

PNC ^丁 SJ 299 84-07

本資料は200/年 7 月 3 / 日付けで
登録区分変更する。 [技術展開部技術協力課]

大型炉の中性子輸送効果の評価に関する研究

(受託研究)

1984年3月

日本情報サービス株式会社



PNC Tsj 299 84-07

1984年3月

大型炉の中性子輸送効果の評価に関する研究*

川畑 博信**
新浜 耕栄**
中挟 義夫**
山本 敏久***
坂東 勝***
竹田 敏一***

要 旨

高速炉炉心計算用の三次元輸送計算コードを作成した。本コードは fine-mesh-rebalancing 法及び system rebalancing 法を用いて解の収束を図っている。この加速法の適用性を調べるための三次元テスト計算を行った。その結果、両 rebalancing を適当に組み合わせて用いると実効増倍率、中性子角分布が早く収束する事が示された。加速法として最近注目をあびている拡散合成法の三次元体系への適用性を調べるための第一段階として、この手法を二次元体系へ拡張した。テスト計算では TWOTRAN-II に比べ 1/10 以下の計算時間で収束する事が示された。

* 本報告書は、日本情報サービス㈱が、大阪大学工学部の協力を得て動力炉・核燃料開発事業団の委託により実施した研究成果である。

** 日本情報サービス株式会社 原子力技術部

*** 大阪大学工学部 原子力工学科

Evaluation of Neutron Transport Effect
in Fast Reactors *

Hironobu Kawabata **
Koei Shinhama **
Yoshio Nakabasami **
Toshihisa Yamamoto ***
Masaru Bando ***
Toshikazu Takeda ***

Abstract

A three-dimensional transport code THOTH has been developed for core calculations of fast reactors. The fine-mesh rebalancing method and the system rebalancing method were adopted to converge the solution. To investigate the applicability of the two methods, a three-dimensional test calculation was performed. The appropriate use of the two rebalancing methods has efficiently converged k_{eff} and neutron angular fluxes.

We extended the diffusion synthetic acceleration method to a two-dimensional system to investigate the applicability of multi-dimensional calculations. In a test calculation, the diffusion synthetic acceleration method required computer time less than a factor of ten compared to the TWOTRAN-II code.

* Study performed under contracts between Power Reactor and Nuclear Fuel Development Corporation, Japan Information Service Ltd. and Osaka University.

** Japan Information Service Ltd.

*** Osaka University, Faculty of Engineering.

目 次

List of Tables

List of Figures

1. 緒 言	1
2. 3次元輸送方程式の S_n 近似	2
2-1. 中性子束分布計算の差分式	2
2-2. 計算手順	5
2-3. 計算結果	9
3. 中性子輸送計算のための拡散合成加速	23
3-1. 2次元 X Y 体系での差分式	26
3-2. 計算結果	32
4. 結言および今後の問題点	33
謝 辞	34
参 考 文 献	35
付 録	
A. 再釣合法の原理	36
B. 拡散合成法の原理	39
C. THOTHコードソースリスト	50

List of Tables

Table 2-1	Mesh sweep for each angular direction
Table 2-2	Comparison of flux
Table 2-3	Comparison of power distributions
Table 3-1	Cross sections
Table 3-2	Computational results

List of Figures

- Figure 2-1 Location of $\psi_{\lambda \pm 1/2}, \psi_{j \pm 1/2}, \psi_{k \pm 1/2}$
- Figure 2-2 Specification of surface
- Figure 2-3 Specification of angular direction
- Figure 2-4 Inner sweep procedure in THOTH
- Figure 2-5 Two-dimensional core model for 3-group test calculations
- Figure 2-6 再釣合法による固有値収束の改善
- Figure 2-7 再釣合法による中性子束収束性の改善
- Figure 2-8 内側反復計算の回数による固有値収束性への影響
- Figure 2-9 S_2 -Initial ϕ による固有値収束性の改善
- Figure 2-10 S_2 -Initial ϕ による中性子束収束性の改善
- Figure 2-11 Three dimensional core model for 3-group calculations
- Figure 2-12 Comparison of eigenvalue convergence (S_4)
- Figure 2-13 Comparison of eigenvalue convergence (S_4)
- Figure 2-14 Comparison of flux convergence (S_4)
- Figure 3-1 Mesh point for diffusion synthetic method in 2-dimensional geometry
- Figure 3-2 Geometry
- Figure 3-3 Comparison of convergent behavior of K_{eff} (3-group calculation)

1. 緒 言

高速増殖炉炉心核特性を精度よく予測するためには中性子輸送効果を考慮した核計算を実行しなければならない。燃焼初期においては制御棒が挿入されているため、制御棒反応度価値、実効増倍率、出力分布等に対する中性子輸送効果が顕著になる。炉中心に制御棒が挿入された場合には二次元 RZ モデルでこの中性子輸送効果を評価できるが、オフセンターの制御棒、特に中途挿入の制御棒に対する中性子輸送効果を二次元体系の輸送計算より評価しようとするともデリングによる近似誤差が生じる。さらに、非均質炉心になると中性子輸送効果はより強く表われるため、三次元中性子輸送を扱える計算コードの開発が望まれる。

このため、三次元輸送コードの開発に関する研究を進めた。本報告では三次元輸送コードを作成するために用いた計算方法について説明する。三次元輸送コードで特に問題となるのは収束性である。拡散計算と異なり S_n 計算では各角度分点に対応する中性子数を計算するので収束は非常におそくなる。このため、まず fine-mesh-rebalancing 法及びこの再釣り合い法を全体系、全エネルギー群に適用する system-rebalancing 法をコードに組みこみ、三次元テスト問題でこの手法の有効性について調べた。

最近、中性子輸送計算の収れん加速法として拡散合成法が注目をあびている。この手法は拡散方程式に似た方程式を輸送方程式から導出し、その方程式を解くことにより加速を行う方式である。この方法は非常に有効であることが一次元体系では確かめられている。これは拡散方程式と類似の式は輸送計算に比べ非常に早く収束できるためである。この拡散合成法を三次元輸送計算に適用した場合の有効性を調べるため、本報告では二次元体系に拡散合成法を拡張し、テスト計算により、その有効性を検討した。

2. 3次元輸送方程式の S_n 近似

2-1. 中性子束分布計算の差分式

3次元 (X, Y, Z) 体系における多群輸送方程式は

$$\mu_m \frac{\partial \psi_m}{\partial X} + \eta_m \frac{\partial \psi_m}{\partial Y} + \xi_m \frac{\partial \psi_m}{\partial Z} + \Sigma_t \psi_m = S_m \quad (2-1)$$

と書き表わされる。

ここで、 μ_m, η_m, ξ_m は中性子飛行方向 Ω_m の X, Y, Z 成分であり、 ψ_m は $\psi^g(X, Y, Z, \Omega_m)$ の角度依存中性子束である。 S_m は散乱および核分裂による中性子源の項である。

(2-1) 式を、差分式にするために、体系をメッシュ (ボックス) に分割する。

$$\begin{aligned} X_{i-\frac{1}{2}} < X_i < X_{i+\frac{1}{2}} & \quad i=1, 2, \dots \quad \text{IT} \\ Y_{j-\frac{1}{2}} < Y_j < Y_{j+\frac{1}{2}} & \quad j=1, 2, \dots \quad \text{JT} \\ Z_{k-\frac{1}{2}} < Z_k < Z_{k+\frac{1}{2}} & \quad k=1, 2, \dots \quad \text{KT} \end{aligned} \quad (2-2)$$

ここで、 $i \pm \frac{1}{2}, j \pm \frac{1}{2}, k \pm \frac{1}{2}$ は、各ボックス表面の中心位置を表わし、 i, j, k はボックスの中心位置を表わす。(Fig. 2-1)

(2-1) 式の差分式は

$$\begin{aligned} \frac{\mu(\psi_{i+\frac{1}{2}} - \psi_{i-\frac{1}{2}})}{\Delta X} + \frac{\eta(\psi_{j+\frac{1}{2}} - \psi_{j-\frac{1}{2}})}{\Delta Y} + \frac{\xi(\psi_{k+\frac{1}{2}} - \psi_{k-\frac{1}{2}})}{\Delta Z} \\ + \Sigma_t \psi = S \end{aligned} \quad (2-3)$$

となる。但し、(2-3) 式では、添字 m は省略している。

さらに、 ΔX , ΔY , ΔZ は

$$\Delta X = X_{i+\frac{1}{2}} - X_{i-\frac{1}{2}}$$

$$\Delta Y = Y_{j+\frac{1}{2}} - Y_{j-\frac{1}{2}}$$

$$\Delta Z = Z_{k+\frac{1}{2}} - Z_{k-\frac{1}{2}} \quad (2-4)$$

を表わしている。

(2-3) 式を計算するために、 $\Psi_{i\pm\frac{1}{2}}$, $\Psi_{j\pm\frac{1}{2}}$, $\Psi_{k\pm\frac{1}{2}}$ と Ψ の関係として、次のダイヤモンド差分近似を用いる。

$$\begin{aligned} 2 \cdot \Psi &= \Psi_{i+\frac{1}{2}} + \Psi_{i-\frac{1}{2}} \\ &= \Psi_{j+\frac{1}{2}} + \Psi_{j-\frac{1}{2}} \\ &= \Psi_{k+\frac{1}{2}} + \Psi_{k-\frac{1}{2}} \end{aligned} \quad (2-5)$$

そうすると、 $\Delta L_m > 0$ の方向に飛んでいる中性子に対しては、(2-5) 式を用いて (2-3) 式より $\Psi_{i+\frac{1}{2}}$, $\Psi_{j+\frac{1}{2}}$, $\Psi_{k+\frac{1}{2}}$ を消去することにより、

$$\Psi = \frac{2 \cdot \left| \frac{\mu}{\Delta X} \right| \cdot \Psi_{i-\frac{1}{2}} + 2 \cdot \left| \frac{\eta}{\Delta Y} \right| \cdot \Psi_{j-\frac{1}{2}} + 2 \cdot \left| \frac{\xi}{\Delta Z} \right| \cdot \Psi_{k-\frac{1}{2}} + S}{2 \cdot \left| \frac{\mu}{\Delta X} \right| + 2 \cdot \left| \frac{\eta}{\Delta Y} \right| + 2 \cdot \left| \frac{\xi}{\Delta Z} \right| + \sum_t} \quad (2-6)$$

として、 Ψ が計算されることになる。(2-6) 式より、 S と \sum_t が決まっていれば、 $\Psi_{i-\frac{1}{2}}$, $\Psi_{j-\frac{1}{2}}$, $\Psi_{k-\frac{1}{2}}$ がすでに計算されていれば、 Ψ が計算できることになる。そのあと、(2-5) 式の関係より、 $\Psi_{i+\frac{1}{2}}$, $\Psi_{j+\frac{1}{2}}$, $\Psi_{k+\frac{1}{2}}$ は

$$\begin{aligned} \Psi_{i+\frac{1}{2}} &= 2 \cdot \Psi - \Psi_{i-\frac{1}{2}} \\ \Psi_{j+\frac{1}{2}} &= 2 \cdot \Psi - \Psi_{j-\frac{1}{2}} \\ \Psi_{k+\frac{1}{2}} &= 2 \cdot \Psi - \Psi_{k-\frac{1}{2}} \end{aligned} \quad (2-7)$$

として、求められる。

$\mu < 0$, $\eta > 0$, $\xi > 0$ の方向に飛んでいる中性子に対しては、(2-6) 式の $\Psi_{i-\frac{1}{2}}$ を $\Psi_{i+\frac{1}{2}}$ に置きかえればよい。他の方向に対しても、同様な方法で Ψ を計算することができる。

一方、この方法では(2-7)式の関係より、中性子束が負になることがある。その場合には、(SET TO ZERO)と呼ばれる手法を使用している。

例として、 $\psi_{i+\frac{1}{2}}$ が負になった場合を説明する。

(2-7)式の結果として、 $\psi_{i+\frac{1}{2}}$ が負になったとすると、 $\psi_{i+\frac{1}{2}}$ をゼロとして(2-3)式を変形すると、(2-6)式のかわりに

$$\psi = \frac{|\frac{\mu}{\Delta X}| \cdot \psi_{i-\frac{1}{2}} + 2 \cdot |\frac{\eta}{\Delta Y}| \cdot \psi_{j-\frac{1}{2}} + 2 \cdot |\frac{\xi}{\Delta Z}| \cdot \psi_{k-\frac{1}{2}} + S}{2 \cdot |\frac{\eta}{\Delta Y}| + 2 \cdot |\frac{\xi}{\Delta Z}| + \Sigma_t} \quad (2-8)$$

として、 Ψ を計算する式が得られる。

この式により、ふたたび Ψ を計算したあと、(2-7)式により $\psi_{i+\frac{1}{2}}$ を計算する。 $\psi_{j+\frac{1}{2}}$ 、 $\psi_{k+\frac{1}{2}}$ に対しても、同じように再計算を行なうことになる。

このような現象は、計算開始時にSがゼロであることから生じることが多いので、注意する必要がある。

2-2. 計算手順

前述の(2-6)式で示した差分式に従い、メッシュ点の中性子を順番に計算していく(スイープ)。このスイープには、(X, Y, Z)の各方向に対して、「+」、「-」の方法がある。例えば、X方向に「+」のスイープはLEFTからRIGHTに向かう(-Xから+Xの方向)ことを表わしている。(Fig.2-2)

角度依存中性子束は、(Fig.2-3)の8象限のどれかに含まれるが、各象限の中性子束の計算では、定められた方向にしかスイープすることができない。

例えば、第4象限の角度依存中性子束に対しては

X方向は「+」方向

Y方向は「-」方向

Z方向は「+」方向

にスイープしなければならない。

同様に、その他の象限の角度依存中性子束に対しては、そのスイープの方向を(Table 2-1)に示す。

次に、(Fig. 2-4)に従って、8方向のスweepの手順を示す。

a)

① : ライン#1から第7象限の方向の角度依存中性子束(第7象限内のすべての方向を考える)について、 $(-, -, -)$ 方向のスweepを行なう。

② : 境界上の中性子束は、境界条件に従い

反射条件 : 反射方向の角度依存中性子束をセットする。

真空条件 : 角度依存中性子束をゼロにする。

上記の方法で、値をセットする。

③ : 同じライン#1を逆にsweepする。この時、 $(+, -, -)$ 方向に、第8象限がsweepされる。

④ : 次のライン#2について、①~③の処理を行なう。

b)

⑤ : X・Y面に接するライン#Jのスweepを完了したあと、②と同じように境界上の角度依存中性子束をセットする。

⑥ : ライン#Jから、 $(-, +, +)$ 方向に第6象限の角度依存中性子束についてsweepを行なう。

⑦ : ②と同じように、境界上のセットを行なったあと、逆に $(+, +, +)$ 方向で第5象限のスweepを行なう。

⑧ : ⑥~⑦をライン#1まで行なう。

c)

⑨ : ①~⑧でsweepが完了した斜線の平面#1から、次の平面#2に移る。そのあと、ライン#J+1よりライン#2Jに向って、第7象限のスweepを行なう。

⑩ : ③と同じように、第8象限のスweepを行なう。

d)

⑪ : ⑨~⑩を繰返して、最上面(平面#k)までのsweepを行なう。

⑫ : ②と同じように境界上のセットを行なう。

⑬ : 次に、①~⑩の処理を平面#kから平面#1まで行なう。この時、sweepされる象限は、1, 2, 3, 4象限である。

各象限の角度依存中性子束は、Sn オーダーのNに対して、

$$MMX = n \cdot (n + 2) / 8 \quad (2-9)$$

の方向が、考えられている。

前述の計算では、各スイープの時に、角度成分として、MMX 個がまとめてスイープされる。

THOTHコードでは、外側反復計算で求めた他群からのソース(QQ)に、内側反復計算で求めた中性子束から計算した自群散乱ソースを加算している。新しく計算されていく角度依存中性子束の効果を、できる限り新しいソース項として用いるために、角度依存中性子束の計算の度に、中性子束の再計算を行なった。この場合、計算時間が、2倍程度になったほか、中性子束が非対称なままで収束してしまうことになった。そのため、現在は体系全体のスイープが1回完了するごとに中性子束の再計算を行なっている。

体系全体のスイープが1回完了すると、中性子束より各セル内での吸収量、もれ量が計算され、再釣合法による中性子束の加速が行なわれる。

再釣合法を適用する場合には、体系全体に対して適用する場合 (system-rebalancing 法) と、メッシュ単位に適用する場合 (fine-mesh-rebalancing 法) の2種類が使い分けられるようになっている。これは、体系の非均質性や大きさによって使い分けなければならない。

体系の固有値(実効増倍率)は、体系全体の核分裂量、吸収量および体系からのもれ量を足し合わせて計算する。

中性子束を出力が1.0になるように規格化したあと、求めた固有値を用いて、核分裂ソース(FISS)および各群のソース(QQ)が次のように計算される。

$$FISS = \sum_{g'} \nu \sum_j \phi_j^{g'} / k_{eff}$$

$$QQ^g = \chi^g \cdot FISS + \sum_{g'} \Sigma_s^{g \rightarrow g'} \cdot \phi_j^{g'} \quad (2-10)$$

外側反復2回目以降は、前回の内側反復計算で、すでに各セルごとの吸収量、もれ量

が求まっているので、これをすべての群について足し合わせたあと、再釣合法を用いてソースとのバランスをとって中性子束および核分裂ソースの加速を行なう。

計算の終了は、内側反復計算が指定された回数に達したときか、中性子束が収束したときである。

中性子束の収束は、求められた新しい中性子束 (SCLFL) と、前回の中性子束 (SCLOL2) を比較し、その変化率の最大値を各群ごとに計算する。この時、固有値 (EIGEN) も、前回の固有値 (EGNOLD) との変化率を計算する。収束判定は、下の2つの条件が満足されているかどうかで調べる。

$$\begin{aligned} \text{Max}_{i,j,k} \left| \frac{\text{SCLFL} - \text{SCLOL2}}{\text{SCLFL}} \right| &< \text{CONCR} \\ \text{Max}_{i,j,k} \left| \frac{\text{EIGEN} - \text{EGNOLD}}{\text{EIGEN}} \right| &< \text{CONCRE} \end{aligned}$$

ここで、CONCR, CONCRE は、それぞれ中性子束、固有値の収束判定基準値であり、 10^{-4} , 10^{-5} が使用される。

THOTHコードでは、計算時間が長くなることを考慮して、次のように計算時間を判定して再計算用のファイル出力を行なう。収束しない場合には、外側反復計算の1回の所要時間と、残り時間を比較し、残り時間が少なければ、次の外側反復計算は行なわずにファイル出力を行なった後、計算を終了する。次回の計算ではそのファイルの値を初期値として計算を行なうことができる。

2-3. 計算結果

(I) 2次元体系での再釣合法による加速の効果

再釣合法による加速の効果を調べるために、(Fig.2-5)に示すように、一片が15.5 cmの燃料集合体と、水反射体による $\frac{1}{4}$ 小型炉心体系で、3群の計算を行なった。収束条件は、中性子束変化率、固有値変化率ともに 10^{-3} を使用した。

再釣合法の効果を調べるために、

- ① 加速しない場合
- ② 内側反復計算のみに再釣合法を使用する場合
- ③ 内側反復計算、外側反復計算の両方に再釣合法を使用する場合

の3ケースについて計算を行なった。

以下にその結果について説明する。

(Fig.2-6)は、固有値の変化の様子を示したもので横軸には計算時間を示している。この図でわかるように内側反復計算のみに再釣合法を適用するだけで、大幅な改善が行なわれたことがわかる。次に(Fig.2-7)に、中性子束変化率の様子を示しているが、この図では、外側反復計算にも再釣合法を適用しなければ十分な加速が行なわれないことがわかる。

いずれの場合も加速しない場合に、収束条件に近づくに従って収束が遅くなるのに対して、再釣合法により、きわめて急速に収束することが示されている。

(Fig.2-8)で、内側反復計算の回数の違いによる固有値の変化の様子を示す。この図において、内側反復計算が5回と7回の場合を示しているが、両者の間には有意な差は見られない。

従って、この体系の計算では内側反復計算の回数は5回で十分である。結論として前述の(Fig.2-6)の場合が、内側反復計算を10回行なっていることを考えると、実際の再釣合法による加速効果は倍程度になると思われる。

次に、 S_8 計算をするために、一担 S_2 計算を行なった場合の効果を述べる。(Fig.2-9)に示すように、 S_2 計算で収束したあと、 S_8 計算では中性子束が急速に収束

することがわかる。この効果は、体系が均質であるため、 S_2 計算と S_8 計算の角度依存中性子束が類似しているためと思われる。非均質な体系にこの方法を適用してその効果を評価したいと思っている。

(ii) TWOTRANコードとの計算結果の比較

THOTHコードにより、2次元体系の計算を行なう際に、XY軸を用いてZ軸にメッシュを1点取ったモデルで計算を行なった。この場合S₄計算で4×4×1メッシュでは収束したが、6×6×1メッシュでは収束しなかった。

原因を調べるために、TWOTRANコードで、1次元体系の計算を行なった。この場合には、5×1メッシュで収束したものの中性子束分布に異常が見られた。

いずれの場合も特に熱群の中性子束の収束性の悪さが目立った。

さらに、TWOTRANコードで5×2メッシュの計算をしたところ正常な結果が得られたので、この現象はS_nコード特有のものか、再釣合法加速に特有のものであると思われる。

次に、THOTHコードで収束した4×4×1メッシュの計算結果をTWOTRANコードの結果と比較した。体系は(Fig.2-5)に示したものとほぼ同じであるが、反射体の厚さを20cm、Z軸方向を1メッシュにしている。

(Table 2-2)に、中性子束分布を示している。特に反射体の所で大きな差異が見られるが、これはスイープの順番の問題で、角度依存中性子束が上下非対象になることから特に中性子束のチルトが大きくなる反射体の部分で、Z軸方向へのもれが生じたものである。

THOTHコードの計算内容の検証のために、計算モデルとしてX-Y体系と同様にX-Z, Y-Z体系で計算したが、結果はまったく問題がなかった。

THOTHコードとTWOTRANコードの固有値は、THOTHコードで0.7711, TWOTRANコードで0.7571となり、これは中性子束分布の違いによるものであると考えられる。

(iii) 3次元非均質体系への再釣合法の適用

Fine-mesh-rebalancing 法は、加速効果は大きいですが、適用する体系によっては発散する場合があります。特に、メッシュの数が多く中性子束分布が大きくひずむ体系に適用する場合には、その安定性について検討する必要があります。

System-rebalancing 法は、体系全体の吸収量、ソース量、体系よりのもれ量から、全エネルギー群で平均されたファクターを計算し、中性子束分布に適用する加速法である。収束に対する安定性は高いが、個々のメッシュ点の中性子束のバランスを考えないために、加速効果はあまり大きくない。この方法は fine-mesh-rebalancing 法で、個々のメッシュ点の中性子束のバランスが取れなかった場合に用いられるのが普通である。

TWOTRAN コードでは、両者の長所を組み合わせるために、内側反復計算の奇数回目には coarse-mesh-rebalancing 法、偶数回目には system-rebalancing 法を用いている。

THOTH コードに、再釣合法を適用した場合の収束の安定性及び有効性を検討するために、次のような3種類の加速法について、テスト計算を行なった。

- ① system-rebalancing 法のみを適用
- ② 奇数回目に fine-mesh-rebalancing 法、偶数回目に system-rebalancing 法を適用
- ③ fine-mesh-rebalancing 法を適用

いずれも、外側反復計算については fine-mesh-rebalancing 法を用いている。方法②は TWOTRAN コードと同じである。

計算体系は (Fig.2-11) に示すような1片 15.5 cm の立方体セルからなる $4 \times 4 \times 4$ の集合体と、それをとり囲む厚さ 15.5 cm の水反射体よりなっている。エネルギー群数は3群で S_4 計算を行ない内側反復計算の回数は最大で5回である。収束条件は、固有値変化率が 10^{-4} 、中性子束変化率が 5×10^{-4} である。

結果としては、方法②、③は、いずれも固有値が振動して収束しなかった。原因は

内側反復計算で中性子束が振動し、収束しないまま外側反復計算に入っていたためである。対策として前回の外側反復計算での中性子束変化率に比べて、今回の内側反復計算の中性子束変化率が大きくなった場合、それ以降は `system-rebalancing` 法のみを用いるようにした。但し、一旦計算を中断した後の再計算では、前回の結果がないために、再度 `fine-mesh-rebalancing` 法が用いられることになる。さらに、`fine-mesh-rebalancing` 法で計算した加速ファクターの値が、2.0 以上の時、又は 0.5 以下の時には、このような振動が発生して収束しないことがわかった。そのため、THOTHコードでは、このファクターの上下限値を指定し、その範囲を出た場合は `system-rebalancing` 法を適用するようにした。

(Fig.2-12) では計算開始後初期の固有値の変化を示し、(Fig.2-13) では計算開始1分後以降の固有値の変化を示している。又、(Fig.2-14) では(Fig.2-13) に対して、中性子束変化率の様子を示している。

方法①では、固有値の変化が少なく、収束が遅いが、途中から `fine-mesh-rebalancing` 法で再計算を行なうと、外側反復計算を1, 2回した後、再度 `system-rebalancing` 法にもどってしまう。しかし、この時固有値が急速に収束していることがわかる。

再計算後も `system-rebalancing` 法のみでは、固有値は収束せず、そのために中性子束の収束も悪くなっている。

方法②では、約4分後に固有値1.0146に収束した。約2分後に山を生じているのは、再計算により `fine-mesh-rebalancing` 法が用いられたためである。

方法③が、計算初期では最もよい収束性を示したが、途中より固有値が発散する方向に動き、中性子束変化率も振動した。

以上の結果より、計算の効率を上げるには、少なくとも計算初期では、数回の `fine-mesh-rebalancing` 法が必要であり、中性子束分布が十分収束した後には、`fine-mesh-rebalancing` 法は用いない方が、収束の安定性の面から有利であることがわかった。

このことから、system-rebalancing 法に加えて、周期的に、fine-mesh-rebalancing 法を用いる加速法が有効であるとわかった。

(Table 2-3) には、方法②で計算した中性子束分布から求めた出力分布と、CITATIONコードで計算した出力分布の比較を示す。結果としては、燃料集合体 A, B の境界や反射体の周辺の輸送効果の大きい所で、出力に差異が生じている。

1 群 CORE CENTER

4.997-3*	3.337-3	7.842-4	3.447-5
4.726-3**	3.365-3	8.374-4	3.871-5
	2.215-3	5.049-4	1.982-5
	2.445-3	5.913-4	2.194-5
(对称性良好)		1.463-4	1.696-5
		1.880-4	2.210-5
			6.365-6
			8.177-6

2 群

2.115-3*	1.403-3	4.949-4	5.400-5
2.000-3**	1.413-3	5.298-4	5.797-5
	9.237-4	3.186-4	3.311-5
	1.009-3	3.737-4	3.751-5
		1.044-4	1.505-5
		1.343-4	1.946-5
			4.939-6
			6.711-6

3 群

1.938-3*	1.626-3	2.530-3	1.453-3
1.732-3**	1.617-3	1.911-3	1.078-3
	1.300-3	1.754-3	9.686-4
	1.524-3	2.236-3	1.246-3
		1.022-3	4.014-4
		1.465-3	5.824-4
			1.116-4
			1.596-4

* THOTH ** TWOTRAN

Table 2-2 Comparison of flux

Table 2-1 Mesh Sweep for Each Angular Direction

Mesh sweep	Angular direction							
	1	2	3	4	5	6	7	8
X	+	-	-	+	+	-	-	+
Y	+	+	-	-	+	+	-	-
Z	+	+	+	+	-	-	-	-

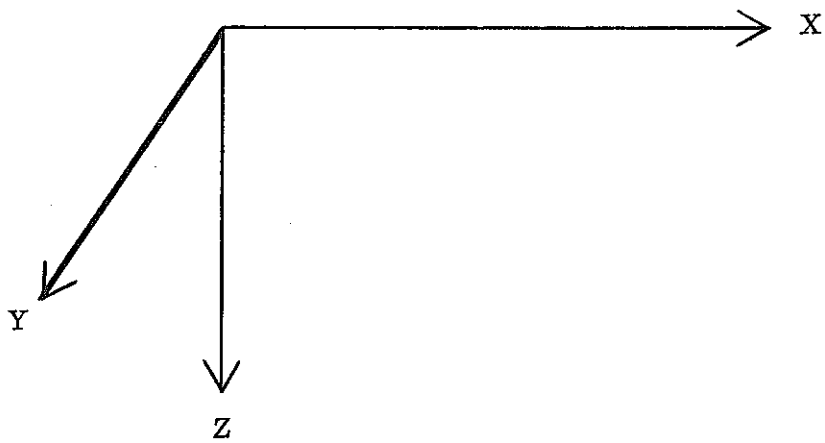
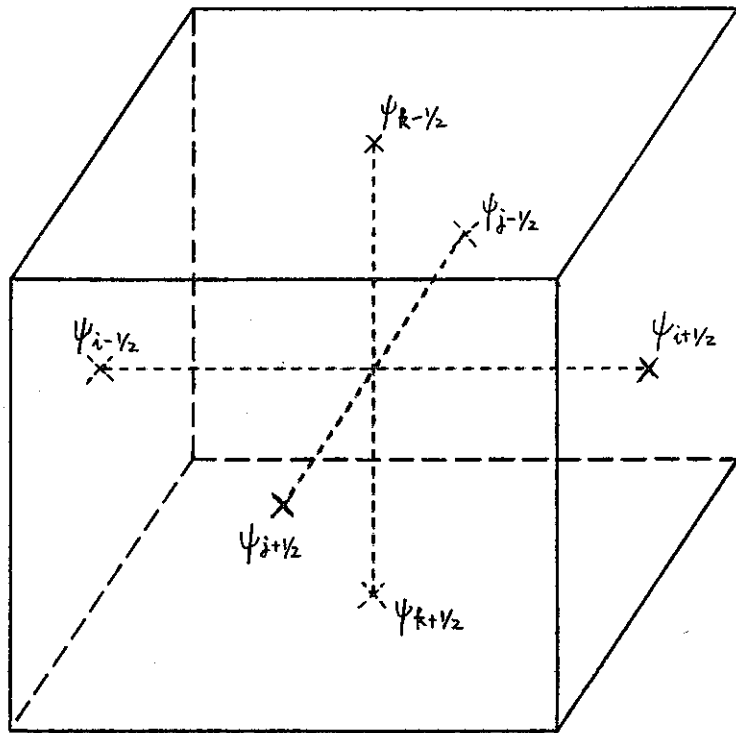


Fig. 2 - 1 Location of $\psi_i \pm \frac{1}{2}$, $\psi_j \pm \frac{1}{2}$ and $\psi_k \pm \frac{1}{2}$

