1667	0.51669E-01	0.51261E-01	0.50837E-01	0.50399E-01	0.49946E-01	0.49482E-01	0.49008E-01	0.48431E-01	0.47734E-01	0.46995E-01
0.5000	0.51669E-01	0.51261E-01	0.50837E-01	0.50399E-01	0.49947E-01	0.49482E-01	0.49008E-01	0.48431E-01	0.47734E-01	0.46995E-01
0.8333	0.51669E-01	0.51261E-01	0.50837E-01	0.50399E-01	0.49946E-01	0.49482E-01	0.49008E-01	0.48431E-01	0.47734E-01	0.46995E-01

J/I	23.5000	24.5000	25.5000	26.5000	27.5000	28.5000	29.5000	30.5000	31.5000	32.5000
0.1667	0.46217E-01	0.45404E-01	0.44554E-01	0.43667E-01	0.42742E-01	0.41774E-01	0.40758E-01	0.39688E-01	0.38554E-01	0.37342E-01
0.5000	0.46217E-01	0.45404E-01	0.44554E-01	0.43668E-01	0.42742E-01	0.41774E-01	0.40758E-01	0.39688E-01	0.38554E-01	0.37342E-01
0.8333	0.46217E-01	0.45404E-01	0.44554E-01	0.43667E-01	0.42742E-01	0.41774E-01	0.40758E-01	0.39688E-01	0.38554E-01	0.37342E-01

J/I	33.5000	34.5000	35.5000	36.5000	37.5000	38.5000	39.5000	40.4166	41.2499	42.0833
0.1667	0.36036E-01	0.34611E-01	0.33034E-01	0.31254E-01	0.29192E-01	0.26708E-01	0.23536E-01	0.20015E-01	0.16966E-01	0.14559E-01
0.5000	0.36036E-01	0.34612E-01	0.33034E-01	0.31254E-01	0.29192E-01	0.26708E-01	0.23536E-01	0.20015E-01	0.16966E-01	0.14559E-01
0.8333	0.36036E-01	0.34611E-01	0.33034E-01	0.31254E-01	0.29192E-01	0.26708E-01	0.23536E-01	0.20015E-01	0.16966E-01	0.14559E-01

J/I	42.9166	43.7499	44.5833	45.4166	46.2499	47.0832	47.9166	48.7499	49.5832	50.4165
0.1667	0.12586E-01	0.10930E-01	0.95175E-02	0.82993E-02	0.72390E-02	0.63085E-02	0.54851E-02	0.47502E-02	0.40881E-02	0.34854E-02
0.5000	0.12586E-01	0.10930E-01	0.95176E-02	0.82993E-02	0.72391E-02	0.63085E-02	0.54851E-02	0.47502E-02	0.40881E-02	0.34854E-02
0.8333	0.12586E-01	0.10930E-01	0.95175E-02	0.82993E-02	0.72390E-02	0.63085E-02	0.54851E-02	0.47502E-02	0.40881E-02	0.34854E-02

J/I	51.2499	52.0832	52.9165	53.7499	54.5832
0.1667	0.29303E-02	0.24117E-02	0.19178E-02	0.14334E-02	0.93509E-03
0.5000	0.29304E-02	0.24118E-02	0.19178E-02	0.14334E-02	0.93510E-03
0.8333	0.29303E-02	0.24117E-02	0.19178E-02	0.14334E-02	0.93509E-03

GROUP 4

J/I	0.3704	1.1111	1.8519	2.5926	3.3333	4.0741	4.8148	5.5555	6.2963	7.0370
0.1667	0.35092E-03	0.35079E-03	0.35053E-03	0.35014E-03	0.34961E-03	0.34895E-03	0.34817E-03	0.34726E-03	0.34621E-03	0.34504E-03
0.5000	0.35092E-03	0.35079E-03	0.35053E-03	0.35014E-03	0.34961E-03	0.34895E-03	0.34817E-03	0.34726E-03	0.34621E-03	0.34504E-03
0.8333	0.35092E-03	0.35079E-03	0.35053E-03	0.35014E-03	0.34961E-03	0.34895E-03	0.34817E-03	0.34726E-03	0.34621E-03	0.34504E-03

J/I	7.7778	8.5185	9.2592	10.0000	10.7407	11.4815	12.2222	12.9629	13.7037	14.4444
0.1667	0.34375E-03	0.34233E-03	0.34079E-03	0.33912E-03	0.33734E-03	0.33544E-03	0.33343E-03	0.33130E-03	0.32907E-03	0.32673E-03
0.5000	0.34375E-03	0.34233E-03	0.34079E-03	0.33912E-03	0.33734E-03	0.33544E-03	0.33343E-03	0.33130E-03	0.32907E-03	0.32673E-03
0.8333	0.34375E-03	0.34233E-03	0.34079E-03	0.33912E-03	0.33734E-03	0.33544E-03	0.33343E-03	0.33130E-03	0.32907E-03	0.32673E-03

J/I	15.1852	15.9259	16.6666	17.4074	18.1481	18.8888	19.6296	20.5000	21.5000	22.5000
0.1667	0.32430E-03	0.32177E-03	0.31916E-03	0.31648E-03	0.31375E-03	0.31101E-03	0.30833E-03	0.30514E-03	0.30110E-03	0.29661E-03
0.5000	0.32430E-03	0.32177E-03	0.31916E-03	0.31648E-03	0.31375E-03	0.31101E-03	0.30833E-03	0.30514E-03	0.30110E-03	0.29661E-03
0.8333	0.32430E-03	0.32177E-03	0.31916E-03	0.31648E-03	0.31375E-03	0.31101E-03	0.30833E-03	0.30514E-03	0.30110E-03	0.29661E-03

J/I	23.5000	24.5000	25.5000	26.5000	27.5000	28.5000	29.5000	30.5000	31.5000	32.5000
-----	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

0.1667	0.29180E-03	0.28672E-03	0.28139E-03	0.27580E-03	0.26995E-03	0.26380E-03	0.25733E-03	0.25045E-03	0.24310E-03	0.23513E-03
0.5000	0.29180E-03	0.28672E-03	0.28139E-03	0.27580E-03	0.26995E-03	0.26381E-03	0.25733E-03	0.25045E-03	0.24310E-03	0.23513E-03
0.8333	0.29180E-03	0.28672E-03	0.28139E-03	0.27580E-03	0.26995E-03	0.26380E-03	0.25733E-03	0.25045E-03	0.24310E-03	0.23513E-03

J/I	33.5000	34.5000	35.5000	36.5000	37.5000	38.5000	39.5000	40.4166	41.2499	42.0833
-----	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

0.1667	0.22637E-03	0.21656E-03	0.20531E-03	0.19202E-03	0.17561E-03	0.15386E-03	0.12112E-03	0.85171E-04	0.59434E-04	0.42846E-04
0.5000	0.22637E-03	0.21656E-03	0.20531E-03	0.19202E-03	0.17561E-03	0.15386E-03	0.12113E-03	0.85171E-04	0.59434E-04	0.42847E-04
0.8333	0.22637E-03	0.21656E-03	0.20531E-03	0.19202E-03	0.17561E-03	0.15386E-03	0.12112E-03	0.85171E-04	0.59434E-04	0.42846E-04

J/I	42.9166	43.7499	44.5833	45.4166	46.2499	47.0832	47.9166	48.7499	49.5832	50.4165
-----	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

0.1667	0.31499E-04	0.23420E-04	0.17523E-04	0.13157E-04	0.98978E-05	0.74523E-05	0.56123E-05	0.42251E-05	0.31774E-05	0.23843E-05
0.5000	0.31499E-04	0.23420E-04	0.17524E-04	0.13158E-04	0.98979E-05	0.74524E-05	0.56123E-05	0.42251E-05	0.31775E-05	0.23843E-05
0.8333	0.31499E-04	0.23420E-04	0.17523E-04	0.13158E-04	0.98978E-05	0.74523E-05	0.56122E-05	0.42251E-05	0.31774E-05	0.23843E-05

J/I	51.2499	52.0832	52.9165	53.7499	54.5832
0.1667	0.17816E-05	0.13203E-05	0.96204E-06	0.67480E-06	0.42737E-06
0.5000	0.17816E-05	0.13203E-05	0.96206E-06	0.67483E-06	0.42735E-06
0.8333	0.17816E-05	0.13203E-05	0.96204E-06	0.67480E-06	0.42737E-06

VOLUME-INTEGRATED FISSION SOURCE

GROUP	FISSIONS
1	0.73270E 00
2	0.76914E-01
3	0.0
4	0.0
5	0.80962E 00

NUMBER OF NEG. COMPONENTS IN GROUP 1	0 POINTS
NUMBER OF NEG. COMPONENTS IN GROUP 2	0 POINTS
NUMBER OF NEG. COMPONENTS IN GROUP 3	0 POINTS
NUMBER OF NEG. COMPONENTS IN GROUP 4	0 POINTS

***** FINAL ITERATION RESULT *****

OUTER ITERATIONS	INNER ITERATIONS	EIGENVALUE	LAMBDA	REDUCT.FACTOR
11	1413	1.23517609	0.10000200E 01	0.12352934E 01
GROUP	INNER ITERATIONS	PER GROUP		
1	1	0.99999064E 00	0	0.0
2	1	0.10014448E 01	0	0.0
3	1	0.10003738E 01	0	0.0
4	1	0.10000763E 01	0	0.0

-----**

TIME USED BY THE CASE WITH ID. 101 5.0062 MINUTES
INNER USED 4.9782 MINUTES, = 99.4 PER CENT

-----**

-----**

LENGTH OF THE MASTER ARRAY 27000 WORDS
PERMANENT RESIDENT MEMBERS 10955 WORDS
FOR TEMPORARY RESIDENT ARRAYS AVAILABLE 16045 WORDS
CROSS SECTION INPUT USED -1 WORDS

(KENN=0/1)=EXTERNAL/CORE STORAGE VERSION OF TSCHEB, KENN = 0
EXTERNAL STORAGE VERSION 4160 WORDS *OPT. NOT USED *
CORE STORAGE VERSION 11044 WORDS *OPT. NOT USED *

-----**

TOTAL NUMBER OF INNER ITERATIONS IN GROUP 1 345
TOTAL NUMBER OF INNER ITERATIONS IN GROUP 2 526
TOTAL NUMBER OF INNER ITERATIONS IN GROUP 3 362
TOTAL NUMBER OF INNER ITERATIONS IN GROUP 4 180

-----**

1.BALANCE TYPE 105 WORDS * USED *
2.BALANCE TYPE 210 WORDS *NOT USED*
3.BALANCE TYPE 379 WORDS *NOT USED*

-----**

TIME USED BY THE WHOLE JOB 5.0065 MINUTES

FREE CORE STORAGE PLACE AT THE END 4 K BYTES

Anhang II: Programmliste

```
SUBROUTINE ADJG(C,XKI,VE,IGP,IHM,MT,IGM)
DIMENSION C(IHM,IGM,MT),XKI(IGP),VE(IGP)
COMMON NACOM,A(1000)
COMMON/FIX/NICOM,IA(300)
EQUIVALENCE(IHS,IA(49)),(IHT,IA(48))
J=1
K=IGM
5 IF(K.LE.J) GOTO 10
T1=XKI(K)
T2=VE(K)
XKI(K)=XKI(J)
VE(K)=VE(J)
XKI(J)=T1
VE(J)=T2
K=K-1
J=J+1
GOTO 5
10 I=1
15 J=IHM
18 IG=1
20 K=IGM
IF(J.LE.IHT) GOTO 30
IF(J.LE.IHS) GOTO 25
IG=J+1-IHS
GOTO 30
25 IC=J-IHS+IGM
IF(IC.LE.0) GOTO 35
K=IC
30 IF(K.LE.IG) GOTO 35
T=C(J,K,I)
C(J,K,I)=C(J,IG,I)
C(J,IG,I)=T
IG=IG+1
K=K-1
GOTO 30
35 J=J-1
IF(J.GE.1) GOTO 18
I=I+1
IF(I.LE.MT) GOTO 15
RETURN
END
```

```
SUBROUTINE ANFISS(RX,RY,DB,XARI,MAI,MAJ,MZ,C,DA,FG,IM,JM,IP,JP,
1 IXM,IYM,MT,IGM,IHM)
  DIMENSION RX(IP),RY(JP),DB(IM),XARI(JM),MAI(IM),MAJ(JM),MZ(IXM,IYM
1),C(IHM,IGM,MT),DA(IM,IYM),FG(IM,JM)
  COMMON NACOM,A(100)
  COMMON/FIX/NICOM,IA(200)
  EQUIVALENCE(IA(78),KWT),(IA(13),IBL),(IA(14),IBR),(IA(15),IBU),
1(IA(16),IBO)
  EQUIVALENCE(IA(44),ITP4),(IA(2),ITH)
  EQUIVALENCE(IA(49),IHS),(IA(48),IHT),(MBK,IA(81))
  IF(ITP4.GT.0.OR.MBK.EQ.4)RETURN
  R1=RX(IP)
  R2=RY(JP)
  IF(IBL.EQ.2) GOTO 101
  IF(IBL.EQ.0.OR.IBL.EQ.4)GO TO 102
  IF(IBR.EQ.0.OR.IBR.EQ.4)GO TO 103
101 CALL PUT(DB,IM,1.)
  GOTO 105
102 IF(IBR.EQ.0.OR.IBR.EQ.4)GO TO 106
  AL=0.5
  AH=-R1
  AA=0.0
  GOTO 107
103 AL=1.0
  AH=R1
  AA=R1
  GOTO 107
106 AL=1.0
  AH=0.0
  AA=R1
107 DO 108 I=1,IM
  V=P1*R1
108 DB(I)=(AL*RX(I)+AH)*(AA-RX(I))/V
105 IF(IBU.EQ.2) GOTO 201
  IF(IBU.EQ.0.OR.IBU.EQ.4)GO TO 202
  IF(IBO.EQ.0.OR.IBO.EQ.4)GO TO 203
201 CALL PUT(XARI,JM,1.)
  GOTO 205
202 IF(IBO.EQ.0.OR.IBO.EQ.4)GO TO 206
  BL=0.5
  BH=-R2
  BA=0.0
  GOTO 207
203 BL=1.0
  BH=R2
  BA=R2
  GOTO 207
206 BL=1.0
  BH=0.0
  BA=R2
207 DO 208 J=1,JM
  V=R2*R2
208 XARI(J)=(BL*RY(J)+BH)*(BA-RY(J))/V
205 CONTINUE
  IF(ITH.NE.0) GOTO 61
  DO 60 I=1,IXM
```



```
DO 60 J=1,IYM
DO 60 K=1,IGM
I3=MZ(I,J)
I3=IABS(I3)
60 DA(I,J)=DA(I,J)+C(IHT-1,K,I3)
GOTO 71
61 DO 62 J=1,IXM
DO 62 J=1,IYM
62 DA(I,J)=1.
71 CONTINUE
DO 70 I=1,IM
DO 70 J=1,JM
I1=MAI(I)
I2=MAJ(J)
70 FG(I,J)=DB(I)*XARI(J)*DA(I1,I2)
CALL PUT(XARI,JM,0.)
RETURN
END
```

```
SUBROUTINE BILAN1(T,ITYP,INDEX,IGP)
DIMENSION T(ITYP,INDEX,IGP)
CALL PUT(T,IGP*ITYP*INDEX,0.)
RETURN
ENTRY BILAN2(IG, S,XSV,XSK,XN,C,XKE,MAI,MAJ,MZ,F,FG,XNEI,XNEJ,
IXNEII,XNEJJ,XKI,L1,L2,L3,L4,L5,L6,L7,L8,L9,IM,JM,IGM,MT,IHM,
2IGP,IXM,IYM,MM,V,DA,DB,W,DE,DU,IP,Q,L10,L11,T,ITYP,INDEX,MWERT,
3MPQ)
DIMENSION Q(IGM,L10,L11),MPQ(IXM,IYM)
DIMENSION DA(IM,IYM,MM),DB(IM,MM),W(MM),DE(MM),DU(MM),V(IM,IYM)
DIMENSION S(IM,JM),XSV(IM,JM),XSK(IM,JM),XN(L7,L8,L9),XKE(IGP),
1C(IHM,IGM,MT),MAI(IM),MAJ(JM),MZ(IXM,IYM),F(IM,JM),FG(IM,JM),
2XNEI(JM,MM),XNEJ(IM,MM),XNEII(L1,L2,L3),XNEJJ(L4,L5,L6),XKI(IGP)
COMMON NACOM,A(200)
COMMON/FIX/NICOM,IA(300)
COMMON/ORG/IO(43),IE(20)
REAL*8 HAUPT(8)/'THE WHOL','E SYSTEM',' MATERIA','L NR. ','
1' ZONE ',' NR. ','PSEUDOZO','NE NR. '/,ADELE(2)
EQUIVALENCE(IHS,IA(49)),(IHT,IA(48))
EQUIVALENCE(IA(8),IEVT)
EQUIVALENCE(IO(23),JDELY),(IO(19),JRY),(IA(4),IGE),(JRX,IO(18))
COMMON/TSCH/ITSCH,RHO,ITP1,ITP2,INORM
EQUIVALENCE(IA(13),IBL),(IA(14),IBR),(IA(15),IBU),(IA(16),IBO)
EQUIVALENCE(KWT,IA(78))
EQUIVALENCE(IQUELL,IA(82)),(ITH,IA(2)),(IGE,IA(4)),(JFX,IO(30))
EQUIVALENCE(JB,IO(31)),(MBT,IA(81))
EQUIVALENCE(RLOG,A(16)),(DZ,A(3)),(EV,A(2))
EQUIVALENCE(RLOG,A(16)),(DZ,A(3)),(EV,A(2))
EQUIVALENCE(A(1),EPS)
EQUIVALENCE(JBKOUT,IA(90))
EQUIVALENCE(KPCH,IA(91))
EQUIVALENCE(JQ,IO(35))
```

EPPD=RLOG*EPS

```
1 CONTINUE
  MBK=IABS(MBT)
  DO 304 I=1,IM
  DO 304 J=1,JM
  I1=MAI(I)
  I2=MAJ(J)
  I3=MZ(I1,I2)
  I3=IABS(I3)
  MM2=1
  IF(MWERT.EQ.1)MM2=I3
  IF(MWERT.EQ.2)MM2=I1+IXM*(I2-1)
  IF(MWERT.GE.3)MM2=MPQ(I1,I2)

**EXTERNAL SOURCE**

  UU=0.
  IF(IQUELL)3301,3302,3303
3301 I4=I1
  I5=I2
  IMMAX=IXM
  GO TO 3305
3303 I4=I
  I5=J
  IMMAX=IM
3305 UU=Q(IG,I4,I5)
  IF(IABS(IQUELL).EQ.2)UU=A(JQ+IG-1)*A(JQ+IGM-1+I4+(I5-1)*IMMAX)
3302 T(1,MM2,IG)=T(1,MM2,IG)+UU*V(I,I2)
  U=XSV(I,J)*V(I,I2)
  IF(ITH.GT.0) GO TO 350

**FISSION PRODUCTION**
**FISSION SOURCE**

  T(6,MM2,IG)=T(6,MM2,IG)+C(IHT-1,IG,I3)*U

  GO TO 351
350 T(6,MM2,IG)=T(6,MM2,IG)+U*XKE(IG)
  DO 352 IGS=1,IGM
352 T(2,MM2,IGS)=T(2,MM2,IGS)+U*XKE(IG)*C(IHT-1,IGS,I3)
351 CONTINUE

**SELFSCATTERING**

  T(7,MM2,IG)=T(7,MM2,IG)+U*C(IHS,IG,I3)

**ABSORPTION**

  T(8,MM2,IG)=T(8,MM2,IG)+U*C(IHT-2,IG,I3)

**FISSION DENSITY**

  CCC=C(IHT=3,IG,I3)
  T(14,MM2,IG)=T(14,MM2,IG)+CCC*U

**CAPTURE DENSITY**

  T(13,MM2,IG)=T(13,MM2,IG)+(C(IHT-2,IG,I3)-CCC)*U
```

BUCKLING LEAKAGE

```
BKLI=0.
QQ=C(IHT-4,IG,I3)
IF(MBT.LT.0)QQ=0.3333333
IF(MBK-1)700,701,702
701 BKLI=QQ*(1.81375/(QQ*DZ+1.4208))**2
GO TO 700
702 IF(MBK-3)703,704,705
703 BKLI=A(JB+IG-1)/(3.*QQ)
GO TO 700
704 BKLI=A(JB-1+IG+(I1-1)*IGM+(I2-1)*IGM*IXM)/(3.*QQ)
GO TO 700
705 BKLI=A(JB-1+IG+(I-1)*IGM ) / (3.*QQ)
700 CONTINUE
T( 9,MM2,IG)=T( 9,MM2,IG)+U*BKLI
```

TOTAL FLUX

```
T( 5,MM2,IG)=T( 5,MM2,IG)+U
```

INSCATTER IN GROUP L

```
IGU=IG+1
IF(IGU.GT.IGM)GO TO 302
DO 303 L=IGU,IGM
303 T(3,MM2,L )=T(3,MM2,L )+U*C(IHS+L-IG,L,I3)
```

OUTSCATTER FROM GROUP IG

```
SSS=0.
DO 306 LIG=IGU,IGM
306 SSS=SSS+C(IHS+LIG-IG,LIG,I3)
T(10,MM2,IG)=T(10,MM2,IG)+U*SSS
302 CONTINUE
304 CONTINUE
```

RIGHT AND LEFT LEAKAGE

```
IF(MWERT.EQ.1.OR.MWERT.GE.3)GO TO 430
MM2=1
I=JRX+IM-1
R=A(I)*6.283185
IF(IGE.EQ.0) R=1.
DO 322 J=1,JM
I4=MAJ(J)
MM2=1
I=JDELY+I4-1
DO 320 M=1,MM
IF(DU(M).LT.0.)GO TO 321
IF(MWERT.EQ.2)MM2=I4*IXM
T(17,MM2,IG)=T(17,MM2,IG)+XNEI(J,M)*W(M)*DU(M)*R*A(I)
GO TO 320
321 IF(IGE.GT.0) GOTO 320
IF(MWERT.EQ.2)MM2=IXM*(I4-1)+1
T(16,MM2,IG)=T(16,MM2,IG)-XNEI(J,M)*W(M)*DU(M)*A(I)
```

320 CONTINUE
322 CONTINUE

UPPER AND LOWER LEAKAGE

```

DO 324 I=1,IM
I4=MAI(I)
MM2=1
DO 325 M=1,MM
IF(DE(M).GT.0.) GO TO 326
IF(MWERT.EQ.2)MM2=I4
T(18,MM2,IG)=T(18,MM2,IG)
1 -XNEJ(I,M)*W(M)*DB(I,M)*DE(M)/(ABS(DE(M))*2.0)
GO TO 325
326 IF(MWERT.EQ.2)MM2=IXM*(IYM-1)+I4
T(19,MM2,IG)=T(19,MM2,IG)
1 +XNEJ(I,M)*W(M)*DB(I,M)*DE(M)/(ABS(DE(M))*2.)
325 CONTINUE
324 CONTINUE

```

```

DO 400 MM2=1,INDEX
DO 400 IK=1,4
KKENN=IA(12+IK)
IF(KKENN=1)400,401,402
401 T(15+IK,MM2,IG)=0.
GO TO 400
402 IF(KKENN.EQ.3)GO TO 401
IF(KKENN.EQ.4)GO TO 400
IF((IK-2)*(IK-4).EQ.0)GO TO 400
SSS=T(16+IK,MM2,IG)-T(15+IK,MM2,IG)
T(16+IK,MM2,IG)=SSS
T(15+IK,MM2,IG)=-SSS
400 CONTINUE

```

TOTAL LEAKAGE

```

DO 1787 MM2=1,INDEX
DO 327 IK=16,19
327 T(11,MM2,IG)=T(11,MM2,IG)+T(1K,MM2,IG)
T(20,MM2,IG)=T(16,MM2,IG)+T(17,MM2,IG)
T(21,MM2,IG)=T(18,MM2,IG)+T(19,MM2,IG)
1787 CONTINUE
383 FORMAT(1H0,4X,'GR. EXTERN.SOURCE    FISSION SOURCE    INSCATTER
1 TOTAL PRODUCT.    TOTAL FLUX    FISSION PROD.    SELFSCATTER')
384 FORMAT(//3X,' GR. ABSORPTION    BUCKL.LEAKAGE    OUTSCATTER
1 TOTAL LEAKAGE    TOTAL LOSSES    CAPTURE DENSITY    FISSION DENSITY'
2)
430 RETURN

```

```
ENTRY BILAN3(T,ITYP,INDEX,IGP,MWERT,TT,IXYM,MPQ,VX,VY)
DIMENSION TT(3,IXYM,IGP),VX(IXM),VY(IYM)
IF(MWERT.NE.0)GO TO 1157
XNF=1.
IF(INORM.NE.2)GO TO 2000
E1=0.
E2=0.
DO 2001 I=1,IM
DO 2001 J=1,JM
I1=MAI(I)
I2=MAJ(J)
I3=MZ(I1,I2)
I3=IABS(I3)
DO 2001 KI=1,IGM
UW=V(I,I2)*XN(I,J,KI)
E1=E1+UW*C(IHT-3,KI,I3)
2001 E2=E2+UW*C(IHT-1,KI,I3)
XNUE=E2/E1
XNF=XNUE*3.125E+10
WRITE(KWT,2002)XNUE
2002 FORMAT(////////10X,'          MITTLERES NUE      = ',E15.7)
2000 CONTINUE
IF(IEVT.EQ.0)GO TO 1155
BB=XNF/A(JFX+IGM)
BB=XNF/(A(JFX+IGM)*EV)
IF(INORM.GE.1)BB=XNF/A(JFX+IGM)
DO 1147 I=1,IGP
1147 A(JFX+I-1)=A(JFX+I-1)*BB
DO 1148 I=1,IM
DO 1148 J=1,JM
F(I,J)=F(I,J)*BB
1148 FG(I,J)=FG(I,J)*BB
DO 1149 IGS=1,IGM
DO 1152 I=1,IM
IF((IBO+IBU).LE.0)GO TO 1200
DO 1150 M=1,MM
1150 XNEJJ(I,M,IGS)=XNEJJ(I,M,IGS)*BB
1200 CONTINUE
DO 1151 J=1,JM
1151 XN(I,J,IGS)=XN(I,J,IGS)*BB
1152 CONTINUE
DO 1154 J=1,JM
IF((IBR+IBL).LE.0)GO TO 1201
DO 1153 M=1,MM
1153 XNEII(J,M,IGS)=XNEII(J,M,IGS)*BB
1201 CONTINUE
1154 CONTINUE
1149 CONTINUE
1157 IF(IEVT.EQ.0)GO TO 1155
DO 1146 MM2=1,INDEX
DO 1146 I=1,ITYP
DO 1146 IGS=1,IGM
1146 T(I,MM2,IGS)=T(I,MM2,IGS)*BB
1155 CONTINUE
```



```
DO 340 MM2=1, INDEX
DO 340 I=1, IYYP
DO 340 IGS=1, IGM
340 T(I, MM2, IGP)=T(I, MM2, IGP)+T(I, MM2, IGS)
IF(MWERT.NE.2)GO TO 3010
```

VERTICAL AND HORIZONTAL BUCKLING

```
CALL PUT(VX, IXM, 0.)
CALL PUT(VY, IYM, 0.)
EQUIVALENCE(JRX, IO(18)), (JRY, IC(19))
DO 3990 I=1, IXM
3990 VX(I)=A(JRY+IYM)*(A(JRX+I)-A(JRX+I-1))
DO 3991 I=1, IYM
3991 VY(I)=A(JRX+IXM)*(A(JRY+I)-A(JRY+I-1))
CALL PUT(TT, 3*(IXM+IYM)*IGP, 0.)
DO 3003 IGS=1, IGP
DO 3000 I=1, IXM
DO 3000 J=1, IYM
VERTICAL
MM2=I+IXM*(J-1)
TT(1, I, IGS)=TT(1, I, IGS)+T(5, MM2, IGS)
HORIZONTAL
TT(1, IXM+J, IGS)=TT(1, IXM+J, IGS)+T(5, MM2, IGS)
3000 CONTINUE
DO 3001 I=1, IXM
TT(1, I, IGS)=TT(1, I, IGS)/VX(I)
TT(2, I, IGS)=T(18, I, IGS)+T(19, I+(IYM-1)*IXM, IGS)
3001 IF(TT(1, I, IGS).GT.1.E-8)TT(3, I, IGS)=TT(2, I, IGS)/TT(1, I, IGS)

DO 3002 J=1, IYM
TT(1, IXM+J, IGS)=TT(1, IXM+J, IGS)/VY(J)
TT(2, IXM+J, IGS)=T(17, IXM*J, IGS)+T(16, 1+(J-1)*IXM, IGS)
3002 IF(TT(1, IXM+J, IGS).GT.1.E-8)
1TT(3, IXM+J, IGS)=TT(2, IXM+J, IGS)/TT(1, IXM+J, IGS)
3003 CONTINUE
3010 CONTINUE
```