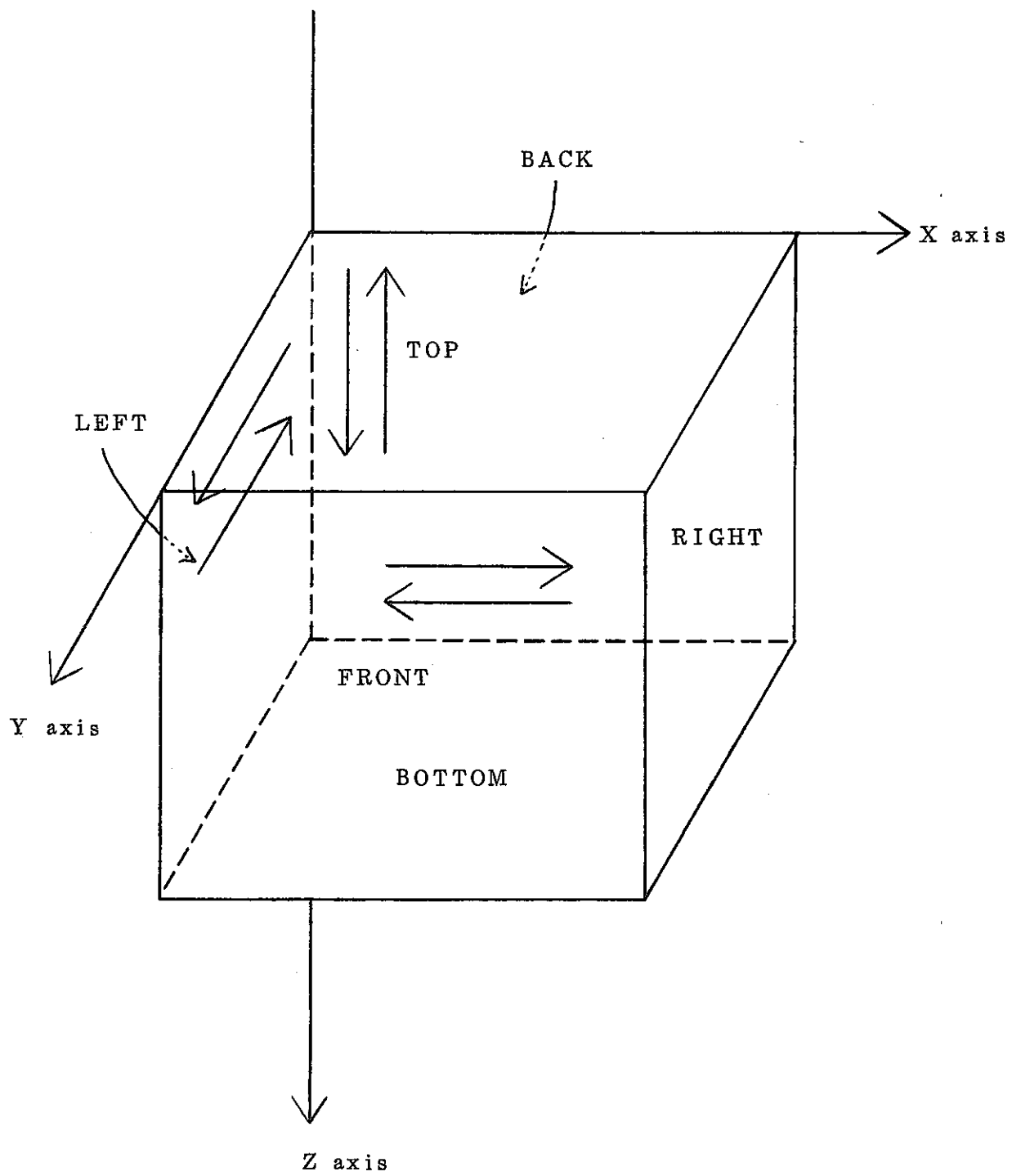
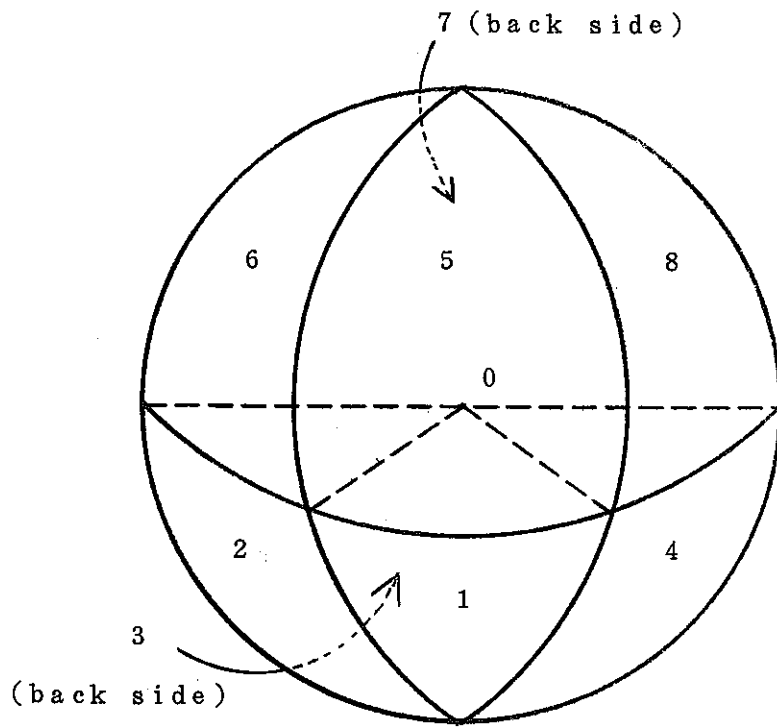


Fig. 2-2 Specification of surface

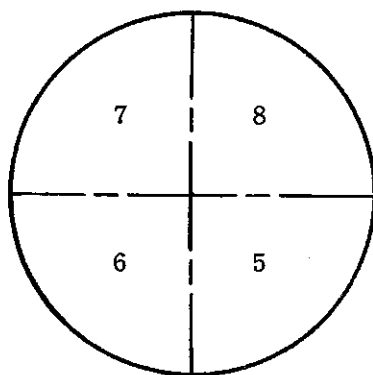
0.63 E-2	0.98	1.78	1.97	R E F L E C T O R
0.68 E-2	0.87	2.24	1.81	
(-7.3 %)	(12.6)	(-20.6)	(8.8)	
	1.27	1.94	1.94	
	1.02	2.26	1.74	
	(24.5)	(-14.1)	(11.5)	
Eigenvalue		1.82	1.58	
1.0146		2.23	1.53	
1.0380		(-18.4)	(3.27)	
(-2.3)				
THOTH			1.18	
CITATION			0.99	
(%-DIF.)			(19.2)	
R E F L E C T O R				

Table 2-3 Comparison of Power Distributions



Definition of each quadrant and origin

Upper view



Lower view

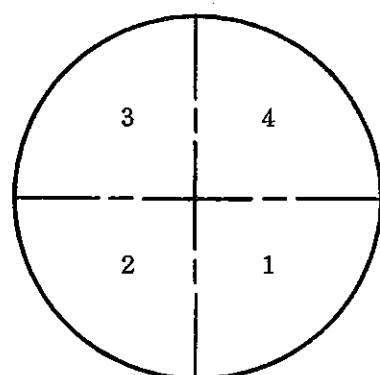


Fig. 2-3 Specification of angular direction

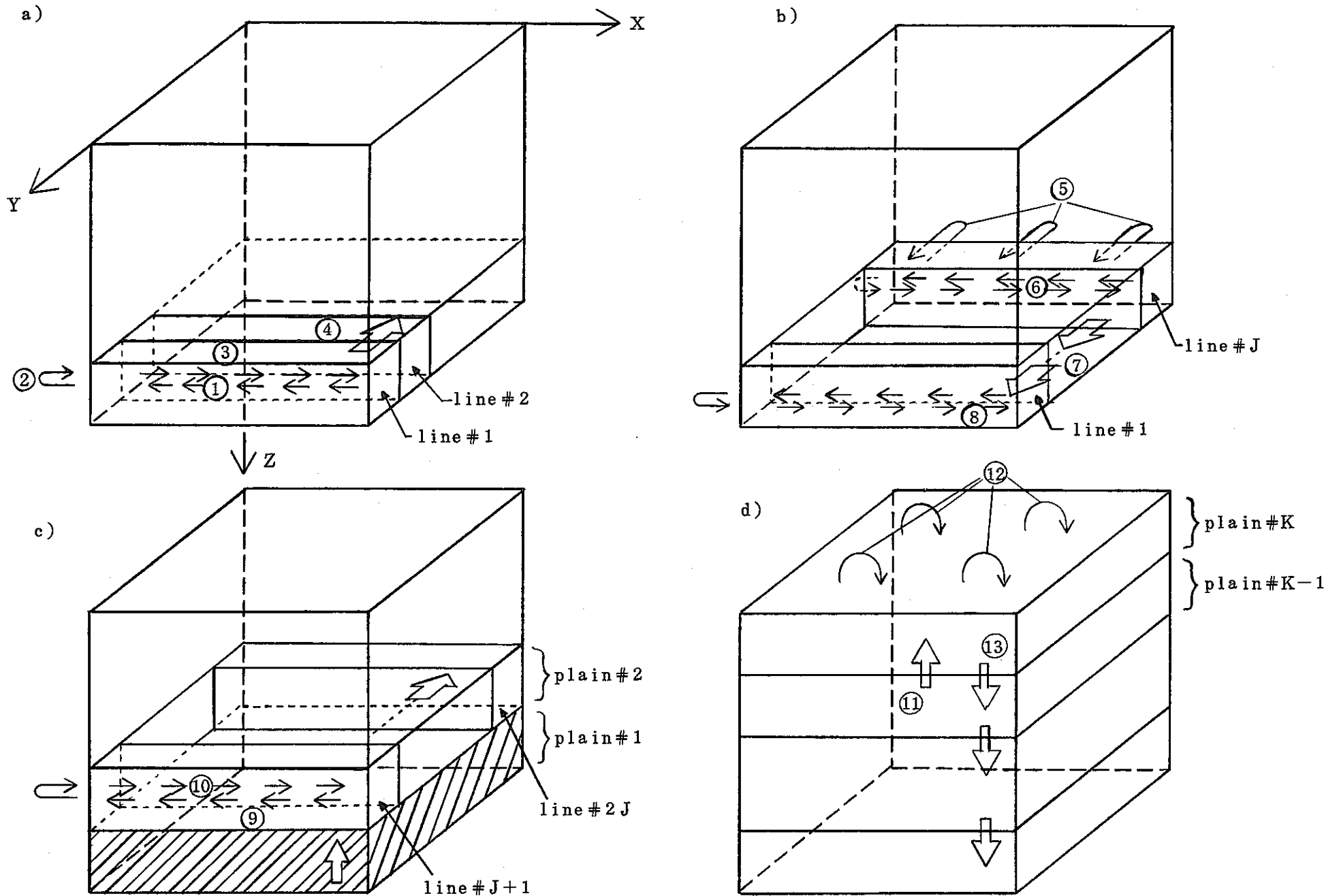


Fig. 2-4 Inner sweep procedure in THOTH

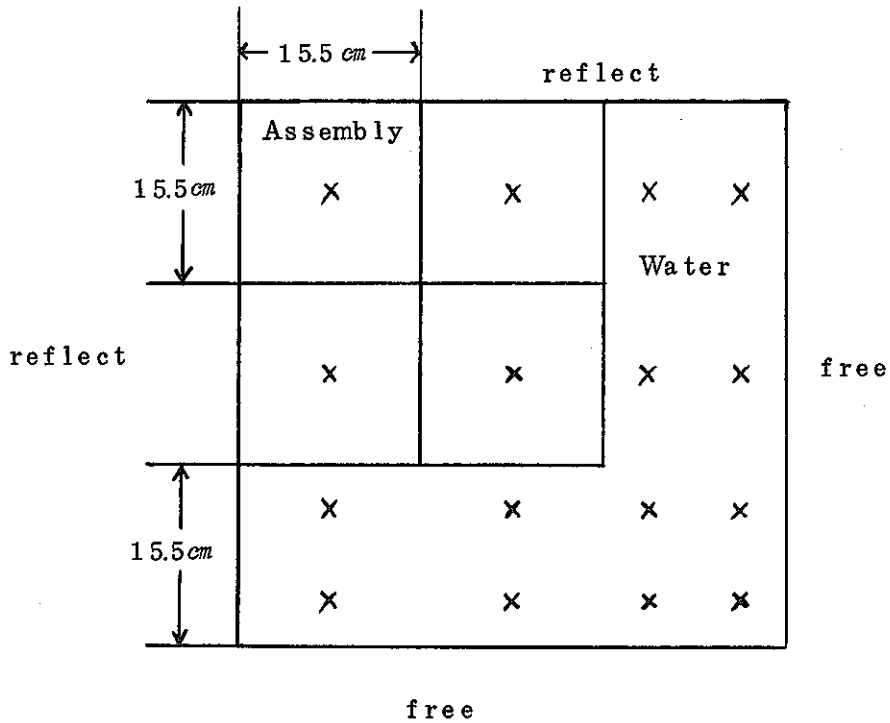
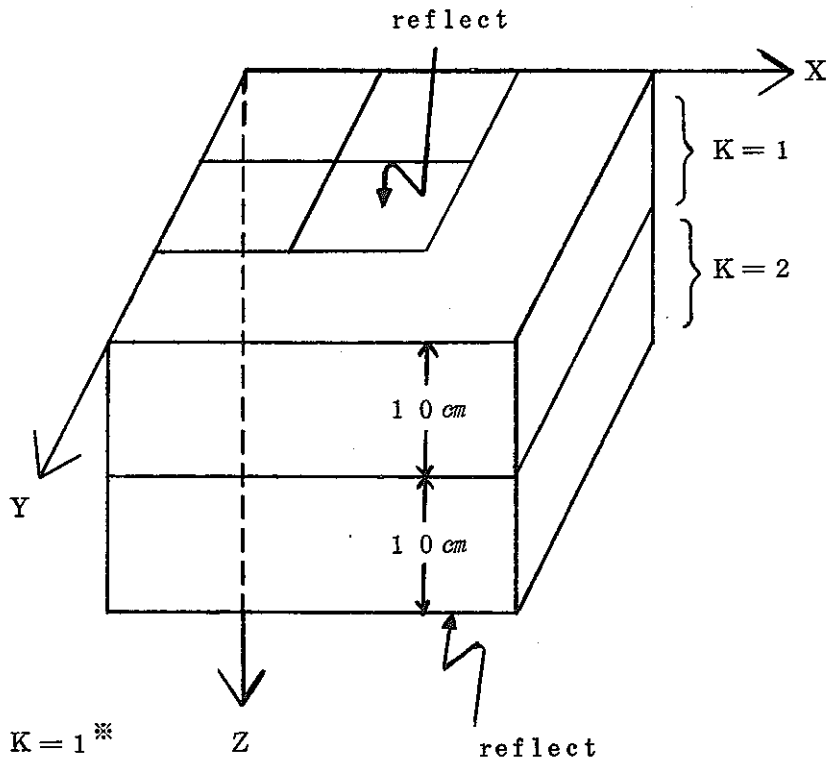


Fig. 2 - 5 Two-dimensional core model
for 3 group test calculation

* For K=2, same as K=1

3. 中性子輸送計算のための拡散合成加速

Sn法で中性子輸送方程式を解く場合、異なる方向の角度依存中性子束（以後、angular flux と書く）は、与えられたソース分布（原子炉で扱う問題では普通等方分布である）に対して、別々に計算されるためにそれらを全ての角度にわたり総和したスカラー量の中性子束（angular flux に対して scalar flux という）の収束の仕方は、拡散方程式の場合に比べて極めて緩慢である。この scalar flux の収束の緩慢さは、直接にソース分布の収束を遅くし、間接的には固有値が、だらだらと変動を続け、いつまでも収束しないという事態を招く。

拡散合成法（diffusion synthetic acceleration method）はこの欠点を克服するための一つの手段である。

この方法を説明するために、まず Sn 法の計算手順を簡単に次式で表わすことにする。

$$\vec{\Omega} \cdot \nabla \tilde{\psi}^{\ell} + \Sigma_t \tilde{\psi}^{\ell} = \Sigma_s \phi^{\ell} + Q \quad (3-1)$$

$$\phi^{\ell+1} = \tilde{\phi}^{\ell} \equiv \int d\Omega \tilde{\psi}^{\ell} \quad (3-2)$$

(3-1) 式で $\tilde{\psi}^{\ell}$ は、 ℓ 回目の Sn 計算で計算される変数であり、正しくはエネルギー群、場所及び方向を添字で明記すべきであるが、見やすくするため省略してある。以後 $\tilde{\psi}^{\ell}$ はすべてのエネルギー群、すべての場所、すべての方向を代表して表わすものとする。

(3-1) 式を解いて得た $\tilde{\psi}^{\ell}$ から (3-2) 式により次回 ($\ell+1$ 回目) の (3-1) 式右辺のソースに使う $\phi^{\ell+1}$ を計算する。

ここで (3-1) 式の Po モーメント（すなわち $\int d\Omega$ の積分）をとると

$$\nabla \cdot \tilde{\mathbf{J}}^{\ell} + \Sigma_t \tilde{\phi}^{\ell} = \Sigma_s \phi^{\ell} + Q \quad (3-3)$$

を得る。ここで $\tilde{\mathbf{J}}^{\ell} \equiv \int d\Omega \vec{\Omega} \tilde{\psi}^{\ell}$ は中性子流を表わす。(3-3) 式において

$$\tilde{\mathbf{J}}^{\ell} = -D \nabla \tilde{\phi}^{\ell} \quad \left(D = \frac{1}{3\Sigma_{tr}} \right)$$

とするのが拡散理論であるが、当然のことながら輸送理論とは一致しない。しかし、輸送理論の中性子流 \vec{j} を

$$\vec{j} = -D \nabla \phi + \vec{R} \quad (\vec{R}: \text{補正項})$$

と表わすことはできる。ところで(3-1)式のP₁モーメント(すなわち $\int \vec{\Omega} d\Omega$ の積分)をとることによって、次式が得られる。

$$\frac{1}{3} \nabla \tilde{\phi}^l + \tilde{\gamma}^l + \Sigma_t \tilde{j}^l = 0 \quad (3-4)$$

ここで $\tilde{\gamma}^l$ は $\tilde{\phi}^l$ には強く依存しない項である。これから

$$\tilde{j}^l = -D^* \nabla \tilde{\phi}^l + \tilde{R}^l \quad \left(\begin{array}{l} D^* \equiv \frac{1}{3\Sigma_t} \\ \tilde{R}^l \equiv \frac{-\tilde{\gamma}^l}{\Sigma_t} \end{array} \right)$$

が導びかれ(3-3)式に用いると、輸送方程式は

$$-\nabla \cdot D^* \nabla \tilde{\phi}^l + (\Sigma_t - \Sigma_s) \tilde{\phi}^l = Q - \nabla \cdot \tilde{R}^l - \Sigma_s (\tilde{\phi}^l - \phi^l) \quad (3-5)$$

と書き直すことができる。いかえれば、この式は(3-1)式を解いて得た $\tilde{\psi}^l$ から計算した新しい $\tilde{\phi}^l$ が満たすべき式である。

$\ell \rightarrow \infty$ (収束時)では $\tilde{\phi}^l = \phi^l$ であるが、当然ながら $\tilde{R}^l \rightarrow 0$ ではない。

そこで、 $\ell \rightarrow \infty$ で $\tilde{\phi}^l = \phi^l$ となることを見込んで次の方程式を考える。

$$-\nabla \cdot D^* \nabla \phi^* + \Sigma_R \phi^* = Q - \nabla \cdot \tilde{R}^l \quad (3-6)$$

ここでは、もはや $\tilde{\phi}^l$ は上の式を満たさないので ϕ^* と、別の変数に置き換えられている。当然ながら $\ell \rightarrow \infty$ で $\phi^* = \phi^l = \tilde{\phi}^l$ 。この式は \tilde{R}^l が求まったあと ($\tilde{R}^l = \tilde{j}^l + D^* \nabla \tilde{\phi}^l$) について解ける式である。

ここで、もし \tilde{R}^{ℓ} の ℓ による変化が $\tilde{\Phi}^{\ell}$ に比べて少ないとすれば、(3-6) 式の解 ϕ^* は近似的に (3-1) 式の収束値のように見なせるから次回 ($\ell + 1$ 回目) の反復計算において ϕ^* を使う方が、 $\tilde{\Phi}^{\ell}$ を使うより有利である。実際 \tilde{R}^{ℓ} は一次元 slab 体系の場合 $\frac{2}{3} \nabla \tilde{\Phi}_2^{\ell}$ ($\Phi_2 = \frac{1}{4\pi} \int_{-1}^1 P_2(\mu) \psi^{\ell}(\mu) d\mu$) となり原子炉のような (拡散的な) 問題では $\tilde{\Phi}_2^{\ell} = 0$ となる領域が多いから有利である。また、仮りに \tilde{R}^{ℓ} の変動が大きくても (3-6) 式は収束の極限でのバランス $\phi^{\ell} = \tilde{\Phi}^{\ell}$ を包含した方程式であるから、真値への漸近を早めることが期待できる。これが拡散合成法の原理であり、ここで紹介したものをその中でも特にソース補正法と呼ぶ。

(3-6) 式を直接解いて解 ϕ^* を $\phi^{\ell+1}$ として用いる方法もあるが、その場合 $\nabla \tilde{R}^{\ell}$ 項の計算に微分が含まれているので、加速の計算がメッシュのとり方に敏感になってくる。そこで今度の計算では (3-6) 式の代わりにこれと等価な (3-6) 式から (3-5) 式を差引いた

$$-\nabla \cdot \mathbf{D}^* \nabla f^{\ell+1} + \Sigma_R f^{\ell+1} = \Sigma_S (\tilde{\Phi}^{\ell} - \phi) \quad (3-7)$$

を用いた。ここで f は

$$\phi^{\ell+1} = \tilde{\Phi}^{\ell} + f^{\ell+1} \quad (3-8)$$

である。すなわち (3-6) 式でなく (3-7) 式を解いて $f^{\ell+1}$ を求め、それから $\phi^{\ell+1}$ を (3-8) 式で求める方法である。こうすることにより (3-6) 式を具体的に数値計算する場合に大きな問題となる、輸送と拡散の差分の違いに基づく加速の不安定性をいくぶんかは確実に取除くことができる。

3-1. 2次元XY体系での差分式

拡散合成法による輸送計算の加速において安定な加速結果を得るには輸送方程式と拡散型加速方程式の間の整合性が必要とされる。これはAlcouffeによって最初に指摘され、離散化した加速拡散方程式の導出は離散化した輸送方程式をその差分の構造を保ちつつ、変形することによって達成される。以下に1次元slab体系でのLarsenの方法を2次元XY体系に拡張し、加速方程式を導出する。(3-1)式に対応して離散化した輸送釣合方程式は

$$\begin{aligned} & \frac{\mu_m}{h_i} (\Psi_{m,i+\frac{1}{2},j}^{\ell+\frac{1}{2}} - \Psi_{m,i-\frac{1}{2},j}^{\ell+\frac{1}{2}}) + \frac{\eta_m}{h_j} (\Psi_{m,i,j+\frac{1}{2}}^{\ell+\frac{1}{2}} - \Psi_{m,i,j-\frac{1}{2}}^{\ell+\frac{1}{2}}) + \sigma_{t,i,j} \Psi_{m,i,j}^{\ell+\frac{1}{2}} \\ & = \sum_{n=0}^N (2n+1) \sigma_{s,n,i,j} \sum_{k=0}^n R_{n,m}^k \phi_{n,i,j}^{k,\ell} + Q_{m,i,j} \end{aligned} \quad (3-9)$$

ここで Ψ_m は方向mの中性子束で(i, j)は空間格子を表わし、半整数は格子境界上の量であることを示す。 ℓ は前と同様に反復回数を示す。 μ_m, η_m は方向mの単位ベクトル Ω_m の、xおよびy方向の成分である。 $R_{n,m}^k$ は方向mに対する球面調和関数値、 $\phi_{n,i,j}^k$ は中性子束の角度モーメントで

$$\phi_{n,i,j}^k = \sum_m R_{nm}^k \Psi_{m,i,j} W_m \quad (3-10)$$

W_m は方向mの荷重であり、次のように規格化されている。

$$\sum_m W_m = 1 \quad (3-11)$$

ダイヤモンド差分式は

$$\Psi_{m,i,j}^{\ell+\frac{1}{2}} = \frac{1}{2} (\Psi_{m,i+\frac{1}{2},j}^{\ell+\frac{1}{2}} + \Psi_{m,i-\frac{1}{2},j}^{\ell+\frac{1}{2}}) \quad (3-12a)$$

$$= \frac{1}{2} (\Psi_{m,i,j+\frac{1}{2}}^{\ell+\frac{1}{2}} + \Psi_{m,i,j-\frac{1}{2}}^{\ell+\frac{1}{2}}) \quad (3-12b)$$

である。

加速の式は P₁ 方程式から導かれる。これは拡散近似が P₁ 近似から導かれたのと同様である。そこで (3-9) 式の次の三つのモーメントをとる。

$$L_0^0 = \sum_m R_{0m}^0 W_m = \sum_m W_m \quad (3-13 a)$$

$$L_1^0 = \sum_m R_{1m}^0 W_m = \sum_m W_m \mu_m \quad (3-13 b)$$

$$L_1^1 = \sum_m R_{1m}^1 W_m = \sum_m W_m \eta_m \quad (3-13 c)$$

得られる三つの式の中には二次以上の高次の中性子束のモーメントが現われるが、それらの項は P₁ 近似が輸送厳密解となるような状況ではゼロとなる。加速の過程で外挿する量はこれら以外のゼロ次及び一次のモーメントだけである。これらの量は次の内側反復のソースとして用いられるので、添字を $(\ell + \frac{1}{2})$ あるいは ℓ から $(\ell + 1)$ に置き換えた式からもとの式を差し引くと高次モーメントは消え次式を得る。

$$\begin{aligned} & \frac{1}{h_i} (f_{1,i+\frac{1}{2},j}^{0,\ell+1} - f_{1,i-\frac{1}{2},j}^{0,\ell+1}) + \frac{1}{h_j} (f_{1,i,j+\frac{1}{2}}^{0,\ell+1} - f_{1,i,j-\frac{1}{2}}^{0,\ell+1}) \\ & + (\sigma_{t ij} - \sigma_{s0 ij}) (\phi_{0,ij}^{0,\ell+1} - \phi_{0,ij}^{0,\ell}) = \sigma_{s0 ij} (\phi_{0,ij}^{0,\ell+\frac{1}{2}} - \phi_{0,ij}^{0,\ell}) \end{aligned} \quad (3-14 a)$$

$$\begin{aligned} & \frac{1}{3h_i} (f_{0,i+\frac{1}{2},j}^{0,\ell+1} - f_{0,i-\frac{1}{2},j}^{0,\ell+1}) + (\sigma_{t ij} - \sigma_{s1 ij}) (\phi_{1,ij}^{0,\ell+1} - \phi_{1,ij}^{0,\ell+\frac{1}{2}}) \\ & = \sigma_{s1 ij} (\phi_{1,ij}^{0,\ell+\frac{1}{2}} - \phi_{1,ij}^{0,\ell}) \end{aligned} \quad (3-14 b)$$

$$\begin{aligned} & \frac{1}{3h_j} (f_{0,i+\frac{1}{2},j}^{0,\ell+1} - f_{0,i-\frac{1}{2},j}^{0,\ell+1}) + (\sigma_{t ij} - \sigma_{s1 ij}) (\phi_{1,ij}^{1,\ell+1} - \phi_{1,ij}^{1,\ell+\frac{1}{2}}) \\ & = \sigma_{s1 ij} (\phi_{1,ij}^{1,\ell+\frac{1}{2}} - \phi_{1,ij}^{1,\ell}) \end{aligned} \quad (3-14 c)$$

ここで f は

$$\phi_{n, i+\frac{1}{2}, j}^{k, l+1} = \phi_{n, i+\frac{1}{2}, j}^{k, l+\frac{1}{2}} + f_{n, i+\frac{1}{2}, j}^{k, l} \quad (n=0, 1: 0 \leq k \leq n) \quad (3-15)$$

等を満たす格子境界上の量である。格子平均の中性子束の値は (3-12) 式に、輸送バランス式に用いたと同じ操作をすることによって

$$\phi_{0ij}^{0, l+1} = \phi_{0ij}^{0, l+\frac{1}{2}} + \frac{1}{2} (f_{0, i+\frac{1}{2}, j}^{0, l+1} + f_{0, i-\frac{1}{2}, j}^{0, l+1}) \quad (3-16a)$$

$$= \phi_{0ij}^{0, l+\frac{1}{2}} + \frac{1}{2} (f_{0, i, j+\frac{1}{2}}^{0, l+1} + f_{0, i, j-\frac{1}{2}}^{0, l+1}) \quad (3-16b)$$

$$\phi_{1ij}^{0, l+1} = \phi_{1ij}^{0, l+\frac{1}{2}} + \frac{1}{2} (f_{1, i+\frac{1}{2}, j}^{0, l+1} + f_{0, i-\frac{1}{2}, j}^{0, l+1}) \quad (3-16c)$$

$$\phi_{1ij}^{1, l+1} = \phi_{1ij}^{1, l+\frac{1}{2}} + \frac{1}{2} (f_{1, i, j+\frac{1}{2}}^{1, l+1} + f_{1, i, j-\frac{1}{2}}^{0, l+1}) \quad (3-16d)$$

より求まる。 $\phi_n^{k, l+\frac{1}{2}}$ は加速の段階では既知である。

(3-14) 式および (3-16) 式の 7 つの式が加速の式である。通常の拡散合成法ではこれらの式から f_1^0 及び f_1^1 を消去し、 f_0^0 に対する拡散型方程式を導き、その解を用いて (3-16) 式から中性子束モーメントの外挿を行う。しかし、得られた拡散型方程式には隣接する 4 格子 ((i, j) , $(i+1, j)$, $(i, j+1)$, $(i+1, j+1)$) のすべての格子境界上の変数が含まれているため (2次元 XY 体系では 12 個、3次元 XYZ 体系では 54 個)、差分に関する新たな近似を入れて格子の角の点に計算点を移さなければ通常の拡散コードが使用できない。(Fig 3-1 参照) この新たな差分の近似を入れることにより加速の式と輸送方程式の間の整合性は悪くなるが、とりあえずこの方式を用いる。加速拡散方程式は以下のようになる。

$$\begin{aligned}
& -\frac{1}{h_{i+\frac{1}{2}}} \left\{ \bar{D}_x^{i+\frac{1}{2}, j+\frac{1}{2}} (f_{0, i+\frac{3}{2}, j+\frac{1}{2}}^{0, l+1} - f_{0, i+\frac{1}{2}, j+\frac{1}{2}}^{0, l+1}) - \bar{D}_x^{i, j+\frac{1}{2}} (f_{0, i+\frac{1}{2}, j+\frac{1}{2}}^{0, l+1} - f_{0, i-\frac{1}{2}, j+\frac{1}{2}}^{0, l+1}) \right\} \\
& -\frac{1}{h_{j+\frac{1}{2}}} \left\{ \bar{D}_y^{i+\frac{1}{2}, j+1} (f_{0, i+\frac{1}{2}, j+\frac{3}{2}}^{0, l+1} - f_{0, i+\frac{1}{2}, j+\frac{1}{2}}^{0, l+1}) - \bar{D}_y^{i+\frac{1}{2}, j} (f_{0, i+\frac{1}{2}, j+\frac{1}{2}}^{0, l+1} - f_{0, i+\frac{1}{2}, j-\frac{1}{2}}^{0, l+1}) \right\} \\
& + \bar{\sigma}_{R, i+\frac{1}{2}, j+\frac{1}{2}} f_{0, i+\frac{1}{2}, j+\frac{1}{2}}^{0, l+1} = \frac{1}{4h_{i+\frac{1}{2}}h_{j+\frac{1}{2}}} \left(g_{0, i, j}^{0, l+\frac{1}{2}} + g_{0, i+\frac{1}{2}, j}^{0, l+\frac{1}{2}} + g_{0, i, j+\frac{1}{2}}^{0, l+\frac{1}{2}} + g_{0, i+\frac{1}{2}, j+\frac{1}{2}}^{0, l+\frac{1}{2}} \right) \\
& -\frac{1}{2h_{i+\frac{1}{2}}h_{j+\frac{1}{2}}} \left\{ \left(g_{1, i+\frac{1}{2}, j+1}^{0, l+\frac{1}{2}} + g_{1, i+\frac{1}{2}, j}^{0, l+\frac{1}{2}} \right) - \left(g_{1, i, j+1}^{0, l+\frac{1}{2}} + g_{1, i, j}^{0, l+\frac{1}{2}} \right) \right\} \\
& -\frac{1}{2h_{i+\frac{1}{2}}h_{j+1}} \left\{ \left(g_{1, i+\frac{1}{2}, j+1}^{1, l+\frac{1}{2}} + g_{1, i, j+1}^{1, l+\frac{1}{2}} \right) - \left(g_{1, i+\frac{1}{2}, j}^{1, l+\frac{1}{2}} + g_{1, i, j}^{1, l+\frac{1}{2}} \right) \right\} \quad (3-17)
\end{aligned}$$

さて

$$h_{i+\frac{1}{2}} = \frac{1}{2}(h_i + h_{i+1}), \quad h_{j+\frac{1}{2}} = \frac{1}{2}(h_j + h_{j+1}) \quad (3-18 a, b)$$

$$\bar{D}_x^{i, j+\frac{1}{2}} = \frac{1}{2h_{j+\frac{1}{2}}} (\bar{D}_{i, j+1} h_{j+1} + \bar{D}_{i, j} h_j) \quad (3-18 c)$$

$$\bar{D}_y^{i+\frac{1}{2}, j} = \frac{1}{2h_{i+\frac{1}{2}}} (\bar{D}_{i+\frac{1}{2}, j} h_{i+1} + \bar{D}_{i, j} h_i) \quad (3-18 d)$$

$$\bar{D}_{ij} = \frac{1}{3(\sigma_{t, ij} - \sigma_{s, ij})} \quad (3-18 e)$$

$$\bar{\sigma}_{R, i+\frac{1}{2}, j+\frac{1}{2}} = \frac{1}{4h_{i+\frac{1}{2}}h_{j+\frac{1}{2}}} (\sigma_{R, ij} h_i h_j + \sigma_{R, i+\frac{1}{2}, j} h_{i+\frac{1}{2}} h_j + \sigma_{R, i, j+\frac{1}{2}} h_i h_{j+\frac{1}{2}} + \sigma_{R, i+\frac{1}{2}, j+\frac{1}{2}} h_{i+\frac{1}{2}} h_{j+\frac{1}{2}}) \quad (3-18 f)$$

$$\sigma_{R, ij} = \sigma_{t, ij} - \sigma_{s, ij} \quad (3-18 g)$$

$$g_{0, i, j}^{0, l+\frac{1}{2}} = \sigma_{s, ij} (\phi_{0, i, j}^{0, l+\frac{1}{2}} - \phi_{0, i, j}^{0, l}) h_i h_j \quad (3-19 a)$$

$$g_{1ij}^{0,l+\frac{1}{2}} = \frac{\sigma_{s1ij}}{\sigma_{t1ij} - \sigma_{s1ij}} (\phi_{1ij}^{0,l+\frac{1}{2}} - \phi_{1ij}^{0,l}) h_j \quad (3-19b)$$

$$g_{1ij}^{1,l+\frac{1}{2}} = \frac{\sigma_{s1ij}}{\sigma_{t1ij} - \sigma_{s1ij}} (\phi_{1ij}^{1,l+\frac{1}{2}} - \phi_{1ij}^{1,l}) h_i \quad (3-19c)$$

であり、また(3-16)式の代わりに次式を用いる。

$$\phi_{0ij}^{0,l+1} = \phi_{0ij}^{0,l+\frac{1}{2}} + \frac{1}{4} (f_{0,i+\frac{1}{2},j+\frac{1}{2}}^{0,l+1} + f_{0,i+\frac{1}{2},j-\frac{1}{2}}^{0,l+1} + f_{0,i-\frac{1}{2},j+\frac{1}{2}}^{0,l+1} + f_{0,i-\frac{1}{2},j-\frac{1}{2}}^{0,l+1}) \quad (3-20a)$$

$$\phi_{1ij}^{0,l+1} = \phi_{1ij}^{0,l+\frac{1}{2}} - \frac{D_{ij}}{2h_i} (f_{0,i+\frac{1}{2},j+\frac{1}{2}}^{0,l+1} + f_{0,i+\frac{1}{2},j-\frac{1}{2}}^{0,l+1} - f_{0,i-\frac{1}{2},j+\frac{1}{2}}^{0,l+1} - f_{0,i-\frac{1}{2},j-\frac{1}{2}}^{0,l+1}) \quad (3-20b)$$

$$\phi_{1ij}^{1,l+1} = \phi_{1ij}^{1,l+\frac{1}{2}} - \frac{D_{ij}}{2h_j} (f_{0,i+\frac{1}{2},j+\frac{1}{2}}^{0,l+1} + f_{0,i-\frac{1}{2},j+\frac{1}{2}}^{0,l+1} - f_{0,i+\frac{1}{2},j-\frac{1}{2}}^{0,l+1} - f_{0,i-\frac{1}{2},j-\frac{1}{2}}^{0,l+1}) \quad (3-20c)$$

内側反復計算は、まず(3-9)式および(3-12)式から $\psi_{mij}^{l+\frac{1}{2}}$ を計算し(3-19)式の拡散、ソース項の計算を経て(3-17)式で外挿量 $f_{0,i+\frac{1}{2},j+\frac{1}{2}}^{0,l+\frac{1}{2}}$ を求め、最後に(3-20)式で $\phi_{0ij}^{0,l+1}$, $\phi_{1ij}^{0,l+1}$, $\phi_{1ij}^{1,l+1}$ を外挿することにより閉じる。もし $\phi_{0ij}^{0,l+1}$ が負になることがあれば、その回は外挿をやめて、回目の(3-9)式右辺ソース項には $\phi_{0ij}^{0,l+1}$ に代えて $\phi_{0ij}^{0,l+\frac{1}{2}}$ そのものを用いる。

内側反復で、自群散乱の変動から外挿量 f_0^0 を解く式(3-17)式を求めたが、同様に核分裂の変動から外側反復の加速拡散方程式を新たに導き検証した。外側反復を表わす輸送方程式を

$$\vec{\Omega} \cdot \nabla \tilde{\Psi}_g^k + \sigma_{tg} \tilde{\Psi}_g^k = \sum_{g'} \int \sigma_{sg' \rightarrow g} \tilde{\Psi}_{g'}^k d\Omega + \frac{\chi_g}{k_{eff}^k} \sum_{g'} \nu \sigma_{fg'} \phi_{g'}^k \quad (3-21)$$

と表わすと、この度導出した外側反復の加速の式は

$$\begin{aligned}
 & -\nabla \cdot D_g \nabla f_g^{k+1} + \sigma_{Rg} f_g^{k+1} \\
 & = \sum_{g' \neq g} \sigma_{s_{g' \rightarrow g}} f_{g'}^{k+1} + \chi_g \left(\frac{\sum_{g'} \nu \sigma_{fg'} \tilde{\phi}_{g'}^k}{\tilde{k}_{\text{eff}}^k} - \frac{\sum_{g'} \nu \sigma_{fg'} \phi_{g'}^k}{k_{\text{eff}}^k} \right) \quad (3-22)
 \end{aligned}$$

で与えることができる。ここで $\tilde{\psi}^k$, $\tilde{\phi}^k$ は k 回目の S_n 外側反復で求められる量であり、 \tilde{k}_{eff}^k はその直後に次式で計算される更新された固有値である。

$$\tilde{k}_{\text{eff}}^k = \frac{\int dV \sum_{g'} \nu \sigma_{fg'} \tilde{\phi}_{g'}^k}{\frac{1}{k_{\text{eff}}^k} \int dV \sum_{g'} \nu \sigma_{fg'} \phi_{g'}^k} \quad (3-23)$$

これをもとに (3-22) 式の核分裂変動項を所与の値として、多群拡散方程式の固定ソース問題を解く、求まった f_g^{k+1} から次式により中性子束及び固有値を外挿する。

$$\phi_g^{k+1} = \tilde{\phi}_g^k + f_g^{k+1} \quad (3-24)$$

$$k_{\text{eff}}^{k+1} = \frac{\int dV \sum_{g'} \nu \sigma_{fg'} \phi_{g'}^{k+1}}{\frac{1}{\tilde{k}_{\text{eff}}^k} \int dV \sum_{g'} \nu \sigma_{fg'} \tilde{\phi}_{g'}^k} \quad (3-25)$$

この ϕ_g^{k+1} 及び k_{eff}^{k+1} が次回 ($k+1$) 回目の外側反復で (3-21) 式ソース項に用いられる。ここでも内側反復の場合と同様に ϕ^{k+1} が負になった場合は外挿をやめ $\tilde{\phi}^k$ として用いる。

3-2. 計算結果

(Fig.3-2) および (Table 3-1) にテスト計算に用いた体系および断面積を示す。計算は拡散合成法による加速について、1) 内側反復及び外側反復の加速を用いた場合、2) 内側反復の加速のみを用いた場合および 3) 基準計算として TWOTRAN コードによる再釣合加速法を用いた場合について行なった。

結果を (Table 3-2) に示す。収束精度はすべて $|\Delta k_{\text{eff}}/k_{\text{eff}}| < 10^{-5}$ である。加速法が異なること以外はすべての条件を同一にしてある。拡散合成法では内側反復でとく拡散方程式の最大反復回数は 5 回とした。外側反復でとく多群拡散方程式はこの体系の場合、上方散乱がなく、固定ソース問題であるのでそれ自身のための外側反復を必要としない。また、外側反復と内側反復の両方を行なった計算では、1 回目の S_n 計算を始める前に多群拡散固有値問題を $|\Delta k_{\text{eff}}/k_{\text{eff}}| < 10^{-4}$ まで解き、その拡散解を S_n 計算の初期値として用いている。これに要した時間も計算時間に含まれている。

(Fig 3-3) に計算時間 (cpu time) に対する固有値 k_{eff} の収束の様子を示す。各外側反復毎に固有値をプロットしてある。(Fig 3-3) から以下の事がわかる。

- 1) 拡散合成法では 1 回の外側反復での変動分 Δk_{eff} が反復初期において大きいため真値への近づき方が早い。しかし k_{eff} が収束するほど変動分は減少し、激しく振動するようことはない。このことは拡散合成法がつねに中性子バランスのとれた加速の式を扱っていることに因ると考えられる。
- 2) 今回導入した外側反復の加速法は、初期値に拡散解を用いたことによる効果を除いて考えても有効であり、1) で述べた性質をより強める働きをすることがわかる。
- 3) 計算に要した時間は、拡散合成法、再釣合法ともほぼ外側反復の回数で判断することができるようである。
- 4) これは具体的には示していないが、3つの計算のいずれも結果で得られる中性子束は各メッシュとも 10^{-8} 以内で一致した。

以上の点から拡散合成法の二次元 XY 体系への応用が有効であることが一応確認されたので、今後、二次元体系に関していっそう研究を進めてゆくとともに、三次元体系への適用性についても研究してゆく予定である。

4. 結言および今後の問題点

高速増殖炉炉心核特性を精度よく予測するために中性子輸送効果を考慮する三次元輸送コードの開発を行なった。中性子輸送効果については、拡散計算（CITATIONコード）との計算結果の比較により示された。輸送計算（Sn）における中性子束分布および固有値の収束性の悪さを改善するために、種々の再釣合法の検討を行なった。

この結果、fine-mesh-rebalancing法とsystem-rebalancing法を、周期的に使い分ける方法が有効であることがわかった。この方法については、テストモデルでは有効性を確認できたが、大型炉に対しては、まだ十分に有効性を確認していない。このため、今後は大型炉を含む種々の炉心に対して再釣合法による中性子束分布や固有値の変化を十分に検討し、最適な加速が行なわれるようにする必要がある。

また、新しい加速法として拡散合成法を二次元XY体系で定式化して、テストを行なった。テストモデルに対してはTWOTRANコードに対して、10倍以上速く収束することがわかった。これも種々の炉心に適用して、その効果を確認し、三次元XYZ体系への拡張を行なう必要がある。今後、この加速法を今回開発した三次元輸送コードに組み込み、大型炉の計算を行なうことにより、その有効性を検証してゆく予定である。

謝 辞

本研究を進めるに当り、有益な情報を提供され、御助言を頂いた動力炉・核燃料開発事業団高速増殖炉開発本部の白方・大谷の各氏に深く感謝の意を表する次第である。

参 考 文 献

- (1) K. Lathrop and F. Brinkley, LA-4848-MS (1973)
- (2) K. Lathrop , LA-6333-MS (1976)
- (3) R. E. Alcouffe , J. Nucl. Sci. Eng. , 64, 344 (1977)
- (4) E. W. Larsen , J. Nucl. Sci. Eng. , 82, 64 (1982)
- (5) E. W. Larsen , J. Nucl. Sci. Eng. , 82, 47 (1982)

Appendix A

再釣合法の原理

再釣合法は、計算体系をいくつかの領域に分け、その各領域の中性子束バランスを強制的にとることによって、中性子束分布をより収束解に近づける、という手法である。具体的には、各領域の中性子束にリバランスファクターを乗じて、各領域の中性子束バランスをとり、そのリバランスファクターを乗じた後の中性子束を新しい中性子束として用いる。内側反復計算においては、各群でのソース分布が与えられており、下のようなバランス式を解くのが目的である。

$$-\sum_i^8 S_i L_{i \rightarrow c}^{\ddagger} + \sum_i^8 S_i L_{c \rightarrow i}^{\ddagger} + \sum_{\text{trc}}^{\ddagger} \phi_c^{\ddagger} V_c = Q_c^{\ddagger} V_c \quad (\text{A-1})$$

ただし、 $L_{i \rightarrow c}$ は、左右上下前後の8つの近接する領域 i から中心 (center) の領域へ流れ込む部分中性子束を示している。 S_i 、 V_c 、 Q_c はそれぞれ近接する領域 i との境界の面積、中心領域の体積、およびソースである。一般に未収束の中性子束では (A-1) 式を満たさないので、中心および近接する8つの中性子束にリバランスファクターを乗じて、その結果、中性子束バランスがとれたと仮定する。

$$L_{i \rightarrow c}^{\ddagger} \propto \phi_i^{\ddagger}, \quad L_{c \rightarrow i}^{\ddagger} \propto \phi_c^{\ddagger} \quad (\text{A-2})$$

であるから、リバランスファクターを乗じた後のバランス式は、下のようになる。

$$-\sum_i^8 S_i f_i^{\ddagger} L_{i \rightarrow c}^{\ddagger} + \sum_i^8 S_i f_c^{\ddagger} L_{c \rightarrow i}^{\ddagger} + \sum_{\text{trc}}^{\ddagger} f_c^{\ddagger} \phi_c^{\ddagger} V_c = Q_c^{\ddagger} V_c \quad (\text{A-3})$$

f は、リバランスファクターである。 f_i は、近接する領域のリバランスファクターでありこれを既知とすれば、中心領域のリバランスファクターは、下の式で与えられる。

$$f_c^{\ddagger} = \frac{Q_c^{\ddagger} V_c + \sum_i^8 S_i f_i^{\ddagger} L_{i \rightarrow c}^{\ddagger}}{\sum_i^8 S_i L_{c \rightarrow i}^{\ddagger} + \sum_{\text{trc}}^{\ddagger} \phi_c^{\ddagger} V_c} \quad (\text{A-4})$$

上式は全領域の f が収束条件を満たすまで繰り返し計算される。

通常は、多い場合でも 100 回以内で収束する（収束条件 10^{-4} の場合）ので、実際のコード上では、200 回計算しても収束しない場合は、リバランスファクターが求まらないとして、加速は行なわないようにしている。

求められたリバランスファクターは、古い中性子束分布にそれぞれ乗じられ、それを新しい中性子束分布として、再び内側反復計算を続ける。

以上の操作は各群で独立して行なわれている。

外側反復計算は、最後の内側反復計算で計算されたもれの量と中性子束分布を用いて、全群について総和した中性子バランス式を解く。

すなわち

$$\begin{aligned} & - \sum_i^g S_i f_i \sum_g^g L_{i \rightarrow c}^g + \sum_i^g S_i f_c \sum_g^g L_{c \rightarrow i}^g + f_c V_c \sum_g^g \Sigma_{a_c}^g \phi_c^g \\ & = \frac{1}{k} f_c V_c \sum_g^g \nu \Sigma_{f_c}^g \phi_c^g \end{aligned} \quad (\text{A-5})$$

$$f_c = \frac{\sum_i^g S_i f_i \sum_g^g L_{i \rightarrow c}^g}{\sum_i^g S_i \sum_g^g L_{c \rightarrow i}^g + V_c \sum_g^g \Sigma_{a_c}^g \phi_c^g - \frac{1}{k} V_c \sum_g^g \nu \Sigma_{f_c}^g \phi_c^g} \quad (\text{A-6})$$

求められたリバランスファクターは、内側反復計算の場合と同様に中性子束分布に用いられる他、核分裂ソース分布にも用いられる。

Fine-mesh-rebalancing 法は、バランスをとる領域を各メッシュごとにとる方法であり、system-rebalancing 法はメッシュの取り方に関係なく体系全体でのバランスをとる方法である。

System-rebalancing 法の場合はリバランスファクターを求めるのに反復計算は不要で下の式によって求められる。

内側反復計算の場合

$$f = \frac{\sum_{\text{SYSTEM}} Q^g \nabla}{\sum_{\text{SYSTEM}} SL_{(net)}^g + \sum_{\text{SYSTEM}} \sum_{tr} \phi^g \nabla} \quad (\text{A-7})$$

外側反復計算の場合

$$f = \frac{\sum_{\text{SYSTEM}} \sum_G \nu \Sigma_f^g \phi^g \nabla / k_{eff}}{\sum_{\text{SYSTEM}} \sum_G SL_{(net)}^g + \sum_{\text{SYSTEM}} \sum_G \Sigma_a^g \phi^g \nabla} \quad (\text{A-8})$$

ただし、 \sum_{SYSTEM} 、 \sum_G はそれぞれ空間及びエネルギー群についての総和を示す。

Appendix B

拡散合成法の原理

多群中性子輸送方程式の数値解は通常 S_n 近似により求められるが多次元体系では収束が遅いため計算時間が膨大になる。このため TWOTRAN 等の輸送コードで拡散理論の結果に対する輸送補正を大型炉心で計算しても、その解が収束しているかどうか、空間メッシュ数は充分かどうか等につき調べる必要がある。多次元輸送コードには粗メッシュ再釣合法及び chebyshev 加速法が収れん加速法としてよく用いられており、特に粗メッシュ再釣合法は多次元体系に應用されているが中性子もれが少ない大型の炉心で衝突当りの吸収割合が少ない場合には、この方法も有効ではなくなり不安定な加速法となってしまう事が Reed により示されている。最近 Reed, Gelbard-Hageman, Alcouffe-Lewis, Alcouffe, Lewis, Painter 等は拡散合成法と称される輸送計算加速法の有効性について調べている。一次元体系での適用例を見ると拡散合成法は他の加速法に比べ非常に有効である事が示されている。ここでは拡散合成法を二次元体系に應用してその有効性につき論じる。

エネルギー群 g 群における中性子輸送方程式を考える。

$$-\Omega \nabla \tilde{\psi}_g^l + \sum_t^g \tilde{\psi}_g^l(r, \Omega) = \sum_s^{g \rightarrow g} \phi_g^{l-1}(r) + Q Q^g(r) \quad (B-1)$$

$\tilde{\psi}_g^l$: g 群 l 番目内側反復での角度依存中性子束

\sum_t^g : g 群の全断面積

$\sum_s^{g \rightarrow g}$: g 群の自群散乱断面積

$Q Q^g$: g 群のソース (核分裂 + 散乱項)

ϕ_g^{l-1} : g 群の中性子束

(B-1) 式の自群散乱項に表われる中性子束 ϕ_g を拡散合成法により予測する。このため輸送方程式 (B-1) を次の拡散方程式に似た形式に変形する。

$$\text{但し} \quad -\nabla \cdot \mathbb{D}^g \nabla \phi_g^l + \Sigma_r^g \phi_g^l = \mathbb{Q} \mathbb{Q}^g(r) - R_g^l \quad (\text{B-2})$$

$$R_g^l = \nabla \cdot \tilde{\mathcal{J}}_g^l + \nabla \cdot \mathbb{D}^g \nabla \tilde{\phi}_g^l \quad (\text{B-3})$$

$$\tilde{\mathcal{J}}^l = \int d\Omega \Omega \tilde{\Psi}_g^l(r, \Omega) \quad (\text{B-4})$$

$$\tilde{\phi}_g^l = \int d\Omega \tilde{\Psi}_g^l(r, \Omega) \quad (\text{B-5})$$

(B-1) 式より $\tilde{\Psi}_g^l$ を求め (B-4) に代入し R_g^l を計算する。つぎに (B-2) 式を解き ϕ_g^l を求め (B-1) 式の右辺の項を求める。この操作をくり返し行ない、 ϕ を収束させる。

Table 3-1 Cross Sections

Region	Group	Σ_a	$V\Sigma_f$	Σ_{tr}	$\Sigma^{1\rightarrow g}$	$\Sigma^{2\rightarrow g}$	$\Sigma^{3\rightarrow g}$
Fuel	1	0.2737-2	0.3714-2	0.3410	3.079-1	0	0
	2	0.2219-1	0.6855-2	0.6869	3.032-2	6.1659-1	0
	3	0.4749-1	0.5001-1	1.0077	1.232-5	4.8136-2	9.602-1
Water	1	0.1003-3	0	0.3699	3.1258-1	0	0
	2	0.2447-3	0	0.8737	5.7159-2	7.7817-1	0
	3	0.6807-2	0	1.4940	2.3630-5	9.5295-2	1.4872

fission spectrum $x^1 = 1.0$ $x^2 = x^3 = 0$

Table 3-2 Computational Results

		Eigenvalue	Number of Outer Iterations	Calculational Time (sec)
Diffusion Synthetic Acceleration	Outer and Inner	0.78956	15	1.73
	Inner only	0.78957	27	4.65
Rebalance Acceleration		0.78961	98	15.23

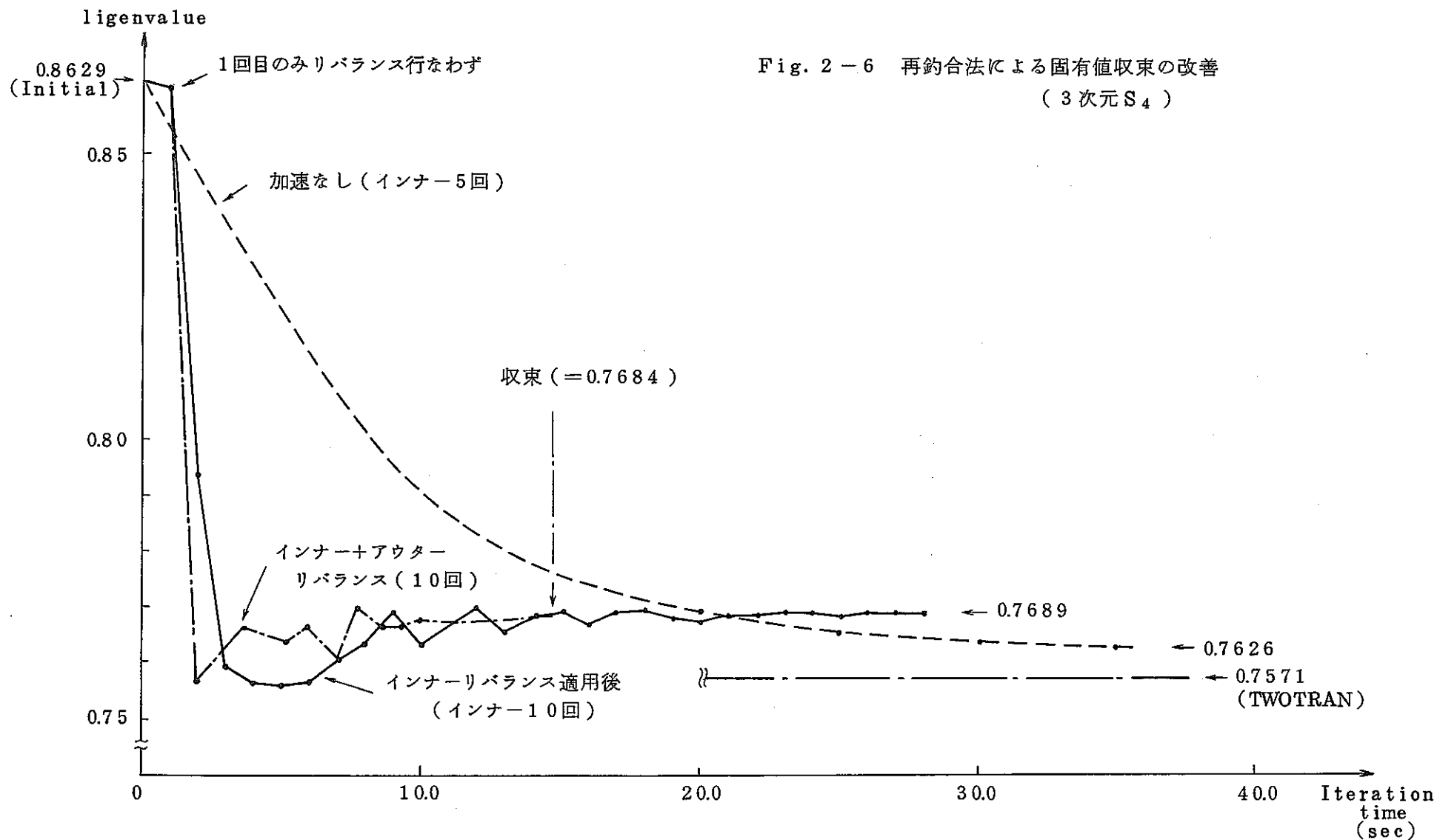
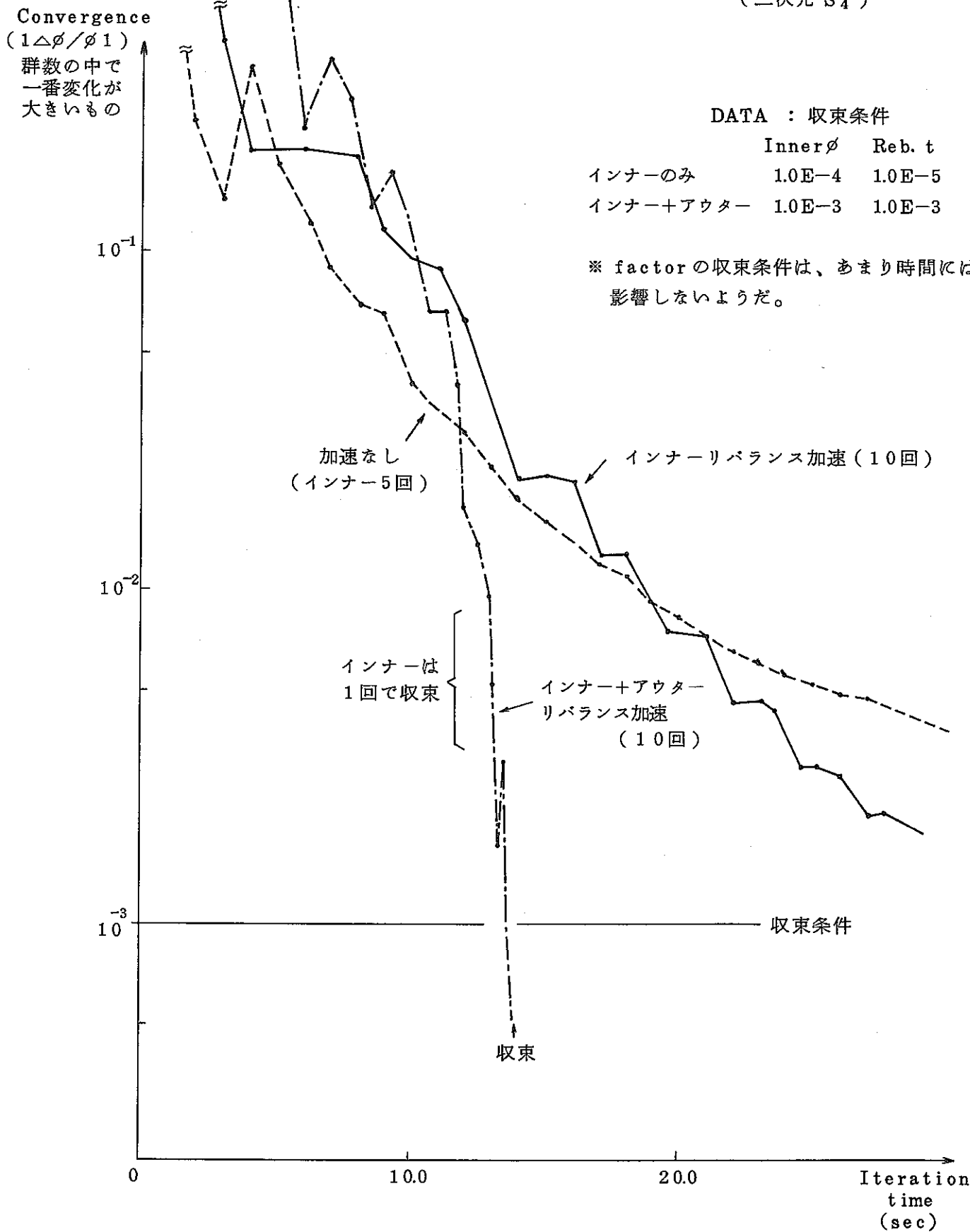


Fig. 2-6 再釣合法による固有値収束の改善 (3次元 S_4)

Fig. 2-7 再釣合法による中性子束収束性の改善
(三次元 S_4)