**FISSION DENSITY **

```
IF(IITH.GT.0) GO TO 360
DO 341 MM2=1, INDEX
DO 341 IGS=1, IGM
341 T(2, MM2, IGS)=T(6, MM2, IGP)*XKE(IGS)
DO 347 MM2=1, INDEX
DO 347 IGS=1, IGM
347 T(2, MM2, IGP)=T(2, MM2, IGP)+T(2, MM2, IGS)
360 CONTINUE
IF(MWERT.NE.0)GO TO 343
MM2=1
DO 342 IGS=1, IGP
T(4, MM2, IGS)=T(1, MM2, IGS)+T(2, MM2, IGS)+T(3, MM2, IGS)
T(12, MM2, IGS)=T(8, MM2, IGS)+T(9, MM2, IGS)+T(10, MM2, IGS)+T(11, MM2, IGS)
1)
IF(T(12, MM2, IGS).GT.RLOG)T(15, MM2, IGS)=T(4, MM2, IGS)/T(12, MM2, IGS)
342 CONTINUE
GO TO 344
343 DO 410 MM2=1, INDEX
DO 410 IGS=1, IGP
```

```
410 T(11,MM2,IGS)=T(1,MM2,IGS)+T(2,MM2,IGS)+T(3,MM2,IGS)-T(8,MM2,IGS)
1 -T(9,MM2,IGS)-T(10,MM2,IGS)
344 CONTINUE
WRITE(KWT,3034)
3034 FORMAT('1')
IF(MWERT.LT.3)GO TO 3033
KARO=MWERT-2
WRITE(KWT,3036)KARO
3036 FORMAT('0',30X,'PICTURE OF THE QUASIZONES, CASE NR.: ',I1//)
CALL MALE(MPQ,MAI,MAJ,IXM,IYM,IXM,IYM,0,Q,IGM)
3033 CONTINUE
WRITE(KWT,3034)
DO 421 MM2=1,INDEX
MWE=MWERT
IF(MWE.GT.3)MWE=3
ADELE(1)=HAUPT(2*MWE +1)
ADELE(2)=HAUPT(2*MWE +2)
WRITE(KWT,422)ADELE(1),ADELE(2),MM2
422 FORMAT(///31X,'NEUTRON BALANCE OF ',2A8,': ',1X,I3/ )
WRITE(KWT,383)
DO 380 I=1,IGP
380 WRITE(KWT,381)I,(T(J,MM2,I),J=1,7)
381 FORMAT( 3X,I4,7(2X,EI4.7))
WRITE(KWT,384)
DO 382 I=1,IGP
382 WRITE(KWT,381)I,(T(J,MM2,I),J=8,14)
IF(MWERT.GT.0)GO TO 421
WRITE(KWT,398)
398 FORMAT(/3X,' GR. NEUTRON BAL. LEFT LEAKAGE',
1 ' RIGHT LEAKAGE LOWER LEAKAGE UPPER LEAKAGE
2 HORIZ. LEAKAGE VERTIC. LEAKAGE')
DO 399 I=1,IGP
399 WRITE(KWT,381)I,(T(J,MM2,I),J=15,21)
421 CONTINUE
CALL WRIT
420 IF(MWERT.NE.2)GO TO 423
WRITE(KWT,3034)
WRITE(KWT,3035)
DO 200 IX=1,IXM
DO 200 JX=1,IYM
MM2 =IX+IXM*(JX-1)
IF((IX-1)*(IX-IXM)*(JX-1)*(JX-IYM).NE.0)GO TO 200
3035 FORMAT(13X,'LEFT-',12X,'RIGHT-',10X,'LOWER-',10X,'UPPER-')
WRITE(KWT,201)MM2
201 FORMAT(/3X,' GR.',23X,'LEAKAGE OF BOUNDARY ZONE ',I3)
DO 202 I=1,IGP
202 WRITE(KWT,381)I,(T(J,MM2,I),J=16,19)
200 CONTINUE

3005 FORMAT(///30X,'VERTICAL BUCKLING, VERTICAL ZONE ',I3//)
3008 FORMAT(///30X,'HORIZONTAL BUCKLING, HORIZONTAL ZONE ',I3//)
3009 FORMAT(///'0',3X,' GR. AVERAGE FLUX LEAKAGE BUCKLING'//)
3040 FORMAT(
1 20X,'COLUMN *BUCKLING* ONLY MEANINGFUL, IF THE VERTICAL BOUND
2ARIES ARE VACUUM'//)
3050 FORMAT(
1 20X,'COLUMN *BUCKLING* ONLY MEANINGFUL, IF THE HORIZONTAL BOUND
2ARIES ARE VACUUM'//)
```



```
WRITE(KWT,3034)
DO 3006 I=1,IXM
WRITE(KWT,3005)I
WRITE(KWT,3009)
WRITE(KWT,3040)
DO 3006 IGS=1,IGP
3006 WRITE(KWT,381)IGS,(TT(J,I,IGS),J=1,3)
EQUIVALENCE(JBKOUT,IA(90))
KPCH=7
IF(JBKOUT.EQ.1.OR.JBKOUT.EQ.3)WRITE(KPCH,3051)((TT(3,I,IGS),I=1,IX
IM),IGS=1,IGM)
3051 FORMAT(6E12.5)
IF(JBKOUT.EQ.2.OR.JBKOUT.EQ.3)WRITE(KPCH,3051)((TT(3,I+IXM,IGS),
1I=1,IYM),IGS=1,IGM)
```

```
WRITE(KWT,3034)
DO 3007 I=1,IYM
WRITE(KWT,3008)I
WRITE(KWT,3009)
WRITE(KWT,3050)
DO 3007 IGS=1,IGP
3007 WRITE(KWT,381)IGS,(TT(J,IXM+I,IGS),J=1,3)
CALL WRIT
423 CONTINUE
RETURN
END
```

```
SUBROUTINE DUMMY
COMMON NACOM,A(27000)
NACOM=27000
COMMON /FIX/NICOM,IA(1000)
NICOM=1000
RETURN
END
```

```
SUBROUTINE FISSN(FX,F,FG,V,XKE,IGP,IM,JM,C,MAI,MAJ,MZ,IXM,IYM,
1IHM,IGM,MT)
DIMENSION FX(IGP),F(IM,JM),FG(IM,JM),V(IM,IYM),XKE(IGP)
DIMENSION C(IHM,IGM,MT),MAI(IM),MAJ(JM),MZ(IXM,IYM)
COMMON NACOM,A(1000)
COMMON /FIX/NICOM,IA(300)
EQUIVALENCE(MBT,IA(81))
EQUIVALENCE(IEVT,IA(8))
COMMON /CONTR/IREBAL,IDIM(26),FUNK(26),DEV1(26),DEV2(26)
EQUIVALENCE(IA(60),NFN), (A(12),FTP),(A(16),RLOG),
1(A(13),XLA),(IA(78),KWT),(A(15),EPG),(A(1),EPS)
3,(IA(2),ITH)
EQUIVALENCE(A(2),EV),(ITP4,IA(44)) ,(ICC,IA(69))
EQUIVALENCE(IA(49),IHS),(IA(48),IHT)
E1=0.0
```

```
5 FTP=FX(IGP)
  IF(ICC.EQ.0.AND.ITP4.GT.0)GO TO 1
  IF(ICC.EQ.0.AND.MBT.EQ.4)GO TO 1
  IF(FTP.LE.RLOG.AND.NFN.NE.0.AND.IEVT.NE.0)GO TO 200
1 CONTINUE
  IF(ITH.NE.0) GOTO 55
10 DO 15 I=1,IM
  DO 15 J=1,JM
  I2=MAJ(J)
  F(I,J)=FG(I,J)*V(I,I2)
15 E1=E1+F(I,J)
  E2=0.0
  DO 20 I=1,IGM
  FX(I)=XKE(I)*E1
20 E2=E2+FX(I)
  FX(IGP)=E2
  IF(NFN)30,100,30
30 E1=FTP
  IF(IEVT.EQ.0)GO TO 36
  IF(ICC.EQ.0.AND.ITP4.GT.0)GO TO 32
  IF(ICC.EQ.0.AND.MBT.EQ.4)GO TO 32
  XLA=FX(IGP)/E1
  BB=1.0/XLA
  GO TO 31
32 BB=1./EV
  XLA=1.
31 DO 50 I=1,IGP
  XKE(I)=BB*XKE(I)
50 FX(I)=BB*FX(I)

  IF(ITH.EQ.0) GOTO 35
  DO 40 I=1,IM
  DO 40 J=1,JM
40 F(I,J)=F(I,J)*BB
35 CONTINUE

  RETURN
36 D1MAX=0.
  D2MAX=0.
  DO 101 IG=1,IGM
  IF(DEV1(IG).GT.D1MAX)D1MAX=DEV1(IG)
  IF(DEV2(IG).GT.D2MAX)D2MAX=DEV2(IG)
101 CONTINUE
  IF(E1.GT.RLOG)GO TO 37
  XLA=1.+AMAX1(D1MAX,D2MAX)
  GO TO 35
37 XLA=FX(IGP)/E1

  GO TO 35
100 NFN=1
  RETURN

55 DO 60 I=1,IM
  DO 60 J=1,JM
  I2=MAJ(J)
```

```
60 F(I,J)=FG(I,J)*V(I,I2)
   E2=0.0
   DO 65 K=1,IGM
   DO 70 I=1,IM
   DO 70 J=1,JM
   I1=MAI(I)
   I2=MAJ(J)
   I3=MZ(I1,I2)
   I3=IABS(I3)
70 E1=E1+F(I,J)*C(IHT-1,K,I3)
   FX(K)=E1
65 E2=E2+FX(K)
   FX(IGP)=E2
   IF(NFN) 30,100,30

200 WRITE(KWT,202)
202 FORMAT(///30X,' **** FISSION SOURCE ZERO *****')
   CALL SAFETY
   STOP
   END
```

```
   SUBROUTINE FOLGE(ISN,IM,JM,EPS,/MFO1/,
1 /MFO2/
,MM,EPG,IP,JP,MRX,MRY,
1 IXM,IYM,ITP3,ITP4,MISN,MIM,MJM,ZEPS,MMRX,MMRY)
   COMMON NACOM,A(1000)
   COMMON/FIX/NICOM,IA(300)
   EQUIVALENCE(EPGA,A(17))
   EQUIVALENCE(IA(79),KRD)
   EQUIVALENCE(KWT,IA(78))
   DIMENSION MRX(IXM),MRY(IYM),MISN(MFO2),MIM(MFO2),MJM(MFO2),
1 ZEPS(MFO2),MMRX(IXM),MMRY(IYM)
70 FORMAT(//20X,'*WARNING*, INTERPOLATING STEP MODIFIED')
   IF(MFO1.NE.0) GOTO 10
   MF=MFO2-1
   MISN(MFO2)=ISN
   MIM(MFO2)=IM
   MJM(MFO2)=JM
   ZEPS(MFO2)=EPS
   DO 4 I=1,MF
   READ(KRD,8)(MISN(I),MIM(I),MJM(I),ZEPS(I))
   IF(MISN(I).GT.ISN)MISN(I)=ISN
   IF(MIM(I).GT.IM)MIM(I)=IM
   IF(MJM(I).GT.JM)MJM(I)=JM
   IF(MISN(I).GT.ISN.OR.MIM(I).GT.IM.OR.MJM(I).GT.JM)WRITE(KWT,70)
4 CONTINUE
8 FORMAT(3I6,E12.5)
   DO 7 I=1,IXM
7 MMRX(I)=MRX(I)
   DO 9 J=1,IYM
9 MMRY(J)=MRY(J)

10 MFO1=MFO1+1
   ISN=MISN(MFO1)
   VI=MIM(MFO1)
   VJ=MJM(MFO1)
   EPS=ZEPS(MFO1)
```

```
MM=(ISN+4)*ISN/2
????????????????
EPG=0.1*EPS
IF(EP SA.NE.0.)EPG=EPSA
IF(MFO1.EQ.1) GOTO 15
I=ITP3
ITP3=ITP4
ITP4=I
```

```
15 CONTINUE
IM=0
JM=0
V2=MIM(MFO2)
V3=VI/V2
DO 20 I=1,IXM
V5=MMRX(I)
V4=V5*V3
MRX(I)=V4
IF(MRX(I).LE.0) MRX(I)=1
20 IM=IM+MRX(I)
V2=MJM(MFO2)
V3=VJ/V2
DO 30 J=1,IYM
V5=MMRY(J)
V4=V5*V3
MRY(J)=V4
IF(MRY(J).LE.0) MRY(J)=1
30 JM=JM+MRY(J)
IP=IM+1
JP=JM+1
COMMON/SPEICH/MBOUND(25)
COMMON/ORG/IO(40),IE(20)
EQUIVALENCE
1 (IA(59),L1),(IA(58),L2),(IA(57),L3),(IA(56),L4),
1(IA(55),L5),(IA(54),L6)
EQUIVALENCE(JXNEII,IO(36)),(JXNEJJ,IO(37)),(JXN,IO(38)),(JFIN,
LIA(46)),(IRIS,IA(42)),(IBL,IA(13)),(IBR,IA(14)),(IBU,IA(15)),
2 (IBO,IA(16))
EQUIVALENCE(IA(7),IGM)
JXNEJJ=JXNEII+JM*MM*IGM
IF(IBL.EQ.0.AND.IBR.EQ.0)GO TO 46
GO TO 47
46 JXNEJJ=JXNEII+1
L1=1
L2=1
L3=1
47 JXN=JXNEJJ+IM*MM*IGM
IF(IBU.EQ.0.AND.IBO.EQ.0)GO TO 50
GO TO 60
50 JXN=JXNEJJ+1
L4=1
L5=1
L6=1
60 JFIN=JXN+IM*JM*IGM
IRIS=NACCM-JFIN
MBOUND(2)=JFIN
MBOUND(16)=IRIS
RETURN
END
```

```

SUBROUTINE GECKO
INTEGER*2 P(80),ME(15)/'1 ','2 ','3 ','4 ','5 ','6 ','7 ','8 ','9
1','.' ','+ ','- ',' ' ','E ','0 ' /
DIMENSION REALS(6),INTEGS(12)
KRD=5
KWT=6
REWIND KRD
READ(KRD,51,END=700)(P(I),I=1,80)
JJ=2
WRITE(KWT,60)
60 FORMAT(///'0')

601 CONTINUE
51 FORMAT(80A1)
READ(KRD,51,END=700)(P(I),I=1,80)
DO 70 K=1,5
IF(P(K).NE.ME(9))GO TO 71
70 CONTINUE
REWIND KRD
RETURN
71 CONTINUE
DO 77 I=1,72
DO 78 J=1,15
IF(P(I).EQ.ME(J))GO TO 77
78 CONTINUE
WRITE(KWT,79)JJ
GO TO 600
77 CONTINUE
79 FORMAT(20X,
1 '***** CARD',I3, ' CONTAINS NONNUMERIC CHARACTER',
2 23X,'*****')
INDEX=1
DO 50 I=1,80
IF(P(I).EQ.ME(10))GO TO 52
GO TO 50
52 INDEX=2
IF(I.LE.18)GO TO 57
INDEX=3
DO 53 J=1,18
IF(P(J).NE.ME(13))GO TO 57
53 CONTINUE
INDEX=2
50 CONTINUE
57 CONTINUE

BACKSPACE KRD
IF(INDEX-2)200,300,400
200 READ(KRD,201,ERR=702)(INTEGS(K),K=1,12)
201 FORMAT(12I6)
DO 202 K=1,12
IF(IABS(INTEGS(K)).LE.117)GO TO 202
WRITE(KWT,203)K,JJ
203 FORMAT(20X, '***** ASCERTAIN THAT POSITION',
```

```
1 I3, ' IN CARD', I4, ' IS CORRECT WRITTEN  *****')
202 CONTINUE
   BACKSPACE KRD
   READ(KRD, 51)(P(I), I=1, 80)
   DO 204 K=1, 12
   IF(P(6*K).NE.ME(13))GO TO 204
   DO 205 J=1, 5
   IF(P(6*K-J).EQ.ME(13))GO TO 205
   WRITE(KWT, 203)K, JJ
   GO TO 204
205 CONTINUE
204 CONTINUE
   GO TO 600
300 READ(KRD, 301, ERR=702) (REALS(K), K=1, 6)
301 FORMAT(6E12.5)
   DO 302 K=1, 6
   IF(ABS(REALS(K)).LT.1.E10.OR .ABS(REALS(K)).GT.1.E-10.OR .
1 REALS(K).EQ.0.)GO TO 302
   WRITE(KWT, 203)K, JJ
302 CONTINUE
   GO TO 600
400 READ(KRD, 401, ERR=702) (INTEGS(K), K=1, 3), ( REALS(I), I=4, 6)
401 FORMAT(3I6, 4E12.5)
   DO 402 K=1, 3
   IF(IABS(INTEGS(K)).LE.70)GO TO 402
   WRITE(KWT, 203)K, JJ
402 CONTINUE
   K=4
   IF(ABS(REALS(K)).LE.1.)GO TO 405
   WRITE(KWT, 203)K, JJ
405 DO 406 K=5, 6
   IF(REALS(K).EQ.0.)GO TO 406
   WRITE(KWT, 203)K, JJ
406 CONTINUE
600 CONTINUE
   JJ=JJ+1
   GO TO 601
700 WRITE(KWT, 701)
701 FORMAT(/20X, '***** MISSING THE 99999, THE JOB WILL DIE ANYWHE
1RE AT ANY TIME *****')
   REWIND KRD
   RETURN
702 WRITE(KWT, 703)
703 FORMAT(20X, '***** ASCERTAIN THAT CARD', I3, ' CONTAINS NO ILLEGAL
1CHARACTER *****')
   GO TO 600
END
```

```

SUBROUTINE GEO(RAX,RAY,MRX,MRY,DELX,DELY,RX,RY,MAI,MAJ,DB,CA,DAL,
1V,IP,JP,IM,JM,IXM,IYM,MM,DU,W,MTETA,ZRAX,ZRAY,DE,ALPHA,MMM)
  DIMENSION ALPHA(IM,IYM,MMM)
  DIMENSION RX(IP),RY(JP),RAX(IP),RAY(JP),DB(IM,MM),DA(IM,IYM,MM),
1DELX(IXM),DELY(IYM),MRX(IXM),MRY(IYM),MAI(IM),MAJ(JM),V(IM,IYM),
2DAL(IM,IYM,MM),ZRAX(IXM),ZRAY(IYM),DU(MM),DE(MM),W(MM),
3MTETA(JM)
  COMMON NACOM,A(1000)
  COMMON/FIX/NICOM,IA(300)
  COMMON /ORG/IO(43),IE(20)
  EQUIVALENCE(IA(8),IEVT),(IA(69),ICC),(IA(78),KWT),(IA(4),IGE)
  EQUIVALENCE(A(16),RLOG),(IA(11),MT),(IA(7),IGM),(IA(75),IHM)
  EQUIVALENCE(JXARI,IO(29)),(JC,IO(16)),(JFG,IO(28)),(KZ,IE(6))
  EQUIVALENCE(NFF,IA(88))
  EQUIVALENCE(JXN,IO(38)),(JXNEJJ,IO(37)),(JXNEII,IO(36))
  IF(NFF.GT.0)CALL PUT(ALPHA,IM*IYM*MMM,0.)
  T2=ZRAX(1)
  T1=0.0
  M1=MRX(1)
  M2=2
  M3=M1+1
  VX=M1
  Z=(T2-T1)/VX
  DELX(1)=Z
  DO 1 I=M2,M3
  I1=I-1
  RX(I)=RX(I-1)+Z
1 MAI(I1)=1
  IF(IXM.LE.1) GOTO 5
  DO 4 J=2,IXM
  T1=T2
  T2=ZRAX(J)
  M1=MRX(J)
  M2=M3+1
  M3=M3+M1
  VX=M1
  Z=(T2-T1)/VX
  DELX(J)=Z
  DO 3 I=M2,M3
  I1=I-1
  RX(I)=RX(I1)+Z
3 MAI(I1)=J
4 CONTINUE
5 T2=ZRAY(1)
  T1=0.0
  M1=MRY(1)
  M2=2
  M3=M1+1
  VX=M1
  Z=(T2-T1)/VX
  DELY(1)=Z
  DO 8 I=M2,M3
  I1=I-1
  RY(I)=RY(I1)+Z
8 MAJ(I1)=1
  IF(IYM.LE.1) GOTO 11
```

```
DO 10 J=2,IYM
T1=T2
T2=ZRAY(J)
M1=MRY(J)
M2=M3+1
M3=M3+M1
VX=M1
Z=(T2-T1)/VX
DELY(J)=Z
DO 12 I=M2,M3
I1=I-1
RY(I)=RY(I1)+Z
12 MAJ(I1)=J
10 CONTINUE
11 CONTINUE
IF(ICC.NE.0) GOTO 135
CALL ANFISS(RX,RY,DB,A(JXARI),MAI,MAJ,IA(KZ),A(JC),DA,A(JFG),
1IM,JM,IP,JP,IXM,IYM,MT,IGM,IHM)
135 CONTINUE
IF(IGE.GT.0) GOTO 101
DO 16 M=1,MM
VU=ABS(DU(M))*2.0
VE=ABS(DE(M))*2.0
DO 18 I=1,IM
K=MAI(I)
DB(I,M)=VE*DELX(K)
DO 18 J=1,IYM
DA(I,J,M)=VU*DELY(J)
18 DAL(I,J,M)=0.0
16 CONTINUE
DO 15 K=1,IM
I=MAI(K)
DO 15 J=1,IYM
15 V(K,J)=DELX(I)*DELY(J)
GOTO 300
101 P1=3.141593
P2=6.283185
DO 104 M=1,MM
VU=ABS(DU(M))*P2
DO 104 I=1,IM
VE=(RX(I)+RX(I+1))*VU
DO 104 J=1,IYM
104 DA(I,J,M)=VE*DELY(J)
CALL PUT(DAL,IYM*IM*MM,0.)
CALL PUT(DB,IM*MM,0.)
DO 105 J=1,IYM
DO 109 I=1,IM
VE=P2*(RX(I+1)-RX(I))
VE=VE*DELY(J)
DO 109 M=2,MM
109 DB(I,M)=DB(I,M-1)-W(M)*DU(M)*VE
DO 119 M=2,MM
VE=W(M)
IF(VE.LE.RLOG) GOTO 119
DO 118 I=1,IM
IF(NFF.GT.0) ALPHA(I,J,M+1)=DB(I,M)
118 DAL(I,J,M)=(DB(I,M)+DB(I,M-1))/VE
119 CONTINUE
105 CONTINUE
```



```
IF(IGE.EQ.2)GOTO 310
DO 208 I=1,IM
K=MAI(I)
VE=P1*DELX(K)*(RX(I+1)+RX(I))
DO 209 M=1,MM
209 DB(I,M)=VE*ABS(DE(M))*2.0
DO 210 J=1,IYM
210 V(I,J)=DELY(J)*VE
208 CONTINUE
GOTO 300

310 DO 211 I=1,IM
K1=MAI(I)
VX=DELX(K1)
VU=VX*2.0
VE=P1*VX*(RX(I+1)+RX(I))
DO 212 J=1,IYM
212 V(I,J)= VE* DELY(J)
DO 214 M=1,MM
214 DB(I,M)=ABS(DE(M))*VU
211 CONTINUE

300 CONTINUE

DO 20 I=1,IM
20 RAX(I)=(RX(I)+RX(I+1))*0.5
DO 21 J=1,JM
21 RAY(J)=(RY(J)+RY(J+1))*0.5
200 CONTINUE
IF(ICE.NE.2)GOTO 41
T1=RY(JP)
T2=T1/2.0
DO 6 I=1,JM
VX=RAY(I)+T2
IF(VX.GT.T1) VX=VX-T1
DO 9 J=2,JP
IF(RY(J).LE.VX) GOTO 9
MTETA(I)=J-1
GOTO 6
9 CONTINUE
6 CONTINUE
41 CONTINUE
30 FORMAT(////1X,10E12.5)
40 FORMAT(////1X,10I12)
RETURN
END

SUBROUTINE IFUNC(XKE,XKI,IGP)
DIMENSION XKI(IGP),XKE(IGP)
IGM=IGP-1
AB=0.0
DO 1 I=1,IGM
AB=AB+XKI(I)
1 XKE(I)=XKI(I)
XKI(IGP)=AB
XKE(IGP)=AB
IF(XKE(IGP).LE.0.)CALL PRINT3(10)
RETURN
END
```

```

SUBROUTINE INNER(XSK,MRM,XNEI,XNEJ,XSV,C,MAI,MAJ,MZ,DU,DE,W,
1DAL,DB,S,V,DA,XARI,XARN,MRMY,IM,JM,MM,IGM,IP,JP,IHM,MT,IXM,IYM,
2MTETA,RX,DELY,ALPHA,MMM,SS,TW,TF,IS)
  DIMENSION RX(IP),DELY(IYM),ALPHA(IM,IYM,MMM)
  DIMENSION XSK(IM,JM),MRM(MM),XNEI(JM,MM),XNEJ(IM,MM),XSV(IM,JM),
1C(IHM,IGM,MT),MAI(IM),MAJ(JM),MZ(IXM,IYM),DU(MM),DE(MM),W(MM),
2S(IM,JM),XARI(JM),XARN(IM,JM),MRMY(MM),DAL(IM,IYM,MM),MTETA(JM),
3DB(IM,MM),DA(IM,IYM,MM),V(IM,IYM)
  DIMENSION SS(IM,JM,MM),TF(IM,JM,IS,IS),TW(IS,IS,MM)
  COMMON NACOM,A(250)
  COMMON/FIX/NICOM,IA(250)
  COMMON/SPEICH/MBOUND(25)
  COMMON /ORG/IO(43),IE(20)
  EQUIVALENCE(MBOUND(18),INNTS)
  EQUIVALENCE(JARN,IO(4)),(JSK,IO(1))
  EQUIVALENCE
  (IA(30),IIL),(IA(80),LC),
1(IA(76),IIC),(A(16),RLOG),(IA(78),KWT),(IA(72),IG),(A(15),EPG)
2,(IA(4),IGE),(IA(13),IBL),(IA(14),IBR),(IA(15),IBU),(IA(16),IBO)
  EQUIVALENCE(IA(49),IHS),(IA(48),IHT),(IA(81),MBT),(A(3),DZ)
  EQUIVALENCE(IO(31),JB),(KWT,IA(78))
  DIMENSION BIM(26),BAM(26)
  COMMON/TSCH/ITSC,H,RHO,ITP1,ITP2,INORM,KLAR(300)
  EQUIVALENCE(IA(61),ICVT)
  EQUIVALENCE(IA(88),NFF)
  EQUIVALENCE(ITP4,IA(44)),(XLA,A(13)),(XLAR,A(14)),(EPS,A(1))
  EQUIVALENCE(IA(107),ISCT)
  MBK=IABS(MBT)
  IMOD=0
1000 IIC=1
  IF(LC.NE.0)GO TO 3006
  DO 3004 I=1,IGM
3004 KLAR(I)=0
3006 CONTINUE
  COMMON /CONTR/IREBAL,IDIM(26),FUNK(26),DEV1(26),DEV2(26),NEGOV(26)
  COMMON/TEMP/GINN
  GNULL=0.
  GN=ZEIT(GNULL)
  IDIFF=LC
  ...
  .....
  BKLI=0.
  QQ=1.
  JBB=JB-1+IG
  IXG=IGM*IXM
5 LC=LC+1
  KLAR(IG)=KLAR(IG)+1
  CALL PUT(XSK,IM*JM,0.)
  MMM=MM/2
  IF(IGE.NE.2) GOTO 401
  DO 410 M=1,MMM
  IF(W(M).LE.RLOG) GOTO 410
  M1=MRM(M)
  M2=MRMY(M1)
  DO 411 J=1,JM
  VE=XNEI(J,M2)
  XNEI(J,M2)=XNEI(J,M)
```

```
411 XNEI(J,M)=VE
410 CONTINUE
    DO 414 M=1,MM
    DO 412 J=1,JM
    K=MTETA(J)
412 XARI(J)=XNEI(K,M)
    DO 413 J=1,JM
413 XNEI(J,M)=XARI(J)
414 CONTINUE
    GOTO 408
401 CONTINUE
    IF(IBL.EQ.1.OR.IBR.EQ.1) GOTO 405
    IF(IBL.EQ.2)GOTO 440
    CALL PUT(XNEI,JM*MM,0.)
    GOTO 440
405 DO 26 M=1,MM
    M1=MRM(M)
    VE=W(M)
    VU=DU(M)
    DO 25 J=1,JM
    IF(VE.LE.RLOG) GOTO 27
    IF(VU.GT.RLOG) GOTO 25
    E1=XNEI(J,M)
    XNEI(J,M)=XNEI(J,M1)
    XNEI(J,M1)=E1

    GOTO 25
27 XNEI(J,M)=XNEI(J,M1)
25 CONTINUE
26 CONTINUE
420 CONTINUE
    IF(IBL.NE.0) GOTO 408
    DO 422 M=1,MM
    VU=DU(M)
    DO 422 J=1,JM
422 IF(VU.GE.RLOG) XNEI(J,M)=0.0
408 IF(IBR.NE.0) GOTO 440
    DO 409 M=1,MM
    VU=DU(M)
    DO 409 J=1,JM
409 IF(VU.LE.RLOG) XNEI(J,M)=0.0
440 CONTINUE

    IF(IBU.EQ.1.OR.IBO.EQ.1) GOTO 460
    IF(IBU.EQ.2) GOTO 480
    CALL PUT(XNEJ,IM*MM,0.)
    GOTO 480
460 MMM=MM/2
    DO 35 M=1,MMM
    M1=MRMY(M)
    DO 35 I=1,IM
    E1=XNEJ(I,M)
    XNEJ(I,M)=XNEJ(I,M1)
35 XNEJ(I,M1)=E1
    IF(IBU.NE.0) GOTO 450
    DO 469 M=1,MM
    VU=DE(M)
    DO 469 I=1,IM
469 IF(VU.GE.RLOG) XNEJ(I,M)=0.0
```

```
450 IF( IBO.NE.0) GOTO 480
    DO 472 M=1,MM
    VE=DE(M)
    DO 472 I=1,IM
472 IF(VE.LE.RLOG) XNEJ(I,M)=0.0
480 CONTINUE
```

BERECHNUNG DER FLUESSE

```
M=1
300 VE=DE(M)
    VU=DU(M)
    IF(VE.GT.RLOG) GOTO 70
    IF(VU.GT.RLOG) GOTO 60
    IND=3
    GOTO 90
60 IND=4
    GOTO 90
70 IF(VU.GT.RLOG) GOTO 80
    IND=2
    GOTO 90
80 IND=1
90 CONTINUE
```

BERECHNE FLUESSE FUER ALLE I UND J

```
IZ=1
VE=W(M)
100 CONTINUE
    GOTO(120,121,121,120),IND
120 I=IZ
    GOTO 122
121 I=IM+1-IZ
122 C3=DB(I,M)
    XNJ=XNEJ(I,M)
    I1=MAI(I)
    DO 400 JJ=1,JM
    GOTO(135,136,137,138),IND
135 J=JJ
    IF(I.EQ.1) XARI(J)=XNEI(J,M)
    GOTO 141
136 J=JJ
    IF(I.EQ.IM)XARI(J)=XNEI(J,M)
    GOTO 141
137 J=JM+1-JJ
    IF(I.EQ.IM)XARI(J)=XNEI(J,M)
    GOTO 141
138 J=JM+1-JJ
    IF(I.EQ.1) XARI(J)=XNEI(J,M)
141 I2=MAJ(J)
    C2=DA(I,I2,M)
    I3=MZ(I1,I2)
    I3=IABS(I3)
    C1=DAL(I,I2,M)
    IF(ISCT.NE.0) GOTO 793
    VS=S(I,J)+C(IHS,IG,I3)*XSV(I,J)*V(I,I2)
    GOTO 794
793 VS=SS(I,J,M)+C(IHS,IG,I3)*XSV(I,J)*V(I,I2)
794 CONTINUE
```

```
IF(MBK-1)700,701,702
701 QQ=C(IHT-4,IG,I3)
BKLI=QQ*(1.81375/(QQ*DZ+1.4208))**2
GO TO 700
702 IF(MBK-3)703,704,705
703 BKLI=A(JBB)
GO TO 707
704 BKLI=A(JBB+(I1-1)*IGM+(I2-1)*IXG)
GO TO 707
705 BKLI=A(JBB+(I-1)*IGM)
707 IF(MBT.GT.0)QQ=C(IHT-4,IG,I3)*3.
BKLI=BKLI/QQ
700 VT=V(I,I2)*(C(IHT,IG,I3)+BKLI)
XNI=XARI(J)
XNM=XARN(I,J)

E2=(C1*XNM+C2*XNI+C3*XNJ+VS)/(C1+C2+C3+VT)
E100=E2

IF(E2.LE.RLOG) E2=0.0
XNJ=E2+E2-XNJ
XARI(J)=E2+E2-XNI
XARN(I,J)=E2+E2-XNM
XSK(I,J)=XSK(I,J)+W(M)*E2
IF(ICVT.LE.0.OR.IIC.GT.1)GO TO 1789
IF(E100.LE.0.)GO TO 1790
IF(XARI(J).LE.0.)GO TO 1790
IF(XARN(I,J).LE.0.)GO TO 1790
IF(XNJ.LE.0.)GO TO 1790
GO TO 1789
1790 IF(W(M).GT.RLOG)NEGQV(IG)=NEGQV(IG)+1
IF(NFF.EQ.0)GO TO 1789
WRITE(KWT,1791)IG,M,I,J,E100,XARI(J),XARN(I,J),XNJ
1791 FORMAT(' WARNING, NEG.F.F.U. ',4(I3,2X),4E17.8)
1789 CONTINUE
IF(XARI(J).LE.RLOG) XARI(J)=0.0
IF(XARN(I,J).LE.RLOG) XARN(I,J)=0.0
IF(XNJ.LE.RLOG) XNJ=0.0
1792 CONTINUE
IF(W(M).LE.RLOG) XARN(I,J)=E2
IF(ISCT.EQ.0) GOTO 400
IF(IMOD.EQ.0) GOTO 400
DO 783 K3=1,IS
DO 783 K4=1,IS
783 TF(I,J,K3,K4)=TF(I,J,K3,K4)+E2*TW(K3,K4,M)
400 CONTINUE
XNEJ(I,M)=XNJ
160 IZ=IZ+1
IF(IZ.LE.IM) GOTO 100
DO 170 J=1,JM
170 XNEI(J,M)=XARI(J)
180 CONTINUE
M=M+1
IF(M.LE.MM) GOTO 300

ENDE- BERECHNUNG DER FLUESSE

IIC=IIC+1
E31=0.
```

```
E21=0.
E2=0.0
E3=0.0
E4=0.
  E5=0.
DO 220 I=1,IM
DO 220 J=1,JM
E11=XSV(I,J)-XSK(I,J)
E1=ABS(E11)
IF(E1.GT.E5)E5=E1
I1=MAI(I)
I2=MAJ(J)
I3=MZ(I1,I2)
I3=IABS(I3)
VE=E1*C(IHS,IG,I3)
E2=E2+VE
E3=E3+E1*C(IHT,IG,I3)-VE
E21=E21+S(I,J)
E31=E31+C(IHS,IG,I3)*E11*V(I,I2)
E4=E4+ABS(XSK(I,J))
220 CONTINUE

.....
.....
  REBALANCING
FUN=E21/(E21+E31)
IF(FUN.LE.0.)FUN=1.
IF(ISCT.NE.0) FUN=1.

IF(IREBAL.GT.0) GO TO 2006
I2=JARN-1
DO 2000 I=JSK,I2
2000 A(I)=A(I)*FUN
E2=E2*FUN
E3=E3*FUN
E4=E4*FUN
2006 CONTINUE

.....
DO 222 I=1,IM
DO 222 J=1,JM
222 XSV(I,J)=XSK(I,J)
IF(E4.LE.RLOG)GO TO 2017
DEV1(IG)=E2/E4
DEV2(IG)=E3/E4
2017 CONTINUE
IF(ISCT.NE.0) GOTO 2018
IF(INNTS.EQ.0)GO TO 2018
CALL TEMP2(E3,IFIRST,IZWEI,IDREI)
CALL TEMP3(XSV,XNEJ,XNEI,A(IFIRST),A(IZWEI),A(IDREI))
2018 CONTINUE
IF(IIC.GT.IIL) GOTO 501
GIM=IM*JM
IF((GIM*E5).GT.(EPG*10.*E4))GO TO 5
501 CONTINUE

.....
IF(ISCT.EQ.0) GOTO 500
IF(IMOD.EQ.1) GOTO 500
DO 784 I=1,IM
```

```
DO 784 J=1,JM
DO 784 K3=1,IS
DO 784 K4=1,IS
784 TF(I,J,K3,K4)=0.0
IMOD=1
GOTO 5
```

```
500 CONTINUE
IDIM(IG)=LC-IDIFF
IF(IREBAL.EQ.0)FUNK(IG)=FUN
FUNK(IG)=FUN
GNUL=ZEIT(GN)
GINN=GINN+GNUL
RETURN
```