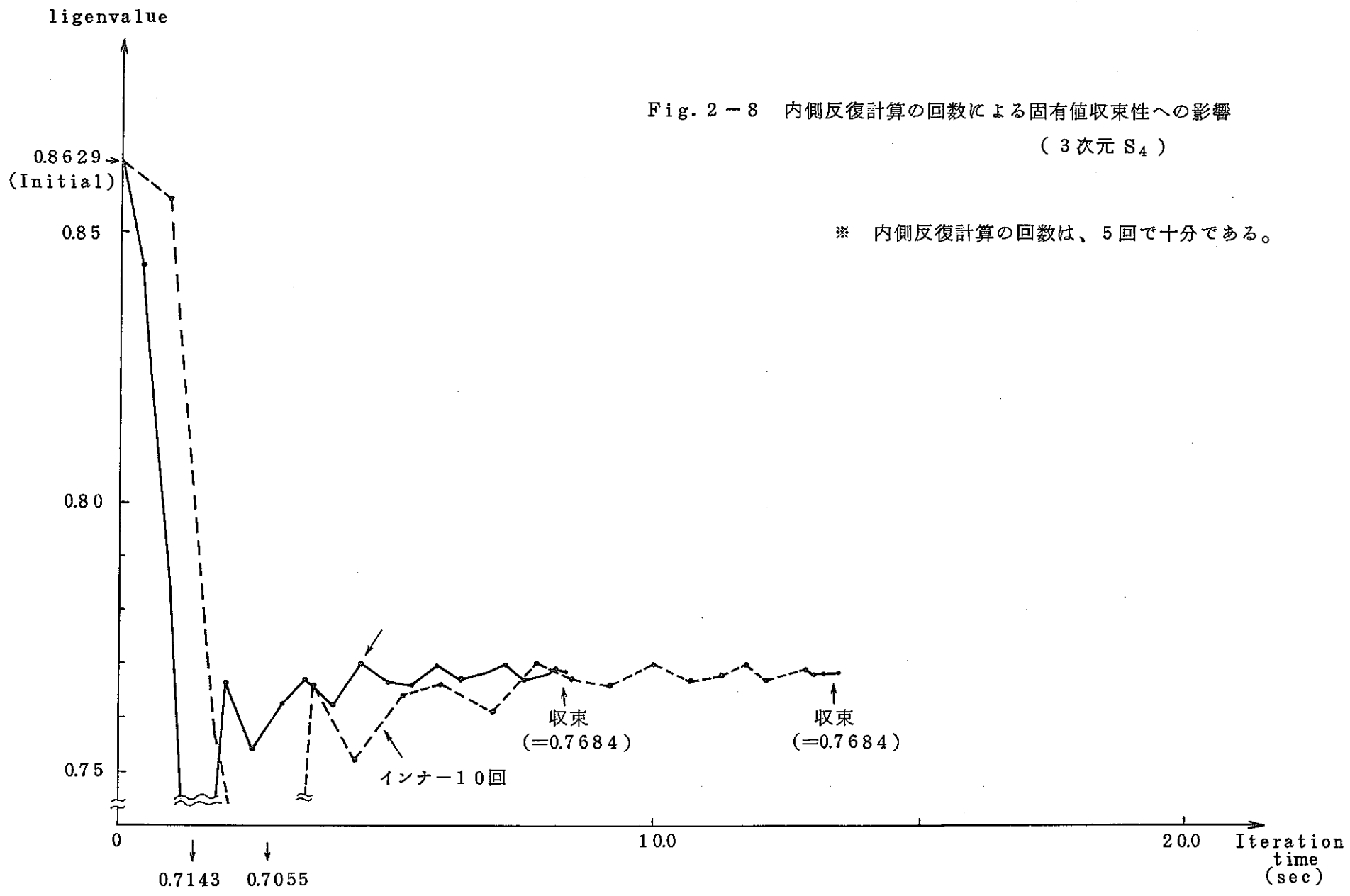


Fig. 2-8 内側反復計算の回数による固有値収束性への影響
(3次元 S_4)

* 内側反復計算の回数は、5回で十分である。

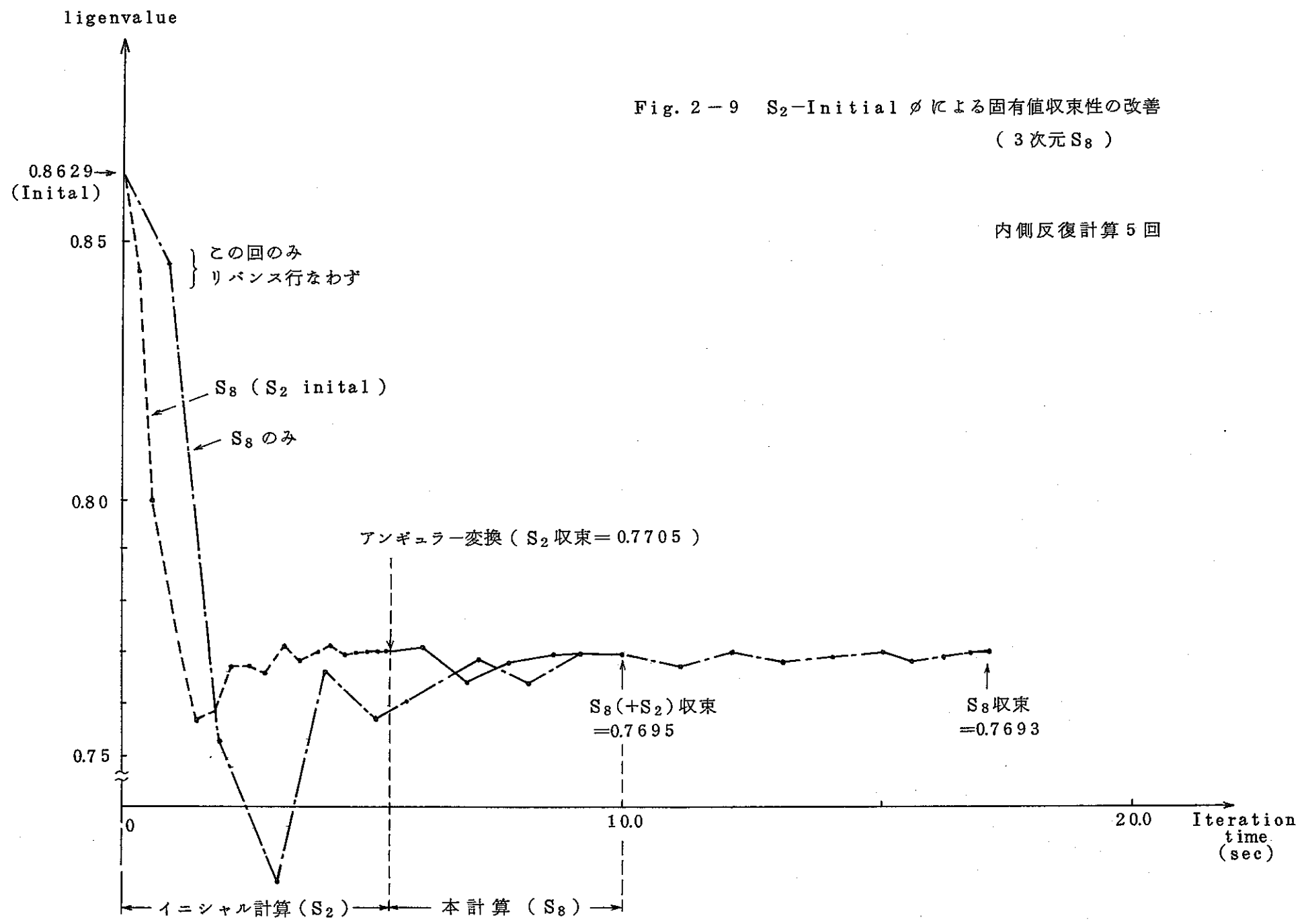


Fig. 2-9 S_2 -Initial ϕ による固有値収束性の改善 (3次元 S_8)

内側反復計算5回

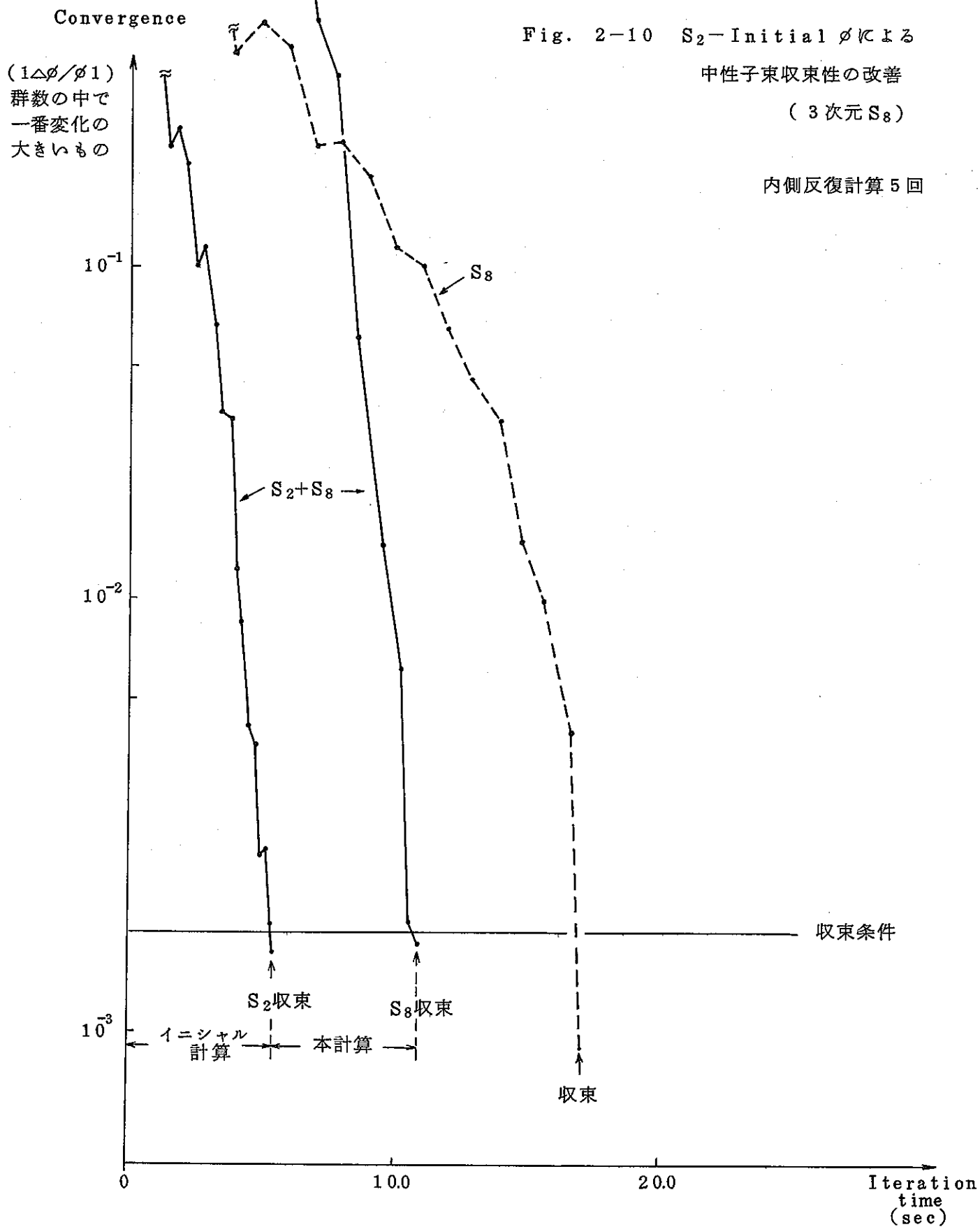
←イニシャル計算 (S_2) → ←本計算 (S_8) →

Fig. 2-10 S_2 -Initial ϕ による

中性子束収束性の改善

(3次元 S_8)

内側反復計算5回



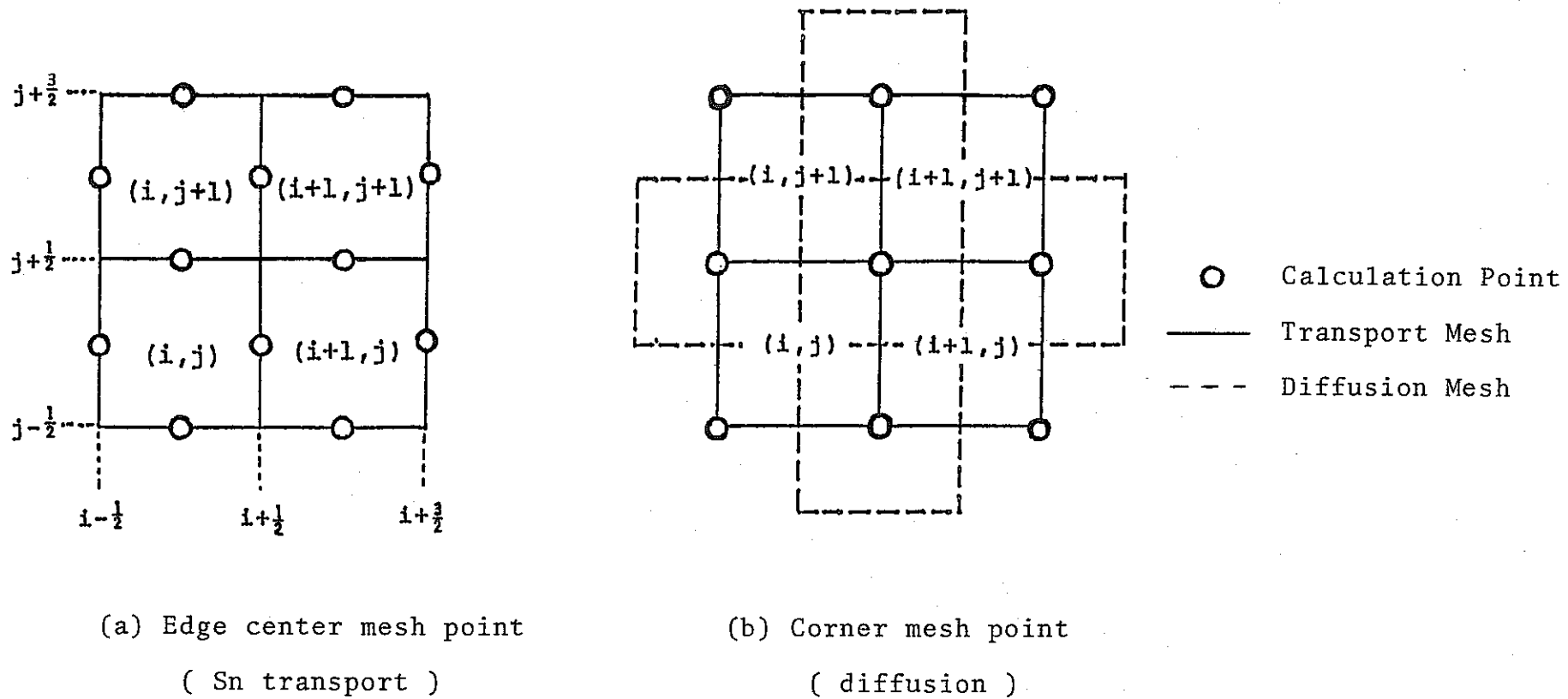
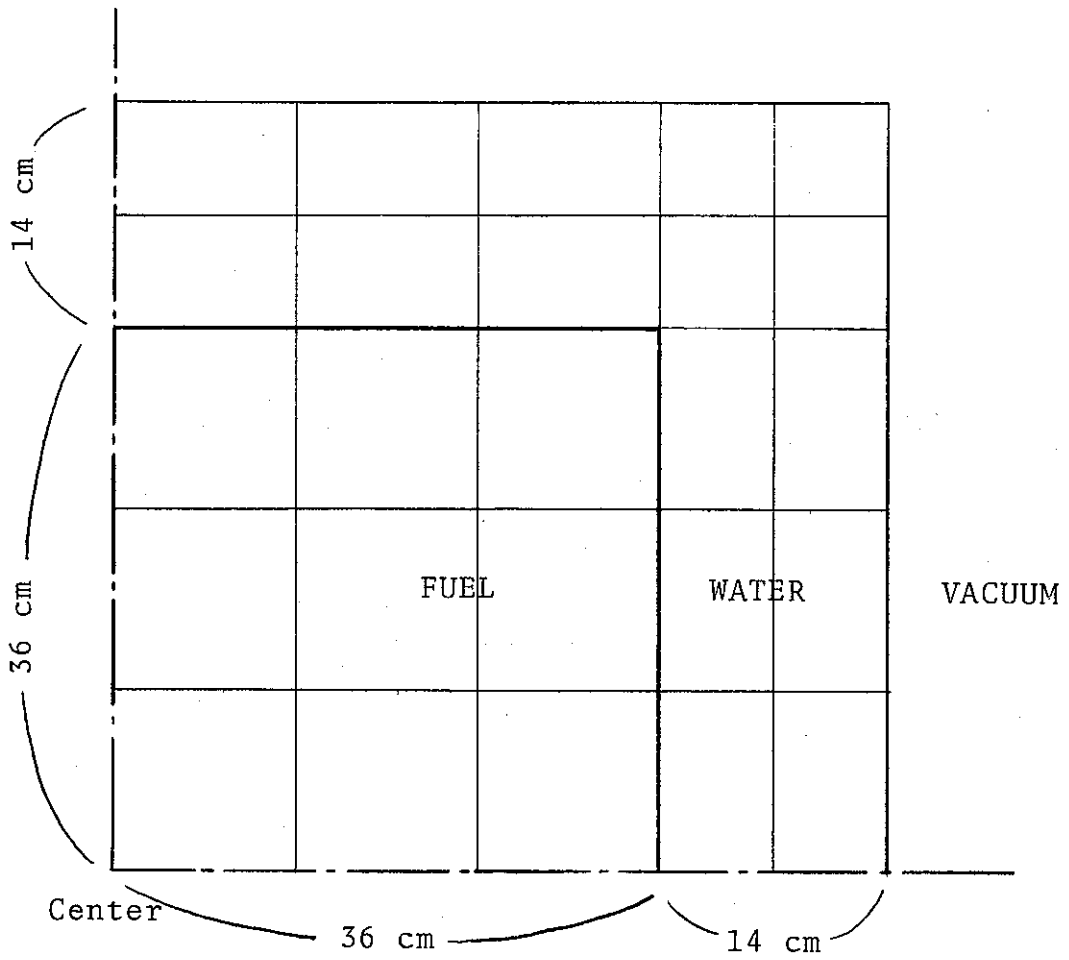


Fig. 3-1 Mesh Point for Diffusion Synthetic Method in 2-Dimensional Geometry



Sample Calculation

2 Dimensional

3 Group Transport Calculation

S4 Keff Calculation

Fig. 3-2 Geometry

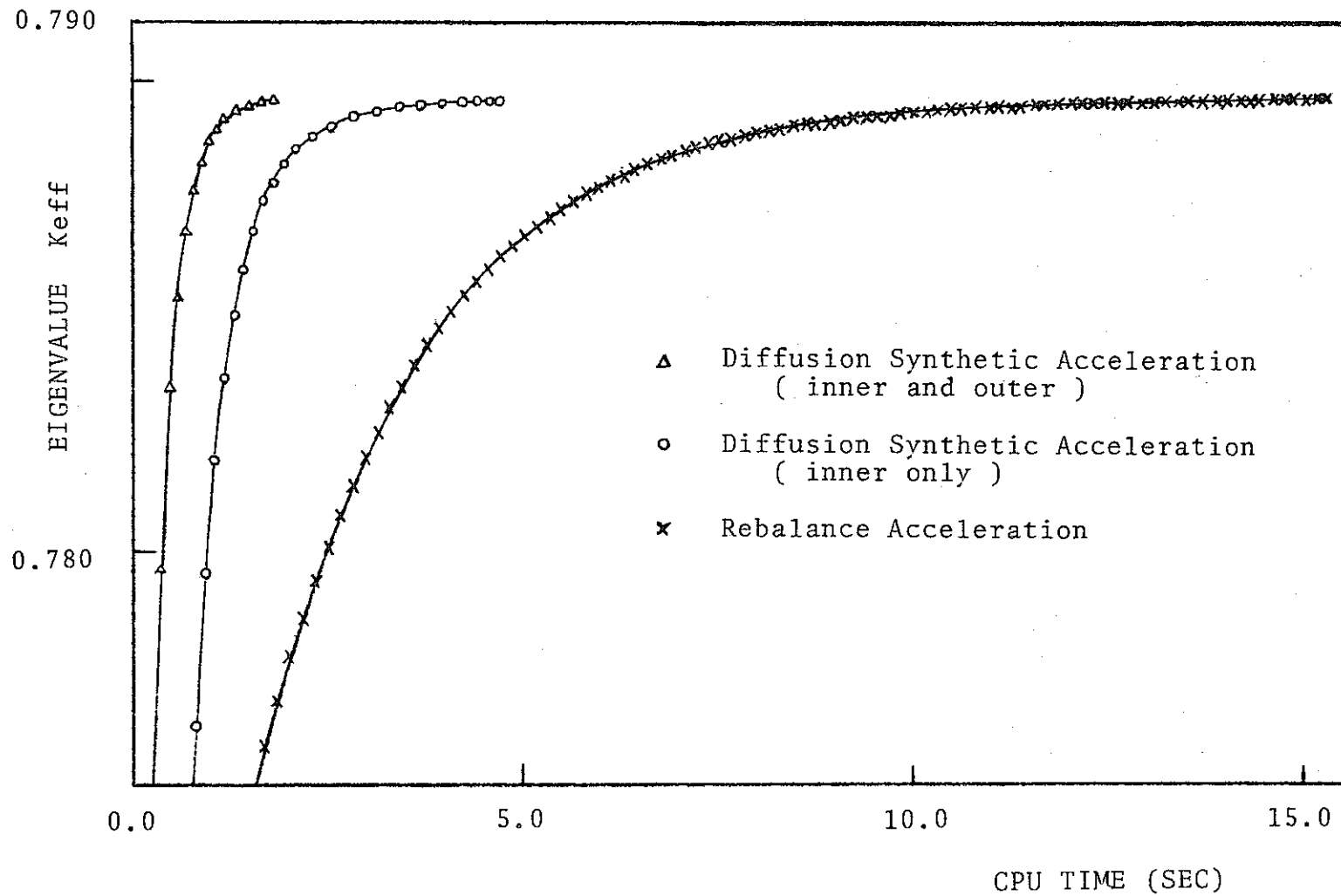


Fig. 3-3 Comparison of Convergent Behavior of K_{eff} (3 Group Calculation)

Appendix C

THOTHコードソースリスト

```

00010 DIMENSION A(200000)
00020 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
& NQUAD(2,2,2)
00030 COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
& CONCR,TIMOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NOW,
00040 & IMXM,JMXX,KMXM,NGMXM
00050 COMMON /OPTON/ISTART,MONT,IROMA,IGRD
00060 COMMON /REBFC/IRBMX,RFCL,RFCLL,RFMX,ICONB,IFLG(70)
00070 LIMIT=200000
00080 READ(5,501) ISTART,MONT,IROMA
00090 READ(5,501) IMX,JMX,KMX,NGMX,MAPMX,NSN
00100 MMX=NSN*(NSN+2)/8
00110 MFIX=MMX
00120 NFIX=NSN
00130 READ(5,502) CONCR,CONCRI,CONCRE,ITINMX,TIMOUT,EIGEN
00140 READ(5,503) IGRD,IRBMX,RFCL,RFCLL,CONCR
00150 501 FORMAT(6I12)
00160 502 FORMAT(3E12.5,I12,2F12.5)
00170 503 FORMAT(2I12,2F12.5,E12.5)
00180 IMXM=IMX
00190 JMXX=JMX
00200 KMXM=KMX
00210 NGMXM=NGMX
00220 IF(MONT.EQ.-1) GOTO 101
00230 IMXM=1
00240 JMXX=1
00250 KMXM=1
00260 NGMXM=1
00270 101 CONTINUE
00280 LDY=LDX+IMX
00290 LDZ=LDY+JMX
00300 LM=LDZ+KMX
00310 LXA=LM+IMX+JMX+KMX
00320 LXF=LXA+NGMX*MAPMX
00330 LXT=LXF+NGMX*MAPMX
00340 LXS=LXT+NGMX*MAPMX
00350 LXI=LXS+NGMX*NGMX*MAPMX
00360 LIB=LXI+NGMX
00370 LAMU=LIB+3*2
00380 LETA=LAMU+MFIX
00390 LAKS=LETA+MFIX
00400 LWGT=LAKS+MFIX
00410 LISX=LWGT+MFIX
00420 LISN=LISX+NGMX*MAPMX
00430 LAI=LISN+NGMX*MAPMX
00440 LAJ=LAI+8*MFIX*(IMX+1)+JMX*KMX+NGMX
00450 LAK=LAJ+8*MFIX*(IMX+1)+KMX*NGMX
00460 LO=LAK+8*MFIX*(IMX+1)+JMX*(KMX+1)+NGMX
00470 LSF=LO+IMX+JMX+KMX+NGMX
00480 LSD1=LSF+IMX+JMX+KMX+NGMX
00490 LSD2=LSD1+IMX+JMX+KMX
00500 LFS=LSD2+IMX+JMX+KMX+NGMX
00510 LOV=LFS+IMX+JMX+KMX
00520 LSS=LOV+IMX+JMX+KMX
00530 LBAX=LSS+IMX+JMX+KMX
00540 LBAY=LBAX+(IMX+1)*JMX+KMX*2
00550 LBAZ=LBAY+IMX*(JMX+1)*KMX*2
00560 LTBX=LBAZ+IMX*JMX*(KMX+1)*2
00570 LTBY=LTBX+(IMX+1)*JMX+KMX*2
00580 LTBI=LTBY+IMX*(JMX+1)*KMX*2
00590 LRF=LTBI+IMX*(JMX+1)*2
00600 LALR=LRF+IMX*(KMX+1)*2
00610 LAUD=LALR+MFIX+IMX
00620 LANF=LAUD+MFIX+IMX
00630 LS=LANF+MFIX+IMX
00640 LX=LS+IMX
00650 LG=LX+IMX
00660 LCLG=30+NGMX
00670 LCMX=LG+30
00680 LC=LCMX+NGMX
00690 LCM=LC+IMX
00700 LMAX=LCM+IMXM+JMXX
00710 WRITE(6,666)
00720 WRITE(6,667)
00730 WRITE(6,668)
00740 666 FORMAT(1H',26(5HTHOTH)/1H',.H',128X,'T'/1H',.O',128X,'H'/1H',.T',
& 10X,'T H R E E D I M E N S I O N A L',40X,
& ' T R A N S P O R T',.C D D E',18X,'D'/1H',.H',128X,'T')
00750 667 FORMAT(1H',T',46X,'***** ** ** * ** * ** * ** * ** *',47X,'H'/
& 1H',.H',42X,' * * * * * * * * * * * * * * *',47X,'T'/
& 1H',.D',42X,' * * * * * * * * * * * * * * *',47X,'H'/
& 1H',.T',42X,' * * * * * * * * * * * * * * *',47X,'D'/
& 1H',.H',42X,' * * * * * * * * * * * * * * *',47X,'T')
00760 668 FORMAT(1H',T',128X,'H'/1H',.H',128X,'T'/1H',.O',30X,
& 'C O M P O S E D B Y T . Y A M A M O T O ( O S A K A',
& ' U N I V .)',27X,'H'/1H',.T',44X,
& 'O C T . 1 9 8 3 S U I T A O S A K A',45X,'T'/1H',.O',128X,
& 'H'/1H',.T',128X,'D'/1H',.H',128X,'T'/1H',26(5HTHOTH)/)
00770 WRITE(6,600) LMAX,LIMIT
00780 600 FORMAT(90X,'MEMORY SIZE REQUIRED =',18,' WORDS'/90X,
& 'MAXIMUM MEMORY SIZE =',18,' WORDS')
00790 IF(LIMIT.LIMIT) STOP 'MEMORY SIZE OVER.'
00800 CALL MAESTOSO(A(LDX),A(LDY),A(LDZ),A(LM),A(LXA),A(LXF),A(LXT),
& A(LXS),A(LXI),A(LIB),A(LAMU),A(LETA),A(LAKS),A(LWGT),A(LISX),
& A(LISN),A(LAI),A(LAJ),A(LAK),A(LO),A(LSF),A(LSD1),A(LSD2),
& A(LFS),A(LOV),A(LSS),A(LBAX),A(LBAY),A(LBAZ),A(LTBX),A(LTBY),
& A(LTBI),A(LRF),A(LALR),A(LAUD),A(LANF),A(LS),A(LX),A(LG),
& A(LT),A(LCMX),A(LC),A(LCM))
00810 CALL RONDO(A(LDX),A(LDY),A(LDZ),A(LM),A(LXA),A(LXF),A(LXT),A(LXS),
& A(LXI),A(LIB),A(LAMU),A(LETA),A(LAKS),A(LWGT),A(LISX),
& A(LISN),A(LAI),A(LAJ),A(LAK),A(LO),A(LSF),A(LSD1),A(LSD2),A(LFS),
& A(LOV),A(LSS),A(LBAX),A(LBAY),A(LBAZ),A(LTBX),A(LTBY),A(LTBI),
& A(LRF),A(LALR),A(LAUD),A(LANF),A(LS),A(LX),A(LG),A(LT),
& A(LCMX),A(LC),A(LCM))
00820 END
00830 SUBROUTINE MAESTOSO(DELX,DELY,DELZ,MAP,XABS,XFIS,XTRA,XSCT,XI,
& IBNDCN,ISCTMX,ISCTMH,ANGFLI,ANGFLJ,ANGFLK,SCLFL,SCLOL2,
& TBALK,TBALY,TBALZ)
00840 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
& NQUAD(2,2,2)
00850 COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
& CONCR,TIMOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NOW,
00860 & IMXM,JMXX,KMXM,NGMXM
00870 COMMON /OPTON/ISTART,MONT,IROMA,IGRD
00880 COMMON /REBFC/IRBMX,RFCL,RFCLL,RFMX,ICONB,IFLG(70)
00890 DIMENSION MAP(IMX,JMX,KMX),XTRA(NGMX,MAPMX),XSCT(NGMX,NGMX,MAPMX)
& ,XFIS(NGMX,MAPMX),XABS(NGMX,MAPMX),XI(NGMX),SCLFL(IMX,JMX,
& KMX,NGMX)
00900 DIMENSION DELX(IMX),DELY(JMX),DELZ(KMX),IBNDCN(3,2),ANGFLI(8,
& MFIX,IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MFIX,IMX,JMX+1,KMX,NGMX),
& ANGFLK(8,MFIX,IMX,JMX,KMX+1,NGMX),ISCTMX(NGMX,MAPMX),
& ISCTMH(NGMX,MAPMX)
00910 DIMENSION SCLOL2(IMX,JMX,KMX,NGMX),TBALK(2,IMX+1,JMX,KMX),
& TBALY(2,IMX,JMX+1,KMX),TBALZ(2,IMX,JMX,KMX+1)

```

```

01300 IFUNC=1START+2
01310 GOTO(1,1,2),IFUNC
01320C****
01330 1 CONTINUE
01340 CALL DUMPRO(DELX,DELY,DELZ,MAP,XABS,XFIS,XTRA,XSCT,XI,IBNDCN,
01350 & ISCTMX,ISCTMN,ANGFLI,ANGFLJ,ANGFLK,SCFL,SCLOL2,TBALX,TBALY,
01360 & TBALZ)
01370 GOTO 999
01380C****
01390 2 CONTINUE
01400 READ(5,501) (DELX(J),J=1,IMX)
01410 READ(5,501) (DELY(J),J=1,JMX)
01420 READ(5,501) (DELZ(K),K=1,KMX)
01430 DO 21 K=1,KMX
01440 DO 21 J=1,JMX
01450 READ(5,502) (MAP(I,J,K),I=1,IMX)
01460 21 CONTINUE
01470 DO 22 NR=1,MAPMX
01480 DO 22 NG=1,NGMX
01490 READ(5,501) XABS(NG,NR),XFIS(NG,NR),XTRA(NG,NR),(XSCT(NN,NG,
01500 NR), NN=1,NGMX)
01510 22 CONTINUE
01520 READ(5,501) (XI(NG),NG=1,NGMX)
01530 READ(5,502) IBL,IBR,IBT,IBB,IBN,IBF
01540 501 FORMAT(16F12.5)
01550 502 FORMAT(24I3)
01560 IBNDCN(1,1)=IBR
01570 IBNDCN(1,2)=IBL
01580 IBNDCN(2,1)=IBB
01590 IBNDCN(2,2)=IBT
01600 IBNDCN(3,1)=IBF
01610 IBNDCN(3,2)=IBN
01620 CALL ISREAD(MAP,ISCTMX,ISCTMN,XSCT)
01630 999 CONTINUE
01640 WRITE(6,601) ISTART,MONT,IROMA
01650 601 FORMAT(10X,'OPTIONS'//20X,'RESTART (0) OR NEW PROBLEM (1)',20X,
01660 & '13/20X,'MONJTOR OPTION YES (-1) OR NO (OTHERS)',9X,13/20X,
01670 & 'INITIAL S-2 CALC. YES (-1) OR NO (OTHERS)',9X,13/)
01680 WRITE(6,602) IMX,JMX,KMX,NGMX,MAPMX,NSN
01690 602 FORMAT(10X,'CONTROL INTEGERS'//20X,'NUMBER OF MESHES (X)',10X,13/
01700 & 20X,'NUMBER OF MESHES (Y)',10X,13/20X,'NUMBER OF MESHES (Z)',
01710 & 10X,13/20X,'NUMBER OF GROUPS',14X,13/20X,
01720 & 'NUMBER OF MATERIALS',11X,13/20X,'NUMBER OF S N CONSTANTS',
01730 & 5X,13//)
01740 WRITE(6,603) CONCR,CONCRI,CONCRE,ITINMX,TIMOUT,EIGEN,IGRD,IRBMX,
01750 & RFCUL,RFCLL,CONCRR
01760 603 FORMAT(10X,'CONVERGENCE CRITERIDNS'//20X,'OUTER ROOP',21X,1PE12.5/
01770 & 20X,'INNER ROOP',21X,E12.5/20X,'EIGENVALUE',15X,E12.5/
01780 & 20X,'MAXIMUM INNER ROOPS',12X,112/20X,'TIME LIMIT',21X,OPF12.5,
01790 & 'MIN.',/20X,'INITIAL EIGEN VALUE',11X,F12.5/20X,
01800 & 'REBALANCE GRADE',16X,112/20X,'MAXIMUM REBALANCE ROOPS',8X,
01810 & 112/20X,'FACTOR UPPER LIMIT',13X,F12.5/20X
01820 & 'FACTOR LOWER LIMIT',13X,F12.5/20X,'REBALANCE CONVERGENCE',
01830 & 10X,1PE12.5//)
01840 WRITE(6,604) (DELX(I),I=1,IMX)
01850 604 FORMAT(10X,'GEOMETRY'//20X,'MESH SIZES (X)'//20X,15F8.2)
01860 WRITE(6,620) (DELY(J),J=1,JMX)
01870 620 FORMAT(/20X,'MESH SIZES (Y)'//20X,15F8.2)
01880 WRITE(6,621) (DELZ(K),K=1,KMX)
01890 621 FORMAT(/20X,'MESH SIZES (Z)'//20X,15F8.2)
01900 WRITE(6,605)
01910 605 FORMAT(/10X,'MATERIAL MAP')
01920 DO 10 K=1,KMX
01930 WRITE(6,611) K
01940 611 FORMAT(/25X,'K=',13/)
01950 DO 10 J=1,JMX
01960 WRITE(6,612) (MAP(I,J,K),I=1,IMX)
01970 10 CONTINUE
01980 612 FORMAT(25X,24I3)
01990 WRITE(6,622)
02000 622 FORMAT(/10X,'MACRO CROSS SECTION SET')
02010 DO 11 MAPI=1,MAPMX
02020 WRITE(6,606) MAPI
02030 606 FORMAT(/20X,'MATERIAL NO.',15//25X,5NGROUP,15H ABSORPTION,
02040 & 15H PRODUCTION,15H TRANSPORT,/)
02050 WRITE(6,623) (NG,XABS(NG,MAPI),XFIS(NG,MAPI),XTRA(NG,MAPI),NG=1,
02060 & NGMX)
02070 623 FORMAT(25X,13,2X,3E15.5)
02080 11 CONTINUE
02090 WRITE(6,624)
02100 624 FORMAT(/10X,'SCATTERING MATRIX')
02110 DO 12 MAPI=1,MAPMX
02120 WRITE(6,625) MAPI
02130 625 FORMAT(/20X,'MATERIAL NO.',15)
02140 DO 13 NG=1,NGMX
02150 WRITE(6,626) (XSCT(NN,NG,MAPI),NN=1,NGMX)
02160 13 CONTINUE
02170 626 FORMAT(1H,10(1PE12.5))
02180 12 CONTINUE
02190 WRITE(6,627)
02200 627 FORMAT(/10X,'FISSION SPECTRUM'//)
02210 WRITE(6,628) (XI(NG),NG=1,NGMX)
02220 628 FORMAT(10X,6(1PE12.5))
02230 WRITE(6,607) (IBNDCN(ID,2),IBNDCN(ID,1),ID=1,3)
02240 607 FORMAT(/10X,'BOUNDARY CONDITIONS'//20X,'FREE (0) OR REFLECT (1)'//
02250 & /25X,'LEFT',5X,13/25X,'RIGHT',5X,13/25X,'FRONT',5X,13/25X,
02260 & 'BACK',5X,13/25X,'TOP',7X,13/25X,'BOTTOM',4X,13/)
02270 CALL CNTRL
02280 RETURN
02290 END
02300 SUBROUTINE ROMANCE(DELX,DELY,DELZ,MAP,XABS,XFIS,XTRA,XSCT,XI,
02310 & IBNDCN,AMU,ETA,AKSI,WGT,ISCTMX,ISCTMN,ANGFLI,ANGFLJ,ANGFLK,
02320 & OQ,SCFL,SCLOL1,SCLOL2,FISS,QV,SS,BALX,BALY,BALZ,TBALX,TBALY,
02330 & TBALZ,RFC,ALR,AUD,ANF,S,XT,GRX,TX,CONVMX,CONV,CONVM)
02340 COMMON /CNTL/IMX,JMX,KMX,MMX,MPIX,NGMX,MAPMX,NSN,NFIS,ITINMX,
02350 & NQUAD(2,2,2)
02360 COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
02370 & CONCRR,TIMOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NDW,
02380 & IMXM,JXMX,KMX,NGMX
02390 COMMON /OPTN/ISTART,MONT,IROMA,IGRD
02400 DIMENSION MAP(IMX,JMX,KMX),OO(IMX,JMX,KMX,NGMX),XTRA(NGMX,MAPMX),
02410 & XSCT(NGMX,NGMX,MAPMX),XFIS(NGMX,MAPMX),XABS(NGMX,MAPMX),
02420 & XI(NGMX),ISCTMX(NGMX,MAPMX),ISCTMN(NGMX,MAPMX),SCFL(IMX,JMX,
02430 & KMX,NGMX),SCLOL1(IMX,JMX,KMX),SCLOL2(IMX,JMX,KMX,NGMX)
02440 DIMENSION GRX(30,NGMX),TX(30),DELX(IMX),DELY(JMX),DELZ(KMX),
02450 & IBNDCN(3,2),ANGFLI(8),MPIX(IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MPIX,
02460 & IMX,JMX+1,KMX,NGMX),ANGFLK(8,MPIX,IMX,JMX,KMX+1,NGMX),
02470 & AMU(MPIX),ETA(MPIX),AKSI(MPIX),WGT(MPIX),CONVMX(NGMX),
02480 & CONV(IMX),CONVM(IMXM,JXMX)
02490 DIMENSION FISS(IMX,JMX,KMX),SS(IMX,JMX,KMX),BALX(2,IMX+1,JMX,KMX),
02500 & BALY(2,IMX,JMX+1,KMX),BALZ(2,IMX,JMX,KMX+1),RFC(IMX,JMX,KMX),
02510 & QV(IMX,JMX,KMX),TBALX(2,IMX+1,JMX,KMX),TBALY(2,IMX,JMX+1,KMX),
02520 & TBALZ(2,IMX,JMX,KMX+1)
02530 DIMENSION ALR(MPIX,IMX),AUD(MPIX,IMX),ANF(MPIX,IMX),S(IMX),XT(IMX)
02540 IF(IROMA.NE.-1) RETURN
02550 MMX=1
02560 NSN=2
02570 CALL CPTIME(IGRN)
02580 CALL SNCON(AMU,ETA,AKSI,WGT,NSN)

```

```

02590 CALL INITAL(ANGFL1,ANGFLJ,ANGFLK,SCLFL,MAP,XTRA,XSCT,XFIS,XI,DELX,
02600 & DELY,DELZ,IBNDEN)
02610 CALL OUTER(MAP,OO,XTRA,XSCT,XFIS,XABS,XI,ISCTMX,ISCTMN,SCLFL,
02620 & GRX,TX,DELX,DELY,DELZ,IBNDEN,ANGFL1,ANGFLJ,ANGFLK,ALR,AUD,ANF,
02630 & S,XT,AMU,ETA,AKSI,WGT,SCLDL1,SCLDL2,FISS,OV,SS,BALX,BALY,BALZ,
02640 & TBALX,TBALY,TBALZ,RFC,CONVMX,CONV,CONVM)
02650 WRITE(6,601)
501 FORMAT(1/H)
02660 & ***** END OF INITIAL GUESS S-2 CALCULATION. ***** (1H)
02670 CALL CPTIME(IEND)
02680 TIMROM=IEND-IBGN
02700 TIMLMT=TIMROM
02710 RETURN
02720 END
02730 SUBROUTINE RONDO(DELX,DELY,DELZ,MAP,XABS,XFIS,XTRA,XSCT,XI,
02740 & IBNDEN,AMU,ETA,AKSI,WGT,ISCTMX,ISCTMN,ANGFL1,ANGFLJ,ANGFLK,
02750 & OO,SCLFL,SCLDL1,SCLDL2,FISS,OV,SS,BALX,BALY,BALZ,TBALX,TBALY,
02760 & TBALZ,RFC,ALR,AUD,ANF,S,XT,GRX,TX,CONVMX,CONV,CONVM)
02770 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITIMX,
02780 & NQUAD(2,2,2)
02790 COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
02800 & CONCR,TIMEOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NOW,
02810 & IMXM,JMXM,KMXM,NGMXM
02820 COMMON /OPTDN/ISTART,MDNT,IROMA,IGRD
02830 DIMENSION MAP(IMX,JMX,KMX),OO(IMX,JMX,KMX),XTRA(NGMX,MAPMX),
02840 & XSCT(NGMX,NGMX,MAPMX),XFIS(NGMX,MAPMX),XABS(NGMX,MAPMX),
02850 & XI(NGMX),ISCTMX(NGMX,MAPMX),ISCTMN(NGMX,MAPMX),SCLFL(IMX,JMX,
02860 & KMX,NGMX),SCLDL1(IMX,JMX,KMX),SCLDL2(IMX,JMX,KMX,NGMX)
02870 DIMENSION GRX(30,NGMX),TX(30),DELX(IMX),DELY(JMX),DELZ(KMX),
02880 & IBNDEN(3,2),ANGFL1(8,MFIX,IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MFIX,
02890 & IMX,JMX+1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,KMX+1,NGMX),
02900 & AMU(MFIX),ETA(MFIX),AKSI(MFIX),WGT(MFIX),CONVMX(NGMX),
02910 & CONV(IMX),CONV(MJMX,MKMX)
02920 DIMENSION FISS(IMX,JMX,KMX),SS(IMX,JMX,KMX),BALX(2,IMX+1,JMX,KMX),
02930 & BALY(2,IMX,JMX+1,KMX),BALZ(2,IMX,JMX,KMX+1),RFC(IMX,JMX,KMX),
02940 & QV(IMX,JMX,KMX),TBALX(2,IMX+1,JMX,KMX),TBALY(2,IMX,JMX+1,KMX),
02950 & TBALZ(2,IMX,JMX,KMX+1)
02960 MMX=MFIX
02970 NSN=MFIX
02980 CALL SNCON(AMU,ETA,AKSI,WGT,NSN)
02990 IF(IROMA.NE.-1.AND.ISTART.EQ.1) CALL INITAL(ANGFL1,ANGFLJ,ANGFLK,
03000 & SCLFL,MAP,XTRA,XSCT,XFIS,XI,DELX,DELY,DELZ,IBNDEN)
03010 IF(IROMA.EQ.-1) CALL CONANG(ANGFL1,ANGFLJ,ANGFLK)
03020 CALL OUTER(MAP,OO,XTRA,XSCT,XFIS,XABS,XI,ISCTMX,ISCTMN,SCLFL,
03030 & GRX,TX,DELX,DELY,DELZ,IBNDEN,ANGFL1,ANGFLJ,ANGFLK,ALR,AUD,ANF,
03040 & S,XT,AMU,ETA,AKSI,WGT,SCLDL1,SCLDL2,FISS,OV,SS,BALX,BALY,BALZ,
03050 & TBALX,TBALY,TBALZ,RFC,CONVMX,CONV,CONVM)
03060 RETURN
03070 END
03080 SUBROUTINE SNCON (U,E,X,W,N)
03090 DIMENSION U(36),E(36),X(36),W(36)
03100 ICEM=1
03110 NN=N
03120 M=(N+(N+2))/8
03130 MM=M
03140 NOUT=6
03150 I=N/2
03160 IF(N.EQ.30)GO TO 310
03170 IF(N.EQ.160)GO TO 330
03180 IF(N.GT.16)GO TO 180
03190 GO TO (100,110,120,130,140,150,160,170),I
03200C =
03210 100 W(1)=1.0
03220 U(1)=.57735027
03230 E(1)=U(1)
03240 GO TO 240
03250C =
03260 110 U(2)=0.90444905
03270 U(1)=0.30163878
03280 W(1)=0.33333333
03290 W(2)=W(1)
03300 W(3)=W(1)
03310 GO TO 190
03320C =
03330 120 U(1)=0.23009194
03340 U(2)=0.68213432
03350 U(3)=0.94557676
03360 W(1)=0.16944656
03370 W(2)=0.16388677
03380 W(3)=W(1)
03390 W(4)=W(2)
03400 W(5)=W(2)
03410 W(6)=W(1)
03420 GO TO 190
03430C =
03440 130 U(1)=0.19232747
03450 U(2)=0.57735027
03460 U(3)=0.79352178
03470 U(4)=0.96229948
03480 W(1)=0.11678847
03490 W(2)=0.09325523
03500 W(3)=W(2)
03510 W(4)=W(1)
03520 W(5)=W(2)
03530 W(6)=0.09010320
03540 W(7)=W(2)
03550 W(8)=W(2)
03560 W(9)=W(2)
03570 W(10)=W(1)
03580 GO TO 190
03590C =
03600 140 U(1)=0.16962228
03610 U(2)=0.50714192
03620 U(3)=0.69686020
03630 U(4)=0.84500612
03640 U(5)=0.97080202
03650 W(1)=0.089842043
03660 W(2)=0.067288705
03670 W(3)=0.055780071
03680 W(4)=W(2)
03690 W(5)=W(1)
03700 W(6)=W(2)
03710 W(7)=0.053133809
03720 W(8)=W(7)
03730 W(9)=W(2)
03740 W(10)=W(3)
03750 W(11)=W(7)
03760 W(12)=W(3)
03770 W(13)=W(2)
03780 W(14)=W(2)
03790 W(15)=W(1)
03800 GO TO 190
03810C =
03820 150 U(1)=0.15295746
03830 U(2)=0.45769112
03840 U(3)=0.62869660
03850 U(4)=0.76226828
03860 U(5)=0.87868027
03870 U(6)=0.97600932

```

```

03880      W(1)=0.07332178
03890      W(2)=0.05266740
03900      W(3)=0.04161495
03910      W(4)=W(3)
03920      W(5)=W(2)
03930      W(6)=W(1)
03940      W(7)=W(2)
03950      W(8)=0.03895667
03960      W(9)=0.03249018
03970      W(10)=W(8)
03980      W(11)=W(2)
03990      W(12)=W(3)
04000      W(13)=W(9)
04010      W(14)=W(9)
04020      W(15)=W(3)
04030      W(16)=W(3)
04040      W(17)=W(8)
04050      W(18)=W(3)
04060      W(19)=W(2)
04070      W(20)=W(2)
04080      W(21)=W(1)
04090      GO TO 190
04100C *
04110      160      U(1)=0.14238965
04120      U(2)=0.42048076
04130      U(3)=0.57735027
04140      U(4)=0.69990185
04150      U(5)=0.80398498
04160      U(6)=0.89605866
04170      U(7)=0.97951538
04180      W(1)=0.062171628
04190      W(2)=0.043325697
04200      W(3)=0.033217605
04210      W(4)=0.031837060
04220      W(5)=W(3)
04230      W(6)=W(2)
04240      W(7)=W(1)
04250      W(8)=W(2)
04260      W(9)=0.030485324
04270      W(10)=0.024545116
04280      W(11)=W(10)
04290      W(12)=W(9)
04300      W(13)=W(2)
04310      W(14)=W(3)
04320      W(15)=W(10)
04330      W(16)=0.019984453
04340      W(17)=W(10)
04350      W(18)=W(3)
04360      W(19)=W(4)
04370      W(20)=W(10)
04380      W(21)=W(10)
04390      W(22)=W(4)
04400      W(23)=W(3)
04410      W(24)=W(9)
04420      W(25)=W(3)
04430      W(26)=W(2)
04440      W(27)=W(2)
04450      W(28)=W(1)
04460      GO TO 190
04470C *
04480      170      U(1)=0.13344572
04490      U(2)=0.39119433
04500      U(3)=0.52689687
04510      U(4)=0.6075610
04520      U(5)=0.74746822
04530      U(6)=0.83302700
04540      U(7)=0.91058181
04550      U(8)=0.98203079
04560      W(1)=0.05415425
04570      W(2)=0.03679653
04580      W(3)=0.02777273
04590      W(4)=0.02580284
04600      W(5)=W(4)
04610      W(6)=W(3)
04620      W(7)=W(2)
04630      W(8)=W(1)
04640      W(9)=W(2)
04650      W(10)=0.02494275
04660      W(11)=0.01962325
04670      W(12)=0.01879762
04680      W(13)=W(11)
04690      W(14)=W(10)
04700      W(15)=W(2)
04710      W(16)=W(3)
04720      W(17)=W(11)
04730      W(18)=0.01544801
04740      W(19)=W(18)
04750      W(20)=W(11)
04760      W(21)=W(3)
04770      W(22)=W(4)
04780      W(23)=W(12)
04790      W(24)=W(18)
04800      W(25)=W(12)
04810      W(26)=W(4)
04820      W(27)=W(4)
04830      W(28)=W(11)
04840      W(29)=W(11)
04850      W(30)=W(4)
04860      W(31)=W(3)
04870      W(32)=W(10)
04880      W(33)=W(3)
04890      W(34)=W(2)
04900      W(35)=W(2)
04910      W(36)=W(1)
04920      GO TO 190
04930C
04940C      SN SET NOT FOUND IN LIBRARY
04950C
04960      180      WRITE (NOUT,280)
04970      CALL EXIT
04980      190      K=I+1
04990      DO 210 J=2,I
05000      LA=I+1-J
05010      DO 200 L=1,LA
05020      U(K)=U(L)
05030      200      K=K+1
05040      210      CONTINUE
05050      K=I
05060      DO 230 J=1,I
05070      LA=I+1-J
05080      DO 220 L=1,LA
05090      E(K)=U(J)
05100      220      K=K+1
05110      230      CONTINUE
05120      240      DD 250 J=1,M
05130      250      W(J)=.25*W(J)
05140C
05150C      REORDER SN LIBRARY
05160C

```

```

05170      K=1
05180      L=MM
05190      260  T=U(K)
05200          U(K)=U(L)
05210          U(L)=T
05220          T=E(K)
05230          E(K)=E(L)
05240          E(L)=T
05250          T=W(K)
05260          W(K)=W(L)
05270          W(L)=T
05280          K=K+1
05290          L=L-1
05300          IF (K.LT.L) GO TO 260
05310          GO TO 320
05320C *
05330      310  N=N/10
05340          M=(N*(N+2))/8
05350          E(1)=0.9305681
05360          E(2)=0.6699905
05370          E(3)=0.6699905
05380          E(4)=0.3300094
05390          E(5)=E(4)
05400          E(6)=E(4)
05410          E(8)=0.0694318
05420          E(7)=E(8)
05430          E(9)=E(7)
05440          E(10)=E(7)
05450          U(1)=0.2568851
05460          U(2)=0.6858601
05470          U(3)=0.2840926
05480          U(4)=0.9118124
05490          U(5)=0.6674930
05500          U(6)=0.2443194
05510          U(7)=0.9784184
05520          U(8)=0.8294630
05530          U(9)=0.5642295
05540          U(10)=0.1946196
05550          W(1)=0.04348186
05560          W(2)=0.04075907
05570          W(3)=0.04075907
05580          W(4)=0.2717272
05590          W(5)=W(4)
05600          W(6)=W(4)
05610          W(7)=0.01087046
05620          W(8)=W(7)
05630          W(9)=W(7)
05640          W(10)=W(7)
05650          GOTD 320
05660C *
05670      330  N=N/10
05680          W(1)=0.01265357
05690          W(2)=0.01389881
05700          W(3)=W(2)
05710          W(4)=0.01307111
05720          W(5)=W(4)
05730          W(6)=W(4)
05740          W(7)=0.01133387
05750          W(8)=W(7)
05760          W(9)=W(7)
05770          W(10)=W(7)
05780          W(11)=0.00906709
05790          W(12)=W(11)
05800          W(13)=W(11)
05810          W(14)=W(11)
05820          W(15)=W(11)
05830          W(16)=0.00653556
05840          W(17)=W(16)
05850          W(18)=W(16)
05860          W(19)=W(16)
05870          W(20)=W(16)
05880          W(21)=W(16)
05890          W(22)=0.00397109
05900          W(23)=W(22)
05910          W(24)=W(22)
05920          W(25)=W(22)
05930          W(26)=W(22)
05940          W(27)=W(22)
05950          W(28)=W(22)
05960          W(29)=0.00158170
05970          W(30)=W(29)
05980          W(31)=W(29)
05990          W(32)=W(29)
06000          W(33)=W(29)
06010          W(34)=W(29)
06020          W(35)=W(29)
06030          W(36)=W(29)
06040          U(1)=0.14020693
06050          U(2)=0.40587392
06060          U(3)=0.16811848
06070          U(4)=0.62463946
06080          U(5)=0.45726783
06090          U(6)=0.16737164
06100          U(7)=0.79066566
06110          U(8)=0.67028552
06120          U(9)=0.44787047
06130          U(10)=0.15727120
06140          U(11)=0.90161681
06150          U(12)=0.81336028
06160          U(13)=0.64548638
06170          U(14)=0.41442776
06180          U(15)=0.14280208
06190          U(16)=0.96314169
06200          U(17)=0.89750519
06210          U(18)=0.77070517
06220          U(19)=0.59138288
06230          U(20)=0.37175881
06240          U(21)=0.12680001
06250          U(22)=0.98866331
06260          U(23)=0.93899261
06270          U(24)=0.84233691
06280          U(25)=0.70344291
06290          U(26)=0.52927537
06300          U(27)=0.32866773
06310          U(28)=0.11138434
06320          U(29)=0.99498865
06330          U(30)=0.95675170
06340          U(31)=0.88174741
06350          U(32)=0.77285808
06360          U(33)=0.63426823
06370          U(34)=0.47130382
06380          U(35)=0.29022745
06390          U(36)=0.09799781
06400C
06410          E(1)=0.98014490
06420          E(2)=0.89833320
06430          E(3)=E(2)
06440          E(4)=0.76276620
06450          E(5)=E(4)

```

```

06460 E(6)=E(4)
06470 E(7)=0.59171730
06480 E(8)=E(7)
06490 E(9)=E(7)
06500 E(10)=E(7)
06510 E(11)=0.40828260
06520 E(12)=E(11)
06530 E(13)=E(11)
06540 E(14)=E(11)
06550 E(15)=E(11)
06560 E(16)=0.23723370
06570 E(17)=E(16)
06580 E(18)=E(16)
06590 E(19)=E(16)
06600 E(20)=E(16)
06610 E(21)=E(16)
06620 E(22)=0.10166670
06630 E(23)=E(22)
06640 E(24)=E(22)
06650 E(25)=E(22)
06660 E(26)=E(22)
06670 E(27)=E(22)
06680 E(28)=E(22)
06690 E(29)=0.01985500
06700 E(30)=E(29)
06710 E(31)=E(29)
06720 E(32)=E(29)
06730 E(33)=E(29)
06740 E(34)=E(29)
06750 E(35)=E(29)
06760 E(36)=E(29)
06770 320 CONTINUE
06780 M=N*(N+2)/8
06790 WRITE (NOUT,290)N
06800 DO 270 J=1,M
06810 W(J)=W(J)/2.0
06820 X(J)=SORT(1.0-U(J)*U(J)-E(J)*E(J))
06830 270 WRITE (NOUT,300)J,U(J),E(J),X(J),W(J)
06840 RETURN
06850C
06860C
06870 280 FORMAT (31HOND SUCH SN CONSTANTS AVAILABLE)
06880 290 FORMAT (////10X,2H S,12,10H CONSTANTS//40X,2HMU,13X,3HETA,13X,
06890 & 3HXS1,12X,6HWEIGHT//)
06900 300 FORMAT (25X,14,4X,4E16.8)
06910 END
06920 SUBROUTINE CNTRL
06930 DIMENSION IDRCT(8),JDRCT(8),KDRCT(8)
06940 COMMON /CNTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
06950 & NQUAD(2,2,2)
06960 DATA IDRCT/ 1,-1,-1, 1, 1,-1,-1, 1/
06970 DATA JDRCT/ 1, 1,-1,-1, 1, 1,-1,-1/
06980 DATA KDRCT/ 1, 1, 1, 1,-1,-1,-1,-1/
06990 DO 1 IQUAD=1,8
07000 ID={IDRCT(IQUAD)+3}/2
07010 JD={JDRCT(IQUAD)+3}/2
07020 KD={KDRCT(IQUAD)+3}/2
07030 NQUAD(ID,JD,KD)=IQUAD
07040 1 CONTINUE
07050 RETURN
07060 END
07070 SUBROUTINE INITAL(ANGFLI,ANGFLJ,ANGFLK,SCLFL,MAP,XTRA,XSCT,XFIS,
07080 & XI,DELX,DELY,DELZ,IBNDCH)
07090 COMMON /CNTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
07100 & NQUAD(2,2,2)
07110 COMMON /MONIT/EGHOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
07120 & CONCR,TIMEOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITRES,ISTA,IPRV,NDW,
07130 & IMXM,JMXM,KMXM,NGMXM
07140 DIMENSION ANGFLI(8,MFIX,IMX+1,JMX,KMX,NGMX), ANGFLJ(8,MFIX,IMX,
07150 & JMX+1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,KMX+1,NGMX), SCLFL(1,IMX,
07160 & JMX,KMX,NGMX),MAP(1,IMX,JMX,KMX),XTRA(NGMX,MAPMX),XFIS(NGMX,
07170 & MAPMX),XSCT(NGMX,NGMX,MAPMX),XI(NGMX),DELX(1,IMX),DELY(1,JMX),
07180 & DELZ(KMX),IBNDCH(3,2)
07190 FVOL=0.0
07200 DO 10 I=1,IMX
07210 DO 10 J=1,JMX
07220 DO 10 K=1,KMX
07230 MAPI=MAP(I,J,K)
07240 IF(XFIS(1,MAPI).LE.0.0) GOTO 100
07250 FVOL=FVOL+DELX(1)*DELY(J)*DELZ(K)
07260 100 CONTINUE
07270 10 CONTINUE
07280 PWRNEW=0.0
07290 DO 1 K=1,KMX
07300 DO 1 J=1,JMX
07310 DO 1 I=1,IMX
07320 MAPI=MAP(I,J,K)
07330 DO 1 NG=1,NGMX
07340 F=1.0/FVOL*XI(NG)
07350 IF(NG.EQ.1) GOTO 101
07360 DO 2 NN=1,NG-1
07370 F=F+XSCT(NN,NG,MAPI)*SCLFL(I,J,K,NN)
07380 2 CONTINUE
07390 101 CONTINUE
07400 XX=XTRA(NG,MAPI)-XSCT(NG,NG,MAPI)
07410 SF=F/XX
07420 IF(XFIS(1,MAPI).LE.0.0) SF=1.0E-5
07430 SCLFL(I,J,K,NG)=SF
07440 PWRNEW=PWRNEW+SF*XFIS(NG,MAPI)*DELX(1)*DELY(J)*DELZ(K)
07450 DO 4 JD=1,2
07460 DO 4 KD=1,2
07470 IO=NQUAD(2,JD,KD)
07480 DO 40 M=1,MMX
07490 ANGFLI(IO,M,1,J,K,NG)=SF
07500 40 CONTINUE
07510 IO=NQUAD(1,JD,KD)
07520 DO 41 M=1,MMX
07530 ANGFLI(IO,M,I+1,J,K,NG)=SF
07540 41 CONTINUE
07550 4 CONTINUE
07560 DO 5 ID=1,2
07570 DO 5 KD=1,2
07580 IO=NQUAD(ID,2,KD)
07590 DO 50 M=1,MMX
07600 ANGFLJ(IO,M,1,J,K,NG)=SF
07610 50 CONTINUE
07620 IO=NQUAD(ID,1,KD)
07630 DO 51 M=1,MMX
07640 ANGFLJ(IO,M,I,J+1,K,NG)=SF
07650 51 CONTINUE
07660 5 CONTINUE
07670 DO 6 ID=1,2
07680 DO 6 JD=1,2
07690 IO=NQUAD(ID,JD,2)
07700 DO 60 M=1,MMX
07710 ANGFLK(IO,M,I,J,K,NG)=SF
07720 60 CONTINUE
07730 IO=NQUAD(ID,JD,1)
07740 DO 61 M=1,MMX