.....
END

.....'IPOLAN' PERFORMS THE ANGULAR INTERPOLATION.....

```
SUBROUTINE IPOLAN(ISN,KM, MMOLD,ARRAY,L1,MM,IGM,QL)
```

```
.....  
INTEGER FIND  
INTEGER*2 KM(1)  
DIMENSION ARRAY(L1,MM,IGM),QL(MM)
```

.....
TOTAL NUMBER OF ORDINATES ON THE HEMISPHERE

```
MMM=MM/2  
MOL=MMOLD/2
```

NUMBER OF LEVELS

```
LEVEL=ISN/2
```

```
DO 3 IG=1,IGM  
DO 3 I=1,L1  
DO 4 M=1,MMOLD  
4 QL(M)=ARRAY(I,M,IG)  
KR=0  
LIND=0
```

```
DO 1 N=1,LEVEL  
FIND=LIND+1  
LIND=FIND+ISN+2-2*N  
MIND=(FIND+LIND)/2
```

```
DO 2 K=FIND,MIND  
KR=KR+1  
INT=KM(KR)
```

```
ARRAY(I,K,IG)=QL(INT)
ARRAY(I,K+MMM,IG)=QL(INT+MOL)
IF(K.EQ.FIND)GO TO 2
IX=2*KM(KR+MIND-K)-INT+1
INDO=LIND+FIND-K
INDO=INDO+1
ARRAY(I,INDO,IG)=QL(IX)
ARRAY(I,INDO+MMM,IG)=QL(IX+MOL)
2 CONTINUE
1 CONTINUE
3 CONTINUE
RETURN
```

END

```
SUBROUTINE PUT(X,LEN,WERT)
DIMENSION X(1)
DO 1 I=1,LEN
1 X(I)=WERT
RETURN
END
```

```
SUBROUTINE LEVLOC(X,Y,Z,M,N,K,MOS,IBILD,IER,IG)
```

```
DIMENSION X(M),Y(N),Z(M,N),KENN(4),PX(4),PY(4),QX(20),QY(20),
1 S(4),T(4),U(4),V(4),W(4)
REAL*4 AZMAX(4)/' ',' ',' ' ..'/,AZMIN(4)/' ',' '
1',' ' ..'/,
1AIDENT(4)/' ',' ',' ' ..'/,INDEX(2)/' ',' ' ..'/,
1 KDENT(2)/' ',' ' ..'/
DIMENSION ILJA(10),IMASS(10)
```

INPUT CONTROL

```
3177 CONTINUE
IER=1
IF(M.LE.0.OR.M.GT.100)IER=IER*2
IF(N.LE.0.OR.N.GT.100)IER=IER*3
IF(K.LE.0.OR.K.GT.40)IER=IER*5
MONI=(MOS-1)*(MOS-2)*(MOS-3)
IF(MONI.NE.0)IER=IER*7
MG=M-1
DO 40 I=1,MG
IF(X(I+1).GE.X(I))GO TO 40
IER=IER*11
GO TO 41
40 CONTINUE
41 CONTINUE
MG=N-1
DO 42 I=1,MG
IF(Y(I+1).GE.Y(I))GO TO 42
IER=IER*13
GO TO 43
42 CONTINUE
```


43 CONTINUE
IF(IER.NE.1)RETURN

PARAMETERS

F=2.
INDZ=1
YDIFF=Y(N)-Y(1)
YAMA=Y(N)+YDIFF/4.
YIMI=Y(1)
XDIFF=X(M)-X(1)
XAMA=X(M)
XIMI=X(1)
ZMAX=-1.E+50
ZMIN=-ZMAX
DO 1 I=1,M
DO 1 J=1,N
IF(Z(I,J).GT.ZMAX)ZMAX=Z(I,J)
IF(Z(I,J).LT.ZMIN)ZMIN=Z(I,J)

1 CONTINUE

IDP=1
QY(1)=Y(1)
QX(1)=X(1)
QX(2)=X(M)
QY(2)=Y(N)
INDZ=1
QX(3)=X(1)+XDIFF*0.1
QY(3)=Y(N)+YDIFF*0.15
QX(4)=X(1)+0.65*XDIFF
QX(5)=QX(4)+XDIFF/10.
QX(6)=QX(4)
QX(7)=QX(5)
QY(4)=Y(N)+YDIFF*0.15
QY(6)=Y(N)+YDIFF/12.
QY(5)=QY(4)
QY(7)=QY(6)
QY(8)=QY(3)
QX(8)=X(1)+XDIFF*0.4
QX(9)=QX(3)
QY(9)=Y(N)+YDIFF/12.
QX(10)=QX(9)+XDIFF/10.
QY(10)=QY(9)
QX(11)=QX(3) -XDIFF/48.
QX(12)=QX(3)+XDIFF/12. -XDIFF/48.
QY(11)=Y(N)+YDIFF/40.
QY(12)=QY(11)
DO 1700 I=1,10
ILJA(I)=2

1700 IMASS(I)=1

CALL CONVX(ZMAX,AZMAX(1),'E12.4')
CALL CONVX(ZMIN,AZMIN(1),'E12.4')
IDENT=101
CALL CONVX(IDENT,AIDENT(1),'I10')
CALL CONVX(IG,INDEX(1),'I4')
CALL CONVX(K,KDENT(1),'I4')
CALL PLOTA(QX(1),QY(1),2,3,4,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
CALL PLOTA(QX(1),QY(1),2,1,4,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,

```
10.,10, IDP,0,0,10,QX(3),QY(3),ILJA(1),IMASS(1),19HIDENTIFICATION =.
2. ,8HFMAX =.,AZMAX(1),8HFMIN =.,AZMIN(1),AIDENT(1),7HGROUP.,
3INDEX(1),KDENT(1), 40HEQUIDISTANT LEVEL CURVES OF NEUT. FLUX..)
  IF(INDZ.NE.0)INDZ=0
  IF(IBILD.EQ.1)GO TO 32
  DO 30 I=1,M
  QX(1)=X(I)
  QX(2)=X(I)
  QY(1)=Y(1)
  QY(2)=Y(N)
  IF(I.EQ.M)GO TO 38
  IKU=M/15+1
  IF(M.GT.15.AND.I-(I/IKU)*IKU .NE.1)GO TO 30
38 CALL PLOTA(QX(1),QY(1),2,3,4,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
  1 0.,10, IDP,0,0,0)
  IF(INDZ.NE.0)INDZ=0
30 CONTINUE
  DO 31 I=1,N
  QX(1)=X(1)
  QX(2)=X(M)
  QY(1)=Y(I)
  QY(2)=Y(I)
  IF(I.EQ.N)GO TO 37
  IKU=N/15 +1
  IF(N.GT.15.AND.(I-(I/IKU)*IKU).NE.1)GO TO 31
37 CALL PLOTA(QX(1),QY(1),2,3,4,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
  1 0.,10, IDP,0,0,0)
31 CONTINUE
32 CONTINUE
  IF(IBILD.EQ.0)GO TO 33
  QX(1)=X(1)
  QX(2)=X(1)
  QX(3)=X(M)
  QX(4)=X(M)
  QY(1)=Y(1)
  QY(3)=Y(N)
  QY(2)=Y(N)
  QY(4)=Y(1)
  QX(5)=QX(1)
  QY(5)=QY(1)
  CALL PLOTA(QX(1),QY(1),5,3,2,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
  1 0.,10, IDP,0,0,0)
33 CONTINUE
  DIFF=ZMAX-ZMIN
  DO 2 IK=1,K
  L=3
  GK=IK
  IF(K.EQ.1)GO TO 11
  IF(MOS.EQ.3)DKOR=DIFF*(F /F **IK)
  IF(MOS.EQ.2)DKOR=DIFF*(1.-F /F **IK)
  IF(MOS.EQ.1)DKOR=DIFF*(GK-1.)/FLOAT(K-1)
  GO TO 12
11 IF(MOS.EQ.1)DKOR=DIFF/2.
  IF(MOS.EQ.2)DKOR= 0.
  IF(MOS.EQ.3)DKOR=DIFF
12 CONTINUE
  ZNIV=ZMAX-DKOR
```

EPS=1.E-5*DIFF
EPS=1.E-20*DIFF

MX=1

8 NX=1

7 CONTINUE

IF(MX.EQ.1.AND.NX.EQ.1)PEPX=X(1)
IF(MX.EQ.1.AND.NX.EQ.1)PEPY=Y(1)

Z1=Z(MX,NX)-ZNIV
Z2=Z(MX+1,NX)-ZNIV
Z3=Z(MX,NX+1)-ZNIV
Z4=Z(MX+1,NX+1)-ZNIV

S(1)=Z1

T(2)=Z1

T(1)=Z2

S(3)=Z2

S(2)=Z3

T(4)=Z3

T(3)=Z4

S(4)=Z4

X1=X(MX)

X2=X(MX+1)

Y1=Y(NX)

Y2=Y(NX+1)

U(1)=X1

V(4)=X1

W(2)=X1

V(1)=X2

U(4)=X2

W(3)=X2

V(2)=Y1

U(3)=Y1

W(1)=Y1

U(2)=Y2

V(3)=Y2

W(4)=Y2

DO 3 I=1,4

KENN(I)=0

DZ=T(I)-S(I)

IF(ABS(S(I))+ABS(T(I)).LE.EPS)GO TO 3

IF(ABS(DZ).LE.EPS)GO TO 3

IF(S(I)*T(I).GT.0.)GO TO 3.

KENN(I)=1

PX(I)=(U(I)*T(I)-V(I)*S(I))/DZ

PY(I)=W(I)

6 IF(I.EQ.1.OR.I.EQ.4)GO TO 4

E1=PX(I)

PX(I)=PY(I)

PY(I)=E1

4 CONTINUE

```
GO TO 3
5 KENN(I)=1
  PX(I)=V(I)
  PY(I)=W(I)
  GO TO 6
3 CONTINUE

DO 100 I=1,3
  IF(KENN(I).EQ.0)GO TO 100
  IST=I+1
  DO 101 J=IST,4
    IF(KENN(J).EQ.0)GO TO 101
    IF(ABS(PX(I)-PX(J))+ABS(PY(I)-PY(J)).GT.1.E-5*ABS(X(M)-X(1)))
1)GO TO 101
  KENN(I)=0
  GO TO 100
101 CONTINUE
100 CONTINUE

J=1
KE=0
DO 102 I=1,4
  KE=KE+KENN(I)
  IF(KENN(I).EQ.0)GO TO 102
  QX(J)=PX(I)
  QY(J)=PY(I)
  J=J+1

102 CONTINUE
  IF((IK-1)*(K-IK).GT.0)GO TO 203
  IF(KE.EQ.0)GO TO 203
  DO 202 I=1,KE
    IF(QX(I).LE.PEPX)GO TO 202
    PEPX=QX(I)
    PEPY=QY(I)
202 CONTINUE
203 CONTINUE
  IF(KE-2)201,103,104
201 IF(KE.EQ.0)GO TO 200
  QX(2)=QX(1)
  QY(2)=QY(1)
  CALL PLOTA(QX(1),QY(1),2,1,L,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
1 0.,10,IDP,0,0,0)
  GO TO 200

103 CALL PLOTA(QX(1),QY(1),2,2,L,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
1 0.,10,IDP,0,0,0)
  GO TO 200
104 IF(KE.EQ.4)GO TO 110
  CALL PLOTA(QX(1),QY(1),2,2,L,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
1 0.,10,IDP,0,0,0)
  CALL PLOTA(QX(2),QY(2),2,2,L,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,
1 0.,10,IDP,0,0,0)
  QX(2)=QX(3)
  QY(2)=QY(3)
```

```
CALL PLOTA(QX(1),QY(1),2,2,L,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,  
1 0.,10,IDP,0,0,0)  
GO TO 200
```

```
110 CALL PLOTA(QX(1),QY(1),4,2,L,1,1,1,INDZ,XAMA,XIMI,0.,YAMA,YIMI,  
1 0.,10,IDP,0,0,0)
```

```
200 CONTINUE
```

```
  NX=NX+1
```

```
  IF(NX.LE.N-1)GO TO 7
```

```
  MX=MX+1
```

```
  IF(MX.LE.M-1)GO TO 8
```

```
  IF((IK-1)*(K-1K).GT.0)GO TO 400
```

```
  QY(1)=PEPY
```

```
  QX(1)=PEPX
```

```
  QX(2)=X(1)+XDIF*0.61
```

```
  IF(IK.EQ.1)QY(2)=Y(N)+YDIF*0.15 +YDIF/60.
```

```
  IF(IK.EQ.K)QY(2)=Y(N)+YDIF/15. +YDIF/40.
```

```
  XD=0.013*XDIFF
```

```
  YD=0.018*XDIFF
```

```
  IF(IK.EQ.K)GO TO 401
```

```
  DO 402 I=1,2
```

```
  QX(5*I-2)=QX(I)+XD
```

```
  QX(5*I-1)=QX(I)-XD
```

```
  QX(5*I)=QX(5*I-1)
```

```
  QX(5*I+1)=QX(5*I-2)
```

```
  QY(5*I-2)=QY(I)+YD
```

```
  QY(5*I-1)=QY(I)-YD
```

```
  QY(5*I)=QY(5*I-2)
```

```
  QY(5*I+1)=QY(5*I-1)
```

```
  CALL PLOTA(QX(5*I-2),QY(5*I-2),2,2,3,3,1,1,INDZ,XAMA,XIMI,0.,  
*YAMA,YIMI,0.,10,IDP,0,0,0)
```

```
  CALL PLOTA(QX(5*I),QY(5*I),2,2,3,3,1,1,INDZ,XAMA,XIMI,0.,  
*YAMA,YIMI,0.,10,IDP,0,0,0)
```

```
402 CONTINUE
```

```
  GO TO 400
```

```
401 DO 403 I=1,2
```

```
  QX(5*I-2)=QX(I)
```

```
  QX(5*I-1)=QX(I)-XD
```

```
  QX(5*I)=QX(I)
```

```
  QX(5*I+1)=QX(I)+XD
```

```
  QX(5*I+2)=QX(I)
```

```
  QY(5*I-2)=QY(I)+YD
```

```
  QY(5*I-1)=QY(I)
```

```
  QY(5*I)=QY(I)-YD
```

```
  QY(5*I+1)=QY(I)
```

```
  QY(5*I+2)=QY(I)+YD
```

```
403 CALL PLOTA(QX(5*I-2),QY(5*I-2),5,2,3,3,1,1,INDZ,XAMA,XIMI,0.,  
*YAMA,YIMI,0.,10,IDP,0,0,0)
```

```
400 CONTINUE
```

```
  2 CONTINUE
```

```
  RETURN
```

```
  END
```

```
CALL PUT(U,IRIS,0.)
+++++
IGEL=IRIS
IF(IINP.GT.0.AND.IINP.LE.50)GO TO 1024
CALL PRINT3(15)
RETURN
1029 CALL PRINT3(14)
RETURN
1024 CONTINUE
REWIND IINP
READ(IINP,ERR=1029,END=1029)IELMA,SIG,NUMB
IF(SIG.EQ.SIGMN)GO TO 1021
CALL MESSAG
WRITE(KWT,1022)SIG
KONTRO=1
RETURN
1022 FORMAT('+',30X,'SIGMN-BLOCK NOT FOUND,THE SUPPLIED BLOCK HAS LABEL
1  ',A8)
1021 CONTINUE
CALL WQORG(IGEL,U,IU,IINP,&1000)
IF(IU(3).NE.MTP)GO TO 1001
IF(IU(4).NE.IGM) GO TO 1002
IF((IU(6)+IU(8)).LT.7) GO TO 1003
J3=JB
J5=JA
```

```
DO 1 J=1,IGM

KENN=1
ITZW=1

CHI

NTYP(1)=DTKTYP(1)
CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
A(J5)=U(LPCR+1)
J5=J5+1

1/V

IF(IEVT.NE.2) GO TO 40
NTYP(1)=DTKTYP(2)
CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1005)
A(J3)=1./U(LPCR+1)
J3=J3+1
40 CONTINUE

SMTOT

NTYP(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
```

```
KENN=1
DO 2 I=4,6
NTYP(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
```

```
KENN=1
NTYP(1)=DTKTYP(9)
CALL WQSKAL(KENN,NTYP,ITZW,J,LPCR,&1004,&1090)
DO 770 N=1,MTP
NA=IPLATZ(N)
770 C(IHT,J,NA)=U(LPCR+N)
SELF SCATTERING
```

```
NTYP(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
```

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

```
NTYP(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
```

+++++

```
IF(ISCT.LE.0) GO TO 1
IF(MTCONT.EQ.MTP)GO TO 1
```

```
DO 28 IK=1,ISCT
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(IMAT(N).GE.0)GO TO 16
NA=IPLATZ(N)+IK
UIK=U(LPCR+N+MTP*(M-NG1))
C(IHS-J+M,M,NA)=UIK
16 CONTINUE
17 CONTINUE
28 CONTINUE
```

```
DO 29 N=1,MTP
IF(IMAT(N).GE.0)GO TO 29
S=0.
```

```
DO 203 IK=1, ISCT
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
```

```
1 CONTINUE
```

*****RETURNS AND ERROR MESSAGES*****

```
WRITE(KWT,1010)
1010 FORMAT( /30X,'CROSS SECTIONS TAKEN FROM NUSYS')
CALL PUT(U,IGEL,0.)
RETURN
```

```
1000 WRITE(KWT,1011)
1011 FORMAT('1'//////////)
1 *****INCORRECT END IN SUBR. LIES2 *****
IUABS=IABS(IU(2))
WRITE(KWT,1012) IUABS
1012 FORMAT(///30X,' ARRAY NAMED U IS TOO SMALL , (' ,I4,' SAETZE )')
GO TO 1030
1001 WRITE(KWT,1011)
WRITE(KWT,1013) IU(3),MTP,MTCONT,MT,ISCT
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///)
GO TO 1030
1002 WRITE(KWT,1011)
WRITE(KWT,1014) IU(4),IGM
1014 FORMAT(///30X,'THE NUMBER OF GROUPS DONT AGREE ',2(2X,I5)//'1')
GO TO 1030
```

```
1003 WRITE(KWT,1011)
      WRITE(KWT,1015) IU(6),IU(8)
1015 FORMAT(///30X,'THERE ARE TOO LITTLE TYPES ON THE CROSS-S. FILE',
      13X,I2,' + ',I2,' TYPES'///'1')
      GO TO 1030
1004 WRITE(KWT,1011)
      WRITE(KWT,1016)
1016 FORMAT(///30X,'*** ERROR IN LIES2,INDICES DONT AGREE'///30X,'OR NUS
      IYS HAS NOT WRITTEN ON IINP'///'1')

1005 WRITE(KWT,1011)
      WRITE(KWT,1017) NTYP(1)
1017 FORMAT(///30X,'***ERROR IN LIES2,SPECIFIED TYPE,',A8,' NOT FOUND')
1030 KONTR0=1
      RETURN
1090 WRITE(KWT,1091)
1091 FORMAT(////////'0          **** ES MUSS ALS TOTALER WIRKUNGSQUERSCHNITT
      I DER NUSYS-TYP @STRTR@ BENUTZT WERDEN ****'/'1')
      GO TO 1030
1019 WRITE(KWT,1011)
      WRITE(KWT,1020)
1020 FORMAT(///30X,' ISCT IST ZU GROS ')
      KONTR0=1
      GO TO 1030
      END

COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
EQUIVALENCE(MFD1,IA(86))
EXTERNAL SECUR
EQUIVALENCE(KWT,IA(78))
CALL FSPIE('MIT ',SECUR)
CALL ERRSET(207,10,2,0,SECUR,210)
KWT=6
CALL PRAEP
CALL DUMMY
363 FORMAT('1',///,30X,'***** VERSION VOM 15.08.71 AUF DV0332 *****')
363 FORMAT('1',///,30X,'***** VERSION VOM 08.09.71 AUF DV0102 *****')
363 FORMAT('1',///,30X,'***** VERSION VOM 01.07.71 AUF G239AP *****')
WRITE(KWT,363)
KRD=5
CALL FREESP(I)
CALL WRIT
WRITE(KWT,6)I
CALL WRIT
6 FORMAT(//'0',29X,'FREE CORE STORAGE PLACE IN THE BEGINNING ',I3,
1' K BYTES')
7 FORMAT(//'0',29X,'FREE CORE STORAGE PLACE AT THE END ',I3,
1' K BYTES')
GNUL=0.
THAUPT=ZEIT(GNUL)
1 CONTINUE
MFO1=0
```

```
MFO2=0
10 CALL SNOW(MFO1,MFO2)
   TH=ZEIT(THAUPT)/60.
   WRITE(KWT,4)TH
4  FORMAT( //30X,'TIME USED BY THE WHOLE JOB      ',F11.4,' MINUTES')
   CALL WRIT
   IF(MFO2.LE.0) GOTO 5
   IF(MFO1.LT.MFO2) GOTO 10
5  CONTINUE
   READ(KRD,2)KKK
   IF(KKK-99999) 1,3,1
3  CALL FREESP(I)
   WRITE(KWT,7)I
   CALL WRIT
   STOP
2  FORMAT(I5)
   END
```

```
SUBROUTINE MALE(MZ,MAI,MAJ,IM,JM,IXM,IYM,MKENN,Q,IGM)
DIMENSION MZ(IXM,IYM),MAI(IM),MAJ(JM),Q(IGM,IXM,IYM)
INTEGER*2 IBUFF(4),BUFF(5,31),KENTA(18)
1/'V ','R ','P ','R ',' ','..',' ' I','--',' '+','-+','VI','RI','PI',
2'RI','IV','IR','IP','IR'/
LOGICAL*1 T(11)/'0','1','2','3','4','5','6','7','8','9',' '/,
1 S(18)/'(',' ','2','X',' ',' ','3','1','(','A','2',' ',' ','I','2','')',
2  ',' ','A','2','')'/,
3  U(10)/'(',' ','2','X',' ',' ','6','3','A','2','')'/
COMMON NACOM,A(1000)
COMMON/ORG/ID(40),IE(20)
COMMON/FIX/NICOM,IA(300)
EQUIVALENCE( IBL, IA(13)),( IBR, IA(14)),( IBU, IA(15)),( IBO, IA(16))
EQUIVALENCE( MAXCH, MMM11), ( M MMM, MINCH), ( KWT, IA(78))
```

```
IF(IM.EQ.1)RETURN
JANZ=(IM- 1)/31+1
```

ANZAHL DER ZEICHNUNGEN

```
DO 1 INZ=1,JANZ
MAXCH=1+MOD(IM-1,31)
IF(INZ.LT.JANZ)MAXCH=31
MVORN=(31-MAXCH)/2
MHINT=31-MAXCH-MVORN
MINCH=MAXCH-1
IFIRST=MAXCH/10
IZWEI=MAXCH-IFIRST*10
IF(IFIRST.EQ.0)IFIRST=10
S(6)=T(IFIRST+1)
S(7)=T(IZWEI+1)
KURUS=2*(31-MAXCH)-1
IF(KURUS.LE.1)KURUS=2
IFIRST=KURUS/10
IZWEI=KURUS-IFIRST*10
IF(IFIRST.EQ.0)IFIRST=10
S(2)=T(IFIRST+1)
S(3)=T(IZWEI+1)
```

```
U(2)=S(2)
U(3)=S(3)
  MULL=2*MAXCH+1
  IFIRST=MULL/10
  IZWEI=MULL-IFIRST*10
  IF(IFIRST.EQ.0)IFIRST=10
  U(6)=T(IFIRST+1)
  U(7)=T(IZWEI+1)
  DO 170 I=1,31
  DO 170 J=1,5
170 BUFF(J,I)=KENTA(9)
```

ERSTE ZEILE

```
  I BUFF(1)=KENTA(5)
  I BUFF(2)=KENTA(5)
  IF(INZ.NE.JANZ)IBUFF(2)=KENTA(6)
  IF(INZ.NE.1)IBUFF(1)=KENTA(6)
  DO 2 I=1,MAXCH
  IBOKA=MOD(IBO,4)
  BUFF(1,I)=KENTA(IBOKA+1)
2  BUFF(2,I)=KENTA(IBOKA+1)
  KOKI=1
  IF(MMMMM.EQ.0)GO TO 1000
  WRITE(KWT,U)IBUFF(1),(BUFF(1,I),BUFF(2,I),I=1,MMMMM),BUFF(1,MMM11)
1,IBUFF(2)
1010 CONTINUE
```

NORMALE DOPPELZEILE

```
  IBLKA=MOD(IBL,4)
  IBRKA=MOD(IBR,4)
  I BUFF(1)=KENTA(1BLKA+11)
  I BUFF(2)=KENTA(1BRKA+15)
  IF(INZ.NE.JANZ)IBUFF(2)=KENTA(6)
  IF(INZ.NE.1)IBUFF(1)=KENTA(6)
  I BUFF(3)=IBUFF(1)
  I BUFF(4)=IBUFF(2)
```