```

```

07750          ANGFLK(10,M,I,J,K+1,NG)=SF
07760          CONTINUE
07770 6        CONTINUE
07780 1        CONTINUE
07790          EIGEN=PWRNEW
07800          WRITE(6,601) PWRNEW
07810 601      FORMAT(///10X,'INITIAL GUESS'/20X,'POWER OR EIGENVALUE',10X,
07820          & F10.5/)
07830          DD 20 NG=1,NGMX
07840          DD 7 J=1,JMX
07850          DD 7 K=1,KMX
07860          DD 7 JD=1,2
07870          DD 7 KD=1,2
07880          IO1=NQUAD(1,JD,KD)
07890          IO2=NQUAD(2,JD,KD)
07900          DD 7 M=1,MMX
07910          IF(1BNDCN(1,2),EO,0) ANGFLI(IO2,M,1,J,K,NG)=0.0
07920          AF=ANGFLI(IO2,M,1,J,K,NG)
07930          ANGFLI(IO1,M,1,J,K,NG)=AF
07940          IF(1BNDCN(1,1),EO,0) ANGFLI(IO1,M,IMX+1,J,K,NG)=0.0
07950          AF=ANGFLI(IO1,M,IMX+1,J,K,NG)
07960          ANGFLI(IO2,M,IMX+1,J,K,NG)=AF
07970 7        CONTINUE
07980          DD 8 K=1,KMX
07990          DD 8 I=1,IMX
08000          DD 8 KD=1,2
08010          DD 8 IO=1,2
08020          IO1=NQUAD(1D,1,KD)
08030          IO2=NQUAD(1D,2,KD)
08040          DD 8 M=1,MMX
08050          IF(1BNDCN(2,2),EO,0) ANGFLJ(IO2,M,1,1,K,NG)=0.0
08060          AF=ANGFLJ(IO2,M,1,1,K,NG)
08070          ANGFLJ(IO1,M,1,1,K,NG)=AF
08080          IF(1BNDCN(2,1),EO,0) ANGFLJ(IO1,M,1,JMX+1,K,NG)=0.0
08090          AF=ANGFLJ(IO1,M,1,JMX+1,K,NG)
08100          ANGFLJ(IO2,M,1,JMX+1,K,NG)=AF
08110 8        CONTINUE
08120          DD 9 I=1,IMX
08130          DD 9 J=1,JMX
08140          DD 9 ID=1,2
08150          DD 9 JD=1,2
08160          IO1=NQUAD(1D,JD,1)
08170          IO2=NQUAD(1D,JD,2)
08180          DD 9 M=1,MMX
08190          IF(1BNDCN(3,2),EO,0) ANGFLK(IO2,M,1,J,1,NG)=0.0
08200          AF=ANGFLK(IO2,M,1,J,1,NG)
08210          ANGFLK(IO1,M,1,J,1,NG)=AF
08220          IF(1BNDCN(3,1),EO,0) ANGFLK(IO1,M,1,J,KMX+1,NG)=0.0
08230          AF=ANGFLK(IO1,M,1,J,KMX+1,NG)
08240          ANGFLK(IO2,M,1,J,KMX+1,NG)=AF
08250 9        CONTINUE
08260 20       CONTINUE
08270          RETURN
08280          END
08290          SUBROUTINE CONANG(ANGFLI,ANGFLJ,ANGFLK)
08300          COMMON /CONT1/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
08310          & NQUAD(2,2,2)
08320          DIMENSION ANGFLI(8,MFIX,IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MFIX,IMX,JMX+
08330          & 1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,KMX+1,NGMX)
08340          DD 1 NG=1,NGMX
08350          DD 1 K=1,KMX+1
08360          DD 1 J=1,JMX+1
08370          DD 1 I=1,IMX+1
08380          IF(J.EQ.JMX+1.DR.K.EQ.KMX+1) GOTO 102
08390          DD 11 IO=1,8
08400          DD 11 M=2,MMX
08410          ANGFLI(10,M,I,J,K,NG)=ANGFLI(10,1,1,J,K,NG)
08420 11      CONTINUE
08430 102     CONTINUE
08440          IF(K.EQ.KMX+1.DR.I.EQ.IMX+1) GOTO 103
08450          DD 12 IO=1,8
08460          DD 12 M=2,MMX
08470          ANGFLJ(10,M,1,J,K,NG)=ANGFLJ(10,1,1,J,K,NG)
08480 12      CONTINUE
08490 103     CONTINUE
08500          IF(1.EQ.IMX+1.DR.J.EQ.JMX+1) GOTO 104
08510          DD 13 IO=1,8
08520          DD 13 M=2,MMX
08530          ANGFLK(10,M,1,J,K,NG)=ANGFLK(10,1,1,J,K,NG)
08540 13      CONTINUE
08550 104     CONTINUE
08560 1        CONTINUE
08570          RETURN
08580          END
08590          SUBROUTINE ISREAD(MAP,ISCTMX,ISCTMN,XSCT)
08600          COMMON /CONT1/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
08610          & NQUAD(2,2,2)
08620          COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
08630          & CONCR,TIMEOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NOW,
08640          & IMXM,JMXM,KMXM,NGMXM
08650          DIMENSION MAP(IMX,JMX,KMX),ISCTMX(NGMX,MAPMX),ISCTMN(NGMX,MAPMX),
08660          & XSCT(NGMX,NGMX,MAPMX)
08670          DD 100 K=1,KMX
08680          DD 100 J=1,JMX
08690          DD 100 I=1,IMX
08700          MAP1=MAP(1,J,K)
08710          DD 200 NG=1,NGMX
08720C ----- E N T R A N C E -----
08730          NN=NGMX
08740          1      ISCTMX(NG,MAP1)=NN
08750          IF(XSCT(NN,NG,MAP1).GT.0.0) GOTO 101
08760          NN=NN-1
08770          GOTO 1
08780C*****
08790 101     NN=1
08800          2      ISCTMN(NG,MAP1)=NN
08810          IF(XSCT(NN,NG,MAP1).GT.0.0) GOTO 201
08820          NN=NN+1
08830          GOTO 2
08840C*****
08850 201     CONTINUE
08860C ----- E X I T -----
08870 200     CONTINUE
08880 100     CONTINUE
08890          RETURN
08900          END
08910          SUBROUTINE OUTER(MAP,QQ,XTRA,XSCT,XFIS,XABS,XI,ISCTMX,ISCTMN,
08920          & SCFL,GRX,YX,DELX,DELY,DELZ,1BNDCN,ANGFLI,ANGFLJ,ANGFLK,ALR,
08930          & AUD,ANF,S,XT,AMU,ETA,AKS1,WGT,SCDL1,SCDL2,FISS,OV,SS,BALX,
08940          & BALY,BALZ,TBALX,TBALY,TBALZ,RFC,CONVMX,CONV,CONVM)
08950          COMMON /CONT1/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
08960          & NQUAD(2,2,2)
08970          COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
08980          & CONCR,TIMEOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NOW,
08990          & IMXM,JMXM,KMXM,NGMXM
09000          COMMON /OPTN/ISTART,MONT,IROMA,IGRD
09010          COMMON /REBFC/IRMX,RFCL,RFCLL,RFPMX,ICONS,IFLG(70)
09020          DIMENSION MAP(IMX,JMX,KMX),QQ(IMX,JMX,KMX,NGMX),XTRA(NGMX,MAPMX),
09030          & XSCT(NGMX,NGMX,MAPMX),XFIS(NGMX,MAPMX),XABS(NGMX,MAPMX),

```



```

09040      & XJ(NGMX),ISCTMX(NGMX,MAPMX),ISCTMN(NGMX,MAPMX),SCLFL(IMX, JMX,
09050      & KMX,NGMX)
09060      DIMENSION GRX(30,NGMX),TX(30),DELX(IMX),DELY(JMX),DELZ(KMX),
09070      & IBNDN(3,2), ANGL1( 8, MFIX,IMX+1,JMX,KMX,NGMX),ANGLJ(8,MFIX,
09080      & IMX, JMX+1,KMX,NGMX), ANGLK(8,MFIX,IMX,JMX,KMX+1,NGMX),
09090      & AMU(MFIX), ETA(MFIX), AKS1(MFIX),WGT(MFIX),SCLD1(IMX,JMX,KMX),
09100      & SCLD2(IMX,JMX,KMX,NGMX),CONVMX(NGMX), CONV(IMX), CONVM(IMXM,
09110      & JMXM)
09120      DIMENSION FISS(IMX,JMX,KMX),SS(IMX,JMX,KMX),BALX(2,IMX+1,JMX,KMX),
09130      & BALY(2,IMX,JMX+1,KMX),BALZ(2,IMX,JMX,KMX+1),RFC(IMX,JMX,KMX),
09140      & QV(IMX,JMX,KMX),TBALX(2,IMX+1,JMX,KMX),TBALY(2,IMX,JMX+1,KMX),
09150      & TBALZ(2,IMX,JMX,KMX+1)
09160      DIMENSION ALR(MFIX,IMX),AUD(MFIX,IMX),ANF(MFIX,IMX),S(IMX),XT(IMX)
09170      ITRDUT=1
09180      CALL CPTIME(ISTA)
09190C     *** D U T E R R O O P E N T R Y S T A R T ***
09200      WRITE(6,505)
09210      605 FORMAT(1H1,25X,'***** C O N V E R G E N C E   I N F O R M A T',
09220      & '   I D N S *****//12H   OUTER   ,12H   INNER   ,
09230      & 12H REBALANCE,12H REBALANCE,18H CPU-TIME   ,24X,
09240      & 12H FLUX   ,6H MAX./12H ITERATION ,12H ITERATIONS,
09250      & 12H ITERATIONS,12H CONVERGENCE,6H HOUR,6H MIN.,6H SEC.,
09260      & 12H EIGENVALUE,12H POWER   ,12H CONVERGENCE,6H GROUP//)
09270      DO 11 NG=1,NGMX
09280      11 CONVMX(NG)=1.0
09290      IFLAG=-IGRD
09300      IF(IGRD.GE.4) IFLAG=-5
09310      DO 10 NG=1,NGMX
09320      10 IFLG(NG)=IFLAG
09330      101 CONTINUE
09340      EDNLD=EIGEN
09350      PWRDLD=PWRNEW
09360C     *** E I G E N V A L U E   C A L C U L A T I O N ***
09370      & CALL EIGENV(FISS,MAP,SCLFL,XFIS,XABS,DELX,DELY,DELZ,TBALX,TBALY,
09380      & TBALZ)
09390      & FNOR=EIGEN/PWRNEW
09400      DO 2 K=1,KMX
09410      DO 2 J=1,JMX
09420      DO 2 I=1,IMX
09430      FISS(I,J,K)=FISS(I,J,K)/EIGEN
09440      2 CONTINUE
09450      IF(ISTART.EQ.1.AND.ITRDUT.EQ.1) GOTO 1003
09460C     *** D U T E R R E B A L A N C E   E N T R Y ***
09470      DO 200 K=1,KMX
09480      DO 200 J=1,JMX
09490      DO 200 I=1,IMX
09500      MAP1=MAP(I,J,K)
09510      VOL=DELX(I)*DELY(J)*DELZ(K)
09520      QV(I,J,K)=FISS(I,J,K)*VOL
09530C     DO 200 NG=1,NGMX
09540      DO 200 NG=1,NGMX
09550      SS(I,J,K)=SS(I,J,K)+XABS(NG,MAP1)*SCLFL(I,J,K,NG)*VOL
09560      SS(I,J,K)=SS(I,J,K)+[XABS(NG,MAP1)*SCLFL(I,J,K,NG)-
09570      & FISS(I,J,K)]*VOL
09580C     200 CONTINUE
09590      ICONB=-5
09600      IF(IGRD.LE.1) ICONB=IGRD
09620      CALL REBAL(QV,SS,TBALX,TBALY,TBALZ,IBNDN,RFC)
09630      IF(ICONB.NE.0) GOTO 203
09640      DO 202 K=1,KMX
09650      DO 202 J=1,JMX
09660      DO 202 I=1,IMX
09670      RFC(I,J,K)=1.0
09680      202 CONTINUE
09690      203 CONTINUE
09700C     *** E I G E N V A L U E   R E C A L C U L A T I O N ***
09710      DO 210 J=1,JMX
09720      DO 210 K=1,KMX
09730      TBALX(2,IMX+1,J,K)=TBALX(2,IMX+1,J,K)*RFC(IMX,J,K)
09740      210 TBALX(1,1,J,K)=TBALX(1,1,J,K)*RFC(1,J,K)
09750      DO 211 I=1,IMX
09760      DO 211 K=1,KMX
09770      TBALY(2,I,JMX+1,K)=TBALY(2,I,JMX+1,K)*RFC(I,JMX,K)
09780      211 TBALY(1,I,1,K)=TBALY(1,I,1,K)*RFC(1,I,K)
09790      DO 212 I=1,IMX
09800      DO 212 J=1,JMX
09810      TBALZ(2,I,J,KMX+1)=TBALZ(2,I,J,KMX+1)*RFC(1,J,KMX)
09820      212 TBALZ(1,I,J,1)=TBALZ(1,I,J,1)*RFC(1,J,1)
09830      & CALL EIGENV(FISS,MAP,SCLFL,XFIS,XABS,DELX,DELY,DELZ,TBALX,
09840      & TBALY, TBALZ)
09850      & FNOR=EIGEN/PWRNEW
09860      DO 220 K=1,KMX
09870      DO 220 J=1,JMX
09880      DO 220 I=1,IMX
09890      RFC(I,J,K)=RFC(I,J,K)*FNOR
09900      220 CONTINUE
09910      IF(ISTART.EQ.1.AND.ITRDUT.EQ.1) GOTO 221
09920      DO 204 K=1,KMX
09930      DO 204 J=1,JMX
09940      DO 204 I=1,IMX
09950      FISS(I,J,K)=FISS(I,J,K)*RFC(I,J,K)
09960      204 CONTINUE
09970      221 CONTINUE
09980      DO 3 NG=1,NGMX
09990      DO 205 K=1,KMX
10000      DO 205 J=1,JMX
10010      DO 205 I=1,IMX
10020      SCLFL(I,J,K,NG)=SCLFL(I,J,K,NG)*RFC(I,J,K)
10030      205 CONTINUE
10040      DO 4 K=1,KMX
10050      DO 4 J=1,JMX
10060      DO 4 JD=1,2
10070      DO 4 KD=1,2
10080      N1=NOUAD(1,JD,KD)
10090      N2=NOUAD(2,JD,KD)
10100      DO 4 M=1,MMX
10110      ANGL1(N1,M,1,J,K,NG)=ANGL1(N1,M,1,J,K,NG)
10120      & =RFC(1, J,K)
10130      ANGL1(N2,M,IMX+1,J,K,NG)=ANGL1(N2,M,IMX+1,J,K,
10140      & NG) =RFC(IMX,J,K)
10150      4 CONTINUE
10160      DO 5 K=1,KMX
10170      DO 5 I=1,IMX
10180      DO 5 ID=1,2
10190      DO 5 KD=1,2
10200      N1=NOUAD(ID,1,KD)
10210      N2=NOUAD(ID,2,KD)
10220      DO 5 M=1,MMX
10230      ANGLJ(N1,M,I,1,K,NG)=ANGLJ(N1,M,I,1,K,NG)
10240      & =RFC(1, I,K)
10250      ANGLJ(N2,M,I,JMX+1,K,NG)=ANGLJ(N2,M,I,JMX+1,K,
10260      & NG) =RFC(1,JMX,K)
10270      5 CONTINUE
10280      DO 6 J=1,JMX
10290      DO 6 I=1,IMX
10300      DO 6 ID=1,2
10310      DO 6 JD=1,2
10320      N1=NOUAD(ID,JD,1)

```

```

10330      N2=NQUAD(1D,JD,2)
10340      DO 6 M=1,MMX
10350      ANGLK(N1,M,I,J,1,NG)=ANGLK(N1,M,I,J,1,NG)
10360      *RFC(1,J,1)
10370      ANGLK(N2,M,I,J,KMX+1,NG)=ANGLK(N2,M,I,J,KMX+1,
10380      NG)*RFC(1,J,KMX)
10390      6 CONTINUE
10400      3 CONTINUE
10410 1003 CONTINUE
10420C **** C L E A R   R E B A L A N C E   A R R A Y S ****
10430      DO 209 L=1,2
10440      DO 206 K=1,KMX
10450      DO 206 J=1,JMX
10460      DO 206 I=1,IMX+1
10470      TBALX(L,I,J,K)=0.0
10480      206 CONTINUE
10490      DO 207 K=1,KMX
10500      DO 207 J=1,JMX+1
10510      DO 207 I=1,IMX
10520      TBALY(L,I,J,K)=0.0
10530      207 CONTINUE
10540      DO 208 K=1,KMX+1
10550      DO 208 J=1,JMX
10560      DO 208 I=1,IMX
10570      TBALZ(L,I,J,K)=0.0
10580      208 CONTINUE
10590 209 CONTINUE
10600C **** E X T E R N A L   S D U R C E   C A L C U L A T I O N ****
10610      DO 7 K=1,KMX
10620      DO 7 J=1,JMX
10630      DO 7 I=1,IMX
10640      MAP2=MAP(I,J,K)
10650      DO 8 NG=1,NGMX
10660      QO(I,J,K,NG)=0.0
10670      IA=ISCTMN(NG,MAP2)
10680      IB=ISCTMX(NG,MAP2)
10690      DO 9 NN=1A,IB
10700      IF(NN.EQ.0) GOTO 9
10710      SCLFLX=SCLFL(I,J,K,NN)
10720      QO(I,J,K,NG)=QO(I,J,K,NG)+XSCT(NN,NG,MAP2)*SCLFLX
10730      9 CONTINUE
10740      QO(I,J,K,NG)=QO(I,J,K,NG)+XI(NG)*FISS(I,J,K)
10750      8 CONTINUE
10760      7 CONTINUE
10770C **** I N N E R   S W E E P   E N T R Y ****
10780      DO 13 NG=1,NGMX
10790      13 CONVMX(NG)=ABS(CONVMX(NG))
10800      CALL INNER(AMU,ETA,AKS1,WGT,DELX,DELY,DELZ,ALR,AUD,ANF,ANGFLI,
10810      ANGLJ,ANGFLK,SCLFL,SCLL1,IBNDCN,MAP,QO,XSCT,XTRA,S,XT,QV,
10820      SS,BALX,BALY,BALZ,TBALX,TBALY,TBALZ,RFC,CONVMX)
10830C **** F L U X   C O N V E R G E N C E   T E S T ****
10840      CALL CONTE(SCLFL,SCLL2,CONVMX,CONV,CONVM)
10850      IF(ISTART.NE.-1) GOTO 102
10860      IF(NSN.NE.NFIX) RETURN
10870      CALL CODA(SCLFL,ANGFLI,ANGFLJ,ANGFLK,FISS,QO)
10880      CALL REACT(SCLFL,ANGFLI,ANGFLJ,ANGFLK,GRX,TX,DELX,DELY,DELZ,
10890      AMU,ETA,AKS1,WGT,XABS,XPIS,XTRA,XSCT,XI,MAP)
10900      STOP '... END OF CALCULATION.'
10910C *****
10920 102 CONTINUE
10930C **** C H E C K   T I M E   F O R   A D D I T I O N A L   S W E E P ****
10940      DATA FACTOR /1.0E-6/
10950      CALL CPTIME(NOW)
10960      TIMCPU=FLOAT(NOW-1STA)/60000.
10970      TIMOIT=FLOAT(NOW-IPRV)/60000.
10980      RESOUR=TIMOUT-TIMCPU
10990      TIMDMP=MMX*IMX*JMX*NGMX*FACTOR
11000      TREQ=TIMLMT+TIMOIT+TIMDMP
11010      IF(RESOUR.GT.TREQ) GOTO 999
11020      CALL DUMPWR(DELX,DELY,DELZ,MAP,XABS,XPIS,XTRA,XSCT,XI,IBNDCN,
11030      ISCTMX,ISCTMN,ANGFLI,ANGFLJ,ANGFLK,SCLFL,SCLL2,TBALX,
11040      TBALY,TBALZ)
11050      CALL REACT(SCLFL,ANGFLI,ANGFLJ,ANGFLK,GRX,TX,DELX,DELY,DELZ,
11060      AMU,ETA,AKS1,WGT,XABS,XPIS,XTRA,XSCT,XI,MAP)
11070      STOP '... TIME LIMIT OVER. TRY RESTART.'
11080 999 IPRV=NOW
11090C **** C O N T I N U E   O U T E R   S W E E P ****
11100      ITROUT=ITROUT+1
11110C      IF(ITROUT.EQ.6) STOP '... END OF OUTER TEST'
11120      GOTO 101
11130C **** O U T E R   R O O P   E N T R Y   E N D ****
11140      END
11150      SUBROUTINE EIGENV(FISS,MAP,SCLFL,XPIS,XABS,DELX,DELY,DELZ,TBALX,
11160      TBALY,TBALZ)
11170      COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
11180      NQUAD(2,2,2)
11190      COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
11200      CONCR,TIMEUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRV,NDW,
11210      IMXM,JMXM,KMXM,NGMXM
11220      DIMENSION FISS(IMX,JMX,KMX),MAP(IMX,JMX,KMX),SCLFL(IMX,JMX,KMX),
11230      NGMX),XABS(NGMX,MAPMX),XPIS(NGMX,MAPMX),DELX(IMX),DELY(JMX),
11240      DELZ(KMX),TBALX(2,IMX+1,JMX,KMX),TBALY(2,IMX,JMX+1,KMX),
11250      TBALZ(2,IMX,JMX,KMX+1)
11260      TF=0.0
11270      TA=0.0
11280      DO 1 K=1,KMX
11290      DO 1 J=1,JMX
11300      DO 1 I=1,IMX
11310      FISS(I,J,K)=0.0
11320      MAP1=MAP(I,J,K)
11330      DO 1 NG=1,NGMX
11340      SF=SCLFL(I,J,K,NG)
11350      FIS=XPIS(NG,MAP1)*SF
11360      FISS(I,J,K)=FISS(I,J,K)+FIS
11370      VOL=DELX(I)*DELY(J)*DELZ(K)
11380      TF=TF+FIS*VOL
11390      TA=TA+XABS(NG,MAP1)*SF*VOL
11400      1 CONTINUE
11410      LRECI=2*(IMX+1)*JMX*KMX
11420      LRECK=2*IMX*JMX*(KMX+1)
11430      XL=0.0
11440      DO 251 K=1,KMX
11450      DO 251 J=1,JMX
11460      XL=XL+TBALX(1,1,J,K)+TBALX(2,IMX+1,J,K)
11470      XL=XL-TBALX(2,1,J,K)-TBALX(1,IMX+1,J,K)
11480      251 CONTINUE
11490      YL=0.0
11500      DO 252 I=1,IMX
11510      DO 252 K=1,KMX
11520      YL=YL+TBALY(1,1,1,K)+TBALY(2,1,JMX+1,K)
11530      YL=YL-TBALY(2,1,1,K)-TBALY(1,1,JMX+1,K)
11540      252 CONTINUE
11550      ZL=0.0
11560      DO 253 I=1,IMX
11570      DO 253 J=1,JMX
11580      ZL=ZL+TBALZ(1,1,J,1)+TBALZ(2,1,J,KMX+1)
11590      ZL=ZL-TBALZ(2,1,J,1)-TBALZ(1,1,J,KMX+1)
11600      253 CONTINUE
11610      EIGEN=TF/(TA+XL+YL+ZL)

```

```

11620 PWRNEW*TF
11630 RETURN
11640 END
11650 SUBROUTINE INNER(AMU,ETA,AKS),WGT,DELX,DELY,DELZ,ALR,AUD,ANF,
11660 & ANGFLI,ANGFLJ,ANGFLK,SCLFL,SCLDL1,IBNDCN,MAP,OO,KSET,XTRA,S,
11670 & XT,QV,SS,BALX,BALY,BALZ,TBALX,TBALY,TBALZ,RFC,CDNYMX)
11680 COMMON /CONT/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSH,MFIX,ITINMX,
11690 & NQUAD(2,2)
11700 COMMON /MONIT/EQNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
11710 & CONCRR,TIMOUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITREB,ISTA,IPRY,NOW,
11720 & IMXM,JMXM,KMXM,NGMXM
11730 COMMON /OPTN/ISTART,MONI,IROMA,IGRD
11740 COMMON /REBFC/IRBMX,RFCUL,RFCLL,RFMX,ICONB,IFLG(70)
11750 COMMON /SWEEP/IDRCT,1,J,K,NG
11760 DIMENSION AMU(MFIX),ETA(MFIX),AKSI(MFIX),WGT(MFIX),DELX(IMX),
11770 & DELY(JMX),DELZ(KMX)
11780 DIMENSION ALR(MFIX,IMX),AUD(MFIX,IMX),ANF(MFIX,IMX),ANGFLI(8,MFIX,
11790 & IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MFIX,IMX,JMX+1,KMX,NGMX),ANGFLK(8,
11800 & MFIX,IMX,JMX,KMX+1,NGMX),SCLFL(IMX,JMX,KMX,NGMX),SCLDL1(IMX,
11810 & JMX,KMX)
11820 DIMENSION IBNDCN(3,2),MAP(IMX,JMX,KMX),OO(IMX,JMX,KMX,NGMX),
11830 & XSCT(NGMX,NGMX,MAPMX),XTRA(NGMX,MAPMX),S(IMX),XT(IMX),SS(IMX,
11840 & JMX,KMX),BALX(2,IMX+1,JMX,KMX),BALY(2,IMX,JMX+1,KMX),BALZ(2,
11850 & IMX,JMX,KMX+1),RFC(IMX,JMX,KMX),QV(IMX,JMX,KMX),TBALX(2,IMX+1,
11860 & JMX,KMX),TBALY(2,IMX,JMX+1,KMX),TBALZ(2,IMX,JMX,KMX+1),
11870 & CONVMX(NGMX)
11880 NG=
11890 11 CONTINUE
11900 ITRIN=0
11910 NTRER=0
11920 10 CONTINUE
11930 DO 12 K=1,KMX
11940 DO 12 J=1,JMX
11950 DO 12 I=1,IMX
11960 SCLDL1(I,J,K)=SCLFL(I,J,K,NG)
11970 12 CONTINUE
11980 IDRCT=-1
11990 JDRCT=-1
12000 KDRCT=-1
12010 1 CONTINUE
12020 K=1
12030 KD=2
12040 IF(KDRCT.EQ.1) GOTO 2
12050 K=KMX
12060 KD=1
12070 2 CONTINUE
12080 J=1
12090 JD=2
12100 IF(JDRCT.EQ.1) GOTO 3
12110 J=JMX
12120 JD=1
12130 3 CONTINUE
12140 I=1
12150 ID=2
12160 IF(IDRCT.EQ.1) GOTO 4
12170 I=IMX
12180 ID=1
12190 4 CONTINUE
12200 IQUAD=NQUAD(ID,JD,KD)
12210***** D A T A M A N A G E M E N T E N T R Y *****
12220 IA=I+2-ID
12230 JA=J+2-JD
12240 KA=K+2-KD
12250----- X B O U N D A R Y O P E R A T I O N -----
12260 IFUNC=IBNDCN(1,IO)
12270 GOTO (101,102),IFUNC+1
12280*****
12290 101 CONTINUE
12300 DO 103 M=1,MMX
12310 ALR(M,1)=0.0
12320 ANGFLI(IQUAD,M,IA,J,K,NG)=ALR(M,1)
12330 103 CONTINUE
12340 GOTO 1000
12350*****
12360 102 CONTINUE
12370 IDR=1
12380 IF(ID.EQ.1) IDR=2
12390 IO=NQUAD(IDR,JD,KD)
12400 DO 104 M=1,MMX
12410 ALR(M,1)=ANGFLI(IO,M,IA,J,K,NG)
12420 ANGFLJ(IQUAD,M,IA,J,K,NG)=ALR(M,1)
12430 104 CONTINUE
12440 1000 CONTINUE
12450 IF(J.EQ.1.AND.JD.EQ.2) GOTO 2000
12460 IF(J.EQ.JMX.AND.JD.EQ.1) GOTO 2000
12470----- D A T A T R A N S F E R T O A U D A R R A Y -----
12480 DO 200 II=1,IMX
12490 DO 200 M=1,MMX
12500 AUD(M,II)=ANGFLJ(IQUAD,M,II,JA,K,NG)
12510 200 CONTINUE
12520 GOTO 2001
12530*****
12540 2000 CONTINUE
12550----- Y B O U N D A R Y O P E R A T I O N -----
12560 IFUNC=IBNDCN(2,JD)
12570 GOTO (201,202),IFUNC+1
12580*****
12590 201 CONTINUE
12600 DO 203 II=1,IMX
12610 DO 203 M=1,MMX
12620 AUD(M,II)=0.0
12630 ANGFLJ(IQUAD,M,II,JA,K,NG)=AUD(M,II)
12640 203 CONTINUE
12650 GOTO 2001
12660*****
12670 202 CONTINUE
12680 JDR=1
12690 IF(JD.EQ.1) JDR=2
12700 IO=NQUAD(ID,JDR,KD)
12710 DO 204 II=1,IMX
12720 DO 204 M=1,MMX
12730 AUD(M,II)=ANGFLJ(IO,M,II,JA,K,NG)
12740 ANGFLK(IQUAD,M,II,JA,K,NG)=AUD(M,II)
12750 204 CONTINUE
12760 2001 CONTINUE
12770 IF(K.EQ.1.AND.KD.EQ.2) GOTO 3000
12780 IF(K.EQ.KMX.AND.KD.EQ.1) GOTO 3000
12790----- D A T A T R A N S F E R T O A N F A R R A Y -----
12800 DO 300 II=1,IMX
12810 DO 300 M=1,MMX
12820 ANF(M,II)=ANGFLK(IQUAD,M,II,J,KA,NG)
12830 300 CONTINUE
12840 GOTO 4000
12850*****
12860 3000 CONTINUE
12870----- Z B O U N D A R Y O P E R A T I O N -----
12880 IFUNC=IBNDCN(3,KD)
12890 GOTO (301,302),IFUNC+1
12900*****