ZWEITE ZEILE

```
  DO 3 JJ=1,JM
  J=JM+1-JJ
  DO 44 I=1,MAXCH
  I1=I+31*(INZ-1)
  I1=MAI(I1)
  I2=MAJ(J)
  IMMAX=IXM
  IF(MKENN.GT.0)GO TO 4
  I1=I
  I2=J
  IMMAX=IM
4  BUFF(3,I)=MZ(I1,I2)
  IF(MKENN.NE.-1)GO TO 44
  QQL=0.
  DO 45 IG=1,IGM
  UUU=Q(IG,I1,I2)
  IF(IABS(IQUELL).EQ.2)UUU=A(JQ+IG-1)*A(JQ+IGM-1+I1+(I2-1)*IMMAX)
```

```
45 QQL=QQL+Q(IG,I1,I2)
   IF(QQL.GT.1.E-8)BUFF(3,I)=111
44 CONTINUE
   DO 6 I=1,MAXCH
   BUFF(1,I)=KENTA(5)
   BUFF(2,I)=KENTA(5)
   IF(J.EQ.JM)GO TO 6
   IF(BUFF(3,I).NE.BUFF(5,I))BUFF(1,I)=KENTA(8)
6 CONTINUE
   DO 25 I=1,MINCH
   BUFF(4,I)=KENTA(5)
   IF(BUFF(3,I).EQ.BUFF(3,I+1))GO TO 25
   BUFF(4,I)=KENTA(7)
   IF(BUFF(5,I).NE.BUFF(5,I+1))BUFF(2,I)=KENTA(7)
25 CONTINUE
   DO 7 I=1,MINCH
   IF(BUFF(1,I).EQ.BUFF(1,I+1).AND.BUFF(1,I).NE.KENTA(5))
1 BUFF(2,I)=BUFF(1,I)
   IF(BUFF(3,I).NE.BUFF(3,I+1).AND.(BUFF(3,I).NE.BUFF(5,I).OR.BUFF(3,
1 I+1).NE.BUFF(5,I+1)).OR.BUFF(5,I).NE.BUFF(5,I+1).AND.(BUFF(5,I).
2NE.BUFF(3,I).OR.BUFF(5,I+1).NE.BUFF(3,I+1)))
2GO TO 20
   GO TO 77
20 BUFF(2,I)=KENTA(9)
77 CONTINUE
   IF(BUFF(1,I).EQ.KENTA(8).AND.BUFF(2,I).EQ.KENTA(9))BUFF(2,I)=KENTA
1(10)
   IF(J.NE.JM)GO TO 7
   BUFF(1,MAXCH)=KENTA(8)
   BUFF(2,I)=KENTA(8)
   BUFF(1,I)=KENTA(8)
7 CONTINUE
   DO 8 I=1,MAXCH
8 BUFF(5,I)=BUFF(3,I)
   KOKI=2
   IF(MMMM.EQ.0)GO TO 1000
   WRITE(KWT,U)IBUFF(1),(BUFF(1,I),BUFF(2,I),I=1,MMMM),BUFF(1,MMM1)
1,IBUFF(2)
1020 CONTINUE
   IF(MMMM.EQ.0)GO TO 1001
   WRITE(KWT,S)IBUFF(3),(BUFF(3,I),BUFF(4,I),I=1,MMMM),BUFF(3,MMM1)
1,IBUFF(4)
1050 CONTINUE
3 CONTINUE

   VORLETZTE ZEILE

   DO 17 I=1,MAXCH
   DO 17 J=1,2
17 BUFF(J,I)=KENTA(8)
   KOKI=3
   IF(MMMM.EQ.0)GO TO 1000
   WRITE(KWT,U)IBUFF(1),(BUFF(1,I),BUFF(2,I),I=1,MMMM),BUFF(1,MMM1)
1,IBUFF(2)
1030 CONTINUE

   LETZTE ZEILE
```

```
IBUFF(1)=KENTA(5)
IBUFF(2)=KENTA(5)
IF(INZ.NE.JANZ)IBUFF(2)=KENTA(6)
IF(INZ.GT.1)IBUFF(1)=KENTA(6)
DO 18 I=1,MAXCH
IBUKA=MOD(IBU,4)
BUFF(1,I)=KENTA(IBUKA+1)
18 BUFF(2,I)=KENTA(IBUKA+1)
KOKI=4
IF(MMMM.EQ.0)GO TO 1000
WRITE(KWT,U)IBUFF(1),(BUFF(1,I),BUFF(2,I),I=1,MMMM),BUFF(1,MMM1)
1,IBUFF(2)
1040 CONTINUE
WRITE(KWT,13)
13 FORMAT( // '0' )
1 CONTINUE
RETURN
1000 WRITE(KWT,U)IBUFF(1),BUFF(1,1),IBUFF(2)
GO TO (1010,1020,1030,1040),KOKI
1001 WRITE(KWT,S)IBUFF(1),BUFF(3,1),IBUFF(2)
GO TO 1050
END
```

```
SUBROUTINE MESSAG
KWT=6
920 FORMAT( 20X,8('*'),80X,8('*'))
WRITE(KWT,920)
RETURN
END
```

```
SUBROUTINE NEWS
21 FORMAT('1'//10X,3('*** SNOW - NEWS ***')//32X,'(25.11.1971)'//)
20 FORMAT(1X,80('-'))
1 FORMAT(/1X, '1. IF EPSA IN K5 IS SPECIFIED NOT EQUAL 0., ITERAT'
*,
1'IONS ARE ONLY TERMINATED'/
2 1X, 'IF IN THE INNER ITERATIONS THE INNER ITERATION MAX'
*,
3'IMUM IIM IS ATTAINED '/
4 1X, 'IN NO GROUP. TO AVOID EXCESSIVE COMPUTING TIME, SP'
*,
5'ECIFY CAREFULLY ICM.'/ )
2 FORMAT(/ 1X, '2. IF IN K5 JBKOUT HAS ONE OF THE VALUES 1,2 OR 3,'
*,
* 'THE CODE PUNCHES OR'/
2 1X, 'WRITES ON A SPECIFIED UNIT, WITH NR. KPCH, T'
*,
4'HE VERTICAL OR HORIZONTAL'/
5 1X, 'BUCKLING, FOR ALL GROUPS AND ALL ZONES OF THE DIRE'
*,
6'CTION IN QUESTION.SOMEWHAT'/
7 1X, 'SIMPLIFIED, THE FORTRAN STATEMENTS RESPECTIVE ARE'/
9/' WRITE(KPCH,1)((B(I,IG),I=1,IXM),IG=1,IGM) ' /
9 ' 1 FORMAT(6E12.5) '
9 / /1X, 'WHERE IGM= NUMBER OF GROUPS, IXM= NUMBER OF ZONES I'
*,
```

```
9'N X=DIRECTION. IF THIS' /
9 1X, 'OPTION IS USED, KPCH MUST BE GT.0 AND A DD-STATEME'
*,
9'NT FOR KPCH MUST BE ' /
9 1X, 'PRESENT IN THE JCL WITH SYSOUT=B.' /)
3 FORMAT( / 1X,'3. IF NFF IN K5 IS EQUAL 1 , A MORE SOPHISTICATED
$ NEGATIVE FLUX FIX-UP'/' IS USED.' / )
4 FORMAT(/ 1X, '4. IF IN K5 KPLOT IS CHOSEN 7 OR -7, PICTURES OF '
*, 'LEVEL CURVES OF THE ' /
*, 1X, 'NEUTRON FLUX FOR SOME IN K19B(NEW INPUT CARD) SP'
*, 'ECIFIED GROUPS OR SUMS' /
* 1X, 'OF GROUPS ARE PRODUCED. IT IS POSSIBLE TO TAKE'//
520X,'//G.FT07FO01 DD SYSOUT=P'//
* 1X, 'TO GET PLOTS OR, IF THE USER HAS THE TCP-IDENTIFI'
*, 'CATION E.G. ABC, TO TAKE' /
920X,'//G.FT07FO01 DD DSN=TCP.ABC.USER(BILD),' /
920X,'// DISP=OLD,LABEL=(,,OUT),UNIT=2314,' /
920X,'// DCB=(RECFM=FB,BLKSIZE=1600,LRECL=80),' /
920X,'// VOL=SER=TCP002'//
* 1X, 'TO BE ABLE TO GET DISPLAYS AT THE TEXTRONIX-DISPLA'
*,
*'Y UNDER TCP, USING THE TCP-')
41 FORMAT(
* 1X, 'STATEMENT'//
*20X,'DE,BILD,P,01.1 A.S.O.'//
*1X, 'TO GET DISPLAYS. BILD IS A USER CHOSEN NAME. FOR I'
$, 'NPUT INFORMATION SEE 5.' /
* 1X, 'THE ROUTINE PLOTA IS USED. IF IB1.GT.0, THE LAST '
*, 'STEP FURNISHES PLOTS.' /)
5 FORMAT( / 1X, '5. NEW INPUT CARD K5, S19A,K19B AND K19C.'//
110X,'K5',10X,'ID1'/22X,'ID2'/22X,'ID3',11X,'AS BEFORE'/22X,'ID4',/
222X,'ID5' /
3 22X,'NFF',11X,'NEGATIVE FLUX FIXUP TRIGGER'/22X,'JBKOUT 1 VE
4RTIVAL BUCKLING OUTPUT ON KPCH'/22X,' 2 HORIZONTAL BUCK
5LING OUTPUT ON KPCH'/23X,' 3 TOTAL BUCKLING OUTPUT ON KP
6CH'/22X,'KPCH UNIT NR. OF BUCKLING OUTPUT,' /
$ 37X , 'WITHOUT MEANING, IF JBKOUT=0.' /
1 22X,'KPLOT UNIT NR. TO RECEIVE GRAPHIC OUTPUT,' /
* 36X,'MUST BE 7 OR -7.' /)
6 FORMAT(10X,'S19A',22X,'IF ABS(KPLOT).EQ.7 READ K19B AND K19C,' /
*36X,'THEN S20, OTHERWISE IMMEDIATELY S20' )
7 FORMAT( / 10X,'K19B', 8X,'N',13X, 'IF KPLOT=7, NUMBER OF GRO
*UPS TO BE PLOTTED,' / 36X,'THEN THESE NUMBERS' / 22X,
* '(IG(I),I=1,N)',1X, 'IF KPLOT=-7,ALL GROUPS WITH POSITIVE OR' /
* 36X, 'NEGATIVE SIGN,GROUPS IMMEDIATELY FOLLOWIN
*G' / 36X, 'EACH OTHER WITHOUT CHANGING SIGN, ARE' /
* 36X, 'ADDED AND THEN PLOTTED.' /)
8 FORMAT(10X,'K19C', 8X,'KNUB', 9X,'NUMBER OF LEVEL CURVES TO BE PL
*OTTED'/22X,'MODUS',5X,'1',3X,'CURVES HAVE EQUIDISTANT LEVEL' /
* 30X,' 2',3X,'LOGARITHMIC LEVEL,FACTOR 2,FROM ABOVE' /
* 30X,' 3',3X,'LOGARITHMIC LEVEL,FACTOR 2,FROM BELOW' /
* 22X,'IBILD',5X,'0',3X,'MESH LINES ARE PLOTTED' /
* 30X,' 1',3X,'MESH LINES ARE NOT PLOTTED' /)
9 FORMAT(/1X, '6. IT IS ALLOWED TO PROVIDE AN EXTERNAL SOURCE IN
* SEPARATED FORM =' /
*1X, ' (SOURCE SPECTRUM) * (GEOMETRIC INTENSITY)' /
*/1X, 'IF IQUELL=2 OR =-2, READ FIRST IN K16 OR K17 THE S
*PECTRUM, THEN (NEW CARD)'
```

* /1X, 'IF IQUELL=2, THE INTENSITY FOR ALL POINTS, ' /1X
* , 'IF IQUELL=-2, THE INTENSITY FOR ALL ZONES.')

```
WRITE(6,21)
WRITE(6,20)
  WRITE(6,1)
WRITE(6,20)
  WRITE(6,2)
WRITE(6,20)
  WRITE(6,3)
WRITE(6,20)
  WRITE(6,4)
WRITE(6,41)
WRITE(6,20)
WRITE(6,5)
WRITE(6,6)
WRITE(6,7)
WRITE(6,8)
WRITE(6,20)
WRITE(6,9)
WRITE(6,20)
RETURN
END
```

```
SUBROUTINE ORDV(MF1,MF2,IGM)
COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
DIMENSION MF1(IGM),MF2(IGM)
EQUIVALENCE(IA(89),KONTR0)
READ(KRD,1) (N,(MF1(I),I=1,N))
IF(N.LE.0.OR.N.GT.IGM) CALL PRINT3(12)
DO 2 J=1,N
  I=MF1(J)
2 MF2(I)=1
1 FORMAT(12I6)
RETURN
END
```

```
SUBROUTINE OUTER(S,XSV,XSK,XN,C,XKE,MAI,MAJ,MZ,F,FG,XNEI,XNEJ,
1XNEII,XNEJJ,XKI,L1,L2,L3,L4,L5,L6,L7,L8,L9,IM,JM,IGM,MT,IHM,
2 IGP,IXM,IYM,MM,V,DA,DB,W,DE,DU,IP,Q,L10,L11,TF,SS,IS)
EQUIVALENCE(IA(42),IRIS),(JFIN,IA(46))
DIMENSION Q(IGM,L10,L11)
DIMENSION DA(IM,IYM,MM),DB(IM,MM),W(MM),DE(MM),DU(MM),V(IM,IYM)
DIMENSION S(IM,JM),XSV(IM,JM),XSK(IM,JM),XN(L7,L8,L9),XKE(IGP),
1C(IHM,IGM,MT),MAI(IM),MAJ(JM),MZ(IXM,IYM),F(IM,JM),FG(IM,JM),
2XNEI(JM,MM),XNEJ(IM,MM),XNEII(L1,L2,L3),XNEJJ(L4,L5,L6),XKI(IGP)
DIMENSION TF(IM,JM,IS,IS),SS(IM,JM,MM)
COMMON NACOM,A(200)
COMMON/FIX/NICOM,IA(300)
COMMON /ORG/IO(43),IE(20)
EQUIVALENCE(IHS,IA(49)),(IHT,IA(48))
EQUIVALENCE(IO(23),JDELY),(IO(19),JRY),(IA(4),IGE),(JRX,IO(18))
COMMON/TSCH/ITSC,RHO,ITP1,ITP2,INORM
EQUIVALENCE(IA(13),IBL),(IA(14),IBR),(IA(15),IBU),(IA(16),IBO),
1(IA(69),ICC),(KTR,IA(12)),(ITP4,IA(44)),
1(IA(29),IIM),(IA(30),IIL),(IA(78),KWT), (IA(73),JP),(
2IA(72),IG),(IA(68),KTP1),(IA(67),KTP2),(IA(66),KTP3),
3(IA(64),KTPP3),(IA(61),ICVT),(A(1),EPS),(A(2),EV),(A(13),XLA),
4(A(14),XLAR),(A(16),RLOG),(IA(65),KTPP2)
```



```
EQUIVALENCE( IA(50),KTPP1), (IA(2), ITH), (A(15), EPG)  
EQUIVALENCE(KMTETA, IE(10)), (JARN, IO(4)), (JDA, IO(7)), (JW, IO(8)),  
1(JAL, IO(9)), (JDU, IO(10)), (JDE, IO(11)), (JDB, IO(13)), (JV, IO(14)),  
2(JF, IO(27)), (JXARI, IO(29)), (JFX, IO(30)), (KMRY, IE(7)), (KRM, IE(3))  
EQUIVALENCE (IO(43), JTW), (IA(82), IQUELL), (IA(107), ISCT),  
1 (IA(108), KTPUN1), (IA(109), KTPUN2), (IO(40), JT)  
EQUIVALENCE(EPSA, A(17))  
COMMON/CONTR/IREBAL, IDIM(26), FUNK(26)  
EQUIVALENCE(IQUELL, IA(82)), (JQ, IO(35))  
CALL PUT(FG, IM*JM, 0.)  
IG=1  
100 CALL PUT(S, IM*JM, 0.)
```

```
GO TO 9060  
ENTRY OUT1(IXM, IYM, MT, IGP)
```

```
INTEGER*4 ITYP(10)/21,14,21,7*14/,KB(11),INDEX(10)  
LOGICAL*1 KTUT(10)  
EQUIVALENCE(KAUSW, IA(87)), (KRD, IA(79)), (MFO1, IA(86))  
DIMENSION IB(800)  
I11=KAUSW/100  
I12=(KAUSW-100*I11)/10  
KFACH=KAUSW-100*I11-10*I12  
IF(KFACH.GT.7)GO TO 9020  
9047 IF(KFACH*IXM*IYM.LE.800)GO TO 9021  
KFACH=KFACH-1  
GO TO 9047  
9021 DO 9011 I=1,10  
9011 KTUT(I)=.FALSE.  
KTUT(1)=.TRUE.  
IF(I11.GT.0)KTUT(2)=.TRUE.  
IF(I12.GT.0)KTUT(3)=.TRUE.  
DO 9012 I=4,10  
IF((I-4).GE.KFACH)GO TO 9013  
9012 KTUT(I)=.TRUE.  
9013 IF(KFACH.EQ.0)GO TO 9014  
IF(MFO1.GT.1)GO TO 9014  
I1=1  
I2=IXM*IYM  
DO 9002 J=1,KFACH  
READ(KRD,9003)INDEX(3+J), (IB(I), I=I1, I2)  
9003 FORMAT(12I6)  
I1=I2+1  
9002 I2=I2+IXM*IYM  
9014 KEND=3+KFACH  
EQUIVALENCE( IB1, IA(5))  
IF(MFO1.EQ.(IABS(IB1)+1).AND.IB1.NE.0)GO TO 9015  
IF(IB1.EQ.0)GO TO 9015  
DO 9016 I=2,10  
9016 KTUT(I)=.FALSE.  
9015 INDEX(1)=1  
INDEX(2)=MT  
INDEX(3)=IXM*IYM  
KB(1)=JFIN  
9004 DO 9000 KR=1,10  
KB(KR+1)=KB(KR)  
IF(.NOT.KTUT(KR))GO TO 9000  
KB(KR+1)=KB(KR)+IGP*ITYP(KR)*INDEX(KR)
```

```
IF(KR.EQ.3)KB(KR+1)=KB(KR+1)+3*(IXM+IYM)*IGP+IXM+IYM
9000 CONTINUE
IF(KB(11).LE.NACOM)GO TO 9001
WRITE(KWT,9022)KAUSW
KTUT(KEND)=.FALSE.
KEND=KEND-1
IF(KEND.NE.0)GO TO 9004
WRITE(KWT,9005)
9005 FORMAT(///' NO EVALUATIONS POSSIBLE')
9001 RETURN
9020 WRITE(KWT,9022)KAUSW
9022 FORMAT('0 TO MANY EVALUATIONS ASKED, KAUSW = ',I6)
KFACH=3
GO TO 9047
9060 CONTINUE
IF(ICVT.NE.2.OR.IG.NE.1)GO TO 1
DO 9006 KR=1,KEND
IF(.NOT.KTUT(KR))GO TO 9006
JFIRST=KB(KR)
CALL BILANI(A(JFIRST),ITYP(KR),INDEX(KR),IGP)
9006 CONTINUE
1 CONTINUE
GO TO 9070
ENTRY OUT2
REAL*8 U1/' USED ',U2/' NOT USED ',U3
DO 1700 I=1,3
KUGEL=INDEX(I)*ITYP(I)*IGP
IF(I.EQ.3)KUGEL=KUGEL+(IXM+IYM)*IGP*3+IXM+IYM
U3=U1
IF(.NOT.KTUT(I))U3=U2
1700 WRITE(KWT,1701)I,KUGEL,U3
CALL WRIT
RETURN
1701 FORMAT(31X,I1,'.BALANCE TYPE ',I6,' WORDS *',A8,'*')
9070 CONTINUE
```

```
-----
IF(ISCT.EQ.0) GOTO 8003
IS=ISCT+1
CALL OUTUN(SS,TF,MAI,MAJ,MZ,C,A(JT),IM,JM,MM,IS,IXM,
1 IYM,IHM,IGM,MT,V)
DO 922 I=1,IM
DO 922 J=1,JM
922 XSV(I,J)=TF(I,J,1,1)
GOTO 120
8003 CONTINUE
IF(ITP4.GT.0)GO TO 4481
IF(ICC.EQ.0)GO TO 80
4481 CONTINUE
IF(KTP1.EQ.0) GOTO 50
12 DO 40 K=1,IG
I1=K-IG+1
IF(I1)16,18,20
16 READ(KTP1)((XSK(I,J),I=1,IM),J=1,JM)
GO TO 22
18 WRITE(KTP1)((XSV(I,J),I=1,IM),J=1,JM)
DO 24 I=1,IM
DO 24 J=1,JM
24 XSK(I,J)=XSV(I,J)
22 DO 26 I=1,IM
```

```
DO 26 J=1,JM
  I1=MAI(I)
  I2=MAJ(J)
  I3=MZ(I1,I2)
  I4=IHS+IG-K
26 S(I,J)=S(I,J)+C(I4,IG,I3)*XSK(I,J)
  GO TO 40
20 READ(KTP1)((XSV(I,J),I=1,IM),J=1,JM)
40 CONTINUE
  GOTO 120
50 DO 55 K=1,IG
  I7=IG-1
  IF(K.EQ.I7)GO TO 60
  IF(K.EQ.IG) GOTO 70
85 CONTINUE
  DO 75 I=1,IM
  DO 75 J=1,JM
  I1=MAI(I)
  I2=MAJ(J)
  I3=MZ(I1,I2)
  I4=IHS+IG-K
75 S(I,J)=S(I,J)+C(I4,IG,I3)*XN(I,J,K)
  GO TO 55
60 DO 65 I=1,IM
  DO 65 J=1,JM
65 XN(I,J,K)=XSV(I,J)
  GO TO 85
70 DO 90 I=1,IM
  DO 90 J=1,JM
90 XSV(I,J)=XN(I,J,K)
55 CONTINUE
  GOTO 120
80 DO 105 I=1,IM
  DO 105 J=1,JM
  I1=MAI(I)
  I2=MAJ(J)
  IF(ITH.NE.0)GO TO 81
  URT=XKE(IG)
  GO TO 82
81 I3=MZ(I1,I2)
  URT=C(IHT-1,IG,I3)
82 U=0.
  IF(IQUELL)104,105,103
104 I4=I1
  I5=I2
  IMMAX=IXM
  GO TO 101
103 I4=I
  I5=J
  IMMAX=IM
101 U=Q(IG,I4,I5)*V(I,I2)
  IF(IABS(IQUELL).EQ.2)U=A(JQ+IG-1)*A(JQ+IGM-1+I4+(I5-1)*IMMAX)
  * *V(I,I2)
105 S(I,J)=F(I,J)*URT+U

  IF(IG.LE.1) GOTO 150
  IF(KTP1.EQ.0) GOTO 110
  WRITE(KTP1)((XSV(I,J),I=1,IM),J=1,JM)
  IF(IG.EQ.IGM) REWIND KTP1
```

```
GOTO 150
110 DO 115 I=1,IM
    DO 115 J=1,JM
115 XN(I,J,IG-1)=XSV(I,J)
    GOTO 150
120 CONTINUE
    DO 130 I=1,IM
    DO 130 J=1,JM
    I1=MAI(I)
    I2=MAJ(J)
    U=0.
    IF(ITH.NE.0)GO TO 125
    URT=XKE(IG)
    GO TO 126
125 I3=MZ(I1,I2)
    I3=IABS(I3)
    URT=C(IHT-1,IG,I3)
126 IF(IQUELL)121,130,123
121 I5=I2
    I4=I1
    IMMAX=IXM
    GO TO 124
123 I5=J
    I4=I
    IMMAX=IM
124 U=Q(IG,I4,I5)
    IF(IABS(IQUELL).EQ.2)U=A(JQ+IG-1)*A(JQ+IGM-1+I4+(I5-1)*IMMAX)
130 S(I,J)=(S(I,J)+U)*V(I,I2)+F(I,J)*URT
    IF(ISCT.EQ.0) GOTO 8043
    R=0.
    DO 8100 I=1,IM
    DO 8100 J=1,JM
    RS=S(I,J)
    DO 8100 M=1,MM
    SS(I,J,M)=SS(I,J,M)+RS
8100 R=R+SS(I,J,M)
    GOTO 8041
8043 CONTINUE
    IF(KTP1.NE.0)REWIND KTP1
150 CONTINUE
    R=0.0
    DO 160 I=1,IM
    DO 160 J=1,JM
160 R=R+S(I,J)
8041 CONTINUE
    IF(R.LE.RLOG) GOTO 195
    IF(ICC.EQ.0.AND.ITP4.EQ.0)GO TO 175
    IF(IBL.EQ.0.AND.IBR.EQ.0) GOTO 175
    IF(IBL.EQ.4)GO TO 175
    IF(KTP2.EQ.0) GOTO 165
    READ(KTP2)((XNEI(J,M),J=1,JM),M=1,MM)
    GOTO 175
165 DO 170 J=1,JM
    DO 170 M=1,MM
170 XNEI(J,M)=XNEII(J,M,IG)
175 IF(ICC.EQ.0.AND.ITP4.EQ.0)GO TO 190
    IF(IBU.EQ.0.AND.IBO.EQ.0) GOTO 190
    IF(IBO.EQ.4)GO TO 190
    IF(KTP3.EQ.0) GOTO 180
```

```
      READ(KTP3)((XNEJ(I,M),I=1,IM),M=1,MM)
      GOTO 190
180  DO 185 I=1,IM
      DO 185 M=1,MM
185  XNEJ(I,M)=XNEJJ(I,M,IG)
190  CONTINUE
      EQUIVALENCE(JALP,IO(5))
      MMM=MM+1
      IS=ISCT+1
      CALL INNER(XSK,IA(KRM),XNEI,XNEJ,XSV,C,MAI,MAJ,MZ,A(JDU),A(JDE),
1A(JW),A(JAL),A(JDB),S,V,A(JDA),A(JXARI),
2A(JARN),IA(KMRY),IM,JM,MM,IGM,IP,JP,IHM,MT,IXM,IYM,IA(KMTETA),
3 A(JRX),A(JDELY),A(JALP),MMM,SS,A(JTW),TF,IS)
      GOTO 200
195  CALL PUT(XSV,IM*JM,0.)
      CALL PUT(XNEI,JM*MM,0.)
      CALL PUT(XNEJ,IM*MM,0.)
      IF(ISCT.EQ.0) GOTO 200
      CALL PUT(TF,IM*JM*IS*IS,0.)
200  CONTINUE
      I1=IGM-IG
      IF(I1.NE.0) GOTO 235
      IF(ISCT.EQ.0) GOTO 8137
      WRITE(KTPUN1)((((TF(I,J,L,M),I=1,IM),J=1,JM),L=1,IS),M=1,IS)
      I1=KTPUN1
      KTPUN1=KTPUN2
      KTPUN2=I1
      GOTO 235
8137 CONTINUE
      IF(KTP1.EQ.0)GOTO 225
      I1=IGM-1
      DO 220 K=1,I1
220  READ(KTP1)
      WRITE(KTP1)((XSV(I,J),I=1,IM),J=1,JM)
      REWIND KTP1
      I1=KTP1
      KTP1=KTPP1
      KTPP1=I1
233  REWIND KTPP1
      GOTO 235
225  DO 230 I=1,IM
      DO 230 J=1,JM
230  XN(I,J,IGM)=XSV(I,J)
235  CONTINUE
      IF(IBL.EQ.0.AND.IBR.EQ.0) GOTO 250
      IF(KTP2.EQ.0) GOTO 240
      WRITE(KTPP2)((XNEI(J,M),J=1,JM),M=1,MM)
      GOTO 250
240  DO 245 J=1,JM
      DO 245 M=1,MM
245  XNEII(J,M,IG)=XNEI(J,M)
250  CONTINUE
      IF(IBU.EQ.0.AND.IBO.EQ.0) GOTO 265
      IF(KTP3.EQ.0) GOTO 255
      WRITE(KTPP3)((XNEJ(I,M),I=1,IM),M=1,MM)
      GOTO 265
255  DO 260 I=1,IM
      DO 260 M=1,MM
260  XNEJJ(I,M,IG)=XNEJ(I,M)
```

```
265 CONTINUE
  IF(ITH.NE.0) GOTO 272
  DO 270 I=1,IM
  DO 270 J=1,JM
  I1=MAI(I)
  I2=MAJ(J)
  I3=MZ(I1,I2)
  I3=IABS(I3)
270 FG(I,J)=FG(I,J)+C(IHT-1,IG,I3)*XSV(I,J)

  GOTO 273
272 DO 274 I=1,IM
  DO 274 J=1,JM
274 FG(I,J)=FG(I,J)+XKE(IG)*XSV(I,J)
273 CONTINUE
```

INIT I I E R E N

```
  IF(ICVT.NE.2)GO TO 300
  DO 9007 KR=1,KEND
  IF(.NOT.KTUT(KR))GO TO 9007
  MWERT=KR-1
  JFIRST=KB(KR)
  CALL BILAN2(IG, S,XSV,XSK,XN,C,XKE,MAI,MAJ,MZ,F,FG,XNEI,XNEJ,
  1XNEII,XNEJJ,XKI,L1,L2,L3,L4,L5,L6,L7,L8,L9,IM,JM,IGM,MT,IHM,
  2IGP,IXM,IYM,MM,V,DA,DB,W,DE,DU,IP,Q,L10,L11,A(JFIRST),ITYP(KR),
  3INDEX(KR),MWERT,IB(1+(KR-4)*IXM*IYM))
9007 CONTINUE

300 CONTINUE
```

```
  IG=IG+1
  IF(IG.LE.IGM) GOTO 100
  IF(KTP2.EQ.0) GOTO 275
  K1=KTP2
  KTP2=KTPP2
  KTPP2=K1
  REWIND KTP2
  REWIND KTPP2
275 IF(KTP3.EQ.0) GOTO 280
  K1=KTP3
  KTP3=KTPP3
  KTPP3=K1
  REWIND KTP3
  REWIND KTPP3
```

```
280 ICC=ICC+1
  E5=ABS(1.-XLAR)*0.2
  XLAR=XLA
  E3=ABS(1.-XLAR)*0.3
  CALL FISSN(A(JFX),F,FG,V,XKE,IGP,IM,JM,C,MAI,MAJ,MZ,IXM,IYM,
  1IHM,IGM,MT)
```

```
IF(ICVT.NE.2)GO TO 399

IF(ISCT.EQ.0) GOTO 7009
REWIND KTPUN1
REWIND KTPUN2
KG=1
7006 READ(KTPUN1)((((TF(I,J,L,K),I=1,IM),J=1,JM),L=1,IS),K=1,IS)
WRITE(KTPUN2)((TF(I,J,1,1),I=1,IM),J=1,JM)
KG=KG+1
IF(KG.LE.IGM) GOTO 7006
REWIND KTPUN2
DO 7008 KG=1,IGM
7008 READ(KTPUN2)((XN(I,J,KG),I=1,IM),J=1,JM)
7009 CONTINUE
DO 9008 KR=1,KEND
IF(.NOT.KTUT(KR)) GOTO 9008
MWERT=KR-1
JFIRST=KB(KR)
JZWEI=JFIRST+IGP*ITYP(KR)*INDEX(KR)
JDREI=JZWEI+3*IGP*(IXM+IYM)
JVIER=JDREI+IXM
CALL BILAN3(A(JFIRST),ITYP(KR),INDEX(KR),IGP,MWERT,A(JZWEI),
1 IXM+IYM, IB(1+(KR-4)*IXM*IYM),A(JDREI),A(JVIER))
9008 CONTINUE

399 CONTINUE
```

```
E2=ABS(1.-XLA)*0.6
EPG=EPG+E2
IF(E2.GT.10.*EPS) GOTO 285
IIL=IIM
285 EV=XKI(IGP)/XKE(IGP)
IF((E2+E3+E5).GT.EPS)GO TO 500
IF(ICVT.EQ.2)GO TO 500
IF(EPSA.EQ.0.)GO TO 502
DO 501 I=1,IGM
IF(IDIM(I).EQ.IIM)GO TO 500
501 CONTINUE
502 CONTINUE
ICVT=1
500 RETURN
END
```

```
SUBROUTINE OUTUN(SS,TF,MAI,MAJ,MZ,C,T,IM,JM,MM,IS,IXM,IYM,IHM,
1 IGM,MT,V)
COMMON/FIX/NICOM,IA(300)
DIMENSION SS(IM,JM,MM),TF(IM,JM,IS,IS),T(IS,IS,MM),MAI(IM),
1 MAJ(JM),MZ(IXM,IYM),C(IHM,IGM,MT),V(IM,IYM)
EQUIVALENCE(IA(107),ISCT),(IA(108),KTPUN1),(IA(49),IHS),
1 (IA(69),ICC),(IA(72),IG),(IA(109),KTPUN2)

DO 5 M=1,MM
DO 5 I=1,IM
DO 5 J=1,JM
5 SS(I,J,M)=0.0
IF(ICC.EQ.0) GOTO 10
P1=1.0
P2=1.0
```

```
IF(IG.EQ.1) GOTO 15
WRITE(KTPUN1)((((TF(I,J,L,M),I=1,IM),J=1,JM),L=1,IS),M=1,IS)
GOTO 20
15 REWIND KTPUN2
GOTO 20
10 IF(IG.EQ.1) RETURN
WRITE(KTPUN1)((((TF(I,J,L,M),I=1,IM),J=1,JM),L=1,IS),M=1,IS)
RETURN
20 REWIND KTPUN1
K=1
500 IF(K.EQ.IG) GOTO 501
READ(KTPUN1)((((TF(I,J,L,M),I=1,IM),J=1,JM),L=1,IS),M=1,IS)
503 CONTINUE
I4=IHS+IG-K
I=1
300 I1=MAI(I)
J=1
400 I2=MAJ(J)
I3=MZ(I1,I2)
IF(I3.LT.0)MW=1
IF(I3.GT.0)MW=0
I3=IABS(I3)
N=1
600 I5=I3+N-1
KH=2*N-1
W=KH
H1=C(I4,IG,I5)
IF(K.EQ.IG.AND.N.EQ.1) GOTO 1000
H2=H1*TF(I,J,N,1)*W
DO 30 N=1,MM
30 SS(I,J,M)=SS(I,J,M)+H2*T(N,1,M)
1000 IF(MW.EQ.0) GOTO 1500
IF(N.EQ.1) GOTO 1400
LL=N-1
DO 50 L=1,LL
L1=LL-L
P1=1.
IF(L1.LE.1) GOTO 44
DO 40 L2=2,L1
Q=L2
40 P1=P1*Q
44 CONTINUE
L1=LL+L
P2=1.
DO 45 L2=2,L1
Q=L2
45 P2=P2*Q
P3=2.0*P1*W*H1/P2
H2=P3*TF(I,J,N,L+1)
H3=P3*TF(I,J,L,N)
DO 47 M=1,MM
47 SS(I,J,M)=SS(I,J,M)+H2*T(N,L+1,M)+H3*T(L,N,M)
50 CONTINUE
1400 N=N+1
IF(N.LE.IS) GOTO 600
1500 CONTINUE
J=J+1
IF(J.LE.JM) GOTO 400
I=I+1
```



```
IF(I.LE.IM) GOTO 300
K=K+1
IF(K.LE.IG) GOTO 500
GOTO 502
501 READ(KTPUN2)((TF(I,J,L,M),I=1,IM),J=1,JM),L=1,IS),M=1,IS)
GOTO 503
502 CONTINUE
DO 60 I=1,IM
DO 60 J=1,JM
I2=MAJ(J)
W=V(I,I2)
DO 60 M=1,MM
60 SS(I,J,M)=SS(I,J,M)*W
RETURN
END
```

```
SUBROUTINE PLACE
COMMON/TSCH/ITSCH,RHO,ITP1,ITP2,INORM,KLAR(300)
COMMON/SPEICH/MBOUND(25)
EQUIVALENCE(MBOUND(18),INNTS),(MBOUND(19),IKENN),(MBOUND(20),IPOSS
1),(MBOUND(21),LENTS),(MBOUND(22),MREAL)
COMMON/FIX/NICOM,IA(1000)
EQUIVALENCE(IBR,IA(14)),(IBL,IA(13)),(IBO,IA(16)),(IBU,IA(15)),
1(KWT,IA(78))
EQUIVALENCE(IGM,IA(7))
REAL*8 MARK(6)/'*THIS OP','T. USED*','*OPT. NO','T USED *',
1'*OTHER O','PT. USED'/
COMPLEX*16 QARK(3),SOOL,POOL
QARK(1)=DCMPLX(MARK(1),MARK(2))
QARK(2)=DCMPLX(MARK(3),MARK(4))
QARK(3)=DCMPLX(MARK(5),MARK(6))
1 FORMAT('1')
WRITE(KWT,1)
CALL WRIT
2 FORMAT(1H0,30X,'LENGTH OF THE MASTER ARRAY',14X,' ',I6,' WORDS
1'/31X,'PERMANENT RESIDENT MEMBERS',14X,' ',I6,' WORDS'/31X,'FOR TE
MPORARY RESIDENT ARRAYS AVAILABLE ',I6,' WORDS')
WRITE(KWT,2)MBOUND(1),MBOUND(2),MBOUND(16)
3 FORMAT(31X,'CROSS SECTION INPUT USED',16X,' ',I6,' WORDS')
WRITE(KWT,3)MBOUND(15)
4 FORMAT('0',30X,'(KENN=0/1)=EXTERNAL/CORE STORAGE VERSION OF TSCHEB
1, KENN = ',I1,/31X,'EXTERNAL STORAGE VERSION ',I6,' WORDS'
2,3X,2A8/31X,'CORE STORAGE VERSION ',I6,' WORDS',3X,
32A8)
POOL=QARK(3)
SOOL=QARK(1)
IF(MBOUND(14).EQ.0)POOL=QARK(1)
IF(MBOUND(14).EQ.0)SOOL=QARK(2)
IF(ITSCH.EQ.0.OR.MBOUND(17).LE.0)POOL=QARK(2)
IF(ITSCH.EQ.0.OR.MBOUND(17).LE.0)SOOL=QARK(2)
WRITE(KWT,4)MBOUND(14),MBOUND(13),POOL, MBOUND(12),SOOL
IF(INNTS.EQ.0)GO TO 20
WRITE(KWT,21)IGM,IPOSS,MREAL,LENTS,IKENN
20 CONTINUE
21 FORMAT('0',30X,'FROM ',I3,' GROUPS',I3,' WOULD HAVE BEEN ABLE TO B
1E ACCELERATED'/31X,'BY TCHEBYCHEW EXTRAPOLATION OF THE INNER ITERA
2TIONS'/31X,'ACTUALLY ',I3,' GROUPS HAVE BEEN ACCELERATED. LENTS =
3',I6,' ', IKENN = ',I3)
```

```
DO 10 I=1,3
  IF(MBOUND(3*I)) 10, 11, 12
11 WRITE(KWT,13) I, MBOUND(3*I), MBOUND(3*I+1), QARK(3),
  MBOUND(3*I+2), QARK(1)
  GO TO 10
12 WRITE(KWT,13) I, MBOUND(3*I), MBOUND(3*I+1), QARK(1), MBOUND(3*I+2),
  QARK(3)
10 CONTINUE
  CALL WRIT
  DO 30 I=1, IGM
30 WRITE(KWT,31) I, KLAR(I)
31 FORMAT(31X, 'TOTAL NUMBER OF INNER ITERATIONS IN GROUP ', I3, ' ', I
  14)
  CALL WRIT
  CALL OUT2
  WRITE(KWT,1)
  RETURN
13 FORMAT('0', 30X, 'PHASE ', I1, ' OF FLUX PASS', ' KLUX =', I2, ' (0/1
  1) = SMALL/BIG VERSION.'/31X, 'IALPH =', I6, 3X, 2A8/31X, 'IBET =', I6,
  23X, 2A8)
  END
```

```
SUBROUTINE PLOT1
COMMON NACOM, A(1000)
COMMON/FIX/NICOM, IA(1000)
COMMON/ORG/IO(40)
EQUIVALENCE( IM, IA(71)), (IGM, IA(7)), (JRAX, IO(20)),
1 (JM, IA(70)), (JRAY, IO(21)), (JXN, IO(38))
EQUIVALENCE( JBKOUT, IA(90)), (KPCH, IA(91)), (KPLOT, IA(92)), (KNUMB, IA(
*93)), (MODUS, IA(94))
EQUIVALENCE( IA(95), IBILD)
EQUIVALENCE( KR0, IA(79))
EQUIVALENCE( JFIN, IA(195))
DIMENSION IRIG(26)

DO 4 I=1, IGM
4 IRIG(I)=0
  READ(KRD,11) N, (IRIG(I), I=1, N)
  READ(KRD,11) KNUMB, MODUS, IBILD
11 FORMAT(12I6)
  IF(KPLOT.EQ.-7) RETURN
  DO 2 I=1, IGM
  IS=IGM+1-I
  IF(IRIG(IS).EQ.0) GO TO 2
  J=IRIG(IS)
  IRIG(J)=1
  IRIG(IS)=0
2 CONTINUE
  RETURN
  ENTRY PLOT2
  IF(KPLOT.NE.7) GO TO 5
```

```
DO 1 I=1,IGM
  JKN=JXN+(I-1)*IM*JM
  IF(IRIG(I).EQ.0)GO TO 1
  IF(IBILD*(IBILD-1).NE.0)IBILD=0
  K=KNUMB
  MOD=MODUS
  CALL LEVLOC(A(JRAX),A(JRAY),A(JKN),IM,JM,K,MOD,IBILD,IER,I)
1 CONTINUE
  RETURN
5 NUMBO=1
  DO 6 I=1,IM
    DO 6 J=1,JM
6 A(JFIN+I-1+(J-1)*IM)=A(JXN+I-1+(J-1)*IM)

DO 7 IG=2,IGM
  JKN=JXN+(IG-1)*IM*JM
  IF(IRIG(IG)*IRIG(IG-1).LE.0)GO TO 8
  DO 9 I=1,IM
    DO 9 J=1,JM
9 A(JFIN+I-1+(J-1)*IM)=A(JKN+I-1+(J-1)*IM)+A(JFIN+I-1+(J-1)*IM)
  GO TO 7
8 K=KNUMB
  MOD=MODUS
  JPN=JFIN
  I=NUMBO
  CALL LEVLOC(A(JRAX),A(JRAY),A(JPN),IM,JM,K,MOD,IBILD,IER,I)
  IF(IGM.EQ.1)GO TO 12
  NUMBO=NUMBO+1
  DO 10 I=1,IM
    DO 10 J=1,JM
10 A(JFIN+I-1+(J-1)*IM)=A(JKN+I-1+(J-1)*IM)
  7 CONTINUE
12 RETURN
  END
```

```
SUBROUTINE PRAEP
  KRD=5
  KWT=6
  INPUT REPRODUCTION
  REAL *8 T(11)/10*0.0D+0,'99999' */
  J=1
  WRITE(KWT,100)
100 FORMAT(1H1/////////30X,' REPRODUCTION OF THE CARD INPUT' //)
  WRITE(KWT,10)
  WRITE(KWT,44)
  10 FORMAT(10X,96(' '))
  20 READ(KRD,11,END= 28,ERR=27) (T(I),I=1,10)
  11 FORMAT(10A8)
  WRITE(KWT,12)J,(T(I),I=1,10)
  12 FORMAT(10X,6H* CARD,13,4X,'I',10A8,'I')
  44 FORMAT(10X,'*',12X,'I',1X,6(5X,1H',5X,'I' ),7X,'I',2(/10X,'*',12X
```

```
1, 'I', 80X, 'I'))
  IF(T(1).EQ.T(11))GO TO 29
  J=J+1
  GO TO 20
29 WRITE(KWT,10)
  GO TO 30
28 CALL FSPIE
  WRITE(KWT,13)
13 FORMAT(100H0THERE HAS BEEN NO FINAL CARD BEGINNING WITH '99999'
  1BEFORE FINDING THE END OF THE INPUT DATA SET /50H AN ATTEMPT IS M
  2ADE TO START THE JOB
  GO TO 30
27 CALL FSPIE
  WRITE(KWT,14)
14 FORMAT(100HOAN I/O ERROR CONDITION HAS BEEN ENCOUNTERED DURING THE
  1 ALPHAMERIC READING OF THE INPUT SET
  STOP
30 REWIND KRD
  CALL GECKO
  RETURN
  END
```

'POLMAN' IS AN INITIATING SUBROUTINE TO PREPARE THE CALLING
OF 'IPOLAN'.

```
SUBROUTINE POLMAN(ARRAY,L1,MM,IGM,ISN,ISNOLD,QL)
DIMENSION ARRAY(L1,MM,IGM),QL(MM)
INTEGER*2 KMPOL(65)
1/1,2,2,1,2,
2 1,2,2,2,1,2,2,1,2,
3 1,2,2,2,2,1,2,2,2,1,2,2,1,2,
4 1,2,3,3,6,7,7,6,7,
5 1,2,2,3,3,1,2,3,3,6,7,7,6,7,
6 1,2,3,4,4,1,2,3,4,8,9,10,13,14/,
7INFO(6)/1,6,15,29,38,52/
```

.....

```
KENN=1
IF(ISN.EQ.6.AND.ISNOLD.EQ.2)KENN=2
IF(ISN.EQ.8.AND.ISNOLD.EQ.2)KENN=3
IF(ISN.EQ.6.AND.ISNOLD.EQ.4)KENN=4
IF(ISN.EQ.8.AND.ISNOLD.EQ.4)KENN=5
IF(ISN.EQ.8.AND.ISNOLD.EQ.6)KENN=6
INF=INFO(KENN)
MMOLD=ISNOLD*(ISNOLD+4)/2
CALL IPOLAN(ISN,KMPOL(INF),MMOLD,ARRAY,L1,MM,IGM,QL)
RETURN
```

END

```
SUBROUTINE PRINT1(ZRAX,ZRAY,MZ,C,IHM,IGM,MT,IXM,IYM,MAI,MAJ,IM,JM)
DIMENSION ZRAX(IXM),ZRAY(IYM),MZ(IXM,IYM),KP(8),MAI(IM),MAJ(JM)
1,C(IHM,IGM,MT)
COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
COMMON /ORG/IO(43),IE(20)
COMMON/SPEICH/MBOUND(25)
COMMON/TSCH/ITSCH,RHO,ITP1,ITP2,INORM,PFORM(300)
COMMON /CONTR/IREBAL,IDIM(26),FUNK(26),DEV1(26),DEV2(26),NEGOV(26)
EQUIVALENCE(IA(88),NFF)
EQUIVALENCE(IA(47),NACO),(IA(46),JFIN),(IA(45),ITP3),(IA(44),
1ITP4),(IA(43),KLUX),(IA(42),IRIS)
2 ,(IA(13),IBL),(IA(14),IBR),(IA(15),IBU),(IA(16),IBO)
EQUIVALENCE(IA(1),ID),(IA(2),ITH),(IA(3),ISN),(IA(4),IGE),(IA(5),
1IB1),(IA(6),IB2),(IA(8),IEVT), (IA(7),IGM),(IA(11),MT),
2(IA(29),IIM),(IA(30),IIL),(IA(28),ICM)
EQUIVALENCE(IA(80),LC),(IA(79),KRD),(IA(78),KWT),(IA(77),KDUM),
1(IA(76),IIC),(IA(75),IHM),(IA(74),IP),(IA(73),JP),(IA(72),IG),
2 (IA(69),ICC),(IA(68),KTP1),(IA(67),KTP2),
3(IA(66),KTP3),(IA(65),KTPP2),(IA(64),KTPP3),(IA(63),IGP),(IA(62),
4MM) ,(IA(61),ICVT),(IA(60),NFN)
EQUIVALENCE(A(1),EPS),(A(2),EV),(A(13),XLA),(A(14),XLAR),(A(15),
1 EPG),(A(16),RLOG),(A(3),DZ)
EQUIVALENCE(IA(59),L1),(IA(58),L2),(IA(57),L3),(IA(56),L4),
1(IA(55),L5),(IA(54),L6),(IA(53),L7),(IA(52),L8),(IA(51),L9)
EQUIVALENCE(IA(50),KTPP1),(IA(12),KTR),(IINP,IA(27))
EQUIVALENCE(KFL1,IE(8)),(KFL2,IE(9))
EQUIVALENCE(IHS,IA(49)),(IHT,IA(48)),(IA(101),IQUER)
EQUIVALENCE(IO(1),JSK),(IO(2),JNEI),(IO(3),JNEJ),(IO(4),JARN),
1(IO(5),JALP),(IO(6),JDAA),(IO(7),JDA),(IO(8),JW),(IO(9),JAL),
2(IO(10),JDU),(IO(11),JDE),(IO(12),JDBB),(IO(13),JDB),(IO(14),JV),
3(IO(15),JS),(IO(16),JC),(IO(17),JSV),(IO(18),JRX),(IO(19),JRY),
4(IO(20),JRAX),(IO(21),JRAY),(IO(22),JDELX),(IO(23),JDELY),
5(IO(24),JXKI),(IO(25),JXKE),(IO(26),JVE),(IO(27),JF),(IO(28),
6JFG),(IO(29),JXARI),(IO(30),JFX),(IO(31),JB),(IO(32),JZEPS),
7(IO(33),JZRAX),(IO(34),JZRAY),(IO(36),JXNEII),(IO(37),JXNEJJ),
8(IO(38),JXN),(IO(35),JQ)
EQUIVALENCE(IE(1),KRX),(IE(2),KRY),(IE(3),KRM),(IE(4),KAI),(IE(5),
1KAJ),(IE(6),KZ),(IE(7),KRMY),(IE(8),KFL1),(IE(9),KFL2),(IE(10),
2KTETA),(IE(11),KMIM),(IE(12),KMJM),(IE(13),KMMRX),(IE(14),KMMRY),
3(IE(15),KMISN),(IE(16),KFIN)
EQUIVALENCE(EP SA,A(17)),(MBT,IA(81)),(IQUELL,IA(82)),(IZT,IA(85)),
1(JBKTP,IA(83)),(JOUT,IA(84)),(A(18),TEPS),(MBOUND(18),INNTS)
2 ,(IA(102),ID1),(IA(103),ID2),(IA(104),ID3),(IA(105),ID4),
3 (IA(106),ID5),(IA(107),ISCT)
1 ,(IA(109),KTPUN2)
EQUIVALENCE(KAUSW,IA(87))
EQUIVALENCE(MF01,IA(86))
EQUIVALENCE(JBKOUT,IA(90)),(KPCH,IA(91)),(KPL0T,IA(92)),(KNU MB,IA(
*93)),(MODUS,IA(94))
EQUIVALENCE(IA(95),IBILD)
```

```
MBK=IABS(MBT)
WRITE(KWT,140)(ID,ITH,ISN,IGE,IGM,IXM,IYM,MT,IM,JM,ICM,IIM)
WRITE(KWT,141)(IIL,IREBAL,ITSCH,KDUM,IINP)
WRITE(KWT,141)IQUER,ITP1,ITP2,ITP3,ITP4,KAUSW
WRITE(KWT,180)IB1,INNTS,IQUELL,MBT,IEVT
WRITE(KWT,185)(NFF,ISCT)
185 FORMAT(20X,'DETAILED NEG. FLUX FIXUP HANDLING(0=NO,1=YES)',I6,2X,
* '( NFF )'/ 20X,'BUCKLING OUTPUT ON KPCH',22X,I6,2X,'(JBKOUT)'/
* 20X,'UNIT TO GET BUCKLINGS',24X,I6,2X,'( KPCH )'/
* 20X,'GRAPHIC OUTPUT, LEVEL CURVES OF NEUTRON FLUX',I7,2X,'( KPLOT
*)'
* /25X,'(IF=7,YES, OTHERWISE NO) '/
*20X,'NUMBER OF LEVEL CURVES PRO PICTURE',11X,I6,2X,'( KNUMB)'/
* 20X,'SPECIFICATION OF CURVES DISTANCE',13X,I6,2X,'( MODUS)'/
* 20X,'PLOT MESHLINES(0=YES,1=NO)',19X,I6,2X,'( IBILD)')
CALL WRIT
WRITE(KWT,142)EPS,DZ,EPSA,TEPS,RHO
CALL WRIT

I1=JRX
I2=JRX+IP-1
WRITE(KWT,143) IP
WRITE(KWT,150)(A(I),I=I1,I2)
I1=JRY
I2=JRY+JM
WRITE(KWT,144) JP
WRITE(KWT,150)(A(I),I=I1,I2)
I1=KAI
I2=KAI+IM-1
WRITE(KWT,145) IM
WRITE(KWT,149) (IA(I),I=I1,I2)
I1=KAJ
I2=KAJ+JM-1
WRITE(KWT,146) JM
WRITE(KWT,149) (IA(I),I=I1,I2)

500 CONTINUE
BOUNDARIES
WRITE(KWT,5400)
IF( IBL.EQ.1)WRITE(KWT,5401)
IF( IBL.EQ.0.OR.IBL.EQ.4)WRITE(KWT,5402)
IF( IBL.EQ.2)WRITE(KWT,5403)
IF( IBL.EQ.0.OR.IBL.EQ.4)WRITE(KWT,5405)
IF( IBL.EQ.1)WRITE(KWT,5404)
IF( IBL.EQ.2)WRITE(KWT,5406)
IF( IBL.EQ.0.OR.IBL.EQ.4)WRITE(KWT,5408)
IF( IBL.EQ.1)WRITE(KWT,5407)
IF( IBL.EQ.2)WRITE(KWT,5409)
IF( IBL.EQ.0.OR.IBL.EQ.4)WRITE(KWT,5411)
IF( IBL.EQ.1)WRITE(KWT,5410)
IF( IBL.EQ.2)WRITE(KWT,5412)

WRITE(KWT,161)
I1=JDU
I2=JDE
I3=KRM
I4=KRM
```

```
I5=JW
DO 165 I=1,MM
WRITE(KWT,160)(A(I1),A(I2),IA(I3),IA(I4),A(I5),I)
I1=I1+1
I2=I2+1
I3=I3+1
I4=I4+1
165 I5=I5+1

WRITE(KWT,158)IGP
I1=JXKI
I2=JXKE-1
WRITE(KWT,150)(A(I),I=I1,I2)
CCCCCCCCCCCCCCCC
III=IXM
II2=IYM
IF(IQUELL.NE.0)WRITE(KWT,1804)
IF(IQUELL)1806,1809,1808
1808 III=IM
II2=JM
1806 IF(IABS(IQUELL).EQ.2)GO TO 1810
1804 FORMAT(//50X,' DISTRIBUTED SOURCE'////)
DO 1803 IG=1,IGM
WRITE(KWT,1805)IG
1805 FORMAT(/14X,' GROUP',I3/)
DO 1700 J=1,II2
1700 WRITE(KWT,1701)(A(JQ+(IG-1)+IGM*(I-1)+IGM*III*(J-1)),I=1,III)
1803 CONTINUE
GO TO 1809
1810 WRITE(KWT,1811)
WRITE(KWT,1701)(A(JQ+I-1),I=1,IGM)
1811 FORMAT(10X,'SOURCE SPECTRUM')
1812 FORMAT(10X,'SOURCE INTENSITY')
WRITE(KWT,1812)
DO 1813 J=1,II2
WRITE(KWT,1701)(A(JQ+IGM-1+I+(J-1)*III),I=1,III)
1813 CONTINUE
1701 FORMAT(2X,10E12.4)
1809 IF(MBK.LE.1)GO TO 203
I2=JZEPS-1
WRITE(KWT,204)
204 FORMAT(//30X,'BUCKLING'//33X,'IF MBK.EQ.2 FOR EACH GROUP'/
133X,'IF MBK.EQ.3 FOR EACH GROUP AND ALL ZONES'/
233X,'IF MBK.EQ.4 FOR EACH GROUP AND ALL Z-POINTS'//)
WRITE(KWT,150)(A(I),I=JB,I2)
203 CONTINUE

IF(IQUER.EQ.0)GO TO 30
IF(IB1.NE.0.AND.MF01.NE.1)GO TO 30

ONLY DOWNSCATTERING

DO 32 I=1,MT
DO 33 J=1,IGM
IF(C(IHT,J,I).LE.RLOG)GO TO 33
BWQ=C(IHT,J,I)-C(IHT-2,J,I)
IU=IHS
```

```
JD=J
34 BWQ=BWQ-C(JU,JD,I)
JU=JU+1
JD=JD+1
IF(JD.LE.IGM.AND.JU.LE.IHM)GO TO 34
BWQ=BWQ/C(IHT,J,I)
IF(ABS(BWQ).LE.EPS)GO TO 33
WRITE(KWT,35)I,J,BWQ
35 FORMAT('0',10X,'MATERIAL',I3,',',GROUP',I3,',(1.0 - CALC TOT X-SEC
IT/INPUT TOT X-SECT) = ',E13.6)
33 CONTINUE
32 CONTINUE
WRITE(KWT,1001)
DO 1008 M=1,MT
I1=1
I2=M INO(IGM,8)
WRITE(KWT,1003) M
1015 DO 1009 K=1,8
1009 KP(K)=I1+K-1
I3=I2-I1+1
WRITE(KWT,1002)(KP(K),K=1,I3)
WRITE(KWT,1005)
DO 1007 I=1,IHM
1007 WRITE (KWT,1004)(C(I,J,M),J=I1,I2)
I1=I1+8
I2=I2+8
IF(IGM.GE.I1) GO TO 1020
GOTO 1008
1020 IF(IGM.LT.I2) I2=IGM
GOTO 1015
1008 CONTINUE
30 CONTINUE

WRITE(KWT,147)
CALL MALE(MZ,MAI,MAJ,IXM,IYM,IXM,IYM,0,A(JQ),IGM)
WRITE(KWT,155)
155 FORMAT(///43X,'MATERIAL DISTRIBUTION ON THE INTERVALS '///)
147 FORMAT(///44X,' MATERIAL DISTRIBUTION ON THE ZONES'///)
CALL MALE(MZ,MAI,MAJ,IM,JM,IXM,IYM,1,A(JQ),IGM)
IF(IQUELL.GE.0)GO TO 157
WRITE(KWT,156)
CALL MALE(MZ,MAI,MAJ,IXM,IYM,IXM,IYM,-1,A(JQ),IGM)
156 FORMAT(///46X,'POSITION OF THE EXTERNAL SOURCE'///)
157 CONTINUE

149 FORMAT(1X,10I8)
150 FORMAT(1X,10E12.5)
1 FORMAT(/////1X,'DISTRIBUTION OF MATERIALS ON ZONES')
2 FORMAT(//4X,'Y,Z,THETA'/4X,'DIRECTION'//9X,1H*)
3 FORMAT(3X,F5.3,1X,1H*)
4 FORMAT(9X,'*')
5 FORMAT(1H+,9X,30A4)
6 FORMAT(1H+,9X,15(4X,A4))
7 FORMAT(1H+,9X,15(I4,A4))
8 FORMAT(6X,15(F5.3,4X))
9 FORMAT(4X,20I5/)
1141 FORMAT(20X,
1 'PRINT CROSS SECTION TABLES (0=NO,1=YES)',6X,I6,2X,'( IQUER)'/20
1X,
```



```
2'FIRST FILE FOR OUTER TCHEBYCHEW EXTRAP.',6X,I6,2X,'( ITP1 )'/20X,  
3'SECOND FILE,SAME PURPOSE,DIFFERENT FROM ITP1',1X,I6,2X,'( ITP2 )'  
1/20X  
6,'FINAL FLUX WRITTEN ON UNIT (IF=0 NOT USED)',3X,I6,2X,'( ITP3 )'/  
1 25X,'AFTER EACH OUTER ITERATION'/  
120X,  
4 'INITIAL FLUX READ FROM UNIT (0=NOT USED)'5X,I6,2X,'( ITP4 )'/20X  
8 ', 'EVALUATION TRIGGER,BALANCES',18X,I6,2X,'( KAUSW)'  
140 FORMAT(/////20X,  
1 'IDENTIFICATION',31X,I6,2X,'( ID )'/20X ,  
1 'THEORY (0=REGULAR,1=ADJOINT)',17X,I6,2X,'( ITH )'/20X,  
1 'SN-ORDER',37X,I6,2X,'( ISN )'/20X,  
1 'GECMETRY (0=(X,Y),1=(R,Z),2=(R,THETA))',7X,I6,2X,'( IGE )'/20X,  
1 'NUMBER OF GROUPS',29X,I6,2X,'( IGM )'/20X,  
3 'NUMBER OF ZONES (X/R-DIRECTION)',14X,I6,2X,'( IZM )'/20X,  
4 'NUMBER OF ZONES (Y/Z/THETA-DIRECTION)',8X,I6,2X,'( IYM )'/20X  
5 ', 'NUMBER OF MATERIALS',26X,I6,2X,'( MT )'/20X,  
5 'NUMBER OF INTERVALS (X/R-DIRECTION)'10X,I6,2X,'( IM )'/20X,  
7 'NUMBER OF INTERVALS(Y/Z/THETA-DIRECTION)',5X,I6,2X,'( JM )'/  
8 20X,  
8 'MAXIMUM NUMBER OF OUTER ITERATIONS',11X,I6,2X,'( ICM )'/20X,  
9 'MAXIMUM NUMBER OF INNER ITERATIONS',11X,I6,2X,'( IIM )'  
141 FORMAT(20X,  
1 'INITIAL-MAXIMUM NUMBER OF INNER ITERATIONS',3X,I6,2X,'( IIL )'/  
12  
10X,'REBALANCING USED (1=NO,0=YES)',16X,I6,2X,'(IREBAL)'/20X,  
1 'TSCHEBYSCHIEFF-ACCELERATION USED (0=NO,1=YES)',1X,I6,2X,'( ITSC  
2H)'/20X,  
3 'CROSS SECTION INPUT BY TAPE (1=NO/0=YES)',5X,I6,2X,'( KDUM )'/  
220X,  
7 'CROSS SECTION TAPE NUMBER',20X,I6,2X,'( IINP )'  
180 FORMAT(20X,  
1 'NUMBER OF INTERPOLATING STEPS',16X,I6,2X,'( IB1 )'/20X,  
2 'TCHEBYCHEW ACCELERATION IN INNER(0=NO)',7X,I6,2X,'( INNTS)'/20X,  
3 'DISTRIBUTED SOURCE (0=NO,1  
4=BY GROUP AND'/25X,'SPACE,-1=BY GROUP AND ZONE',14X,I6,2X,'(IQUELL  
5)'/20X,  
3'BUCKLING(0=NO,1=YES,2=BY GROUP,3=BY GROUP AND ZONE,4=BY GRO  
7UP AND SPACE',11X,I6,2X,'( MBK )'/20X,'PROBLEM TYPE',33X,I6,2X,'( I  
8 IEVT )'  
142 FORMAT( / 20X,  
1 'CONVERGENCE CRITERION',18X,E12.5,2X,'( EPS )'/20X,  
2 'BUCKLING HEIGHT',24X,E12.5,2X,'( DZ )'/20X,  
3 'FLUX CONVERGENCE CRITERION',13X,E12.5,2X,'( EPSA )'/20X,  
4'CRITERION FOR ACCEL. PARAMETER SEARCH', 2X,E12.5,2X,'( TEPS )'/20  
6X,  
5 'ESTIMATED OUTER EXTRAP. PARAMETER', 6X,E12.5,2X,'( RHO )'/)  
  
143 FORMAT(/////1X,'RADII (X/R-DIRECTION) ',I4/)  
144 FORMAT(/////1X,'RADII (Y/Z/THETA-DIRECTION) ',I4/)  
145 FORMAT(/////1X,'THE',I5,' INTERVALS IN (X/R)-DIRECTION ARE LOCATED  
1 IN THE ZONES WITH NUMBER'/)  
146 FORMAT(/////1X,'THE',I5,' INTERVALS IN (Y/Z/THETA)-DIRECTION ARE L  
1OCATED IN THE ZONES WITH NUMBER'/)  
5400 FORMAT(/////28X,' BOUNDARY CONDITIONS'/)  
5401 FORMAT(30X,' X=0: REFLEXION')  
5402 FORMAT(30X,' X=0: VACUUM')  
5403 FORMAT(30X,' X=0: PERIODIC')  
5404 FORMAT(30X,' X=R: REFLEXION')
```

```
5405 FORMAT(30X,' X=R: VACUUM')
5406 FORMAT(30X,' X=R: PERIODIC')
5407 FORMAT(30X,' Y=0: REFLEXION')
5408 FORMAT(30X,' Y=0: VACUUM')
5409 FORMAT(30X,' Y=0: PERIODIC')
5410 FORMAT(30X,' Y=R: REFLEXION')
5411 FORMAT(30X,' Y=R: VACUUM')
5412 FORMAT(30X,' Y=R: PERIODIC')
160 FORMAT(23X,E12.5,E13.5,2I8,8X,E12.5,I8)
161 FORMAT(/////////30X,'DIRECTIONS',11X,'REFL.DIRECTIONS',8X,'WEIGHTS
1'//27X,'MUE',10X,'ETA',10X,'MUE',5X,'ETA')
158 FORMAT(////////1X,'FISSION FRAC',I6/)
1001 FORMAT(////////1X,'CROSS SECTIONS'//)
1005 FORMAT(/)
1002 FORMAT(3X,8(5X,5HGROUP,I3))
1003 FORMAT(////////1X,8HMATERIAL,I4///)
1004 FORMAT(1X,8E13.5)
RETURN
ENTRY PRINT2(XSV,XN,RAX,RAY,FX,MAI,MAJ,V,IM,JM,IGP,IYM,TF,IS)
VERSION FOR TAPEOPTION MUST STILL BE INTEGRATED
DIMENSION XSV(IM,JM),XN(IM,JM,IGM),RAX(IM),RAY(JM),FX(IGP),
1 V(IM,IYM),TF(IM,JM,IS,IS)
FLUX PRINT
801 FORMAT( 1H1 , 'FLUX BY GROUP AND SPACE'//1X,'(I=X/R-DIRECTION,J=
1Y/Z/THETA-DIRECTION)')
802 FORMAT(///1X,'GROUP',I3)
803 FORMAT(///4X,'J/I',5X,9(F8.4,4X),F8.4/)
804 FORMAT(1X,F8.4,2X,10E12.5)
IG=1
600 IF(KTP1.EQ.0) GOTO620
REWIND KTP1
READ(KTP1)((XSV(I,J),I=1,IM),J=1,JM)
GOTO 630
620 DO 625 I=1,IM
DO 625 J=1,JM
625 XSV(I,J)=XN(I,J,IG)
630 CONTINUE
IF(KTR.EQ.0) GOTO 606
I1=KFL2+IG-1
I2=IA(I1)
IF(IG.NE.1) GOTO 605
WRITE(KWT,801)
605 IF(I2.NE.1) GOTO 603
WRITE(KWT,802) IG
I1=1
I2=MINO(10,IM)
610 WRITE(KWT,803)(RAX(I),I=I1,I2)
DO 615 J=1,JM
615 WRITE(KWT,804)(RAY(J),(XSV(I,J),I=I1,I2))
I1=I1+10
I2=I2+10
I2=MINO(I2,IM)
IF(IM.GE.I1) GOTO 610
603 CONTINUE
IG=IG+1
IF(IG.LE.IGM) GOTO 600
```

VOLUME- INTEGRATED FISSIONS

```
805 FORMAT(/////1X,'VOLUME-INTEGRATED FISSION SOURCE'//1X,'GROUP',3X,  
1'FISSIONS'//)  
806 FORMAT(1X,I3,5X,E12.5)  
606 WRITE(KWT,805)  
DO 645 I=1,IGP  
AA=FX(I)  
645 WRITE(KWT,806)(I,AA)  
WRITE(KWT,1467)  
1467 FORMAT(///'0')  
WRITE(KWT,407)(K,NEGOV(K),K=1,IGM)  
407 FORMAT( 10X,'NUMBER OF NEG. COMPONENTS IN GROUP ',I2,2X,I5,' POI  
INTS')
```