```

```

12910 301 CONTINUE
12920 DD 303 I1=1,IMX
12930 DD 303 M=1,MMX
12940 ANF(M,I1)=0.0
12950 ANGFLK(IQUAD,M,I1,J,KA,NG)=ANF(M,I1)
12960 303 CONTINUE
12970 GOTO 4000
12980*****
12990 302 CONTINUE
13000 KDR=1
13010 IF(KD.EQ.1) KDR=2
13020 I0=NQUAD(JD,JD,KDR)
13030 DD 304 I1=1,IMX
13040 DD 304 M=1,MMX
13050 ANF(M,I1)=ANGFLK(I0,M,I1,J,KA,NG)
13060 ANGFLK(IQUAD,M,I1,J,KA,NG)=ANF(M,I1)
13070 304 CONTINUE
13080 4000 CONTINUE
13090***** L I N E M E S H S W E E P E N T R Y *****
13100 SQ=DELY(J)=DELZ(K)
13110 DD 20 I1=1,IMX
13120 MAPI=MAP(I1,J,K)
13130 S(I1)=QO(I1,J,K,NG)
13140 SFL=SCLFL(I1,J,K,NG)
13150 SSS=XSCCT(NG,NG,MAPI)*SFL
13160 S(I1)=S(I1)+SSS
13170 XT(I1)=XTRA(NG,MAPI)
13180 VOL=SQ+DELX(I1)
13190C SS(I1,J,K)=(XT(I1)*SFL-SSS)*VOL
13200 OV(I1,J,K)=QO(I1,J,K,NG)*VOL
13210 20 CONTINUE
13220 CALL LINER(ALR,AUD,ANF,AMU,ETA,AKSI,DELX,DELY,DELZ,S,XT)
13230 KA=K-1+KD
13240 JA=J-1+JD
13250 DD 40 I=1,IMX
13260 IA=I-1+ID
13270 DD 40 M=1,MMX
13280 ANGFLI(IQUAD,M,IA,J,K,NG)=ALR(M,I)
13290 ANGFLJ(IQUAD,M,I,JA,K,NG)=AUD(M,I)
13300 ANGFLK(IQUAD,M,I,J,KA,NG)=ANF(M,I)
13310 40 CONTINUE
13320C----- T U R N I N G P O I N T F O R I -----
13330 IDRCT=IDRCT*(-1)
13340 IF(IDRCT.EQ.1) GOTO 3
13350C----- S W E E P N E X T L I N E -----
13360 J=J+JDRCT
13370 IF(J.EQ.0.OR.J.EQ.JMX+1) GOTO 5
13380 GOTO 3
13390C*****
13400 5 CONTINUE
13410C----- T U R N I N G P O I N T F O R J -----
13420 JDRCT=JDRCT*(-1)
13430 IF(JDRCT.EQ.1) GOTO 2
13440C----- S W E E P N E X T P L A N E -----
13450 K=K+KDRCT
13460 IF(K.EQ.0.OR.K.EQ.KMX+1) GOTO 6
13470 GOTO 2
13480C*****
13490 6 CONTINUE
13500C----- T U R N I N G P O I N T F O R K -----
13510 KDRCT=KDRCT*(-1)
13520 IF(KDRCT.EQ.1) GOTO 1
13530C----- S C A L A R F L U X R E F L E S H M E N T -----
13540 DD 33 K=1,KMX
13550 DD 33 J=1,JMX
13560 DD 33 I=1,IMX
13570 SCLFLX=0.0
13580 DD 34 I0=1,8
13590 DD 34 M=1,MMX
13600 AF1=ANGFLI(I0,M,I,J,K,NG)
13610 AF12=ANGFLI(I0,M,I+1,J,K,NG)
13620 AFJ1=ANGFLJ(I0,M,I,J,K,NG)
13630 AFJ2=ANGFLJ(I0,M,I,J+1,K,NG)
13640 AFK1=ANGFLK(I0,M,I,J,K,NG)
13650 AFK2=ANGFLK(I0,M,I,J,K+1,NG)
13660 SCLFLX=SCLFLX+(AF1+AF12+AFJ2+AFK1+AFK2)=WGT(M)
13670 34 CONTINUE
13680 SCLFL(I,J,K,NG)=SCLFLX/6.0
13690 33 CONTINUE
13700C----- R E B A L A N C E E N T R Y -----
13710 DD 21 K=1,KMX
13720 DD 21 J=1,JMX
13730 DD 21 I=1,IMX
13740 RFC(I,J,K)=1.0
13750 MAPI=MAP(I,J,K)
13760 SS(I,J,K)=(XTRA(NG,MAPI)-XSCCT(NG,NG,MAPI))*DELX(I)
13770 *DELY(J)=DELZ(K)=SCLFL(I,J,K,NG)
13780 21 CONTINUE
13790 DD 22 J=1,JMX
13800 DD 22 K=1,KMX
13810 DD 22 IA=1,IMX+1
13820 SQ=DELY(J)=DELZ(K)
13830 BALX(1,IA,J,K)=0.0
13840 BALX(2,IA,J,K)=0.0
13850 DD 22 JD=1,2
13860 DD 22 KD=1,2
13870 I01=NQUAD(1,JD,KD)
13880 I02=NQUAD(2,JD,KD)
13890 DD 22 M=1,MMX
13900 WAM=WGT(M)=AMU(M)*SQ
13910 AF=ANGFLI(I01,M,IA,J,K,NG)
13920 BALX(1,IA,J,K)=BALX(1,IA,J,K)+AF+WAM
13930 AF=ANGFLI(I02,M,IA,J,K,NG)
13940 BALX(2,IA,J,K)=BALX(2,IA,J,K)+AF+WAM
13950 22 CONTINUE
13960 DD 23 K=1,KMX
13970 DD 23 I=1,IMX
13980 DD 23 JA=1,JMX+1
13990 SQ=DELZ(K)=DELX(I)
14000 BALY(1,I,JA,K)=0.0
14010 BALY(2,I,JA,K)=0.0
14020 DD 23 KD=1,2
14030 DD 23 ID=1,2
14040 I01=NQUAD(ID,1,KD)
14050 I02=NQUAD(ID,2,KD)
14060 DD 23 M=1,MMX
14070 WET=WGT(M)=ETA(M)*SQ
14080 AF=ANGFLJ(I01,M,I,JA,K,NG)
14090 BALY(1,I,JA,K)=BALY(1,I,JA,K)+AF+WET
14100 AF=ANGFLJ(I02,M,I,JA,K,NG)
14110 BALY(2,I,JA,K)=BALY(2,I,JA,K)+AF+WET
14120 23 CONTINUE
14130 DD 24 I=1,IMX
14140 DD 24 J=1,JMX
14150 DD 24 KA=1,KMX+1
14160 SQ=DELX(I)=DELY(J)
14170 BALZ(1,I,J,KA)=0.0
14180 BALZ(2,I,J,KA)=0.0
14190 DD 24 ID=1,2

```

```

14200          DD 24 JD=1,2
14210          IQ1=NOUAD(ID,JD,1)
14220          IQ2=NOUAD(ID,JD,2)
14230          DD 24 M=1,MMX
14240          WAK=WGT(M)*AKSI(M)+SO
14250          AF=ANGFLK(IQ1,M,1,J,KA,NG)
14260          BALZ(1,I,J,KA)=BALZ(1,I,J,KA)+AF*WAK
14270          AF=ANGFLK(IQ2,M,1,J,KA,NG)
14280          BALZ(2,I,J,KA)=BALZ(2,I,J,KA)+AF*WAK
14290 24      CONTINUE
14300          IF(ISTART.EQ.1.AND.ITROUT.EQ.1) GOTO 701
14305**** R E B A L A N C E   G R A D E   D E T E R M I N A T I O N ****
14310          ICONB=IFLG(NG)
14320          IF(IGRD.NE.2) GOTO 401
14330          IH=ITRIN/2
14340          IH=IH*2
14350          IF(IH.NE.ITRIN) GOTO 401
14360          ICONB=-IFLG(NG)
14370 401      IF(IFLG(NG).GE.-2) GOTO 45
14375          CUL=CONVMX(NG)
14380          IF(IGRD.EQ.3.DR.CMX.LE.CUL) GOTO 46
14410          IFLG(NG)=-4
14420          GOTO 45
14430C*****
14440 46      IF(IFLG(NG).LE.-5.OR.CUL.LT.1.OE-3) GOTO 45
14450          CONE=ABS(1.O-EGNOLD/EIGEN)
14460          IF(CONE.GT.1.OE-3) GOTO 45
14470          ICONB=-5
14480          IF(IGRD.EQ.3) GOTO 47
14490          IFLG(NG)=-5
14500 45      ICONB=IFLG(NG)
14510 47      CALL REBAL(QV,SS,BALX,BALY,BALZ,IBNDCN,RFC)
14520 701      CONTINUE
14530          IF(ICONB.NE.0) GOTO 801
14540          DD 27 K=1,KMX
14550          DD 27 J=1,JMX
14560          DD 27 I=1,IMX
14570          RFC(I,J,K)*1.0
14580 27      CONTINUE
14590          GOTO 901
14600C*****
14610 801      CONTINUE
14620          DD 28 K=1,KMX
14630          DD 28 J=1,JMX
14640          DD 28 I=1,IMX
14650          SCLFL(I,J,K,NG)=SCLFL(I,J,K,NG)*RFC(I,J,K)
14660 28      CONTINUE
14670          IF(IBNDCN(1,1).EQ.0) GOTO 1029
14680          DD 29 K=1,KMX
14690          DD 29 J=1,JMX
14700          DD 29 JD=1,2
14710          DD 29 KD=1,2
14720          N2=NOUAD(2,JD,KD)
14730          DD 29 M=1,MMX
14740          ANGFLI(N2,M,IMX+1,J,K,NG)=ANGFLI(N2,M,IMX+1,J,K,
14750          NG)*RFC(IMX,J,K)
14760 29      CONTINUE
14770 1029      IF(1BNDCN(2,1).EQ.0) GOTO 1030
14780          DD 30 K=1,KMX
14790          DD 30 I=1,IMX
14800          DD 30 ID=1,2
14810          DD 30 KD=1,2
14820          N2=NOUAD(ID,2,KD)
14830          DD 30 M=1,MMX
14840          ANGFLJ(N2,M,I,JMX+1,K,NG)=ANGFLJ(N2,M,I,JMX+1,K,
14850          NG)*RFC(I,JMX,K)
14860 30      CONTINUE
14870 1030      IF(1BNDCN(3,1).EQ.0) GOTO 901
14880          DD 31 J=1,JMX
14890          DD 31 I=1,IMX
14900          DD 31 ID=1,2
14910          DD 31 JD=1,2
14920          N2=NOUAD(ID,JD,2)
14930          DD 31 M=1,MMX
14940          ANGFLK(N2,M,I,J,KMX+1,NG)=ANGFLK(N2,M,I,J,KMX+1,
14950          NG)*RFC(I,J,KMX)
14960 31      CONTINUE
14970C*****
14980 901      CONTINUE
14990C**** E N D   O F   I N N E R   R O U P ****
15000          ITRIN=ITRIN+1
15010          NTREB=NTREB+ITREB
15020          IF(ITRIN.EQ.ITINMX) GOTO 8
15030          CMX=0
15040          DD 32 J=1,JMX
15050          DD 32 K=1,KMX
15060          DD 32 I=1,IMX
15070          ERR=ABS(1.O-SCLFL(I,J,K,NG)/SCLDL(I,J,K))
15080          CMX=AMAX1(CMX,ERR)
15090 32      CONTINUE
15100          IF(CMX.LE.CONCRI) GOTO 8
15110          GOTO 10
15120C*****
15130 8      CONTINUE
15140          WRITE(6,610) NG,ITRIN,NTREB,RFMX,CMX
15150 610      FORMAT(12X,I5,I7,I8,8X,1PE12.3,42X,E12.3)
15160          DD 41 K=1,KMX
15170          DD 41 J=1,JMX
15180          DD 41 I=1,IMX
15190          RF=RFC(I,J,K)
15200          TBALX(1,I,J,K)=TBALX(1,I,J,K)+BALX(1,I,J,K)*RF
15210          TBALX(2,I+1,J,K)=TBALX(2,I+1,J,K)+BALX(2,I+1,J,K)*RF
15220          TBALY(1,I,J,K)=TBALY(1,I,J,K)+BALY(1,I,J,K)*RF
15230          TBALY(2,I,J+1,K)=TBALY(2,I,J+1,K)+BALY(2,I,J+1,K)*RF
15240          TBALZ(1,I,J,K)=TBALZ(1,I,J,K)+BALZ(1,I,J,K)*RF
15250 41      TBALZ(2,I,J,K+1)+TBALZ(2,I,J,K+1)+BALZ(2,I,J,K+1)*RF
15260          DD 42 J=1,JMX
15270          DD 42 K=1,KMX
15280          TBALX(2,1,J,K)=TBALX(2,1,J,K)+BALX(2,1,J,K)*RFC(1,J,K)
15290 42      TBALX(1,IMX+1,J,K)=TBALX(1,IMX+1,J,K)+BALX(1,IMX+1,J,K)
15300          *RFC(IMX,J,K)
15310 8      DD 43 I=1,IMX
15320          DD 43 K=1,KMX
15330          TBALY(2,1,1,K)+TBALY(2,1,1,K)+BALY(2,1,1,K)*RFC(1,1,K)
15340 43      TBALY(1,1,JMX+1,K)+TBALY(1,1,JMX+1,K)+BALY(1,1,JMX+1,K) *RFC(1,
15350          JMX,K)
15360 8      DD 44 I=1,IMX
15370          DD 44 J=1,JMX
15380          TBALZ(2,1,J,1)+TBALZ(2,1,J,1)+BALZ(2,1,J,1)*RFC(1,J,1)
15390 44      TBALZ(1,1,J,KMX+1)+TBALZ(1,1,J,KMX+1)+BALZ(1,1,J,KMX+1) *RFC(1,
15400          J,KMX)
15410C---- S W E E P   N E X T   G R O U P ----
15420          NG=NG+1
15430          IF(NG.EQ.NGMX+1) GOTO 7
15440          GOTO 11
15450C*****
15460 7      CONTINUE
15470          RETURN
15480          END

```

```

15490 SUBROUTINE LINER(ALR,AUD,ANF,AMU,ETA,AKSI,DELX,DELY,DELZ,S,XT)
15500 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
15510 & NQUAD(2,2,2)
15520 COMMON /SWEEP/IDRCT,I,J,K,NG
15530 DIMENSION ALR(MFIX,IMX),AUD(MFIX,IMX),ANF(MFIX,IMX),AMU(MFIX),
15540 & ETA(MFIX),AKSI(MFIX),DELX(IMX),DELY(JMX),DELZ(KMX),S(IMX),
15550 & XT(IMX)
15560 DIMENSION F1(3,2),F2(3,2)
15570 1 CONTINUE
15580 DD 2 M=1,MMX
15590 AFID=ALR(M,I)
15600 AFJD=AUD(M,I)
15610 AFKD=ANF(M,I)
15620 A=2.0*AMU(M)/DELX(I)
15630 B=2.0*ETA(M)/DELY(J)
15640 C=2.0*AKSI(M)/DELZ(K)
15650 AFC=(A*AFID+B*AFJD+C*AFKD+S(I))/(A+B+C+XT(I))
15660 AFC=AFC+AFC
15670 AFIN=AFC-AFID
15680 AFJN=AFC-AFJD
15690 AFKN=AFC-AFKD
15700C**** N E G A T I V E F L U X F I X - U P E N T R Y ****
15710C-----
15720C FIXUP POSITIVE AF. NEGATIVE AF.
15730C FACTORS I J K I J K
15740 DATA F1 /1.0,1.0,1.0,0.5,0.5,0.5/
15750 DATA F2 /1.0,1.0,1.0,0.0,0.0,0.0/
15760C-----
15770 IF(AFIN.LE.0.0.OR.AFJN.LE.0.0.OR.AFKN.LE.0.0) GOTO 101
15780 GOTO 103
15790C*****
15800 101 IFLAG=0
15810 10 CONTINUE
15820 IFI=1
15830 IFJ=1
15840 IFK=1
15850 IF(AFIN.LT.0.0) IFI=2
15860 IF(AFJN.LT.0.0) IFJ=2
15870 IF(AFKN.LT.0.0) IFK=2
15880 AFC=(A*FI(1,IFI)+B*FJ(2,IFJ)+C*FK(3,IFK)+XT(I))
15890 S(I)=S(I)/(A*FI(1,IFI)+B*FJ(2,IFJ)+C*FK(3,IFK)+XT(I))
15900 AFIN=F2(1,IFI)*(2.0-AFC-AFID)
15910 AFJN=F2(2,IFJ)*(2.0-AFC-AFJD)
15920 AFKN=F2(3,IFK)*(2.0-AFC-AFKD)
15930 IF(AFIN.LT.0.0.OR.AFJN.LT.0.0.OR.AFKN.LT.0.0) IFLAG=IFLAG+1
15940 IF(IFLAG.GE.1.AND.IFLAG.LE.3) GOTO 10
15950 103 CONTINUE
15960 ALR(M,I)=AFIN
15970 AUD(M,I)=AFJN
15980 ANF(M,I)=AFKN
15990 2 CONTINUE
16000 I=I+IDRCT
16010 IF(I.EQ.0.OR.I.EQ.IMX+1) GOTO 102
16020 IOLD=I-IDRCT
16030 DD 20 M=1,MMX
16040 ALR(M,I)=ALR(M,IOLD)
16050 20 CONTINUE
16060 GOTO 1
16070C*****
16080 102 CONTINUE
16090 RETURN
16100 END
16110 SUBROUTINE REBAL(QV,SS,BALX,BALY,BALZ,IBNDCN,RFC)
16120 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
16130 & NQUAD(2,2,2)
16140 COMMON /MORIT/ECNOLD,EIGEN,PWRDLD,PWRNEW,CDNCR,CONCRI,CONCRE,
16150 & CONCR,TIMEOUT,TIMLNT,TIMCPU,ITRDT,ITRIN,ITREB,ISTA,IPRV,NOW,
16160 & IMXM,JMNM,KMXM,NGMXM
16170 COMMON /RBFRC/IRBMX,RFCUL,RFCLL,RPMX,ICONB,IFLG(70)
16180 COMMON /SWEEP/IDRCT,I,J,K,NG
16190 DIMENSION QV(IMX,JMX,KMX),BALX(2,IMX+1,JMX,KMX),BALY(2,IMX,JMX+1,
16200 & KMX),BALZ(2,IMX,JMX,KMX+1),RFC(IMX,JMX,KMX),SS(IMX,JMX,KMX),
16210 & IBNDCN(3,2)
16220 IF(ICONB.EQ.0) RETURN
16230 IF(ICONB.GT.-5) GOTO 302
16240 ITREB=0
16250 1001 CONTINUE
16260 RFMX=0.0
16270 DD 5 K=1,KMX
16280 DD 5 J=1,JMX
16290 DD 5 I=1,IMX
16300 RFCO=RFC(I,J,K)
16310 FL1=BALX(1,I,J,K)
16320 FL2=BALX(2,I,J,K)
16330 FR1=BALX(1,I+1,J,K)
16340 FR2=BALX(2,I+1,J,K)
16350 FU1=BALY(1,I,J,K)
16360 FU2=BALY(2,I,J,K)
16370 FD1=BALY(1,I,J+1,K)
16380 FD2=BALY(2,I,J+1,K)
16390 FN1=BALZ(1,I,J,K)
16400 FN2=BALZ(2,I,J,K)
16410 FF1=BALZ(1,I,J,K+1)
16420 FF2=BALZ(2,I,J,K+1)
16430 A=QV(I,J,K)
16440 B=SS(I,J,K)
16450 IF(A.GT.0.0.AND.B.GT.0.0) GOTO 301
16460 GOTO 303
16470C*****
16480 301 IF(I.EQ.IMX) GOTO 101
16490 A=A+RFC(I+1,J,K)*FR1
16500 GOTO 102
16510C*****
16520 101 IF(IBNDCN(1,1).NE.1) GOTO 102
16530 B=B+FR2-FR1
16540 GOTO 100
16550C*****
16560 102 B=B+FR2
16570 100 IF(I.EQ.1) GOTO 103
16580 A=A+RFC(I-1,J,K)*FL2
16590 GOTO 104
16600C*****
16610 103 IF(IBNDCN(1,2).NE.1) GOTO 104
16620 B=B+FL1-FL2
16630 GOTO 106
16640C*****
16650 104 B=B+FL1
16660 105 IF(J.EQ.JMX) GOTO 106
16670 A=A+RFC(I,J+1,K)*FD1
16680 GOTO 107
16690C*****
16700 106 IF(IBNDCN(2,1).NE.1) GOTO 107
16710 B=B+FD2-FD1
16720 GOTO 110
16730C*****
16740 107 B=B+FD2
16750 110 IF(J.EQ.1) GOTO 111
16760 A=A+RFC(I,J-1,K)*FU2
16770 GOTO 112

```

```

16780C*****
16790 111 IF(IBNDCH(2,2).NE.1) GOTO 112
16800 B=B+FU1-FU2
16810 GOTO 115
16820C*****
16830 112 B=B+FU1
16840 115 IF(K.EQ.KMX) GOTO 116
16850 A=A+RFC(1,J,K+1)+FF1
16860 GOTO 117
16870C*****
16880 116 IF(IBNDCH(3,1).NE.1) GOTO 117
16890 B=B+FF2-FF1
16900 GOTO 120
16910C*****
16920 117 B=B+FF2
16930 120 IF(K.EQ.1) GOTO 121
16940 A=A+RFC(1,J,K-1)+FN2
16950 GOTO 122
16960C*****
16970 121 IF(IBNDCH(2,2).NE.1) GOTO 122
16980 B=B+FN1-FN2
16990 GOTO 125
17000C*****
17010 122 B=B+FN1
17020 125 CONTINUE
17030 IF(B.LE.O.O) GOTO 302
17040 F=A/B
17050 IF(F.GT.RFCUL) GOTO 302
17060 IF(F.LT.RFCLL) GOTO 302
17070 RFC(1,J,K)=F
17080 RFCNV=ABS((RFC(1,J,K)-RFCO)/RFC(1,J,K))
17090 RFMX=AMAX1(RFCNV,RFMX)
17100 5 CONTINUE
17110 IF(RFMX.LE.CONCRR) RETURN
17120 ITRB=ITRB+1
17130 IF(ITRB.LE.IRBMX) GOTO 1001
17140C**** T O T A L R E B A L A N C E O N L Y ****
17150 302 TLK=O.O
17160 TOV=O.O
17170 TSS=O.O
17180 DD 131 K=1,KMX
17190 DD 131 J=1,JMX
17200 131 TLK=TLK+BALX(1,1,J,K)-BALX(2,1,J,K)-BALX(1,IMX+1,J,K)+ BALX(2,
17210 & IMX+1,J,K)
17220 DD 132 K=1,KMX
17230 DD 132 I=1,IMX
17240 132 TLK=TLK+BALY(1,1,1,K)-BALY(2,1,1,K)-BALY(1,1,JMX+1,K)+ BALY(2, 1,
17250 & JMX+1,K)
17260 DD 133 I=1,IMX
17270 DD 133 J=1,JMX
17280 133 TLK=TLK+BALZ(1,1,J,1)-BALZ(2,1,J,1)-BALZ(1,1,J,KMX+1)+ BALZ(2, 1,
17290 & J, KMX+1)
17300 DD 134 K=1,KMX
17310 DD 134 J=1,JMX
17320 DD 134 I=1,IMX
17330 TOV=TOV+OV(1,J,K)
17340 134 TSS=TSS+SS(1,J,K)
17350 F=TOV/(TSS+TLK)
17360 DD 135 K=1,KMX
17370 DD 135 J=1,JMX
17380 DD 135 I=1,IMX
17390 135 RFC(1,J,K)=F
17400 RETURN
17410C*****
17420 303 ICONB=O
17430 RETURN
17440 END
17450 SUBROUTINE CONTE(SCLFL,SCLLD,CONVMX,CONV,CONV)
17460 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,MFIX,ITINMX,
17470 & NQUAD(2,2)
17480 COMMON /SWEEP/IDRCT,I,J,K,NG
17490 COMMON /MONIT/EGHOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
17500 & CONCR,TIMOUT,TIMLMT,TIMCPU, ITROUT,ITRIN,ITRB,ISTA,IPRV,NDW,
17510 & IMXM,JMXM,KMXM,NGMXM
17520 COMMON /DPTON/ISTART,MONT,IROMA,IGRD
17530 COMMON /REBFC/IRBMX,RFCUL,RFCLL,RFMX,ICONB,IFLG(70)
17540 DIMENSION SCLFL(IMX,JMX,KMX,NGMX),SCLLD(IMX,JMX,KMX,NGMX),
17550 & CONVMX(NGMX),CONV(IMX),CONVM(IMXM,JMXM),MONI(10),MONJ(10),
17560 & MDNK(10),MDNG(10)
17570 DATA MONI /1,2,3,4,5,6,-1,-1,-1,-1/
17580 DATA MONJ /1,2,3,4,5,6,-1,-1,-1,-1/
17590 DATA MDNK /1, 2, 3,-1,-1,-1,-1,-1,-1,-1/
17600 DATA MDNG /1, 3,-1,-1,-1,-1,-1,-1,-1,-1/
17610 ISTART=-1
17620 IF(MONT.NE.-1) GOTO 1001
17630C ---- M O N I T O R S E C T I O N N O . 1 ----
17640 MDNIMX=O
17650 MDNJMX=O
17660 MDNKMx=O
17670 MDNGMX=O
17680 DD 11 I1=1,10
17690 IF(MONI(I1).GT.O) MDNIMX=MDNIMX+1
17700 IF(MONJ(I1).GT.O) MDNJMX=MDNJMX+1
17710 IF(MDNK(I1).GT.O) MDNKMx=MDNKMx+1
17720 IF(MDNG(I1).GT.O) MDNGMX=MDNGMX+1
17730 IF(MONI(MDNIMX).GT.IMX) MDNIMX=MDNIMX-1
17740 IF(MDNJ(MDNJMX).GT.JMX) MDNJMX=MDNJMX-1
17750 IF(MDNK(MDNKMx).GT.KMX) MDNKMx=MDNKMx-1
17760 IF(MDNG(MDNGMX).GT.NGMX) MDNGMX=MDNGMX-1
17770 11 CONTINUE
17780 WRITE(6,601) ITROUT
17790 601 FORMAT(/H, '*** CONVERGENCE INFORMATIONS OF OUTER ITR. NO. ',
17800 & I4, ' ***'/)
17810C ---- C O N V E R G E N C E T E S T S E C T I O N ----
17820 1001 MG=1
17830 DD 3 NG=1,NGMX
17840 CONVMX(NG)=O.O
17850 MK=1
17860 DD 3 K=1,KMX
17870 MJ=1
17880 DD 2 J=1,JMX
17890 DD 1 I=1,IMX
17900 SCLFLX=SCLFL(I,J,K,NG)
17910 SCLD=SCLLD(I,J,K,NG)
17920 CONV(I)=(SCLFLX-SCLD)/SCLFLX
17930 SCLLD(I,J,K,NG)=SCLFLX
17940 CNV=ABS(CONV(I))
17950 IF(CNV.GE.CONVMX(NG)) CONVMX(NG)=CONV(I)
17960 1 CONTINUE
17970 IF(MONT.NE.-1) GOTO 2
17980C ---- M O N I T O R S E C T I O N N O . 2 ----
17990 IF(NG.NE.MDNG(MG)) GOTO 2
18000 IF(K.NE.MDNK(MK)) GOTO 2
18010 IF(J.NE.MDNJ(MJ)) GOTO 2
18020 DD 21 MI=1,MDNIMX
18030 CONVM(MI,MJ)=CONV(MONI(MI))
18040 21 CONTINUE
18050 MJ=MJ+1
18060C ---- N E X T J

```

```

18070 2 CONTINUE
18080 IF(MONT.NE.-1) GOTO 3001
18090C ---- M O N I T O R S E C T I O N N O . 3 ----
18100 IF(NG.NE.MONG(MG)) GOTO 3001
18110 IF(K.NE.MONK(MK)) GOTO 3001
18120 WRITE(6,602) NG,K
18130 602 FORMAT(5X,' NG=',13,3X,' K =',13//)
18140 WRITE(6,603) (MDN(MI),MI=1,MONIMX)
18150 603 FORMAT(5X,10(13,3X)/)
18160 DO 31 MJ=1,MDNJMK
18170 DO 32 MI=1,MONIMX
18180 C=ABS(CONVM(MI,MJ))
18190 CONVM(MI,MJ)=ALOG(C)/ALOG(10.0)
18200 32 CONTINUE
18210 WRITE(6,604) MONJ(MJ), (CONVM(MI,MJ),MI=1,MONIMX)
18220 604 FORMAT(13,2X,10(F6.2))
18230 31 CONTINUE
18240 MK=MK+1
18250 IF(MK.LE.MONKMX) GOTO 301
18260 MG=MG+1
18270 301 CONTINUE
18280C ---- J U D G E I F A L L F L U X E S C O N V E R G E D ----
18290 3001 CNVM=ABS(CONVMX(NG))
18300 CONE=ABS(1.0-EGNOLD/EIGEN)
18310 IF(CNVM.GT.CONCR.OR.CONE.GT.CONCRE) ISTART=0
18320C ---- N E X T K , N C
18330 3 CONTINUE
18340C ---- M O N I T O R S E C T I O N N O . 4 ----
18350C ( S T A N D A R D D P T I O N )
18360C
18370 CMX=0.0
18380 DO 41 NG=1,NGMX
18390 CNV=ABS(CONVMX(NG))
18400 IF(CMX.GE.CNV) GOTO 101
18410 CMX=CNV
18420 NGX=NG
18430 101 CONTINUE
18440 41 CONTINUE
18450 IHOOR=INT(TIMCPU/60.)
18460 MIN=INT(TIMCPU-FLOAT(IHOOR*60))
18470 SEC=60.0-TIMCPU-FLOAT(3600-IHOOR*60-MIN)
18480 WRITE(6,606) ITROUT,ITRES,RFMX,IHOOR,MIN,SEC,EIGEN,PWRNEW,
18490 & CONVMX(NGX),NGX
18500 606 FORMAT(1H,19,3X,12X,15,6X,1PE12,3,216,OPF6,2,2F12,7,1PE12,3,16)
18510 RETURN
18520 END
18530 SUBROUTINE CDDA(SCLFL,ANGFLI,ANGFLJ,ANGFLK,FISS,QQ)
18540 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
18550 & NQUAD(2,2)
18560 DIMENSION SCLFL(IMX,JMX,KMX,NGMX), ANGFLI(8,MFIX,IMX+1,JMX,KMX,
18570 & NGMX),ANGFLJ(8,MFIX,IMX,JMX+1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,
18580 & KMX+1,NGMX),QQ(IMX,JMX,KMX,NGMX),IDPT(6),NDOUT(6)
18590C -----
18600C DUMP ARRAYS / FLUX,ANI,ANJ,ANK,FIS,SRG/
18610C DATA IDPT / -1,0,0,0,-1,0 /
18620C DATA NDUT / 6,9,9,9,6,0 /
18630C
18640C IF(NDOUT) (TAKE DUMP ON 'NDUT' FILE + PRINT OUT,
18650C & TAKE NO DUMP,
18660C & TAKE DUMP ON 'NDUT' FILE ONLY.)
18670C -----
18680C
18690 WRITE(6,666)
18700 666 FORMAT(1H1,35X,10(1H*),' F L U X C O N V E R G E D ',10(1H*)//)
18710 CALL REWER(IDPT,NDOUT)
18720 IF(IDPT(1).NE.-1) GOTO 101
18730 LREC=IMX+JMX+KMX+NGMX
18740 NF=IABS(NDOUT(1))
18750 IF(NF.EQ.0.OR.NF.EQ.6) GOTO 102
18760 CALL DUMPER(NF,2,FLUX,LREC,4H FL,4HUX,IMX,JMX,KMX,NGMX,-1,-
18770 & 1,4H I,4H J,4H K,4H NG,4H )
18780 102 CONTINUE
18790 IF(NDOUT(1).LT.0) GOTO 101
18800 CALL DUMPER(6,2,SCLFL,LREC,4H FL,4HUX,IMX,JMX,KMX,NGMX,-1,-
18810 & 1,4H I,4H J,4H K,4H NG,4H )
18820 101 CONTINUE
18830 IF(IDPT(2).NE.-1) GOTO 201
18840 LREC=8*MMX+(IMX+1)*JMX+KMX+NGMX
18850 NF=IABS(NDOUT(2))
18860 IF(NF.EQ.0.OR.NF.EQ.6) GOTO 202
18870 CALL DUMPER(NF,2,ANGFLI,LREC,4HANG,4HFL,I,8,MMX,IMX+1,JMX,
18880 & KMX,NGMX,4H IO,4H M,4H I,4H J,4H K,4H NG )
18890 202 CONTINUE
18900 IF(NDOUT(2).LT.0) GOTO 301
18910 CALL DUMPER(6,2,ANGFLI,LREC,4HANG,4HFL,I,8,MMX,IMX+1,JMX,
18920 & KMX,NGMX,4H IO,4H M,4H I,4H J,4H K,4H NG )
18930 201 CONTINUE
18940 IF(IDPT(3).NE.-1) GOTO 301
18950 LREC=8*MMX+IMX*(JMX+1)+KMX+NGMX
18960 NF=IABS(NDOUT(3))
18970 IF(NF.EQ.0.OR.NF.EQ.6) GOTO 302
18980 CALL DUMPER(NF,2,ANGFLJ,LREC,4HANG,4HFL,I,8,MMX,IMX,JMX+1,
18990 & KMX,NGMX,4H IO,4H M,4H I,4H J,4H K,4H NG )
19000 302 CONTINUE
19010 IF(NDOUT(3).LT.0) GOTO 301
19020 CALL DUMPER(6,2,ANGFLJ,LREC,4HANG,4HFL,I,8,MMX,IMX,JMX+1,
19030 & KMX,NGMX,4H IO,4H M,4H I,4H J,4H K,4H NG )
19040 301 CONTINUE
19050 IF(IDPT(4).NE.-1) GOTO 401
19060 LREC=8*MMX+IMX*(JMX+1)+KMX+NGMX
19070 NF=IABS(NDOUT(4))
19080 IF(NF.EQ.0.OR.NF.EQ.6) GOTO 402
19090 CALL DUMPER(NF,2,ANGFLK,LREC,4HANG,4HFL,I,8,MMX,IMX,JMX,
19100 & KMX+1,NGMX,4H IO,4H M,4H I,4H J,4H K,4H NG )
19110 402 CONTINUE
19120 IF(NDOUT(4).LT.0) GOTO 401
19130 CALL DUMPER(6,2,ANGFLK,LREC,4HANG,4HFL,I,8,MMX,IMX,JMX,
19140 & KMX+1,NGMX,4H IO,4H M,4H I,4H J,4H K,4H NG )
19150 401 CONTINUE
19160 IF(IDPT(5).NE.-1) GOTO 501
19170 LREC=IMX+JMX+KMX
19180 NF=IABS(NDOUT(5))
19190 IF(NF.EQ.0.OR.NF.EQ.6) GOTO 502
19200 CALL DUMPER(NF,2,FISS,LREC,4H FIS,4HSION,IMX,JMX,KMX,-1,-1,-
19210 & 1,4H I,4H J,4H K,4H ,4H ,4H )
19220 502 CONTINUE
19230 IF(NDOUT(5).LT.0) GOTO 501
19240 CALL DUMPER(6,2,FISS,LREC,4H FIS,4HSION,IMX,JMX,KMX,-1,-1,-1,
19250 & 4H I,4H J,4H K,4H ,4H ,4H )
19260 501 CONTINUE
19270 IF(IDPT(6).NE.-1) GOTO 701
19280 LREC=IMX+JMX+KMX+NGMX
19290 NF=IABS(NDOUT(6))
19300 IF(NF.EQ.0.OR.NF.EQ.6) GOTO 702
19310 CALL DUMPER(NF,2,QQ,LREC,4H SOU,4HRCE,IMX,JMX,KMX,NGMX,-1,-
19320 & 1,4H I,4H J,4H K,4H NG,4H ,4H )
19330 702 CONTINUE
19340 IF(NDOUT(6).LT.0) GOTO 701
19350 CALL DUMPER(6,2,QQ,LREC,4H SOU,4HRCE,IMX,JMX,KMX,NGMX,-1,-1,