ACTIVITY PRINT

```
I=ID1+ID2+ID3+ID4+ID5  
IF(I.EQ.0) GOTO 799  
810 FORMAT(/////1X,'ACTIVITY PRINT'//1X, 'CROSS SECTION TYPE  
11 = STR'/ 1X,'CROSS SECTION TYPE 2 = SFISS'/1X,'CROSS SECTION TYPE  
2 3 = SCAPT+SFISS'/1X,'CROSS SECTION TYPE 4 = NUSF'/1X,'CROSS SECTI  
3ON TYPE 5 = STRTR')  
WRITE(KWT,810)  
L=0  
700 L=L+1  
GOTO(701,702,703,704,705,799),L  
701 IF(ID1.EQ.0) GOTO 700  
GOTO 706  
702 IF(ID2.EQ.0) GOTO 700  
GOTO 706  
703 IF(ID3.EQ.0) GOTO 700  
GOTO 706  
704 IF(ID4.EQ.0) GOTO 700  
GOTO 706  
705 IF(ID5.EQ.0) GOTO 700  
706 LL=0  
GOTO(707,708,709,710,711),L  
707 IF(ID1.EQ.2) LL=1  
GOTO 712  
708 IF(ID2.EQ.2) LL=1  
GOTO 712  
709 IF(ID3.EQ.2) LL=1  
GOTO 712  
710 IF(ID4.EQ.2) LL=1  
GOTO 712  
711 IF(ID5.EQ.2)LL=1  
812 FORMAT(///1X,'ENERGY-INTEGRAL OF FLUX-WEIGHTED CROSS SECTIONS,TYPE  
1',I3)  
814 FORMAT(///1X,'ENERGY-INTEGRAL OF FLUX- AND VOLUME-WEIGHTED CROSS S  
LECTIONS,TYPE',I3)  
712 IF(LL) 713,713,714  
713 WRITE(KWT,812) L  
GOTO 715  
714 WRITE(KWT,814) L  
715 CONTINUE  
DO 720 I=1,IM  
I1=MAI(I)  
DO 720 J=1,JM  
I2=MAJ(J)  
I3=M7(I1,I2)
```

```
T=0.0
DO 720 K=1,IGM
T=T+C(L,K,I3)*XN(I,J,K)
720 XSV(I,J)=T
IF(LL.EQ.0) GOTO 721
DO 722 I=1,IM
DO 722 J=1,JM
I2=MAJ(J)
722 XSV(I,J)=V(I,I2)*XSV(I,J)
815 FORMAT (/1X,' I=X/R-DIRECTION,J=Y/Z/THETA-DIRECTION'//4X,'J/I',5X,
19(F8.4,4X),F8.4/)
721 I1=1
I2=MINO(10,IM)
724 WRITE(KWT,815)(RAX(I),I=I1,I2)
DO 725 J=1,JM
725 WRITE(KWT,804)(RAY(J),(XSV(I,J),I=I1,I2))
I1=I1+10
I2=I2+10
I2=MINO(I2,IM)
IF(IM.GE.I1) GOTO 724
GOTO 700

799 CONTINUE
RETURN
ENTRY PRINT3(I)
EQUIVALENCE(IA(89),KONTRO)
IF(KONTRO.EQ.1) GO TO 899
CALL MESSAG
CALL MESSAG
WRITE(KWT,900)
CALL MESSAG
900 FORMAT('+',30X,' ERROR IN SNOW WHICH CAUSES THE JOB TO BE TERMINAT
LED')
899 CONTINUE
CALL MESSAG
IF(I.EQ.1) WRITE(KWT,901)
IF(I.EQ.2) WRITE(KWT,902)
IF(I.EQ.3) WRITE(KWT,903)
IF(I.EQ.4) WRITE(KWT,904)
IF(I.EQ.5) WRITE(KWT,905)
IF(I.EQ.6) WRITE(KWT,906)
IF(I.EQ.7) WRITE(KWT,907)
IF(I.EQ.8) WRITE(KWT,908)
IF(I.EQ.9) WRITE(KWT,909)
IF(I.EQ.10) WRITE(KWT,910)
IF(I.EQ.11) WRITE(KWT,911)
IF(I.EQ.12) WRITE(KWT,912)
IF(I.EQ.13) WRITE(KWT,913) ITP4
IF(I.EQ.14) WRITE(KWT,9014) IINP
IF(I.EQ.15) WRITE(KWT,9015) IINP
IF(I.EQ.16) WRITE(KWT,9016)
IF(I.EQ.17) WRITE(KWT,9017)
901 FORMAT('+',30X,' INCORRECT SN-ORDER')
IF(I.EQ.18) WRITE(KWT,9018) KAUSW,KPCH
IF(I.EQ.19) WRITE(KWT,9019) KPLOT
902 FORMAT('+',30X,' NUMBER OF ZONE INTERVALS NOT POSITIVE')
903 FORMAT('+',30X,' INCONSISTENCY OF BUCKLINGS')
904 FORMAT('+',30X,' INCONSISTENCY OF ZONE RADII')
905 FORMAT('+',30X,' MATERIAL NUMBERS INCORRECT')
```

```
906 FORMAT('+',30X,'IEVT=0 AND IQUELL=0 NOT PERTITTED')
907 FORMAT('+',30X,'INCGNSISTENCY OF THE BOUNDARY CONDITIONS')
908 FORMAT('+',30X,'UNQUALIFIED PRECISION DEMAND')
909 FORMAT('+',30X,'NOT ALLOWED TO USE ONLY ONE POINT IN A DIRECTION')
910 FORMAT('+',30X,'FISSION SPECTRUM VANISHES')
911 FORMAT('+',30X,'EXTERNAL SOURCE IS NOT POSITIVE')
912 FORMAT('+',30X,'NUMBER OF FLUXES TO WRITE OUT NOT CORRECT')
913 FORMAT('+',30X,'FAILED AT PROV1, ITAPE1 = ',I5)
9014 FORMAT('+',30X,' DD-STATEMENT MISSING FOR IINP = ',I10,' OR NUSYS-
1STEP HAS FAILED TO WRITE ON IINP')
9015 FORMAT('+',30X,' INCORRECT VALUE OF IINP, IINP = ',I10)
9016 FORMAT('+',30X,'THERE SEEMS NO CORRECT UNIT TO BE SPECIFIED FOR FL
LUX INPUT')
9017 FORMAT('+',30X,'INCOSISTENT SPECIFICATION OF GEOMETRY AND BOUNDARY
1 CONDITIONS')
9018 FORMAT('+',30X,'BUCKLING OUTPUT OPTION DELETED, KAUSW = ',I4,
* ' KPCH = ',I4)
9019 FORMAT('+',30X,'INCORRECT KPLOT INPUT, KPLOT = ',I4,' KPLOT MUST B
*E 7')
KONTRO=1
RETURN
END
```

```
.....
.....
.. -----
..
.. THE ROUTINE 'PROV1', WITH ENTRIES 'PROV2' AND 'PROV3'
.. HAS THE FOLLOWING PURPOSE.
.. 1.) TO ACCEPT FROM A FORMER CASE THE SCALAR FLUX
.. 2.) TO ACCEPT THE ANGULAR FLUX AT BOUNDARIES, WHERE
.. NO VACUUM BOUNDARY CONDITION IS GIVEN.
.. 3.) TO INTERPOLATE THESE GIVEN NUMBERS, IF
.. A) IMOLD NOT EQUAL IM
.. B) JMOLD NOT EQUAL JM
.. C) ISNOLD NOT EQUAL ISN
..
.. IT IS ALSO ASSUMED THAT ALL FLUXES 'XN', 'XNEII' AND 'XNEJJ' ARE
.. HOLD IN THE CORE STORAGE.
..
.. -----
.....
.....
..
```

```
SUBROUTINE PROV1(RXMAX,RYMAX,S,XN,RX,RY,IMOLD,JMOLD,IRIS,FG,MAI,MA
IJ,MZ,C,V,AFF,FX,IGP,IM,JM,IHM,IGM,MT,IXM,IYM,RHO,XKE)
COMMON NACOM,A(1000)
COMMON/FIX/NICOM,IA(300)
COMMON /ORG/IO(43),IE(20)
COMMON /SPEICH/MBOUND(25)

DIMENSION V(IM,IYM),AFF(IM,JM),FX(IGP),S(IRIS)
DIMENSION XKE(1)
DIMENSION FG(IM,JM),C(IHM,IGM,MT),MAI(IM),MAJ(JM),MZ(IXM,IYM)
DIMENSION RX(IXM),RY(IYM),XN(IM,JM,IGM)
```

```
EQUIVALENCE( ITH, IA(2))
EQUIVALENCE(ITAPE1, IA( 44)), (KWT, IA(78)),
1(KLUX, IA( 43)), (JFIN, IA( 46)), (ISN, IA(3)), (ID, IA(1)), (IBL, IA(13)),
2(IBR, IA(14)), (IBU, IA(15)), (IBO, IA(16)), (IHT, IA(48))
EQUIVALENCE(IA(89), KONTRO)
EQUIVALENCE(KRD, IA(79))
```

.....

4747 CONTINUE

```
NUMS=0
GO TO 201
ENTRY PROVO(IMOLD, JMOLD)
IF(ITAPE1.GT.0) GO TO 200
```

223 CALL PRINT3(16)

RETURN

200 IF(ITAPE1.GT.50.OR.ITAPE1.EQ.KWT.OR.ITAPE1.EQ.KRD)GO TO 223

```
REWIND ITAPE1
READ(ITAPE1,END=1000,ERR=1000)IDOLD,ICCOLD,
1 IMOLD, JMOLD, IGMOLD, IXMOLD, IYMO
ILD, ISNOLD, MMOLD, IBLOLD, IBROLD, IBUOLD, IBOOLD
```

RETURN

201 CONTINUE

```
KARATE=0
GABS=0.
IF(IGMOLD.NE.IGM) GO TO 100
IF(IXMOLD.NE.IXM) GO TO 100
IF(IYMOLD.NE.IYM) GO TO 100
IF(IBLOLD.NE.IBL)GO TO 100
IF(IBROLD.NE.IBR)GO TO 100
IF(IBUOLD.NE.IBU)GO TO 100
IF(IBOOLD.NE.IBO)GO TO 100
```

KARATE=1

```
READ(ITAPE1) (S(I),I=1,IXMOLD)
DO 101 I=1,IXM
GABS=ABS(RX(I)-S(I))
IF(GABS.GT.1.E-1) GO TO 100
```

101 CONTINUE

```
KARATE=2
READ(ITAPE1) (S(I),I=1,IYMOLD)
DO 102 I=1,IYM
GABS=ABS(RY(I)-S(I))
IF(GABS.GT.1.E-1) GO TO 100
```

102 CONTINUE

```
IBET=IMOLD*JMOLD
IALPH=IGM*IBET
KARATE=3
KLUX=1
MBOUND(4)=IALPH+IMOLD+JMOLD
MBOUND(5)=IBET +IMOLD+JMOLD
IF(IRIS.LT.MBOUND(4))KLUX=0
MBOUND(3)=KLUX
IF(IRIS.LT.MBOUND(5)) GO TO 100
```

KLUX = 1, SIMULTANEOUSLY READ IN THE OLD FLUXES.
KLUX = 0, READ IN THE OLD FLUXES FOR EACH GROUP SEPARATELY.

RETURN

```
ENTRY PROV2 (RXOLD,RYOLD,Q,IMOLD,JMOLD,RXNEU,RYNEU)
DIMENSION RXOLD(IMOLD),RYOLD(JMOLD),Q(IMOLD,JMOLD,IGM)
DIMENSION RXNEU(IM),RYNEU(JM)
READ(ITAPE1)(RXOLD(I),I=1,IMOLD)
READ(ITAPE1)(RYOLD(J),J=1,JMOLD)
READ(ITAPE1)A(2),RHO
IF(KLUX.EQ.0) GO TO 300
DO 301 IG=1,IGM
READ(ITAPE1)((Q(I,J,IG),I=1,IMOLD),J=1,JMOLD)
301 CONTINUE
300 CONTINUE
```

SPATIAL INTERPOLATION FOR 'XN'.

```
DO 400 IG=1,IGM
IF(KLUX.EQ.1) GO TO 401
I1=1
READ(ITAPE1)((Q(I,J,I1),I=1,IMOLD),J=1,JMOLD)
401 CONTINUE
DO 1 I1=1,IM
DO 1 J1=1,JM
X=RXNEU(I1)
Y=RYNEU(J1)
K=0
L=0
IF(RXMAX.LE.(2.*X))K=1
IF((2.*Y).GT.RYMAX)L=1
```

```
DO 2 M=1,IMOLD
IF(X.GT.RXOLD(M)) GO TO 2
IGXO=M
GO TO 3
2 CONTINUE
IGXO=IMOLD+1
3 DO 4 M=1,JMOLD
IF(Y.GT.RYOLD(M)) GO TO 4
IGYO=M
GO TO 5
4 CONTINUE
IGYO=JMOLD+1
5 CONTINUE
```

```
LIM=(IGXO-1)*(IGXO-IMOLD-1)*(IGYO-1)*(IGYO-JMOLD-1)
IF(LIM.EQ.0) GO TO 11
```

```
KL=K+L
IF(KL.NE.1) GO TO 8
```

U=(RYOLD(IGYO-1)-Y)*(RXOLD(IGXO)-RXOLD(IGXO-1))-(RYOLD(IGYO)-RYOLD
1(IGYO-1))*(RXOLD(IGXO)-X)

IXB=IGXO
IYB=IGYO-1
IXC=IXB-1
IYC=IGYO

7 IF(U.GT.0.) GO TO 6

IU1=IXB
IU2=IYB
IXB=IXC
IYB=IYC
IXC=IU1
IYC=IU2

6 IXA=IXC

IYA=IYB

GO TO 40

8 U=(RYOLD(IGYO)-RYOLD(IGYO-1))*(X-RXOLD(IGXO-1))-(Y-RYOLD(IGYO-1))*
1(RXOLD(IGXO)-RXOLD(IGXO-1))

IXB=IGXO-1
IYB=IGYO-1
IXC=IGXO
IYC=IGYO
GO TO 7

11 IT1=0

IT2=0

IF(IGXO.EQ.1)IT1=1

IF(IGXO.EQ.(IMOLD+1))IT1=2

IF(IGYO.EQ.1)IT2=3

IF(IGYO.EQ.(JMOLD+1))IT2=6

KU=IT1+IT2

IF(KU.LE.0.DR.KU.GT.8)WRITE(KWT,126)KU

126 FORMAT(10X,16HFEHLER IN PROV ,I2)

GO TO (21,22,23,24,25,26,27,28),KU

24 IXA=1

IYA=1

IXB=2

IYC=2

GO TO 30

27 IXA=1

IYA=JMOLD

IXB=2

IYC=JMOLD-1

GO TO 30

25 IXA=IMOLD

IYA=1

IXB=IMOLD-1

IYC=2


```
GO TO 30

28 IXA=IMOLD
   IYA=JMOLD
   IXB=IMOLD-1
   IYC=JMOLD-1
   GO TO 30

21 IXA=1
   IXB=2
32 IYA=IGYO
   IYC=IGYO-1
   IF(L.EQ.0) GO TO 30
   IU=IYA
   IYA=IYC
   IYC=IU
   GO TO 30
22 IXA=IMOLD
   IXB=IMOLD-1
   GO TO 32

23 IYA=1
   IYC=2
36 IXA=IGXO
   IXB=IGXO-1
   IF(K.EQ.0) GO TO 30
   IU=IXA
   IXA=IXB
   IXB=IU
   GO TO 30
26 IYA=JMOLD
   IYC=JMOLD-1
   GO TO 36
```

.....
.....

```
30 CONTINUE
40 CONTINUE
```

```
XA=RXOLD(IXA)
YA=RYOLD(IYA)
XB=RXOLD(IXB)
YC=RYOLD(IYC)
IF(KLUX.EQ.1) GO TO 405
F1=Q(IXA,IYA,1)
F2=Q(IXB,IYA,1)
F3=Q(IXA,IYC,1)
GO TO 404
405 F1=Q(IXA,IYA,IG)
    F2=Q(IXB,IYA,IG)
    F3=Q(IXA,IYC,IG)
404 CONTINUE
XZ1=(X-XA)/(XB-XA)
YZ1=(Y-YA)/(YC-YA)
IF(IMOLD.EQ.1)XZ1=0.
```

```
IF(JMOLD.EQ.1)YZ1=0.
U=F1+(F2-F1)*XZ1+(F3-F1)*YZ1
IF(U.LE.0.)U=0.
XN(IIII,JJJJ,IG)=U
1 CONTINUE
400 CONTINUE
WRITE(KWT,4731) IDOLD,ICCOLD,ITAPE1
CALL WRIT
4731 FORMAT( 30X,'FLUX FROM PREVIOUS CALCULATION'/30X,'PROBLEM WITH I
IDENTIFICATION NR. ',I6 /
1 30X,'FORMER OUTER ITERATION NR. ',I3/
1 30X,'ON UNIT NR. ',I3)
CALL PUT(FG,IM*JM,0.)
IF(IITH.GT.0)GO TO 280
DO 270 IG=1,IGM
DO 270 I=1,IM
DO 270 J=1,JM
I1=MAI(I)
I2=MAJ(J)
I3=MZ(I1,I2)
270 FG(I,J)=FG(I,J)+C(IHT-1,IG,I3)*XN(I,J,IG)
GO TO 282
280 E1=0.
DO 281 IG=1,IGM
DO 281 I=1,IM
DO 281 J=1,JM
281 FG(I,J)=FG(I,J)+XKE(IG)*XN(I,J,IG)
282 CONTINUE
CALL FISSN(FX,AFF,FG,V,XKE,IGP,IM,JM,C,MAI,MAJ,MZ,IXM,IYM,IHM,IGM,
IMT)
RETURN
100 WRITE(KWT,111)IGMOLD,IGM,IXMOLD,IXM,IYMOLD,IYM,JFIN,KARATE,IRIS
1,IMOLD,JMOLD
KONTRO=1
RETURN
111 FORMAT(30X,30HINCORRECT END IN PROV ,2X//30X,
1' IGMOLD IGM IXMOLD IXM IYMOLD IYM JFIN KARATE IRIS'
2//30X,12I7)
```

.....

ENTRY PROV3(ARRAY,ARQ,MM,QL,L1,RXYNEU,RXYOLD,L2)

.....
L1 IS IM OR JM
L2 IS IMOLD OR JMOLD

.....
DIMENSION ARRAY(L1,MM,IGM),ARQ(L2,MM,IGM),QL(MM),RXYNEU(L1),
IRXYOLD(L2)

ORGANIZATION

KARATE=6
IBET=MM*L2
IALPH=IBET*IGM
KLUX=1

```
MBOUND(8+3*NUMS)=IBET+MM+IMOLD+JMOLD
MBOUND(7+3*NUMS)=IALPH+MM+IMOLD+JMOLD
IF(IRIS.LT.MBOUND(7+3*NUMS))KLUX=0
MBOUND(6+3*NUMS)=KLUX
IF(IRIS.LT.MBOUND(8+3*NUMS))GO TO 100
```

.....

```
IF(KLUX.EQ.0)GO TO 1002
DO 1003 IG=1,IGM
1003 READ(ITAPE1)((ARQ(J,M,IG),J=1,L2),M=1,MMOLD)
1002 CONTINUE
```

```
DO 2000 IG=1,IGM
IF(KLUX.NE.0)GO TO 2002
I1=1
READ(ITAPE1)((ARQ(J,M,I1),J=1,L2),M=1,MMOLD)
2002 CONTINUE
```

SPATIAL INTERPOLATION FOR 'XNEII' AND 'XNEJJ'.

```
IANF=1
DO 2010 I=1,L1
Y=RXYNEU(I)
DO 2011 J=IANF,L2
IF(Y.GT.RXYOLD(J))GO TO 2011
IY=J
GO TO 2012
2011 CONTINUE
IY=L2+1
2012 CONTINUE
IANF=MAX0(1,IY-1)
```

```
KNUR=(IY-1)*(L2+1-IY)
IF(KNUR.GT.0)GO TO 2020
IF(IY.EQ.1)GO TO 2030
IYB=L2
GO TO 2040
2030 IYB=2
GO TO 2040
2020 IYB=IY
2040 CONTINUE
IYA=IYB-1
```

```
DO 2050 M=1,MMOLD
I1=IG
IF(KLUX.EQ.0)I1=1
ZZ1=(Y-RXYOLD(IYA))/(RXYOLD(IYB)-RXYOLD(IYA))
IF(L2.EQ.1)ZZ1=0.
U=ARQ(IYA,M,I1)+(ARQ(IYB,M,I1)-ARQ(IYA,M,I1))*ZZ1
IF(U.LE.0.)U=0.
ARRAY(I,M,IG)=U
2050 CONTINUE
2010 CONTINUE
```

2000 CONTINUE

ANGULAR INTERPOLATION FOR 'XNEII' AND 'XNEJJ'

IF(ISN.LT.ISNOLD) GO TO 2070
IF(ISN.EQ.ISNOLD)GO TO 2060
CALL POLMAN(ARRAY,L1,MM,IGM,ISN,ISNOLD,QL)

2060 NUMS=NUMS+1
RETURN
2070 WRITE(KWT,2071)
2071 FORMAT(30X,'AT THIS TIME IMPOSSIBLE TO INTERPOLATE CASES WHERE ISN
1.LT.ISNOLD')
KONTRQ=1
RETURN
1000 CALL PRINT3(13)
RETURN
END

*****'PROV4' INITIATES FLUXES PASSED OVER FROM OTHER GEOMETRY*****

SUBROUTINE PROV4(XN,XNEII,IM,JM,MM,IGM,F,FG,C,XKE,MAI,MAJ,MZ,FX,V,
1 IHM,MT,IXM,IYM,IGP)
DIMENSION XN(IM,JM,IGM),XNEII(JM,MM,IGM)
DIMENSION F(IM,JM),FG(IM,JM),C(IHM,IGM,MT),XKE(1),MAI(IM),MAJ(JM),
1FX(1),V(IM,IYM),MZ(IXM,IYM)
EQUIVALENCE(JFIN,IA(46)),(JBKTP,IA(83)),(IRIS,IA(42)),(KWT,IA(78))
EQUIVALENCE(NFN,IA(60))
EQUIVALENCE(IHT,IA(48))
EQUIVALENCE(ITH,IA(2))
NFN=1
COMMON NACOM,A(100)
COMMON/FIX/NICOM,IA(300)

I2=JFIN+IM*IGM-1
IF(I2.GT.NACOM)GO TO 1
READ(JBKTP)(A(I),I=JFIN,I2)
DO 2 IG=1,IGM
DO 2 I=1,IM
U=A(JFIN-1+I+(IG-1)*IM)
DO 2 J=1,JM
2 XN(I,J,IG)=U

I2=JFIN+MM*IGM-1
IF(I2.GT.NACOM)GO TO 1
READ(JBKTP)(A(I),I=JFIN,I2)
DO 3 IG=1,IGM
DO 3 M=1,MM
U=A(JFIN-1+M+(IG-1)*MM)
DO 3 J=1,JM
3 XNEII(J,M,IG)=U

```
CALL PUT(FG,IM*JM,0.)
IF(IHT.GT.0)GO TO 280
DO 270 IG=1,IGM
DO 270 I=1,IM
DO 270 J=1,JM
I1=MAI(I)
I2=MAJ(J)
I3=MZ(I1,I2)
I3=IABS(I3)
270 FG(I,J)=FG(I,J)+C(IHT-1,IG,I3)*XN(I,J,IG)
GO TO 282
280 E1=0.
DO 281 IG=1,IGM
DO 281 I=1,IM
DO 281 J=1,JM
281 FG(I,J)=FG(I,J)+XKE(IG)*XN(I,J,IG)
282 CONTINUE
CALL FISSN(FX,F ,FG ,V,XKE,IGP,IM,JM,C,MAI,MAJ,MZ,IXM,IYM,IHM,IGM,
1MT)
```

```
5 RETURN
1 WRITE(KWT,4)
4 FORMAT('0',30X,'IMPOSSIBLE TO PASS FLUXES IN PROV4')
GO TO 5
END
```

```
SUBROUTINE REMUS
RETURN
END
```

```
SUBROUTINE RICH2(DU,DE,W,MRM,MM,MRMY)
DIMENSION DU(MM),DE(MM),W(MM),MRM(MM),MRMY(MM)
DO 1 I=1,6
1 W(I)=0.25
W(1)=0.0
W(4)=0.0
DO 2 I=1,3
J=I+3
DE(I)=-0.5773503
2 DE(J)=0.5773503
DU(1)=-0.8164946
DU(2)=-0.5773503
DU(3)=0.5773503
DU(4)=-0.8164966
DU(5)=-0.5773503
DU(6)=0.5773503
MRM(1)=3
MRM(2)=3
MRM(3)=2
MRM(4)=6
MRM(5)=6
MRM(6)=5
DO 3 I=1,3
J=I+3
MRMY(I)=J
3 MRMY(J)=I
```

RETURN
END

```
SUBROUTINE RICH4(DU,DE,W,MRM,MM,MRMY)
DIMENSION DU(MM),DE(MM),W(MM),MRM(MM),MRMY(MM),Z(16),NK(16)
DATA NK /5,5,4,3,2,8,8,7,13,13,12,11,10,16,16,15/
DATA Z / -0.942809,-0.8819171,-0.3333333,0.3333333,0.8819171,
1-0.4714045,-0.3333333,0.3333333,-0.942809,-0.8819171,-0.3333333,
20.3333333,0.8819171,-0.4714045,-0.3333333,0.3333333/
DO 5 M=1,16
5 DU(M)=Z(M)
DO 7 M=1,MM
7 MRM(M)=NK(M)
DO 10 J=1,5
10 DE(J)=-0.3333333
DO 11 J=6,8
11 DE(J)=-0.8819171
DO 12 J=9,13
12 DE(J)= 0.3333333
DO 13 J=14,16
13 DE(J)=0.8819171
DO 14 J=1,8
I=J+8
MRMY(I)=J
14 MRMY(J)=I
DO 1 I=1,MM
1 W(I)=0.0833333
W(1)=0.0
W(6)=0.0
W(9)=0.0
W(14)=0.0
RETURN
END
```

```
SUBROUTINE RICH6(DU,DE,W,MRM,MM,MRMY)
DIMENSION DU(MM),DE(MM),W(MM),MRM(MM),MRMY(MM),Z(60),NK(30)
DATA Z /0.0,0.040215,0.043118,0.043118,2*0.0402153,0.0431180,0.0402153,
10.0,4*0.043118,0.0,2*0.0402153,0.0,0.0402153,0.043118,0.0402153,
20.0402153,0.043118,0.0402153,0.0,4*0.043118,0.0,2*0.0402153,
3-0.9660918,-0.9309493,-0.68313,-0.2581989,0.2581989,0.68313,
40.9309493,-0.7302967,-0.68313,-0.2581989,0.2581989,0.68313,
5-0.3651484,-0.2581989,0.2581989,-0.966092,-0.9309493,-0.68313,
6-0.2581989,0.2581989,0.68313,0.9309493,-0.7302967,-0.68313,
7-0.2581989,0.2581989,0.68313,-0.3651484,-0.2581989,0.2581989/
DATA NK /7,7,6,5,4,3,2,12,12,11,10,9,15,15,14,22,22,21,20,19,18,
117,27,27,26,25,24,30,30,29/
DO 45 I=1,30
W(I)=Z(I)
J=I+30
DU(I)=Z(J)
45 MRM(I)=NK(I)
DO 46 I=1,15
J=I+15
MRMY(I)=J
46 MRMY(J)=I
DO 47 I=1,7
```

```
J=I+15
DE(J)=+0.2581989
47 DE(I)=-0.2581989
DO 48 I=8,12
J=I+15
DE(J)=0.68313
48 DE(I)=-0.68313
DO 49 I=13,15
J=I+15
DE(J)=0.9309493
49 DE(I)=-0.9309493
RETURN
END
```

```
SUBROUTINE RICH8(DU,DE,W,MRM,MM,MRMY)
DIMENSION DU(MM),DE(MM),W(MM),MRM(MM),MRMY(MM),Z(48),NK(48)
DATA Z / -0.9759,-0.9511897,-0.7867958,-0.5773503,-0.2182179,
10.2182179,0.5773503,0.7867958,0.9511897,-0.8164966,-0.7867958,
2-0.5773503,-0.2182179,0.2182179,0.5773503,0.7867958,-0.6172134,
3-0.5773503,-0.2182179,0.2182179,0.5773503,-0.3086067,-0.2182179,
40.2182179,-0.9759,-0.9511898,-0.7867958,-0.5773503,-0.2182179,
50.2182179,0.5773503,0.7867958,0.9511898,-0.8164966,-0.7867958,
6-0.5773503,-0.2182179,0.2182179,0.5773503,0.7867958,-0.6172134,
7-0.5773503,-0.2182179,0.2182179,0.5773503,-0.3086067,-0.2182179,
80.2182179 /
DATA NK /9,9,8,7,6,5,4,3,2,16,16,15,14,13,12,11,21,21,20,19,18,
124,24,23,33,33,32,31,30,29,28,27,26,40,40,39,38,37,36,35,
245,45,44,43,42,48,48,47 /
DO 1 I=1,48
1 MRM(I)=NK(I)
DO 2 I=1,48
2 DU(I)=Z(I)
DO 3 I=1,24
J=I+24
MRMY(I)=J
3 MRMY(J)=I
DO 4 I=1,9
J=I+24
DE(I)=-0.2182179
4 DE(J)=0.2182179
DO 5 I=10,16
J=I+24
DE(I)=-0.5773503
5 DE(J)=0.5773503
DO 6 I=17,21
J=I+24
DE(I)=-0.7867958
6 DE(J)=0.7867958
DO 7 I=22,24
J=I+24
DE(I)=-0.9511897
7 DE(J)=0.9511897
DO 8 I=1,48
8 W(I)=0.0252932
W(1)=0.0
W(2)=0.0266502
W(5)=0.0266502
```

W(6)=0.0266502
W(9)=0.0266502
W(10)=0.0
W(12)=0.0182904
W(15)=0.0182904
W(17)=0.0
W(22)=0.0
W(23)=0.0266502
W(24)=0.0266502
W(25)=0.0
W(26)=0.0266502
W(29)=0.0266502
W(30)=0.0266502
W(33)=0.0266502
W(34)=0.0
W(36)=0.0182904
W(39)=0.0182904
W(41)=0.0
W(46)=0.0
W(47)=0.0266502
W(48)=0.0266502
RETURN
END

SUBROUTINE SECRET
COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
COMMON/ORG/IO(44),IE(20)
COMMON/TSCH/ITSCH,RHO,ITP1,ITP2
COMMON/CONTR/IREBAL,IDIM(26),FUNK(26)
EQUIVALENCE(KRD,IA(79))
INTEGER*4 I1/0/,I2/1/,I3/43/,I4/1/,I5/15/,KWT/6/
11 FORMAT(1X,'-----'
1-----'
2-----')
IEND=0
WRITE(KWT,11)
READ(KRD,1)I1,I2,I3,I4,I5
1 FORMAT(5I6)
GO TO 40

ENTRY SECUR
IEND=1
GO TO 40
ENTRY SAFETY
IEND=0
40 CONTINUE
IO(44)=IA(46)+1
KAD=MINO(2000,IA(42))
IO40=IO(39)+KAD
WRITE(KWT,11)
WRITE(KWT,11)
15 FORMAT(// 5(' A(',I2,') = ',G14.6,' *'))
31 FORMAT(///7(' IE(',I2,') = ',I7,' *'))


```
30 FORMAT( //7(' IO(' ,I2,' ) =',I7,' *') )
17 FORMAT(// 7(' IA(' ,I3,' ) =',I6,' *') )
  WRITE(KWT,17)(I,IA(I),I=1,120)
  WRITE(KWT,11)
  WRITE(KWT,15)(I,A(I),I=1,20)
  WRITE(KWT,11)
  WRITE(KWT,16)ITSCH,IREBAL,ITP1,ITP2,RHO
16 FORMAT(// ' ITSCH = ',I2,' IREBAL = ',I2,' ITP1 = ',I3,' ITP2
  1 = ',I3,' RHO = ',G15.8)
  WRITE(KWT,11)
  WRITE(KWT,30)(I,IO(I),I=1,43)
  WRITE(KWT,31)(I,IE(I),I=1,16)
  WRITE(KWT,11)
  WRITE(KWT,11)
  IF(I1)2,2,3
  2 K=I2
12 IF(K.GT.I3)GO TO 13
  KL=K
  N1=IO(K)
  K=K+1
  N2=IO(K)-1
  WRITE(KWT,11)
  WRITE(KWT,6)KL
  WRITE(KWT,4)(A(J),J=N1,N2)
  GO TO 12
  4 FORMAT(1X,10E12.5)
13 IF(I1.LT.0)GO TO 10
  3 K=I4
23 IF(K.GT.I5)GO TO 10
  KL=K
  N1=IE(K)
  K=K+1
  N2=IE(K)-1
  WRITE(KWT,11)
  WRITE(KWT,20)KL
  WRITE(KWT,5)(IA(J),J=N1,N2)
  GO TO 23
10 WRITE(KWT,11)
  5 FORMAT(1X,10I6)
  WRITE(KWT,11)
  CALL PLACE
  IF(IEND.EQ.1)STOP
  RETURN
  6 FORMAT(1H0,20X,' ARRAY A(IO(' ,I3,' ))'//)
20 FORMAT(1H0,20X,' ARRAY IA(IE(' ,I3,' ))'//)
  END
```

SUBROUTINE TEMP1 (KLAR, TSFAKT, TMOB, ITKENN, BIM, BAM)

THE ROUTINE "TEMP1" WITH ENTIES "TEMP2" AND "TEMP3" PERFORMS AND CONTROLS TCHEBYCHEFF EXTRAPOLATION OF THE INNER ITERATIONS.