```



```

19360      &      4H 1 ,4H J ,4H K ,4H NG ,4H      ,4H      ]
19370 701 CONTINUE
19380 RETURN
19390 END
19400 SUBROUTINE REACT(SCLFL,ANGFLI,ANGFLJ,ANGFLK,GRX,TX,DELX,DELY,DELZ,
19410 & AMU,ETA,AKSI,WGT,KBS,XFIS,XTRA,XSCT,XI,MAP)
19420 DIMENSION SCLFL(1,MX,JMX,KMX,NGMX),ANGFLI(8,MFIX,IMX+1,JMX,KMX),
19430 & NGMX),ANGFLJ(8,MFIX,IMX,JMX+1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,
19440 & KMX+1,NGMJ)
19450 DIMENSION GRX(30,NGMX),TX(30),DELX(IMX),DELY(JMX),DELZ(KMX),
19460 & AMU(MFIX),ETA(MFIX),AKSI(MFIX),WGT(MFIX),XABS(NGMX,MAPMX),
19470 & XFIS(NGMX,MAPMX),XTRA(NGMX,MAPMX),XSCT(NGMX,NGMX,MAPMX),
19480 & XI(NGMX),MAP(IMX,JMX,KMX)
19490 COMMON /CONT/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
19500 & NQUAD(2,2,2)
19510 DD 300 IX=1,30
19520 TX(IX)=0.0
19530 DD 300 NG=1,NGMX
19540 GRX(IX,NG)=0.0
19550 300 CONTINUE
19560 DD 1 NG=1,NGMX
19570 DD 1 K=1,KMX
19580 DD 1 J=1,JMX
19590 DD 1 I=1,IMX
19600 FLXVOL=SCLFL(I,J,K,NG)*DELX(I)*DELY(J)*DELZ(K)
19610 MAPI=MAP(I,J,K)
19620 GRX(1,NG)=GRX(1,NG)+FLXVOL
19630 GRX(2,NG)=GRX(2,NG)+XFIS(NG,MAPI)*FLXVOL
19640 GRX(3,NG)=GRX(3,NG)+XABS(NG,MAPI)*FLXVOL
19650 GRX(4,NG)=GRX(4,NG)+XTRA(NG,MAPI)*FLXVOL
19660 GRX(25,NG)=GRX(25,NG)+XSCT(NG,NG,MAPI)*FLXVOL
19670 DD 310 NN=1,NG-1
19680 FLXVOL=SCLFL(I,J,K,NN)*DELX(I)*DELY(J)*DELZ(K)
19690 SSS=XSCT(NN,NG,MAPI)*FLXVOL
19700 GRX(26,NG)=GRX(26,NG)+SSS
19710 310 CONTINUE
19720 DD 311 NN=NG+1,NGMX
19730 FLXVOL=SCLFL(I,J,K,NN)*DELX(I)*DELY(J)*DELZ(K)
19740 SSS=XSCT(NN,NG,MAPI)*FLXVOL
19750 GRX(27,NG)=GRX(27,NG)+SSS
19760 311 CONTINUE
19770 1 CONTINUE
19780 DD 2 NG=1,NGMX
19790 DD 301 J=1,JMX
19800 DD 301 K=1,KMX
19810 SQ=DELY(J)*DELZ(K)
19820 DD 301 JD=1,2
19830 DD 301 KD=1,2
19840 IQ1=NOUAD(1,JD,KD)
19850 IQ2=NOUAD(2,JD,KD)
19860 DD 301 M=1,MMX
19870 WAM=WGT(M)*AMU(M)*SQ
19880 AF=ANGFLI(IQ2,M,1,J,K,NG)
19890 GRX(6,NG)=GRX(6,NG)+AF*WAM
19900 AF=ANGFLI(IQ1,M,1,J,K,NG)
19910 GRX(12,NG)=GRX(12,NG)+AF*WAM
19920 AF=ANGFLI(IQ1,M,IMX+1,J,K,NG)
19930 GRX(7,NG)=GRX(7,NG)+AF*WAM
19940 AF=ANGFLI(IQ2,M,IMX+1,J,K,NG)
19950 GRX(13,NG)=GRX(13,NG)+AF*WAM
19960 301 CONTINUE
19970 GRX(18,NG)=GRX(12,NG)-GRX(6,NG)
19980 GRX(19,NG)=GRX(13,NG)-GRX(7,NG)
19990 DD 302 K=1,KMX
20000 DD 302 I=1,IMX
20010 SQ=DELZ(K)*DELX(I)
20020 DD 302 KD=1,2
20030 DD 302 ID=1,2
20040 IQ1=NOUAD(ID,1,KD)
20050 IQ2=NOUAD(ID,2,KD)
20060 DD 302 M=1,MMX
20070 WET=WGT(M)*ETA(M)*SQ
20080 AF=ANGFLJ(IQ2,M,1,I,K,NG)
20090 GRX(8,NG)=GRX(8,NG)+AF*WET
20100 AF=ANGFLJ(IQ1,M,1,I,K,NG)
20110 GRX(14,NG)=GRX(14,NG)+AF*WET
20120 AF=ANGFLJ(IQ1,M,1,JMX+1,I,K,NG)
20130 GRX(9,NG)=GRX(9,NG)+AF*WET
20140 AF=ANGFLJ(IQ2,M,I,JMX+1,I,K,NG)
20150 GRX(15,NG)=GRX(15,NG)+AF*WET
20160 302 CONTINUE
20170 GRX(20,NG)=GRX(14,NG)-GRX(8,NG)
20180 GRX(21,NG)=GRX(15,NG)-GRX(9,NG)
20190 DD 303 I=1,IMX
20200 DD 303 J=1,JMX
20210 SQ=DELX(I)*DELY(J)
20220 DD 303 ID=1,2
20230 DD 303 JD=1,2
20240 IQ1=NOUAD(ID,JD,1)
20250 IQ2=NOUAD(ID,JD,2)
20260 DD 303 M=1,MMX
20270 WAK=WGT(M)*AKSI(M)*SQ
20280 AF=ANGFLK(IQ2,M,I,J,I,NG)
20290 GRX(10,NG)=GRX(10,NG)+AF*WAK
20300 AF=ANGFLK(IQ1,M,I,J,I,NG)
20310 GRX(16,NG)=GRX(16,NG)+AF*WAK
20320 AF=ANGFLK(IQ1,M,I,J,KMX+1,NG)
20330 GRX(11,NG)=GRX(11,NG)+AF*WAK
20340 AF=ANGFLK(IQ2,M,I,J,KMX+1,NG)
20350 GRX(17,NG)=GRX(17,NG)+AF*WAK
20360 303 CONTINUE
20370 GRX(22,NG)=GRX(16,NG)-GRX(10,NG)
20380 GRX(23,NG)=GRX(17,NG)-GRX(11,NG)
20390 DD 304 IX=1,30
20400 TX(IX)=TX(IX)+GRX(IX,NG)
20410 304 CONTINUE
20420 2 CONTINUE
20430 TLEAK=0.0
20440 DD 305 IX=18,23
20450 TLEAK=TLEAK+TX(IX)
20460 305 CONTINUE
20470 EIGEN=TX(2)/(TX(3)+TLEAK)
20480 PWRNEW=TX(2)
20490 DD 306 NG=1,NGMX
20500 GRX(29,NG)=TX(2)*XI(NG)/EIGEN
20510 GRX(30,NG)=GRX(4,NG)-GRX(25,NG)
20520 DD 306 IX=18,23
20530 GRX(30,NG)=GRX(30,NG)+GRX(IX,NG)
20540 306 CONTINUE
20550 DD 307 NG=1,NGMX
20560 DD 307 K=1,KMX
20570 DD 307 J=1,JMX
20580 DD 307 I=1,IMX
20590 SCLFL(I,J,K,NG)=SCLFL(I,J,K,NG)/PWRNEW
20600 307 CONTINUE
20610 WRITE(6,600)
20620 600 FORMAT(1H1,20X,17(1H*))
20630 & ' R E A C T I O N R A T E S T A B L E ',17(1H*)/////
20640 WRITE(6,601)

```

```

20650 601 FORMAT(12X,5HGROUP,12H FLUX=VOLUME,12H FISSION
20660 & 12H ABSORPTION,12H REMOVAL,12H TRANSPORT,
20670 & 12H SELF SCAT.,12H UP SCAT.,12H DOWN SCAT./)
20680 DO 320 NG=1,NGMX
20690 WRITE(6,602) NG,GRX(1,NG),GRX(29,NG),GRX(3,NG),GRX(30,NG),GRX(4,
20700 NG),GRX(25,NG),GRX(27,NG),GRX(26,NG)
20710 320 CONTINUE
20720 602 FORMAT(12X,13,2X,6(1PE12.4))
20730 WRITE(6,603) TX(1),TX(29),TX(3),TX(30),TX(4),TX(25),TX(27),TX(26)
20740 603 FORMAT(/12X,5HTOTAL,6(1PE12.4)/)
20750 WRITE(6,604)
20760 604 FORMAT(5X,'IN FLOWS'/)
20770 WRITE(6,605)
20780 605 FORMAT(24X,5HGROUP,12H LEFT,12H RIGHT
20790 & 12H FRONT,12H BACK,12H TOP
20800 & 12H BOTTOM /)
20810 DO 321 NG=1,NGMX
20820 WRITE(6,606) NG,GRX(6,NG),GRX(7,NG),GRX(8,NG),GRX(9,NG),GRX(10,
20830 NG),GRX(11,NG)
20840 321 CONTINUE
20850 606 FORMAT(24X,13,2X,6(1PE12.4))
20860 WRITE(6,607) TX(6),TX(7),TX(8),TX(9),TX(10),TX(11)
20870 607 FORMAT(/24X,5HTOTAL,6(1PE12.4)/)
20880 WRITE(6,608)
20890 608 FORMAT(5X,'OUT FLOWS'/)
20900 WRITE(6,605)
20910 DO 322 NG=1,NGMX
20920 WRITE(6,606) NG,GRX(12,NG),GRX(13,NG),GRX(14,NG),GRX(15,NG),
20930 GRX(16,NG),GRX(17,NG)
20940 322 CONTINUE
20950 WRITE(6,607) TX(12),TX(13),TX(14),TX(15),TX(16),TX(17)
20960 WRITE(6,605)
20970 609 FORMAT(5X,'NET FLOWS'/)
20980 WRITE(6,605)
20990 DO 323 NG=1,NGMX
21000 WRITE(6,606) NG,GRX(18,NG),GRX(19,NG),GRX(20,NG),GRX(21,NG),
21010 GRX(22,NG),GRX(23,NG)
21020 323 CONTINUE
21030 WRITE(6,607) TX(18),TX(19),TX(20),TX(21),TX(22),TX(23)
21040 WRITE(6,610) TX(2),TX(3),TLEAK,EIGEN
21050 610 FORMAT(/10X,'TOTAL FISSION',1PE12.4/10X,
21060 & 'TOTAL ABSORPTION',1PE12.4/10X,'TOTAL LEAKAGE',1PE12.4/
21070 & 10X,'EIGEN VALUE',OPF11.4//1H1,20X,10(1H=),
21080 & 'POWER AVERAGE FLUX DISTRIBUTION',BUT10',
21090 & 'NS',10(1H=))
21100 LREC=IMX*JMX=KMX=NGMX
21110 CALL DUMPER(6,2,SCLFL,LREC,4H SCL,4HFL,IMX,JMX,KMX,NGMX,-1,-1,
21120 & 4H I,4H J,4H K,4H NG,4H ,4H )
21130 RETURN
21140 END
21150 SUBROUTINE REWER(IOPT,ROUT)
21160 DIMENSION IOPT(6),ROUT(6)
21170 DO 1 IOP=1,6
21180 IF(IOPT(IOPT),LE,0) GOTO 101
21190 NF=ABS(ROUT(IOPT(IOPT)))
21200 REWIND NF
21210 101 CONTINUE
21220 1 CONTINUE
21230 RETURN
21240 END
21250 SUBROUTINE DUMPRD(DELX,DELY,DELZ,MAP,XABS,XFIS,XTRA,XSCT,XI,
21260 & IBNDCN,ISCTMX,ISCTMN,ANGFLI,ANGFLJ,ANGFLK,SCLFL,SCLL2,BALX,
21270 & BALY,BALZ)
21280 DIMENSION DELX(IMX),DELY(JMX),DELZ(KMX),MAP(IMX,JMX,KMX),
21290 & XABS(NGMX,MAPMX),XFIS(NGMX,MAPMX),XTRA(NGMX,MAPMX),XSCT(NGMX,
21300 & NGMX,MAPMX),XI(NGMX),IBNDCN(3,2),ISCTMX(NGMX,MAPMX),
21310 & ISCTMN(NGMX,MAPMX)
21320 DIMENSION ANGFLI(8,MFIX,IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MFIX,IMX,JMX+
21330 & 1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,KMX+1,NGMX)
21340 DIMENSION SCLFL(IMX,JMX,KMX,NGMX),SCLL2(IMX,JMX,KMX,NGMX),
21350 & BALX(2,IMX+1,JMX,KMX),BALY(2,IMX,JMX+1,KMX),BALZ(2,IMX,JMX,KMX+
21360 & 1)
21370 COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
21380 & NQUAD(2,2,2)
21390 COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
21400 & CONCR,TIMDUT,TIMLMT,TIMCPU,ITROUT,ITRIN,ITRES,ISTA,IPRV,NOW,
21410 & INMX,IMXM,KMX,NGMX
21420 REWIND 1
21430 READ(1) IMXD,JMXD,KMXD,MMXD,MFIXD,NGMXD,MAPMXD,NSND,NFIXD
21440 IF(IMX.NE.IMXD) STOP '... INPUT MISMATCH IN *IMX*'
21450 IF(JMX.NE.JMXD) STOP '... INPUT MISMATCH IN *JMX*'
21460 IF(KMX.NE.KMXD) STOP '... INPUT MISMATCH IN *KMX*'
21470 IF(NGMX.NE.NGMXD) STOP '... INPUT MISMATCH IN *NGMX*'
21480 IF(MAPMX.NE.MAPMXD) STOP '... INPUT MISMATCH IN *MAPMX*'
21490 IF(NSN.NE.NSND) STOP '... INPUT MISMATCH IN *NSN*'
21500 READ(1) EGNOLD,EIGEND,PWROLD,PWRNEW
21510 IF(EIGEN,LE,0,0) EIGEN=EIGEND
21520 READ(1) (DELX(I),I=1,IMX)
21530 READ(1) (DELY(J),J=1,JMX)
21540 READ(1) (DELZ(K),K=1,KMX)
21550 DO 1 K=1,KMX
21560 DO 1 J=1,JMX
21570 READ(1) (MAP(I,J,K),I=1,IMX)
21580 1 CONTINUE
21590 DO 10 M=1,MAPMX
21600 READ(1) (XABS(N,M),N=1,NGMX)
21610 READ(1) (XFIS(N,M),N=1,NGMX)
21620 READ(1) (XTRA(N,M),N=1,NGMX)
21630 10 CONTINUE
21640 DO 2 M=1,MAPMX
21650 DO 2 N=1,NGMX
21660 READ(1) (XSCT(L,N,M),L=1,NGMX)
21670 2 CONTINUE
21680 READ(1) (XI(N),N=1,NGMX)
21690 DO 11 J=1,2
21700 READ(1) (IBNDCN(I,J),I=1,3)
21710 11 CONTINUE
21720 DO 12 M=1,MAPMX
21730 READ(1) (ISCTMX(N,M),N=1,NGMX)
21740 READ(1) (ISCTMN(N,M),N=1,NGMX)
21750 12 CONTINUE
21760 DO 3 N=1,NGMX
21770 DO 3 K=1,KMX
21780 DO 3 J=1,JMX
21790 DO 3 I=1,IMX+1
21800 DO 3 M=1,MFIX
21810 READ(1) (ANGFLI(L,M,J,J,K,N),L=1,8)
21820 3 CONTINUE
21830 DO 4 N=1,NGMX
21840 DO 4 K=1,KMX
21850 DO 4 J=1,JMX+1
21860 DO 4 I=1,IMX
21870 DO 4 M=1,MFIX
21880 READ(1) (ANGFLJ(L,M,J,J,K,N),L=1,8)
21890 4 CONTINUE
21900 DO 5 N=1,NGMX
21910 DO 5 K=1,KMX+1
21920 DO 5 J=1,JMX
21930 DO 5 I=1,IMX

```

```

21940      DO 5 M=1,MFIX
21950          READ(1) (ANGFLK(L,M,I,J,K,N),L=1,8)
21960  5 CONTINUE
21970      DO 7 N=1,NGMX
21980          DO 7 K=1,KMX
21990              DO 7 J=1,JMX
22000                  READ(1) (SCLL2(I,J,K,N),I=1,IMX)
22010  7 CONTINUE
22020      DO 8 N=1,NGMX
22030          DO 8 K=1,KMX
22040              DO 8 J=1,JMX
22050                  READ(1) (SCLFL(I,J,K,N),I=1,IMX)
22060  8 CONTINUE
22070      DO 14 K=1,KMX
22080          DO 14 J=1,JMX
22090              DO 14 I=1,IMX+1
22100                  READ(1) (BALX(ID,I,J,K),ID=1,2)
22110 14 CONTINUE
22120      DO 15 K=1,KMX
22130          DO 15 J=1,JMX+1
22140              DO 15 I=1,IMX
22150                  READ(1) (BALY(ID,I,J,K),ID=1,2)
22160 15 CONTINUE
22170      DO 16 K=1,KMX+1
22180          DO 16 J=1,JMX
22190              DO 16 I=1,IMX
22200                  READ(1) (BALZ(ID,I,J,K),ID=1,2)
22210 16 CONTINUE
22220      WRITE(6,601)
22230 601 FORMAT(///1H ,132(1H*)//25X,
22240      & ' R E S T A R T   D A T A   G O T T E N ',
22250      & '   F R O M   U N I T   N O . 1'//1H ,132(1H*)//)
22260      REWIND 1
22270      RETURN
22280      END
22290      SUBROUTINE DUMPWR(DELX,DELY,DELZ,MAP,XABS,XFIS,XTRA,XSCT,X1,
22300      & IBNDCN,ISCTMX,ISCTMN,ANGFLI,ANGFLJ,ANGFLK,SCLFL,SCLL2, BALX,
22310      & BALY,BALZ)
22320      DIMENSION DELX(IMX),DELY(JMX),DELZ(KMX),MAP(IMX,JMX,KMX),
22330      & XABS(NGMX,MAPMX),XFIS(NGMX,MAPMX),XTRA(NGMX,MAPMX),XSCT(NGMX,
22340      & NGMX,MAPMX),XI(NGMX),IBNDCN(3,2),ISCTMX(NGMX,MAPMX),
22350      & ISCTMN(NGMX, MAPMX)
22360      DIMENSION ANGFLI(8,MFIX,IMX+1,JMX,KMX,NGMX),ANGFLJ(8,MFIX,IMX,JMX+
22370      & 1,KMX,NGMX),ANGFLK(8,MFIX,IMX,JMX,KMX+1,NGMX)
22380      DIMENSION SCLFL(IMX,JMX,KMX,NGMX),SCLL2(IMX,JMX,KMX,NGMX),
22390      & BALX(2,IMX+1,JMX,KMX),BALY(2,IMX,JMX+1,KMX),BALZ(2,IMX,JMX,KMX+
22400      & 1)
22410      COMMON /CONTL/IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX,ITINMX,
22420      & NQUAD(2,2,2)
22430      COMMON /MONIT/EGNOLD,EIGEN,PWROLD,PWRNEW,CONCR,CONCRI,CONCRE,
22440      & CNCCR,TIMOUT,TIMLMT,TIMCPU,ITRDT,ITRIN,ITREB,ISTA,IPRY,NDW,
22450      & IMXM,JMXM,KMXM,NGMXM
22460      REWIND 2
22470      WRITE(2) IMX,JMX,KMX,MMX,MFIX,NGMX,MAPMX,NSN,NFIX
22480      WRITE(2) EGNOLD,EIGEN,PWROLD,PWRNEW
22490      WRITE(2) (DELX(I),I=1,IMX)
22500      WRITE(2) (DELY(J),J=1,JMX)
22510      WRITE(2) (DELZ(K),K=1,KMX)
22520      DO 1 K=1,KMX
22530          DO 1 J=1,JMX
22540              WRITE(2) (MAP(I,J,K),I=1,IMX)
22550  1 CONTINUE
22560      DO 10 M=1,MAPMX
22570          WRITE(2) (XABS(N,M),N=1,NGMX)
22580          WRITE(2) (XFIS(N,M),N=1,NGMX)
22590          WRITE(2) (XTRA(N,M),N=1,NGMX)
22600 10 CONTINUE
22610      DO 2 M=1,MAPMX
22620          DO 2 N=1,NGMX
22630              WRITE(2) (XSCT(L,N,M),L=1,NGMX)
22640  2 CONTINUE
22650      WRITE(2) (XI(N),N=1,NGMX)
22660      DO 11 J=1,2
22670          WRITE(2) (IBNDCN(I,J),I=1,3)
22680 11 CONTINUE
22690      DO 12 M=1,MAPMX
22700          WRITE(2) (ISCTMX(N,M),N=1,NGMX)
22710          WRITE(2) (ISCTMN(N,M),N=1,NGMX)
22720 12 CONTINUE
22730      DO 3 N=1,NGMX
22740          DO 3 X=1,KMX
22750              DO 3 J=1,JMX
22760                  DO 3 I=1,IMX+1
22770                      DO 3 M=1,MFIX
22780                          WRITE(2) (ANGFLI(L,M,I,J,K,N),L=1,8)
22790  3 CONTINUE
22800          DO 4 N=1,NGMX
22810              DO 4 K=1,KMX
22820                  DO 4 J=1,JMX+1
22830                      DO 4 I=1,IMX
22840                          DO 4 M=1,MFIX
22850                              WRITE(2) (ANGFLJ(L,M,I,J,K,N),L=1,8)
22860  4 CONTINUE
22870          DO 5 N=1,NGMX
22880              DO 5 X=1,KMX+1
22890                  DO 5 J=1,JMX
22900                      DO 5 I=1,IMX
22910                          DO 5 M=1,MFIX
22920                              WRITE(2) (ANGFLK(L,M,I,J,K,N),L=1,8)
22930  5 CONTINUE
22940          DO 7 N=1,NGMX
22950              DO 7 K=1,KMX
22960                  DO 7 J=1,JMX
22970                      WRITE(2) (SCLL2(I,J,K,N),I=1,IMX)
22980  7 CONTINUE
22990          DO 8 N=1,NGMX
23000              DO 8 K=1,KMX
23010                  DO 8 J=1,JMX
23020                      WRITE(2) (SCLFL(I,J,K,N),I=1,IMX)
23030  8 CONTINUE
23040          DO 14 K=1,KMX
23050              DO 14 J=1,JMX
23060                  DO 14 I=1,IMX+1
23070                      WRITE(2) (BALX(ID,I,J,K),ID=1,2)
23080 14 CONTINUE
23090          DO 15 K=1,KMX
23100              DO 15 J=1,JMX+1
23110                  DO 15 I=1,IMX
23120                      WRITE(2) (BALY(ID,I,J,K),ID=1,2)
23130 15 CONTINUE
23140          DO 16 K=1,KMX+1
23150              DO 16 J=1,JMX
23160                  DO 16 I=1,IMX
23170                      WRITE(2) (BALZ(ID,I,J,K),ID=1,2)
23180 16 CONTINUE
23190      WRITE(6,601)
23200 601 FORMAT(///1H ,132(1H*)//25X,' R E S T A R T   D U M P   T A K E N ',
23210      & '   O N   U N I T   N O . 2'//1H ,132(1H*)//)
23220      REWIND 2

```

```

23230      END
23240      SUBROUTINE DUMPER(NFOUT,IFUNC,ARRAY,LREC,NAME1,NAME2,IDMX1, IDMX2,
23250      2 IDMX3, IDMX4, IDMX5, IDMX6, IDNAM1, IDNAM2, IDNAM3, IDNAM4, IDNAM5,
23260      8 IDNAM6)
23270      DIMENSION IDMX(6), IDNAME(6), IDNUMB(6)
23280      DIMENSION ARRAY(LREC)
23290      CHARACTER*4 IDNAME, NAME1, NAME2
23300      CHARACTER*4 IDNAM1, IDNAM2, IDNAM3, IDNAM4, IDNAM5, IDNAM6
23310      IDMX(1)=IDMX1
23320      IDMX(2)=IDMX2
23330      IDMX(3)=IDMX3
23340      IDMX(4)=IDMX4
23350      IDMX(5)=IDMX5
23360      IDMX(6)=IDMX6
23370      IDNAME(1)=IDNAM1
23380      IDNAME(2)=IDNAM2
23390      IDNAME(3)=IDNAM3
23400      IDNAME(4)=IDNAM4
23410      IDNAME(5)=IDNAM5
23420      IDNAME(6)=IDNAM6
23430      IF(NFOUT.EQ.6) GOTO 3
23440      GOTO(1,2),IFUNC
23450C *
23460      1 CONTINUE
23470C*  INTEGER DUMP * * * * *
23480      DO 11 ID=1,LREC
23490      WRITE(NFOUT) INT(ARRAY(ID))
23500      11 CONTINUE
23510      RETURN
23520C *
23530      2 CONTINUE
23540C*  REAL DUMP * * * * *
23550      DO 21 ID=1,LREC
23560      WRITE(NFOUT) ARRAY(ID)
23570      21 CONTINUE
23580      RETURN
23590C *
23600      3 CONTINUE
23610C*  BLOCKED DATA AREA * * * * *
23620      WRITE(6,601) NAME1,NAME2, IDNAM1, IDNAM2
23630      601 FORMAT(//1H ,10(2H= ),' D U M P L I S T D F ',2A4,10(2H =//
23640      8 ' 5X',1 ' A4,' FOR HORIZONTAL ,',A4,' FOR VERTICAL ARRAYS.)/)
23650      LBREC=IDMX(1)+IDMX(2)+IDMX(3)+IDMX(4)+IDMX(5)+IDMX(6)
23660      IF(LBREC.LT.0) LBREC=-LBREC
23670      IF(LREC.NE.LBREC) STOP '...RECORD SIZE ERROR.'
23680      IDIM=0
23690      DO 300 ID=1,6
23700      IF[IDMX(ID).EQ.-1] GOTO 300
23710      IF[IDMX(ID).EQ.0.OR.IDMX(ID).LT.-1]
23720      8 STOP '...DUMP OPTION ERROR.'
23730      IDIM=IDIM+1
23740      300 CONTINUE
23750      DO 303 ID=1,6
23760      IDMX(ID)=IABS(IDMX(ID))
23770      303 CONTINUE
23780      DO 30 ID6=1, IDMX(6)
23790      IDNUMB(6)=ID6
23800      DO 30 ID5=1, IDMX(5)
23810      IDNUMB(5)=ID5
23820      DO 30 ID4=1, IDMX(4)
23830      IDNUMB(4)=ID4
23840      IF(IDIM.LT.4) GOTO 5
23850      WRITE(6,602) [IDNAME(IDS), IDNUMB(IDS), IDS*4, IDIM]
23860      602 FORMAT(5X,3(A4,'-',13,3X)/)
23870      5 CONTINUE
23880      DO 30 ID3=1, IDMX(3)
23890      WRITE(6,603) IDNAM3, ID3
23900      IF(IFUNC.EQ.1) GOTO 6
23910      WRITE(6,603) [ID1, ID1=1, IDMX(1)]
23920      603 FORMAT(5X,10(16,6X)/)
23930      6 CONTINUE
23940      GOTO(31,32),IFUNC
23950C *
23960      31 CONTINUE
23970C*  BLOCKED INTEGER DUMP * * * * *
23980      DO 301 ID2=1, IDMX(2)
23990      ID1=[([([([ID6-1)+IDMX(5)+ID5-1)+IDMX(4)+ID4-1)+IDMX(3)+
24000      8 ID3-1)+IDMX(2)+ID2-1)+IDMX(1)]
24010      WRITE(6,604) [INT(ARRAY(IDD+ID1)), ID1=1, IDMX(1)]
24020      604 FORMAT(1H ,3014)
24030      301 CONTINUE
24040      GOTO 999
24050C *
24060      32 CONTINUE
24070C*  BLOCKED REAL DUMP * * * * *
24080      DO 302 ID2=1, IDMX(2)
24090      ID1=[([([([([ID6-1)+IDMX(5)+ID5-1)+IDMX(4)+ID4-1)+IDMX(3)+
24100      8 ID3-1)+IDMX(2)+ID2-1)+IDMX(1)]
24110      WRITE(6,605) ID2, [ARRAY(IDD+ID1), ID1=1, IDMX(1)]
24120      605 FORMAT(13,2X,10(1PE12.5))
24130      302 CONTINUE
24140      999 CONTINUE
24150      30 CONTINUE
24160      RETURN
24170      END

```