TEMP1 : INITIATION, GLOBAL CONTROL.
TEMP2 : CONTROL IF GROUP IG BECOMES ACTIVE.
TEMP3 : PERFORMS DATA HANDLING AND EXTRAPOLATION.

COMMON/FIX/NICOM, IA(300)
COMMON NACOM, A(1000)
COMMON/SPEICH/MBOUND(25)
COMMON/TSCH/ITSCH

DIMENSION KLAR(1), ITKENN(1), TMOB(1), TSFAKT(1), BIM(1), BAM(1)

EQUIVALENCE(JFIN, IA(46)), (IG, IA(72)), (KWT, IA(78)), (MBOUND(13), IMAX
1), (MBOUND(14), KENN), (MBOUND(12), ITOTL), (IGM, IA(7)), (IRIS, IA(42)), (2
INNTS, MBOUND(18)), (IKENN, MBOUND(19)), (IPOSS, MBOUND(20)),
3 (IM, IA(71)), (JM, IA(70)), (A(18), TEPS), (MBOUND(21), LENTS)
EQUIVALENCE(IA(62), MM), (MREAL, MBOUND(22))
EQUIVALENCE(IA(76), IIC), (IA(61), ICVT)
EQUIVALENCE(A(16), RLOG)
EQUIVALENCE(EPS, A(1))

IF(INNTS.EQ.0)RETURN
IF(TEPS.EQ.0.)TEPS=0.006
LENTS=IM*JM+(IM+JM)*MM
IRSMAX=2*LENTS
MREAL=0
IKENN=0
IPOSS=IGM
GUK=0.1

ITKENN(IG) = 0 NOCH NICHTS GEFUNDEN

ITKENN(IG).GT.IGM NICHT BENUTZT ODER AUSGESCHALTET
 = 200 + IG E3 ZU KLEIN
 = 300 + .. RHO.LE.0.1 ODER RHO.GT.0.999
 = 400 + .. DURCH EINGABE AUSGESCHALTET
 = 600 + .. ZU SCHLECHT, AUSSER BETRIEB
 = 700 + .. WIE 3000

2 KONK=ITOTL
IF(KENN.EQ.0)KONK=IMAX

```
IF(ITSCH.EQ.0)KONK=0
IF(KONK+IRSMAX-IRIS)10,10,4
4 IF(ITSCH.EQ.0)GO TO 9
IF(KENN)8,8,7
7 KENN=0
GO TO 2
8 IF(IRSMAX.GT.IRIS)GO TO 9
IKENN=1
KONK=0
GO TO 10
9 INNTS=0
RETURN
10 JFIRST=JFIN+KONK
CALL PUT(A(JFIRST),NACOM-JFIRST+1,-1.E+20)
RETURN
```

```
ENTRY TEMP2(E3,IFIRST,I2,I3)
```

```
CALL TEMP2(E3,IFIRST,IZWEI,IDREI)
LCG=IIC-2
KENKO=MOD(LCG,2)
IF(ICVT.EQ.2)RETURN
IF(INNTS.NE.-1)GO TO 11
IF(LCG.NE.3)GO TO 11
IF(ICC.GT.0)GO TO 11
U=TSFAKT(IG)
IF(U.GT.0.999.OR.U.LE.0.1)GO TO 111
ITKENN(IG)=- (MREAL+1)
MREAL=MREAL+1
GO TO 11
111 IF(U.LE.0.)ITKENN(IG)=400+IG
11 CONTINUE
IF(IABS(ITKENN(IG)).GT.IPOSS)RETURN
12 IF(ITKENN(IG))15,13,15
13 IF(KLAR(IG)-8)40,14,16
14 BIM(IG)=E3
BAM(IG)=E3
15 IFIRST=JFIRST+KENKO*LENTS
I2=IFIRST+IM*JM
I3=I2+IM*MM
EQUIVALENCE(IA(69),ICC)
IF(LCG.LE.4)GO TO 40
IF(E3.GT.0.01.AND.ICC.GT.4)ITKENN(IG)=600+IG
40 RETURN

16 IF(BIM(IG).LT.(RLOG*EPS))GO TO 105
U=(E3/BIM(IG))**(1./(KLAR(IG)-8))
IF(ABS(U-BAM(IG))-TEPS)18,18,17
17 BAM(IG)=U
RETURN
18 TSFAKT(IG)=U
200 TMOB(IG)=2.
TSFAKT(IG)=U*(1.-BAM(IG))/(1.-U)
201 IF(TSFAKT(IG).LT.0.93)
1TSFAKT(IG)=TSFAKT(IG)*(1.+3.*TEPS)
IF(TSFAKT(IG).LT.GUK.OR.TSFAKT(IG).GT.0.999)GO TO 102
```

```
ITKENN(IG)=- (MREAL+1)
MREAL=MREAL+1
GO TO 15

ENTRY TEMP3(X,Y,Z,D,E,F)

CALL TEMP3(XSK,XNEJ,XNEI,A(IFIRST),A(IZWEI),A(IDREI))

DIMENSION X(1),Y(1),Z(1),D(1),E(1),F(1)
IF(ICVT.EQ.2)RETURN
IF(IABS(ITKENN(IG)).GT.IPOSS)RETURN
IF(ITKENN(IG).EQ.0)RETURN
IF(LCG.GT.0)GO TO 800
ITKENN(IG)=-IABS(ITKENN(IG))
J1=IFIRST+(1-2*KENKO)*LENTS+1
A(J1)=-1.E+20
TMOB(IG)=2.
800 CONTINUE
J1=IM*JM
J2=IM*MM
J3=JM*MM
IF(ITKENN(IG))23,28,19
19 W=TMOB(IG)
DO 20 I=1,J1
X(I)=W*X(I)+D(I)*(1.-W)
20 CONTINUE
DO 21 I=1,J2
Y(I)=Y(I)*W+(1.-W)*E(I)
21 CONTINUE
DO 22 I=1,J3
Z(I)=Z(I)*W+(1.-W)*F(I)
22 CONTINUE
TMOB(IG)=1./(1.-(W*TSFAKT(IG)**2/4.))
23 DO 24 I=1,J1
24 D(I)=X(I)
DO 25 I=1,J2
25 E(I)=Y(I)
DO 26 I=1,J3
26 F(I)=Z(I)
IF(ITKENN(IG))27,28,28
27 J1=IFIRST+(1-2*KENKO)*LENTS+1
27 J1=IFIRST+(1-2*KENKO)*IPOSS*LENTS+1
IF(A(J1).GE.0.)ITKENN(IG)=-ITKENN(IG)
28 RETURN
100 WRITE(KWT,101)IG
CALL WRIT
101 FORMAT(30X,'UNDETERMINABLE ERROR IN TEMP',2X,I3)
RETURN
102 ITKENN(IG)=IPOSS+300
104 RETURN
105 ITKENN(IG)=IPOSS+200
GO TO 104
*****
*****
END
```

```

SUBROUTINE TOMBO(RAY,
1          XN,XNEII,IM,JM,MM,IGM,IZT,BUFF,IRIS)
DIMENSION RAY(1),XN(IM,JM,IGM),XNEII(JM,MM,IGM),BUFF(IGM,IM)
COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
EQUIVALENCE(KWT,IA(78)),(JOUT,IA(84))
REWIND JOUT
IF(IZT.EQ.1.OR.IZT.EQ.JM)GO TO 10
Z1=RAY(IZT+1)
Z2=RAY(IZT-1)
Z=RAY(IZT)
G1=Z1-Z2
G2=Z1-Z
G3=Z-Z2
GM=2./(G1*G2*G3)
DO 1 IG=1,IGM
DO 1 I=1,IM
U=XN(I,IZT,IG)
IF(U.LE.0.)GO TO 2
BUFF(IG,I)=(GM/U)*(XN(I,IZT+1,IG)*G3-U*G1+XN(I,IZT-1,IG)*G2)
GO TO 1
2 BUFF(IG,I)=0.
1 CONTINUE
WRITE(KWT,3) IZT,JOUT
3 FORMAT(///30X,'BUCKLING OF THE J-TRAVERSE NR.',I4 /30X,'WRITTEN ON
1 UNIT NR. ',I3)
DO 4 IG=1,IGM
4 WRITE(KWT,5) IG,(BUFF(IG,I),I=1,IM)
5 FORMAT(///30X,'BUCKLING IN GROUP ',I4//((2X, 10(1X,E11.3)))
WRITE(JOUT)((BUFF(IG,I),IG=1,IGM),I=1,IM)
WRITE(JOUT)((XN(I,IZT,IG),I=1,IM),IG=1,IGM)
WRITE(JOUT)((XNEII(IZT,M,IG),M=1,MM),IG=1,IGM)
RETURN
10 WRITE(KWT,11)
11 FORMAT(///30X,'UNPERMITTED BUCKLING OUTPUT OPTION USED')
RETURN
END
```

'TOPROV' WRITES SOME IDENTIFYING NUMBERS, THE ZONE- AND THE INTERVAL-RADII,KEFF AND THE LATEST REDUCTION FACTOR, THE SCALAR FLUX AND THE ANGULAR FLUX AT THE BOUNDARIES,IF NECESSARY, ON UNIT NR. 'ITP3'.

IT IS ASSUMED THAT THE ARRAYS 'XN','XNEII' AND 'XNEJJ' HAVE BEEN HOLD COMPLETELY IN THE CORE STORAGE.

SUBROUTINE TOPROV(XN,XNEII,XNEJJ,RAX,RAY,S,IRIS,MM,IP,JP,IGM,IXM,I
IYM,IM,JM,ROM,ZRAX,ZRAY)

COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
COMMON/ORG/IO(43),IE(20)
DIMENSION XN(IM,JM,IGM),XNEII(JM,MM,IGM),XNEJJ(IM,MM,IGM),S(IRIS)
DIMENSION ZRAX(IXM),ZRAY(IYM),RAX(IP),RAY(JP)

EQUIVALENCE(JFIN,IA(46)),(ITP3,IA(45)),(ID,IA(1)),
1 (KWT,IA(78)),(KTP1,IA(68)),(ISN,IA(3)),(IBL,IA(13))
1), (IBR,IA(14)),(IBU,IA(15)),(IBO,IA(16))
EQUIVALENCE(ICC,IA(69))
EQUIVALENCE(IA(89),KONTRO)
EQUIVALENCE(KWT,IA(78))
EQUIVALENCE(KRD,IA(79))

GO TO 7
ENTRY TOP
IF(ITP3.GT.50.OR.ITP3.EQ.KWT.OR.ITP3.EQ.KWT)GO TO 20
RETURN

7 REWIND ITP3
WRITE(ITP3,ERR=20,END=20)ID,ICC,
1 IM,JM,IGM,IXM,IYM,ISN,MM,IBL,IBR,IBU,I
1BO

.....
ZONE RADII X AND Y-DIRECTION

WRITE(ITP3)(ZRAX(I),I=1,IXM)

WRITE(ITP3)(ZRAY(I),I=1,IYM)

.....
INTERVAL-RADII

WRITE(ITP3)(RAX(I),I=1,IM)

WRITE(ITP3)(RAY(I),I=1,JM)

.....
WRITE(ITP3)A(2),ROM
IF(KTP1.EQ.0)GO TO 4
WRITE(KWT,8)

8 FORMAT(/30X,'AT THIS TIME NOT POSSIBLE TO WRITE OUT FLUXES'/30X,
1 ' ON TAPE,IF ALL FLUXES HAVE NOT BEEN STORED IN THE'/30X,
2 'CORE STORAGE')
ITP3=0
GO TO 6

.....
4 DO 5 IG=1,IGM
5 WRITE(ITP3)((XN(I,J,IG),I=1,IM),J=1,JM)

.....
ANGULAR FLUXES AT THE BOUNDARY

```
IF((IBL+IBR).LE.0.OR.IBR.EQ.4)GO TO 10
DO 1 IG=1,IGM
1 WRITE(ITP3)((XNEII(J,M,IG),J=1,JM),M=1,MM)
10 CONTINUE
IF((IBO+IBU).LE.0.OR.IBO.EQ.4)GO TO 11
DO 2 IG=1,IGM
2 WRITE(ITP3)((XNEJJ(I,M,IG),I=1,IM),M=1,MM)
11 CONTINUE
GO TO 6
20 CALL FSPIE
KONTRO=1
WRITE(KWT,9)ITP3
9 FORMAT(20X,'***** THE PROGRAM WILL NOT BE ABLE TO STORE FLUSES O
IN TAPE NR. ',I5,' *****')
ITP3=0
6 CONTINUE
RETURN
-----
END
```

'TSCHEB' PERFORMS TSCHBYCHEV EXTRAPOLATION FOR THE OUTER ITERATIONS

SUBROUTINE TSCHEB(IGM,IM,JM,MM,ICC,BUFF,ITM)

INTEGER TAPE1,TAPE2

```
COMMON/TSCH/ITSCH,RHO,ITP1,ITP2
COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
COMMON NACOM,A(1)
COMMON/FIX/NICOM,IA(1)
COMMON/ORO/IO(43),IE(20)
COMMON /SPEICH/MBOUND(25)
EQUIVALENCE(MBOUND(17),IBUT)
EQUIVALENCE(MBOUND(13),IMAX),(MBOUND(14),KENN),(MBOUND(12),ITOTL)
DIMENSION BUFF(ITM)
DIMENSION LENGTH(5),IBEGIN(5)
EQUIVALENCE
1 (IA(13),IBL),(IA(14),IBR),(IA(15),IBU),(IA(16),IBO)
EQUIVALENCE(XLA,A(13)),(XLAR,A(14)),(KWT,IA(78)),
2(JFIN,IA(46)),(ITP1,TAPE1),(ITP2,TAPE2)
EQUIVALENCE(JF,IO(27)),(JFG,IG(28)),
1 (IO(29),JXARI),(IO(30),JFX),(IO(31),JB),(IO(32),JZEPS),
7(IO(33),JZRAX),(IO(34),JZRAY),(IO(36),JXNEII),(IO(37),JXNEJJ),
8(IO(38),JXN),(IO(35),JQ)
EQUIVALENCE(KRD,IA(79))
```

```
GO TO 170
ENTRY TSCHE1( ITM, IGM, IM, JM, MM)
NLEN=5
```

```
LENGTH(1)=IM*JM*IGM
LENGTH(2)=IM*JM
LENGTH(3)=IM*JM
LENGTH(4)=JM*MM*IGM
LENGTH(5)=IM*MM*IGM
IF((IBO+IBU).LE.0)LENGTH(5)=1
IF((IBR+IBL).LE.0)LENGTH(4)=1
```

```
IBEGIN(1)=JXN
IBEGIN(2)=JF
IBEGIN(3)=JFG
IBEGIN(4)=JXNEII
IBEGIN(5)=JXNEJJ
```

```
KENN=0
ITOTL=0
IMAX=MAXO(LENGTH(1),LENGTH(2),LENGTH(3),LENGTH(4),LENGTH(5))
DO 4 I=1,NLEN
4 ITOTL=ITOTL+LENGTH(I)
LHALF=ITOTL
ITOTL=ITOTL*2
IF(ITSCH.EQ.0)RETURN
IF(ITOTL.LE.ITM)KENN=1
IF(IMAX.GT.ITM)GO TO 10
IF(KENN.NE.0)GO TO 12
IF(ITP1.LE.0.OR.ITP2.LE.0)GO TO 10
IF(ITP1.EQ.ITP2)GO TO 10
IF(ITP1.GT.50.OR.ITP1.EQ.KWT.OR.ITP1.EQ.KRD)GO TO 10
IF(ITP2.GT.50.OR.ITP2.EQ.KWT.OR.ITP2.EQ.KRD)GO TO 10
WRITE(ITP1,ERR=13,END=13)RHO
WRITE(ITP2,ERR=13,END=13)RHO
12 RETURN
13 WRITE(KWT,14)
14 FORMAT(20X,'***** THERE ARE PERHAPS INVALID OR NO DD STATEMENTS
1 FOR ITP1 AND/OR ITP2 *****')
10 CALL WRIT
WRITE(KWT,11)ITSCH, ITOTL, IMAX, ITM, RHO, ITP1, ITP2
CALL WRIT
11 FORMAT(30X,'***** IMPOSSIBLE TO PERFORM TSCHEB. EXTRAPOLATION F
1 OR OUTER *****'/30X,' ITSCH = ',I1,
1 'ITOTL = ',I6,' IMAX = ',I6,' ITM = ',I6,
12X/30X,' RHO = ',
1 E14.4,' ITP1 = ',I6,' ITP2 = ',I6)
ITSCH=0
RETURN
170 IF(ITSCH.EQ.0.AND.ICC.NE.0)RETURN
IF(ICC-1)1,2,3
1 RETURN
2 IBIS=1
```



```
W=2.
IBUT=-1
RO=-0.1
ROALT=18.
IWR=TAPE1
JTCOM=JFIN
CALL WRITER(IWR, IBEGIN, LENGTH, NLEN, KENN, JFIN, JTCOM)
RETURN
```

```
3 IF(IBIS.NE.1)GO TO 7
  IBIS=2
  IWR=TAPE2
  JTCOM=JFIN +LHALF
  CALL WRITER(IWR, IBEGIN, LENGTH, NLEN, KENN, JFIN, JTCOM)
  IWR=TAPE1
  RETURN
```

```
7 IBIS=IBIS+1
  IF(IBUT.GT.0) GO TO 51
  IF(RHO.NE.0.) GO TO 52
  ROUR=ROALT
  ROALT=RO
  IF(ABS(1.-XLAR).GT.1.E-8)RO=(1.-XLA)/(1.-XLAR)
  EPO=1.
  IF(RO.GT.0.1)EPO=ABS((RO-ROALT)/RO)
  IF(EPO.GT.0.1.OR.ICC.LT.4) GO TO 63
  IF(RO.LE.0) GO TO 63
  RHO=RO*1.04
  IF(RO.GT.0.92)RHO=RO
  IF(RHO.GT.0.99 )GO TO 10
```

```
52 CONTINUE
  CALL WRIT
  WRITE(KWT,126)ICC,RHO,KENN
  CALL WRIT
```

```
126 FORMAT(//30X,'TSCHEB. EXTRPOLATION OF THE OUTER ITERATIONS'//
  130X,'FROM THE',I3,'. ITERATION WITH RHO = ',E15.8,3X,I2//)
```

.....
.....

```
51 IBUT=1
  W2=1.-W
  JTCOM=JFIN -1
  IF(IWR.EQ.TAPE2)JTCOM=JFIN+LHALF-1
  DO 78 IS=1,5
  ILEN=LENGTH(IS)
  ICOM=IBEGIN(IS)-1
  IF(KENN.EQ.1)GO TO 79
  JTCOM=JFIN-1
  CALL READER(IWR,ILEN,BUFF)
79 CONTINUE
  DO 77 IND1=1,ILEN
  U=W*A(ICOM+IND1)+W2*A(JTCOM+IND1)
  IF(U.LE.0.)U=0.
  A(ICOM+IND1)=U
77 CONTINUE
  JTCOM=JTCOM +ILEN
78 CONTINUE
  W=1./(1.-(RHO*RHO*W/4.))
```

```
63 JTCOM=JFIN
   IF(IWR.EQ.TAPE2)JTCOM=JFIN +LHALF
   CALL WRITER(IWR,IBEGIN,LENGTH,NLEN,KENN,JFIN,JTCOM)
   IF(IWR.EQ.TAPE1) GO TO 70
   IWR=TAPE1
   GO TO 71
70 IWR=TAPE2
71 RETURN
```

.....

END

```
      SUBROUTINE ANISO(DE,DU,W,T,TW,IS,ISCT,MM,CO,SI)
      DIMENSION DE(MM),DU(MM),T(IS,IS,MM)
      DIMENSION CO(ISCT),SI(ISCT),W(MM),TW(IS,IS,MM)
      M=1
100  E1=DE(M)*DE(M)+DU(M)*DU(M)
      E2=SQRT(1.-E1)
      E3=SQRT(E1)
      T(1,1,M)=1.
      T(2,1,M)=E2
      M1=IS-1
      DO 10 N=1,M1
      V=N
      T(N,N+1,M)=0.0
10  T(N+1,N+1,M)=(2.*V-1.)*E3*T(N,N,M)
      M1=IS-2
      IF(M1.EQ.0) GOTO 105
      DO 20 N=1,M1
      H1=N
      H2=2.*H1+1.
      H3=H1+1.
      H4=H2*E2*T(N+1,1,M)-H1*T(N,1,M)
20  T(N+2,1,M)=H4/H3
      M1=IS-2
      DO 30 J=1,M1
      DO 30 N=J,M1
      H1=N
      H2=2.*H1+1.
      H3=J
      V=H1-H3+1.
      H4=H2*E2*T(N+1,J+1,M)-(H1+H3)*T(N,J+1,M)
30  T(N+2,J+1,M)=H4/V
105 CONTINUE
      M=M+1
      IF(M.LE.MM) GOTO 100
```

21-2,3,4: BERECHNUNG VON T UND TW

```
      M=1
200  E3=SQRT(DE(M)*DE(M)+DU(M)*DU(M))
      E1=DU(M)/E3
      E2=DE(M)/E3
      SI(1)=E1
      CO(1)=E2
      M1=IS-1
```

```
IF(M1.LT.2) GOTO 104
DO 310 J=2,M1
CO(J)=CO(J-1)*E2-SI(J-1)*E1
310 SI(J)=SI(J-1)*E2+CO(J-1)*E1
104 CONTINUE
DO 320 N=1,M1
K=N+1
DO 320 J=K,IS
320 T(N,J,M)=T(J,N+1,M)*SI(N)
DO 340 N=2,IS
DO 340 J=2,N
340 T(N,J,M)=T(N,J,M)*CO(J-1)
DO 1200 I=1,IS
DO 1200 J=1,IS
I1=I+J
V=I1
V=V/2.
I2=V
I2=I2*2
IF(I1.EQ.I2.AND.J.GT.I) GOTO 1100
IF(I1.NE.I2.AND.J.LE.I) GOTO 1100
GOTO 1200
1100 T(I,J,M)=0.
1200 CONTINUE
DO 350 N=1,IS
DO 350 J=1,IS
350 TW(N,J,M)=T(N,J,M)*W(M)
M=M+1
IF(M.LE.MM) GOTO 200

RETURN
END
```

```
SUBROUTINE WRIT
COMMON/FIX/NICOM,IA(300)
COMMON NACOM,A(1000)
EQUIVALENCE(KWT,IA(78))
WRITE(KWT,1)
1 FORMAT( /17X,'**',85('-'),'**'/)
RETURN
END
```

```
QUERSCHNITTS - ORGANISATION
SUBROUTINE WQORG(LDIM,S,LS,NFL,*)
COMMON KN(90)
DIMENSION S(1),LS(1),LX(250),NTYP(1)
EQUIVALENCE (KN(3),ICR),(KN(32),NG)
CALL FSPIE
REWIND NFL
ADJUSTED STATEMENT
```

```
READ(NFL)

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

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+I+I)-LS(LPGA1+I)
GO TO 12
11 ND=LS(2)-LS(LPGA1+I)
12 LX(I)=ND
13 MX=MAX0(MX,ND)
LDA=LS(1)+1+MX
NADI=LS(1)+1
DO 14 I=1,NGP
K=NGP+1-I
LDA=LDA+LX(K)
IF(LCA.GT.LDIM) GO TO 15
14 CONTINUE
GO TO 19

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+I+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
   NI=0
   GO TO 17

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)

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

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.NE.0) RETURN 2
ICR=-1
GO TO 98
29 LPCR=LS(LPGBL+3)+(JTY-1)*NMI
98 RETURN
```

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
```

'WRITER' WRITES THE FIVE ESSENTIEL ARRAYS FOR THE EXTRAPOLATION
IN THE A-ARRAY END OR ON TAPE 'ITAPE'.

SUBROUTINE WRITER(ITAPE,IBEGIN,LENGTH,NLEN,KENN,JFIN,IBIS)

```
COMMON NACOM,A(1000)
COMMON/FIX/NICOM,IA(300)
DIMENSION IBEGIN(NLEN),LENGTH(NLEN)
```

.....

```
IF(KENN.EQ.0)REWIND ITAPE
DO 60 IJ=1,5
IAB=IBEGIN(IJ)
IKZ=LENGTH(IJ)
IF(KENN.EQ.1)GO TO 1
```

```
IZZ=IAB+IKZ-1
WRITE(ITAPE)(A(J),J=IAB,IZZ)
GO TO 60
1 DO 2 I=1,IKZ
2 A(IBIS+I-1)=A(IAB+I-1)
  IBIS=IBIS+IKZ
60 CONTINUE
IF(KENN.EQ.0)REWIND ITAPE
RETURN
```

END

```
*****
SUBROUTINE READER(ITAPE,ILEN,BUF)
DIMENSION BUF(ILEN)
READ(ITAPE)(BUF(IJ),IJ=1,ILEN)
RETURN
END
NUMBER OF CARDS
```

```
SUBROUTINE SNOW(/MF01/,/MF02/)
COMMON NACOM,A(200)
COMMON/FIX/NICOM,IA(300)
COMMON/TEMP/GINN
COMMON/ORG/IO(43),IE(20)
COMMON /CONTR/IREBAL,IDIM(26),FUNK(26),DEV1(26),DEV2(26),NEGOV(26)
COMMON/TSCH/ITSCH,RHO,ITP1,ITP2,INORM,PFORM(300)
COMMON /SPEICH/MBOUND(25)
DIMENSION IPFORM(300)
EQUIVALENCE(PFORM(1),IPFORM(1))
DIMENSION IU(1)
```

```
*****
*****
THE IO-ARRAY CONTAINS THE FIRST ADDRESSES OF THE REAL ARRAYS.
```

```
*****
IO( 1) - JSK      NEW GROUP FLUXES
IO( 2) - JNEI    GROUP BOUNDARY FLUXES R. AND L.
IO( 3) - JNEJ    GROUP BOUNDARY FLUXES U. AND L.
IO( 4) - JARN
IO( 5) - JALP
IO( 6) - JDAA
IO( 7) - JDA
IO( 8) - JW
IO( 9) - JAL
IO(10) - JDU
IO(11) - JDE
IO(12) - JOBB
```

IO(13) - JDB
IO(14) - JV
IO(15) - JS
IO(16) - JC
IO(17) - JSV
IO(18) - JRX
IO(19) - JRY

CONTINUE

IO(20) - JRAX
IO(21) - JRAY
IO(22) - JDELX
IO(23) - JDELY
IO(24) - JXKI
IO(25) - JXKE
IO(26) - JVE
IO(27) - JF
IO(28) - JFG
IO(29) - JXARI
IO(30) - JFX
IO(31) - JB
IO(32) - JZEPS
IO(33) - JZRAX
IO(34) - JZRAY
IO(35) - JQ
IO(36) - JXNEII
IO(37) - JXNEJJ
IO(38) - JXN

CONTINUE

IO(39) - JSS
IO(40) - JT
IO(41) - JCO
IO(42) - JST
IO(43) - JTW

CONTINUE

IA(1) - ID
IA(2) - ITH
IA(3) - ISN
IA(4) - IGE
IA(5) - IBI
IA(6) - IB2
IA(7) - IGM
IA(8) - IEVT
IA(9) - IXM
IA(10) - IYM
IA(11) - MT
IA(12) - KTR
IA(13) - IBL
IA(14) - IBR
IA(15) - IBU
IA(16) - IBO

CONTINUE

IA(27) - IINP
IA(28) - ICM
IA(29) - IIM
IA(30) - IIL
IA(42) - IRIS
IA(43) - KLUX
IA(44) - ITP4
IA(45) - ITP3

IA(46) - JFIN
IA(47) - NACO
IA(48) - IHT
IA(49) - IHS

CONTINUE

IA(50) - KTPP1
IA(51) - L9
IA(52) - L8
IA(53) - L7
IA(54) - L6
IA(55) - L5
IA(56) - L4
IA(57) - L3
IA(58) - L2
IA(59) - L1
IA(60) - NFN
IA(61) - ICVT
IA(62) - MM
IA(63) - IGP
IA(64) - KTPP3
IA(65) - KTPP2
IA(66) - KTP3
IA(67) - KTP2

CONTINUE

IA(70) - JM
IA(68) - KTP1
IA(69) - ICC
IA(71) - IM
IA(72) - IG
IA(73) - JP
IA(74) - IP
IA(75) - IHM
IA(76) - IIC
IA(77) - KDUM
IA(78) - KWT
IA(79) - KR D
IA(80) - LC
IA(81) - MBT
IA(82) - IQUELL
IA(83) - JBKTP
IA(84) - JOUT
IA(85) - IZT
IA(86) - MFO1
IA(87) - KAUSW
IA(88) - NFF
IA(89) - KONTRD
IA(90) - JBKOUT

CONTINUE

IA(101) - IQUER
IA(102) - ID1
IA(103) - ID2
IA(104) - ID3
IA(105) - ID4
IA(106) - ID5

CONTINUE

ALPHABETIC CATALOG OF THE INTEGER COMMON VARIABLES

CONTINUE

IA(6) - IB2
IA(5) - IB1
IA(13) - IBL
IA(16) - IBO
IA(14) - IBR
IA(15) - IBU
IA(69) - ICC
IA(28) - ICM
IA(61) - ICVT
IA(01) - ID
IA(102) - ID1
IA(103) - ID2
IA(104) - ID3
IA(105) - ID4
IA(106) - ID5

CONTINUE

IA(8) - IEVT
IA(72) - IG
IA(4) - IGE
IA(7) - IGM
IA(63) - IGP
IA(75) - IHM
IA(49) - IHS
IA(48) - IHT
IA(76) - IIC
IA(29) - IIM
IA(30) - IIL
IA(27) - IINP
IA(71) - IM

CONTINUE

IA(74) - IP
IA(82) - IQUELL
IA(101) - IQUER
IA(42) - IRIS
IA(3) - ISN
IA(2) - ITH
IA(45) - ITP3
IA(44) - ITP4
IA(9) - IXM
IA(10) - IYM

CONTINUE

IA(85) - IZT
IO(9) - JAL
IO(5) - JALP
IO(4) - JARN
IO(31) - JB
IA(90) - JBKOUT
IA(83) - JBKTP
IO(16) - JC
IO(7) - JDA
IO(6) - JDAA
IO(13) - JDB
IO(12) - JOBB
IO(11) - JDE

CONTINUE

IO(22) - JDELX
IO(23) - JDELY
IO(10) - JDU
IO(27) - JF

	IO(28)	-	JFG	
	IA(46)	-	JFIN	
	IO(30)	-	JFX	
	IA(70)	-	JM	
	IO(2)	-	JNEI	GROUP BOUNDARY FLUXES R. AND L.
	IO(3)	-	JNEJ	GROUP BOUNDARY FLUXES U. AND L.
CONTINUE				
	IA(84)	-	JOUT	
	IA(73)	-	JP	
	IO(35)	-	JQ	
	IO(20)	-	JRAX	
	IO(21)	-	JRAY	
CONTINUE				
	IO(18)	-	JRX	
	IO(19)	-	JRY	
	IO(15)	-	JS	
	IO(1)	-	JSK	NEW GROUP FLUXES
	IO(17)	-	JSV	
	IO(14)	-	JV	
	IO(26)	-	JVE	
	IO(8)	-	JW	
	IO(29)	-	JXARI	
CONTINUE				
	IO(25)	-	JXKE	
	IO(24)	-	JXKI	
	IO(38)	-	JXN	
	IO(36)	-	JXNEII	
	IO(37)	-	JXNEJJ	
	IO(32)	-	JZEPS	
	IO(33)	-	JZRAX	
	IO(34)	-	JZRAY	
	IA(87)	-	KAUSW	
	IA(77)	-	KDUM	
	IA(43)	-	KLUX	
	IA(89)	-	KGNTRO	
CONTINUE				
	IA(79)	-	KRD	
	IA(68)	-	KTP1	
	IA(64)	-	KTPP3	
	IA(65)	-	KTPP2	
	IA(66)	-	KTP3	
	IA(67)	-	KTP2	
	IA(50)	-	KTPP1	
	IA(12)	-	KTR	
	IA(78)	-	KWT	
	IA(80)	-	LC	
	IA(59)	-	L1	
CONTINUE				
	IA(58)	-	L2	
	IA(57)	-	L3	
	IA(56)	-	L4	
	IA(55)	-	L5	
	IA(54)	-	L6	
	IA(53)	-	L7	
	IA(52)	-	L8	
	IA(51)	-	L9	
	IA(81)	-	MBT	
	IA(86)	-	MFO1	
	IA(62)	-	MM	

IA(11) - MT
CONTINUE
IA(47) - NACO
IA(88) - NFF
IA(60) - NFN
IA(107) - ISCT
IA(108) - KTPUN1
IA(109) - KTPUN2

EQUIVALENCE (A(1),IU(1))
EQUIVALENCE (IA(47),NACO),(IA(46),JFIN),(IA(45),ITP3),(IA(44),
1ITP4),(IA(43),KLUX),(IA(42),IRIS)
2 ,(IA(13),IBL),(IA(14),IBR),(IA(15),IBU),(IA(16),IBO)
EQUIVALENCE (IA(1),ID),(IA(2),ITH),(IA(3),ISN),(IA(4),IGE),(IA(5),
1IB1),(IA(6),IB2),(IA(8),IEVT),(IA(9),IXM),(IA(7),IGM),(IA(11),MT),
2(IA(29),IIM),(IA(30),IIL),(IA(28),ICM),(IA(10),IYM)
EQUIVALENCE(IA(80),LC),(IA(79),KRD),(IA(78),KWT),(IA(77),KDUM),
1(IA(76),IIC),(IA(75),IHM),(IA(74),IP),(IA(73),JP),(IA(72),IG),
2(IA(71),IM),(IA(70),JM),(IA(69),ICC),(IA(68),KTP1),(IA(67),KTP2),
3(IA(66),KTP3),(IA(65),KTPP2),(IA(64),KTPP3),(IA(63),IGP),(IA(62),
4MM) ,(IA(61),ICVT),(IA(60),NFN)
EQUIVALENCE(A(1),EPS),(A(2),EV),(A(13),XLA),(A(14),XLAR),(A(15),
1 EPG),(A(16),RLOG),(A(3),DZ)
EQUIVALENCE(IA(59),L1),(IA(58),L2),(IA(57),L3),(IA(56),L4),
1(IA(55),L5),(IA(54),L6),(IA(53),L7),(IA(52),L8),(IA(51),L9)
EQUIVALENCE(IA(50),KTPP1),(IA(12),KTR),(IINP,IA(27))
EQUIVALENCE(KFL1,IE(8)),(KFL2,IE(9))
EQUIVALENCE(IHS,IA(49)),(IHT,IA(48)),(IA(101),IQUER)
EQUIVALENCE(IO(1),JSK),(IO(2),JNEI),(IO(3),JNEJ),(IO(4),JARN),
1(IO(5),JALP),(IO(6),JDAA),(IO(7),JDA),(IO(8),JW),(IO(9),JAL),
2(IO(10),JDU),(IO(11),JDE),(IO(12),JDBB),(IO(13),JDB),(IO(14),JV),
3(IO(15),JS),(IO(16),JC),(IO(17),JSV),(IO(18),JRX),(IO(19),JRY),
4(IO(20),JRAX),(IO(21),JRAY),(IO(22),JDELX),(IO(23),JDELY),
5(IO(24),JXKI),(IO(25),JXKE),(IO(26),JVE),(IO(27),JF),(IO(28),
6JFG),(IO(29),JXARI),(IO(30),JFX),(IO(31),JB),(IO(32),JZEPS),
7(IO(33),JZRAX),(IO(34),JZRAY),(IO(36),JXNEII),(IO(37),JXNEJJ),
8(IO(38),JXN),(IO(35),JQ)
EQUIVALENCE(IO(39),JSS),(IO(40),JT),(IO(41),JCO),
1 (IO(42),JSI),(IO(43),JTW)
EQUIVALENCE(IE(1),KRX),(IE(2),KRY),(IE(3),KRM),(IE(4),KAI),(IE(5),
1KAJ),(IE(6),KZ),(IE(7),KRMY),(IE(8),KFL1),(IE(9),KFL2),(IE(10),
2KTETA),(IE(11),KMIM),(IE(12),KMJM),(IE(13),KMMRX),(IE(14),KMMRY),
3(IE(15),KMISN),(IE(16),KFIN)
EQUIVALENCE(EP5A,A(17)),(MBT,IA(81)),(IQUELL,IA(82)),(IZT,IA(85)),
1(JBKTP,IA(83)),(JOUT,IA(84)),(A(18),TEPS),(MBOUND(18),INNTS)
2 ,(IA(102),ID1),(IA(103),ID2),(IA(104),ID3),(IA(105),ID4),
3 (IA(106),ID5),(IA(107),ISCT),(IA(108),KTPUN1),(IA(109),KTPUN2)
4 ,(IA(109),KTPUN2),(IA(110),MTP)
EQUIVALENCE(IA(87),KAUSW)
EQUIVALENCE(IA(88),NFF)
EQUIVALENCE(IA(89),KONTRO)
EQUIVALENCE(JBKOUT,IA(90))
ROM=0.
GINN=0.
RHO=0.
GNULL=0.
IF(MFO2.NE.0)GO TO 6010
DO 6009 LI=1,25
6009 MBOUND(LI)=-1

```
MBOUND(1)=NACOM
6010 CONTINUE
TSUB=ZEIT(GNULL)
.....
CALL PUT(IDIM,130,0.)
IF(MF02.EQ.0) GOTO 3
I=MF01+1
WRITE(KWT,2) I
2 FORMAT('1 ****ITERATION STEP NR.',I4,'****'////////)
GOTO 3242
3 CONTINUE
KRD=5
KWT=6
READ(KRD,5)(A(I),I=1,20)
5 FORMAT(20A4)
10 FORMAT(1H1,5X,10('*'),80('-'),10('*')/6X,10('*'),20A4,10('*')/
1 6X,10('*'),80('-'),10('*'))
WRITE(KWT,10)(A(I),I=1,20)
CALL WRIT
CALL PUT(A,NACOM,0.)
CALL PUT(IA,NICOM,0.)
KRD=5
KWT=6
KLUX=-1
NACO=NACOM-5
READ(KRD,25)(IA(I),I=1,11),ICM,IIM,IIL,KTR,KDUM,IINP,IREBAL
1,ITSCH,ITP1,ITP2,ITP3,ITP4,INORM
IS=IABS(ISN)
IF(IS.EQ.2.OR.IS.EQ.4) I=9851
IF(IS.EQ.6.OR.IS.EQ.8) I=9851
IF(I.NE.9851) CALL PRINT3(1)
MF02=IABS(IB1)+1
IF(MF02.EQ.1) MF02=0
IF(IB1.EQ.0) GOTO 28
IA(41)=ITP4

IF(IB1.GT.0)ITP4=0
28 CONTINUE
READ(KRD,25) IBL,IBR,IBU,IBO,IQUELL,MBT,JBKTP,JOUT,IZT,INNTS,IQUER
1,KAUSW
MBK=IABS(MBT)
IF(IBU.EQ.2.AND.IBO.NE.2.OR.IBO.EQ.2.AND.IBU.NE.2)CALL PRINT3(7)
IF(IBL.EQ.4.AND.IBR.NE.4.OR.IBL.NE.4.AND.IBR.EQ.4)CALL PRINT3(7)
IF(IBU.EQ.4.AND.IBO.NE.4.OR.IBU.NE.4.AND.IBO.EQ.4)CALL PRINT3(7)
IF(IGE.GT.0.AND.(IBL.EQ.0.OR.IBL.EQ.4))CALL PRINT3(17)
READ(KRD,25)(ID1,ID2,ID3,ID4,ID5,NFF,JBKOUT,ISCT)
1 ,KTPUN1,KTPUN2,MTP
READ(KRD,25)(KPCH,KPLOT)
IS=ISCT+1
25 FORMAT(12I6)
IF(ISCT.EQ.0) MTP=MT
30 FORMAT(6E12.5)
READ(KRD,30) EPS,EV,DZ,RHO,EPSA,TEPS
IF(MBK.EQ.4.AND.JBKTP.EQ.0)CALL PRINT3(3)
IF(EPS.LE.0.)CALL PRINT3(8)
IF(IEVT.EQ.0.AND.IQUELL.EQ.0)CALL PRINT3(6)
JOHANN=MOD(KAUSW,100)
JOHANN=JOHANN-MOD(KAUSW,10)
```

```
IF(JBKOUT.NE.0.AND.JOHANN.EQ.0)CALL PRINT3(18)
IF(JBKOUT.NE.0.AND.KPCH.LE.0)CALL PRINT3(18)
IF(INNTS.EQ.-1)READ(KRD,30)(PFORM(IGM+I),I=1,IGM)
KTPP1=9
KTP1=10
32 KTP2=11
KTP3=12
KTPP2=13
KTPP3=14
RLOG=1.E-10
BERECHNE PARAMETER
.....
IGP=IGM+1
IHM=IGM+5
IHT=5
IHS=6
EPG=0.1*EPS
IF(EPSA.NE.0.)EPG=EPSA
IS=IABS(ISN)
MM=(IS+4)*IS/2
MMM=MM+1
LIES ZONENINTERVALLE
KRX=220
KRY=KRX+IXM
KRM=KRY+IYM
LL=KRY-1
LR=KRM-1
READ(KRD,25)(IA(I),I=KRX,LL)
READ(KRD,25)(IA(I),I=KRY,LR)
DO 35 I=KRX,LL
IF(IA(I).LE.0) CALL PRINT3(2)
35 IM=IM+IA(I)
DO 40 I=KRY,LR
IF(IA(I).LE.0) CALL PRINT3(3)
40 JM=JM+IA(I)
IF(IM.EQ.1.OR.JM.EQ.1)CALL PRINT3(9)
IP=IM+1
JP=JM+1
IJM=IM*JM
ADRESSEN
KAI=KRM+MM
KAJ=KAI+IM
KZ=KAJ+JM
KRMV=KZ+IXM*IYM
KFL1=KRMV+MM
KFL2=KFL1+IGM
KTETA=KFL2+IGM
KMIM=KTETA+JM
KMJM=KMIM+MFO2
KMMRX=KMJM+MFO2
KMMRY=KMMRX+IXM
KMISN=KMMRY+IYM
IF(ISCT.EQ.0) GOTO 29
KIMAT=KMMRY+IYM
KIPLAT=KIMAT+MTP
KMISN=KIPLAT+MTP
29 KFIN=KMISN+MFO2
IF(IB1.EQ.0) KFIN=KMIM
```

```
L1=JM
L2=MM
L3=IGM
L4=IM
L5=MM
L6=IGM
L7=IM
L8=JM
L9=IGM
JSK=51
JNEI=JSK+IJM
JNEJ=JNEI+JM*MM
JARN=JNEJ+IM*MM
JALP=JARN+IJM
JDAA=JALP
IF(NFF.GT.0)JDAA=JALP+IM*IYM*MM
JDA=JDAA
JW=JDA+IM*IYM*MM
JAL=JW+MM
JDU=JAL+IM*MM*IYM
JDE=JDU+MM
JDBB=JDE+MM
JDB=JDBB
JV=JDB+IM*MM
JS=JV+IM*IYM
JC=JS+IJM
JSV=JC+IHM*IGM*MT
JRX=JSV+IJM
JRY=JRX+IP
JRAX=JRY+JP
JRAY=JRAX+IP
JDELX=JRAY+JP
JDELY=JDELX+IXM
JXKI=JDELY+IYM
JXKE=JXKI+IGP
JVE=JXKE+IGP
JF=JVE+IGP
JFG=JF+IJM
JXARI=JFG+IJM
JFX=JXARI+JM
JB=JFX+IGP
JSS=51
JT=51
JTW=51
IF(ISCT.EQ.0) GOTO 641
IS=ISCT+1
JSS=JFX+IGP
JT=JSS+IM*JM*MM
IS=ISCT+1
JCO=JT+IS*IS*MM
JSI=JCO+ISCT
JTW=JSI+ISCT
JB=JTW+IS*IS*MM
641 CONTINUE
IMFORT=1
IF(MBK.EQ.0)GO TO 41
IF(MBK-3)42,43,44
42 IMFORT=IGM
IF(MBK.EQ.1)IMFORT=1
```

```
GO TO 41
43 IMFORT=IGM*IXM*IYM
GO TO 41
44 IMFORT=IGM*IM
41 JZEPS=JB+IMFORT
JZRAX=JZEPS+MFO2
JZRAY=JZRAX+IXM
JQ=JZRAY+IYM
JQUELL=1
IF(IQUELL)48,52,49
48 L10=IXM
L11=IYM
GO TO 51
49 L10=IM
L11=JM
51 JQR=L10*L11
JQUELL=IGM*JQR
52 JXNEII=JQ+JQUELL
JXNEJJ=JXNEII+JM*MM*IGM
IF(JXNEJJ.GT.NACO) GOTO 46
IF(IBM.EQ.0.AND.IBR.EQ.0) GOTO 46
GOTO 47
46 JXNEJJ=JXNEII+1
L1=1
L2=1
L3=1
47 IF(JXNEJJ.LE.NACO) KTP2=0
JXN=JXNEJJ+IM*MM*IGM
IF(JXN.GT.NACO) GOTO 50
IF(IBM.EQ.0.AND.IBO.EQ.0) GOTO 50
GOTO 60
50 JXN=JXNEJJ+1
L4=1
L5=1
L6=1
60 IF(JXN.LE.NACO) KTP3=0
JFIN=JXN+IJM*IGM
IF(ISCT.NE.0) JFIN=MAX0(JXN+IJM*IS*IS,JXN+IGM*IJM)
IF(JFIN.GT.NACO) GOTO 55
KTP1=0
GOTO 70
55 JFIN = JXN+1
L7=1
L8=1
L9=1
70 CONTINUE
75 IF(KTP1.NE.0) REWIND KTP1
IF(KTP2.EQ.0) GOTO 80
REWIND KTP2
REWIND KTP2
I1=JFIN-JM*MM+1
I2=JFIN
DO 85 J=1,IGM
85 WRITE(KTP2)(A(I),I=I1,I2)
80 IF(KTP3.EQ.0) GOTO 90
REWIND KTP3
REWIND KTP3
I1=JFIN-IM*MM+1
I2=JFIN
```



```
DO 95 J=1,IGM
95 WRITE(KTP3)(A(I),I=I1,I2)
90 I1=0
  IF(KTP1.NE.0) I1=I1+2
  IF(KTP2.NE.0) I1=I1+2
  IF(KTP3.NE.0) I1=I1+2
  IF(I1.EQ.0)GO TO 97
  CALL WRIT
  WRITE(KWT,98)
  CALL WRIT
98 FORMAT(/31X,'WARNING, THIS CASE MUST NOT USE FLUX PASS'/)
97 CONTINUE
  WRITE(KWT,100)(JFIN,KFIN,I1)
  CALL WRIT
  IRIS=NACOM-JFIN
  MBOUND(2)=JFIN
  MBOUND(16)=IRIS
100 FORMAT(///32X,'REAL CORE STORAGE',14X,I8/32X,'INTEGER CORE STORAG
  IE',11X,I8/32X,'NUMBER OF INTERNAL TAPES USED',2X,I8)
110 FORMAT(///30X,'*****STORAGE EXCEEDED*****')
  IF(JFIN.LE.NACOM.AND.KFIN.LE.NICOM) GOTO 105
  WRITE(KWT,110)
  KONTR0=1
105 CONTINUE
```

```
EINGABE
IF(ISN.LT.0) CALL SECRET
ISN=IABS(ISN)
```

```
ZONENRADIEN
I1=JZRAX
I2=JZRAX+IXM-1
READ(KRD,30)(A(I),I=I1,I2)
DO 106 I=I1,I2
  E1=A(I)-A(I-1)
  IF(I.EQ.I1) E1=A(I1)
106 IF(E1.LE.RLOG) CALL PRINT3(4)
  I1=JZRAY
  I2=JZRAY+IYM-1
  READ(KRD,30)(A(I),I=I1,I2)
  DO 107 I=I1,I2
    E1=A(I)-A(I-1)
    IF(I.EQ.I1) E1=A(I1)
107 IF(E1.LE.RLOG) CALL PRINT3(4)
```

MATERIAL NUMBERS

```
I1=KZ
I2=KZ+IXM*IYM-1
READ(KRD,25)(IA(I),I=I1,I2)
IF(ISCT.NE.0) GOTO 156
DO 155 I=I1,I2
155 IF(IA(I).LE.0.OR.IA(I).GT.MT)CALL PRINT3(5)
156 CONTINUE
```

```
IF(ISCT.EQ.0) GOTO 1198
I1=KIMAT
```

```
I2=I1+MTP-1
READ(KRD,25)(IA(I),I=I1,I2)
1198 CONTINUE

IF(MBK.LE.1)GO TO 200
I2=JB+IMFORT-1
IF(MBK.EQ.4)GO TO 201
READ(KRD,30)(A(I),I=JB,I2)
GO TO 200
201 IF(JBKTP.LE.0.OR.JBKTP.GT.50)GO TO 207
REWIND JBKTP
READ(JBKTP)(A(I),I=JB,I2)
GO TO 200
207 WRITE(KWT,208)MBK,JBKTP
208 FORMAT(/ /10X,' ERROR IN BUCKLING OPTION ',2I11)
KONTRO=1
200 CONTINUE

IF(IQUELL.EQ.0)GO TO 202
IF(IABS(IQUELL).EQ.2)GO TO 204
I2=JQ+JQUELL-1
READ(KRD,30)(A(I),I=JQ,I2)
GO TO 205

204 I2=JQ+IGM-1
I3=I2+1
I4=JQ+JQUELL-1
READ(KRD,30)(A(I),I=JQ,I2)
READ(KRD,30)(A(I),I=I3,I4)
205 E1=0.

E1=0.
DO 203 I=JQ,I2
203 E1=E1+A(I)
IF(E1.LE.0.)CALL PRINT3(11)
202 CONTINUE
FLUSSAUSGABE

IF(KTR) 101,102,103
101 DO 104 I=1,IGM
I1=KFL2+I-1
104 IA(I1)=1
GOTO 102
103 CALL ORDV(IA(KFL1),IA(KFL2),IGM)
102 CONTINUE

IF(KPLOT.EQ.0)GO TO 300
IF(IABS(KPLOT).NE.7)GO TO 301
CALL PLOT1
GO TO 300
301 CALL PRINT3(19)
KPLOT=0
300 CONTINUE
QUERSCHNITTE
IF(KDUM) 602,603,602
602 CONTINUE
```

```
I1=JXKI
I2=JXKE-2
READ(KRD,30)(A(I),I=I1,I2)

I3=IHM*IGM
I1=JC
I2=JC+I3-1
DO 120 M=1,MT
READ(KRD,30)(A(I),I=I1,I2)
I1=I1+I3
120 I2=I2+I3
GOTO 135
603 CALL LIES2(A(JC),IHM,IGM,MT,A(JFIN),IU(JFIN),IRIS,
1 IA(KIMAT),IA(KIPLAT),MTP)

135 CONTINUE
IF(ITH.EQ.0) GOTO 133
CALL ADJG(A(JC),A(JXKI),A(JVE),IGP,IHM,MT,IGM)
133 CONTINUE

IF(IB1.EQ.0) GOTO 3134
3242 CONTINUE
LC=0
ICC=0
IIC=0
ICVT=0
IF(MFO1.EQ.1) ITP4=IA(41)
CALL FOLGE(ISN,IM,JM,EPS,MFO1,MFO2,MM,EPG,IP,JP,IA(KRX),IA(KRY),
1 IXM,IYM,ITP3,ITP4,IA(KMISN),IA(KMIM),IA(KMJM),A(JZEPS),IA(KMMRX),
2 IA(KMMRY))
L1=JM
L2=MM
L3=IGM
L4=IM
L5=MM
L6=IGM
L7=IM
L8=JM
L9=IGM
3134 CONTINUE
MKLAR=1
MFAKT=IGM+1
MTMOB=2*IGM+1
MITK=3*IGM+1
MBIM=4*IGM+1
MBAM=5*IGM+1
CALL PUT(PFORM,IGM,0.)
IF(INNTS.NE.-1)CALL PUT(PFORM(IGM+1),IGM,0.)
CALL PUT(PFORM(2*IGM+1),4*IGM,0.)

CALL WRIT

IF(ITSCH.EQ.0) GO TO 4240
WRITE(KWT,4241)
4241 FORMAT(30X,'TSCHEBYCHEFF EXTRAPOLATION SHALL BE PERFORMED')
4243 FORMAT(/30X,'REBALANCING IS MADE')
4240 CONTINUE
```

```
IF(IREBAL.EQ.1) GO TO 4242
WRITE(KWT,4243)
4242 CONTINUE
```

```
CONTROL CALLS
IF(ITP3.GT.0)CALL TOP
IF(ITP4.GT.0)CALL PROVO(IMOLD,JMOLD)
IF(ISCT.NE.0) GOTO 555
CALL TSCHE1(IRIS,IGM,IM,JM,MM)
555 CONTINUE
CALL OUT1(IXM,IYM,MT,IGP)
```

```
CALL WRIT
RICHTUNGEN UND GEWICHTE
```

```
IS=ISN/2
GOTO(136,137,138,139),IS
136 CALL RICH2(A(JDU),A(JDE),A(JW),IA(KRM),MM,IA(KRMY))
GOTO 134
137 CALL RICH4(A(JDU),A(JDE),A(JW),IA(KRM),MM,IA(KRMY))
GOTO 134
138 CALL RICH6(A(JDU),A(JDE),A(JW),IA(KRM),MM,IA(KRMY))
GO TO 134
139 CALL RICH8(A(JDU),A(JDE),A(JW),IA(KRM),MM,IA(KRMY))
134 CONTINUE
CALL IFUNC(A(JXKE),A(JXKI),IGP)
IF(ITP4.LE.0) GO TO 4310
RXMAX=A(JZRAX+IXM-1)
RYMAX=A(JZRAY+IYM-1)
NFN=1
*****
DER PROV1-AUFRUF MUSS IM HAUPTPROGRAMM VOR DEM GEO-AUFRUF STEHEN
*****
CALL PROV1(RXMAX,RYMAX,A(JFIN),A(JXN),
3 A(JZRAX),A(JZRAY),IMOLD,JMOLD,IRIS,
1A(JFG),IA(KAI),IA(KAJ),IA(KZ),A(JC),A(JV),A(JF ),A(JFX),IGP,IM,
2JM,IHM,IGM,MT,IXM,IYM,ROM,A(JXKE))
4310 CONTINUE
```

```
130 CALL GEO(A(JRAX),A(JRAY),IA(KRX),IA(KRY),A(JDELX),A(JDELY),
1A(JRX),A(JRY),IA(KAI),IA(KAJ),A(JDB),A(JDA),A(JAL),A(JV),IP,
2JP,IM,JM,IXM,IYM,MM,A(JDU),A(JW),IA(KTETA),A(JZRAX),A(JZRAY),
3A(JDE),A(JALP),MMM)
```

```
IF(ISCT.EQ.0) GOTO 4298
IS=ISCT+1
CALL ANISO(A(JDE),A(JDU),A(JW),A(JT),A(JTW),IS,
1 ISCT,MM,A(JCO),A(JSI))
4298 CONTINUE
```

```
IF(ITP4.LE.0)GO TO 4311
IND1=JFIN+IMOLD
IND2=IND1+JMOLD
IND3=IND2+MM
CALL PROV2(A(JFIN),A(IND1),A(IND2),IMOLD,JMOLD,A(JRAX),A(JRAY))
IF((IBR+IBL).LE.0.OR.IBR.EQ.4)GO TO 4400
KLUX=-1
```

```
CALL PROV3(A(JXNEII),A(IND3),MM,A(IND2),JM,A(JRAY),A(IND1),JMOLD)
4400 CONTINUE
IF((IBO+IBU).LE.0.OR.IBO.EQ.4)GO TO 4401
KLUX=-1
CALL PROV3(A(JXNEJJ),A(IND3),MM,A(IND2),IM,A(JRAX),A(JFIN),IMOLD)
4401 CONTINUE
4311 CONTINUE
IF(ITP4.EQ.0.AND.MBK.EQ.4)CALL PROV4(A(JXN),A(JXNEII),IM,JM,MM,IGM
1,A(JF),A(JFG),A(JC),A(JXKE),IA(KAI),IA(KAJ),IA(KZ),A(JFX),A(JV),
2IHM,MT,IXM,IYM,IGP)

CALL PRINT1(A(JZRAX),A(JZRAY),IA(KZ),A(JC),IHM,IGM,MT,IXM,IYM,
1 IA(KAI),IA(KAJ),IM,JM)
IF(ICM.LT.0)CALL SECRET
ICM=IABS(ICM)
TRM=IABS(ICM)

CALL FISSN(A(JFX),A(JF),A(JFG),A(JV),A(JXKE),IGP,IM,JM,A(JC),
1IA(KAI),IA(KAJ),IA(KZ),IXM,IYM,IHM,IGM,MT)
IF(ICC.NE.0)GOTO 175
IF(KONTRO.NE.0)CALL SECUR
WRITE(KWT,171)
171 FORMAT('1',40X,5('*'),' ITERATION MONITOR ',5('*'))
170 FORMAT(/////19X,'OUTER',11X,'INNER',12X,'EIGENVALUE',8X,'LAMBDA',
114X,'REDUCT.FACTOR'/17X,'ITERATIONS',5X,'ITERATIONS'/)
WRITE(KWT,170)
175 CONTINUE
IF(ICVT.EQ.1)ICVT=2
IF(ABS(1.-XLAR).GT.RLOG)ROM=(1.-XLA)/(1.-XLAR)
WRITE(KWT,180)ICC,LC,EV,XLA,ROM
...
...
4203 FORMAT(1H0)
WRITE(KWT,4203)
WRITE(KWT,4202)
WRITE(KWT,4201)(I,IDIM(I),FUNK(I),IPFORM(3*IGM+I),PFORM(IGM+I),
1I=1,IGM)
WRITE(KWT,4203)
...
...
4202 FORMAT(25X,5HGROU,3X,26HINNER ITERATIONS PER GROUP)
4201 FORMAT(25X,I4,13X,I4,4X,E15.8,14X,I5,3X,E15.8)
180 FORMAT(
1 14X,I9,5X,I10,5X,F19.8,5X,E15.8,5X,E15.8)
.....
IF(ICC.GT.ICM)GOTO 501
IF(ICC.GT.IRM)GOTO 501
IF(ISCT.NE.0)GOTO 504
503 CALL TSCHEB(IGM,IM,JM,MM,ICC,A(JFIN),IRIS)
IF(ICC.EQ.0)CALL TEMP1(IPFORM(1),PFORM(MFAKT),PFORM(MTMOB),IPFORM(
1MITK),PFORM(MBIM),PFORM(MBAM))
504 CONTINUE
CALL OUTER(A(JS),A(JSV),A(JSK),A(JXN),A(JC),A(JXKE),IA(KAI),
1IA(KAJ),IA(KZ),A(JF),A(JFG),A(JNEI),A(JNEJ),A(JXNEII),
2A(JXNEJJ),A(JXKI),L1,L2,L3,L4,L5,L6,L7,L8,L9,IM,JM,IGM,MT,IHM,
3IGP,IXM,IYM,MM,A(JV),A(JDA),A(JDB),A(JW),A(JDE),A(JDU),IP,A(JQ),
4 L10,L11,A(JXN),A(JSS),IS)
II=876543
```

```

IF(IA(1).EQ.II) CALL SECRET
IF(ITP3.LE.0)GO TO 4302
CALL TOPROV(A(JXN),A(JXNEII),A(JXNEJJ),A(JRAX),A(JRAY),A(JFIN),
1 IRIS,MM,IP,JP,IGM,IXM,IYM,IM,JM,ROM,A(JZRAX),A(JZRAY))
4302 CONTINUE
IF(ICVT.LE.1)GO TO 175
CALL PRINT2(A(JSV),A(JXN),A(JRAX),A(JRAY),A(JFX),IA(KAI),IA(KAJ),
1 A(JV),IM,JM,IGP,IYM,A(JXN),IS)
WRITE(KWT,172)
WRITE(KWT,170)
.....
IF(ABS(1.-XLAR).GT.RLOG)ROM=(1.-XLA)/(1.-XLAR)
WRITE(KWT,180) ICC,LC,EV,XLA,ROM
WRITE(KWT,4203)
WRITE(KWT,4202)
WRITE(KWT,4201)(I, IDIM(I),FUNK(I),IPFORM(3*IGM+I ),PFORM(IGM+I ),
1 I=1,IGM)
WRITE(KWT,4203)
WRITE(KWT,4203)
CALL WRIT
.....
172 FORMAT('1',35X,10('*'),' FINAL ITERATION RESULT ',10('*'))
GINN=GINN/60.
TSUBA=ZEIT(TSUB) /60.
TSUB=(GINN/TSUBA)*100.
WRITE(KWT,4210)ID,TSUBA,GINN,TSUB
CALL WRIT
IF(JOUT.GT.0)CALL TOMBO(A(JRAY),A(JXN),A(JXNEII),IM,JM,MM,IGM,IZT,
1A(JFIN),IRIS)
4210 FORMAT( 30X,' TIME USED BY THE CASE WITH ID.',I6,5X,F11.4,' MINUT
1ES'
1 /30X,' INNER USED ',F11.4,' MINUTES, =' ,F6.1,' PER CENT')

GOTO 500
501 WRITE(KWT,502)
502 FORMAT(///30X,'*****WARNING*****'/30X,'FINISHED BY MAXIMUM NUMBER
1 OF OUTER ITERATIONS'//30X,'A FINAL ITERATION IS PERFORMED')
ICVT=2
GO TO 503
500 CALL PLACE
IF(IABS(KPLOT).EQ.7
* .AND.(MFO1.EQ.(IABS(IB1)+1).OR.IB1.EQ.0))CALL PLOT2
CALL REMUS
RETURN
END

```